Ensemble System based on Genetic Algorithm for Stock Market Forecasting

Rafael Thomazi Gonzalez
Institute of Informatics
Federal University of Rio Grande do Sul
Porto Alegre, RS, Brazil
rthomazigonzalez@gmail.com

MSc. Carlos Alberto Padilha
Institute of Informatics
Federal University of Rio Grande do Sul
Porto Alegre, RS, Brazil
carlos.engcomp@gmail.com

Prof. Dr. Dante Augusto Couto Barone
Institute of Informatics
Federal University of Rio Grande do Sul
Porto Alegre, RS, Brazil
dante.barone@gmail.com

***Abstract*—Financial time series forecasting is regarded as one of the most challenging applications of time series forecasting. Many researchers have been focusing on this topic due to the potential of yielding significant profits on the invested money in a short time frame. Believing in the predictability of stock markets, traders have been using Technical Analysis tools for a very long time to analyze and predict the behavior of stocks, aiming to make the best investment decisions possible with this information. In addition, applying machine learning techniques to predict stock market movements has become an area of research that has received a lot of attention in recent years. Popular algorithms such as Artificial Neural Networks and Support Vector Machines have been widely used in this area and they have been reporting satisfactory performances. As an attempt to improve the accuracy of these algorithms, researchers have been proposing techniques to combine them, forming Ensemble Systems. This work presents the design of an Ensemble System based on Genetic Algorithm for forecasting the weekly prices' trend in the Sao Paulo Stock Exchange Index (Ibovespa Index). In order to evaluate the performance of the proposed method, experiments were conducted to compare it with other popular ensemble methods (e.g., Bagging, Boosting and Random Forests). Finally, the empirical results show that the proposed model outperforms the other ensemble methods. Therefore, this implies that the proposed approach can be used by traders as a promising tool for forecasting stock market prices.**

Keywords—genetic algorithm, ensemble system, financial market, technical analysis, forecasting

I. INTRODUCTION

Financial markets are considered to be complex and non-linear dynamical systems [1]. They are also characterized by data intensity, noise, a high degree of uncertainty, and hidden relationships [2]. Moreover, financial markets can be influenced by many factors such as political, economic, and psychological, which can further increase its volatility. Forecasting financial time series is therefore a difficult task. However, predicting is important in the sense that it provides concrete data for market participants to make more informed and accurate investment decisions.

Technical Analysis has been widely used by traders as a tool for predicting the future behavior of the stock prices [3]. Recently, applications of machine learning techniques to stock market prediction has been receiving a lot of attention, as it has been showing better predictive accuracy. In [4], a review is presented of these techniques and its applications in stock markets. There have been many studies using Artificial Neural Networks (ANNs) for financial time series modeling and forecasting. Some successful examples are: [5], [6], [7], and [8]. However, ANNs have some disadvantages including the need for the determination of the number of processing elements in the hidden layers, and the value of controlling parameters, which can lead to overfitting [9]. In order to overcome these issues, researchers have been using Support Vector Machines (SVMs), a specific type of learning algorithm developed by Vapnik (1995) that implements the structural risk minimization principle, i.e. it aims to minimize an upper bound of generalization error. This method has become very popular and some applications of SVM to financial forecasting problems have been reported in [9], [10], [11]. As an attempt to improve the accuracy and stability of a classification system generated by single machine learning algorithms, many researchers have been proposing the combination of multiple classifiers, forming Ensembles [12]. In recent years, these systems have become a popular topic. Their application in predicting financial time series has also shown successful results such as in [13] and [14].

Evolutionary Computation (EC) techniques, another subfield of Artificial Intelligence, have been proving to be a powerful tool kit for economic analysis. Researchers have been using these techniques to approach financial problems such as modeling and forecasting financial markets [15-16], creating and optimizing trade strategies [17], and portfolio management [18].

This paper aims to apply an Ensemble of SVMs to predicting the movement direction of stock price index, and a Genetic algorithm to perform feature selection and parameters optimization for each member of this ensemble. This paper also examines the feasibility of applying the proposed method in financial forecasting by comparing it with other methods of forming ensemble such as Bagging, AdaBoost and Random Forest.

This paper is organized as follows. Section II introduces the key factors on Ensemble Systems development, the SVM algorithm and some of its characteristics, and also has a brief explanation on genetic algorithms and its mechanics. Section III describes the proposed method. Section IV details the experimental procedure and analyzes the experimental results. Finally, Section V presents the conclusion of the paper.

978-1-4799-7492-4/15/$31.00 ©2015 IEEE

II. THEORETICAL BACKGROUND

A. Ensemble Systems

In recent years, researchers have been proposing methods to improve the accuracy and stability of the predictive systems generated by machine learning algorithms. One method that stands out is the combination of multiple learning algorithms, forming Ensembles or Committees [12]. The main idea of using Ensembles is that different individual models can offer complementary information about unknown instances, improving the quality of the overall classification process in terms of generalization and accuracy [19]. In the regular architecture of a committee, a new input pattern is presented to all classifiers. Then the individual models provide their outputs and send them to a combination method, which is responsible for providing the final output of the system.

Two points are important when designing ensemble systems: the base classifiers and the combination method [19]. In relation to the ensemble components, it is fundamental to have set of individual models whose errors are at least somewhat uncorrelated. Thus, when combining them, individual failures will be minimized. In other words, the individual models should be diverse among themselves, since the combination of identical classifiers does not improve the accuracy [12]. Diversity among the base classifiers is the key to achieving high accuracy in ensembles. Diversity can be reached when the base models are built under different circumstances, such as in the following ways:

- ***Different training sets*:** multiple training sets are created by resampling the original data according to some sampling distribution. A classifier is then built from each training set. Learning strategies such as *Bagging* [20] and *Boosting* [21] achieve diversity through the use of training sets with different instances;
- ***Different input features*:** a subset of input features is chosen to form each training set. *Random Forest* [22], which uses decision trees as its base classifier, is a well-known ensemble method that uses this approach;
- ***Different parameters settings*:** diversity can be achieved by changing the initial parameter setting of the base classifiers. For instance, an artificial neural network can generate different models by either changing the initial weights of the links between the neurons or its network topology;
- ***Different learning algorithms*:** diversity can be achieved by using different types of individual classifiers, also called heterogeneous ensembles. For instance, usually an ensemble that is composed of a decision tree and a neural network is likely to be more diverse than ensembles composed only of either neural networks, or decision trees;

Once the set of base classifiers has been created, the next step is to choose an effective way of combining their outputs. The choice of the best combination method for an ensemble requires exhaustive training and testing [19]. There are a great number of combination methods reported in literature. Three of the most used methods for classification problems are: Majority Voting, Weighted Voting and Naive Bayesian [12].

B. Support Vector Machine

Support Vector Machine (SVM) was first proposed by Vladimir Vapnik [23]. The basic idea of SVM is to use a linear model to implement nonlinear class boundaries through some nonlinear mapping of the input vector into a high-dimensional feature space. Thus, a linear model constructed in the new space can represent a nonlinear decision boundary in the original space. In the new space, an optimal separating hyperplane is constructed. Therefore, SVM is known as the algorithm that finds a special kind of linear model, i.e. the maximum margin hyperplane, which gives the maximum separation between the decision classes. The margin of the classes is defined by the support vectors, which are the marginal examples of a given class in the training data, i.e. input examples closer to the other class. All other training examples are irrelevant for defining the binary class boundaries.

For the linearly separable case, a hyperplane separating the binary decision classes in the two-attribute case can be represented as the following equation:

$$y = w_0 + w_1 x_1 + w_2 x_2 \quad (1)$$

Where y is the outcome, x_i are the feature values and w_i are the weight values that represent the hyperplane, and should be learned by the algorithm. The maximum margin hyperplane can be described as the following equation in terms of the support vectors:

$$y = b + \sum_{i=0} \alpha_i y_i x(i) \cdot x \quad (2)$$

Where $x(i)$ is the ith support vector, y_i is the class value of training example $x(i)$, $\cdot$ represents the dot product and the vector x represents a test instance. In this equation, b and α_i are parameters that determine the hyperplane. From the implementation point of view, finding the support vectors and determining the parameters b and α_i are equivalent to solving a linearly constrained quadratic programming.

As mentioned above, SVM transform the inputs into the high-dimensional feature space in order to construct a linear model to implement nonlinear class boundaries. For the nonlinearly separable case, a high-dimensional version of (2) is represented as follows:

$$y = b + \sum_{i=0} \alpha_i y_i K(x(i), x) \quad (3)$$

The function $K(x(i), x)$ is defined as the kernel function. There are different kernels for generating the inner products to construct SVMs with different types of nonlinear decision surfaces in the input space. Common examples of kernel functions are the Polynomial kernel $K(x, y) = (xy + 1)^d$ and the Gaussian Radial Basis Function (RBF) $K(x, y) = exp(-\|x - y\|^2/2\sigma^2)$, where d is the degree of the polynomial kernel, and σ is the bandwidth of the Gaussian radial basis function kernel [24].

For the separable case, the coefficient α_i in (3) has a lower bound equal to zero. For the non-separable case, SVM can be

generalized by placing an upper bound C on the coefficients α_i in addition to the lower bound [25]. The coefficient C, also known as the penalty factor, controls the trade-off between achieving a low error rate on the training data and the model complexity. For small values of C, the number of training errors increases, since it generates a larger-margin separating hyperplane. Conversely, large values of C will lead to a smaller-margin hyperplane, which ends up penalizing non-separable points more severely [23].

C. Genetic Algorithm

Genetic Algorithms (GAs) were first introduced by Holland [26]. They are a class of Evolutionary Algorithms, which are general adaptive optimization search methodologies based on a direct analogy to Darwinian natural selection and genetics in biological systems. GAs have been considered a promising alternative to conventional heuristic methods. The relative insensitivity of GAs to noise and the requirement of no domain knowledge make them a powerful tool for optimization problems. Moreover, associated with the characteristics of exploration (the process of visiting new regions of a search space) and exploitation (the process of visiting those regions of a search space within the neighborhood of previously visited points), GAs are capable of dealing with large search spaces efficiently [27]. Therefore, they have been widely used in different tasks, including numerical and combinatorial optimization, optimization of neural networks, multi-agent systems, economics, and bioinformatics [28-30].

The simple Genetic Algorithm works with a set of candidate solutions called a population, in which each chromosome (individual) represents a possible solution for the given problem. The individuals are assessed through a fitness function, which informs the goodness of a chromosome in the solution of the optimization task. In addition, this is the function optimized (minimized or maximized) by the GA. Based on fitness values, individuals are selected and some genetic operators (crossover and mutation) can be applied to them with certain probabilities, forming new ones. These operators are successively applied to the population in a loop, being each run of the loop called generation. The main idea is that these individuals evolve, tending to create better ones, until acceptable results are obtained, or some stop condition is met. At the end, the fittest chromosome found during all generations is the GA's answer to the given [31].

III. Proposed Method

The main objective of this paper is to apply a genetic algorithm to enhance the classification performance a set of SVM in forecasting stock indexes variations. In this approach, GA performs, simultaneously, feature selection and parameter optimization, in order to generate different subsets of inputs for each base classifier. The overall architecture of the proposed method (*GAENSEMBLE*) is shown in Fig 1.

A. Ensemble System

As mentioned previously, an effective ensemble should consist of a set of classifiers that are not only highly accurate, but ones that also make their errors in different parts of the input space as well. The ensemble used in the proposed method is composed of a set of 10 SVMs. In order to increase the diversity of the ensemble, each SVM is constructed using different sets of attributes and parameters. After all SVMs have been trained, their outputs are combined by Majority Voting.

In [32], it is shown that the Gaussian Radial Basis Function is the SVM kernel that allows for higher diversity among the most popular ones, because its Gaussian width parameter promotes a more detailed tuning. Thus, this kernel is used in all SVMs of the ensemble. Finally, for each SVM there are two parameters to be optimized by the Genetic Algorithm: σ and C.

B. Feature Selection and Parameters Optimization

Feature selection aims to reduce the dimensionality of the attributes of a dataset, looking for the best ones and excluding the irrelevant or redundant attributes. There is a vast number of works in the literature using feature selection in ensemble systems, GA being one of the most used methods, such as in [33] and [34]. In the context of ensemble systems, feature

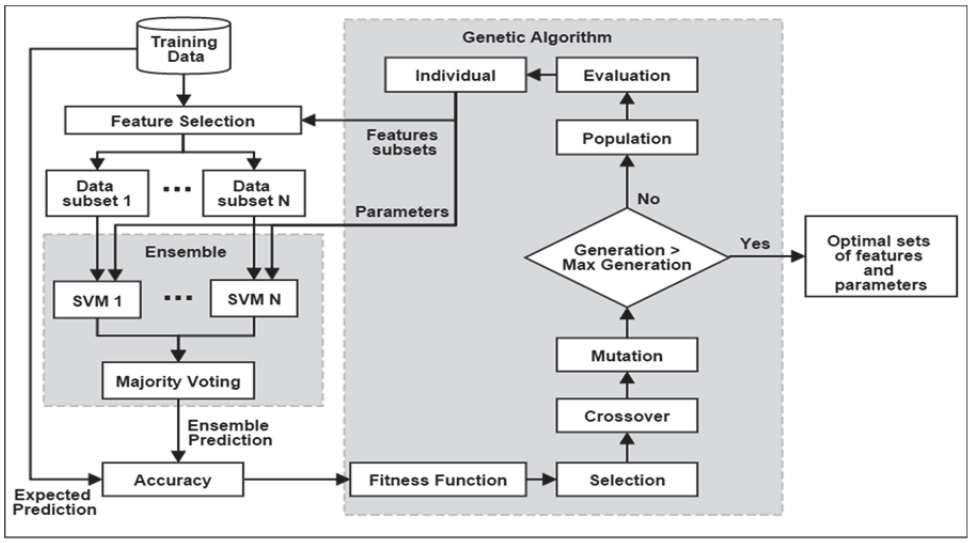

Fig. 1. Overall architecure of the proposed method

selection methods are used to provide different subsets of features for the base classifiers, aiming to increase the diversity of the ensemble, and to reduce redundancy among the attributes of a pattern. This is because the individual classifiers will classify the same input patterns, but these patterns will have been built using different subsets of features. In addition, by reducing the dimensionality of the base classifiers, feature selection also helps to reduce the overall complexity of the ensemble.

In addition to the feature selection, proper parameter settings can improve the SVM classification accuracy and avoid either over-fitting, or under-fitting, on the training data. As mentioned previously, diversity can also be promoted in ensemble systems by creating base classifiers with different parameter settings. At this level, GA tries to find the best values of σ and C for each SVM in the ensemble, aiming to increase the accuracy of the final classifier.

C. Genetic Algorithm

The GA used for feature selection and parameters optimization in the proposed ensemble system is similar to a standard genetic algorithm. The main contributions are in the representation of the individuals and, consequently, in the crossover and mutation operators. A hybrid chromosome is used to represent a possible solution (individual) for this problem. Individuals are code part as binary, and part as floating-point values. The size of the chromosome is $(F \times K) + (2 \times K)$ genes, where F represents the number of features of the dataset, and K represents the number of classifiers. The binary part represents the subset of features selected for each classifier, which '1' corresponds to a selected feature and '0' to a non-selected one. The first K bits represent the feature subset for classifier k_1, followed by K bits for classifier k_2, and so on. Similarly, the real part contains the two parameters to be optimized for each SVM of the ensemble, in which the first two values (σ_1 and C_1) represent the parameters for the classifier k_1, followed by other two values for classifier k_2, and so on.

Fig. 2 shows an example of an individual that represents an ensemble composed of three SVMs (k_1, k_2 and k_3) and a dataset with 5 attributes. In this example, the attributes 1, 3 and 5 are assigned to classifier k_1, attributes 2 and 4 are assigned to classifier k_2 and, attributes 1, 2 and 4 are assigned to classifier k_3. Also, classifier k_1 has $\sigma_1 = 0.05$ and $C_1 = 500$, classifier k_2 has $\sigma_2 = 0.1$ and $C_2 = 1000$, and classifier k_3 has $\sigma_3 = 0.1$ and $C_3 = 20$.

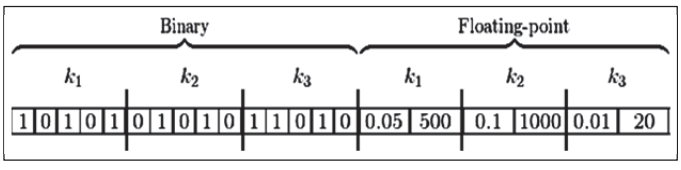

Fig. 2. Reprentation of the hybrid chromosome

In order to select the two parents for the crossover process, the *K-Tournament Selection* [27] method is used. This method establishes several engagements involving n individuals chosen at random from the population. In each combat, the individual with the best fitness value is selected. In this project K is fixed to 3. Moreover, aiming to assure that the best individuals always survive unaltered to the next generation, the *elitism* technique [27] is also applied, which copies the best individual of the population in generation t to the population in generation $t + 1$.

Since individuals are coded using two different types of representation, crossover and mutation are applied using different methods for each type of representation. In the case of the crossover, the *Scattered Crossover* technique [35] is used for the binary part, while the *BLX-Alpha-Beta Crossover* technique [36] is used for the real-coded part. Similar to the crossover step, the binary part of the chromosome is mutated by randomly flipping some bits of the string, while the real-coded part is mutated using the technique called *Non-Uniform Mutation* [37].

Classification accuracy of the ensemble classifier is used to evaluate the fitness of individuals. The proposed GA proceeds with the next generation until the process reaches the maximum number of generations. After manual adjustments, the main parameters were set as follow: 25 as population size; 80% as crossover rate; 10% as mutation rate; and 50 as maximum number of generation.

IV. Experimental Procedure

A. Research Data

This project aims to develop an algorithm to predict whether the next-week price of the Bovespa Index (Ibovespa) will be higher or lower than the current price. Ibovespa is the main indicator of the Brazilian stock market's average performance. This index is calculated and disseminated by the Brazilian stock exchange, known as BM&FBOVESPA. It is calculated as a weighted average of a theoretical portfolio which contains the more actively traded, and more representative stocks of the Brazilian stock market [38].

As globalization has been deepening the interactions between global financial markets, nowadays, almost no market is isolated. Economic data, political perturbation, and any other oversea affairs could cause dramatic fluctuations in domestic markets. Therefore, in order to enhance the accuracy of the proposed method, in this project it is proposed to also use global financial data, such as major world stock indices and foreign currency as input data. Thus, the research dataset is composed of the following financial data:

- ***World Stock Indices:*** Ibovespa Index (Brazil), S&P 500 Index (U.S.), Dow Jones Industrial Average (U.S.), FTSE 100 (U.K.), DAX Index (Germany), Nikkei 225 (Japan), Hang Seng Index (China);
- ***Currency:*** USD, EUR and CNY.

The historical dataset used in this project was downloaded from [39]. Since the price value alone lends little insight into future price movements, the goal becomes to develop features that provide information on not only past and current price movements, but also its future behavior. Because of that, this study selects 7 technical indicators to make up the initial attributes, as determined by the review of domain experts and

prior research [3, 40]. Table 1 presents the technical indicators considered in this study.

TABLE I. THECNICAL INDICATORS USED FOR FEATURE EXTRACTION

Indicator	Description
Price Rate of Change (ROC)	It measures de difference between the current price and the price n days ago.
Ratio	It is the ratio between two ROC s calculated over different time intervals. Particularly, ROC_n/ROC_m for $m > n$ is informative because it lends insight into how the change in price is changing over time
Disparity	It displays the distance of the current price from the moving average of n days (MA_n).
Price Oscillator	It measures the difference between two moving averages of a security's price (MA_n/MA_m for $m > n$).
Stochastic Oscillator	It compares where a stock's price closed relative to its price range over a given time period.
MACD Histogram	It indicates changes in the strength and direction of a trend in security's price.
Parabolic SAR	It trails price as the trend extends over time, aiming to show the direction of a stock's trend and potential points where this trend has a higher-than-normal probability of switching directions.

The first two technical indicators, ROC and $Ratio$, are applied to all stock indices and currencies, while the other five are only applied to the Ibovespa Index. Finally, the feature set is then scaled into the range of [-1, 1]. The goal of linear scaling is to independently normalize each feature component to the specified range. It ensures the larger value input attributes do not overwhelm smaller value inputs, and also helps reduce prediction errors.

This study is to predict the directions of weekly change of the stock price index. Therefore, the instances are categorized as "0" or "1" in the research data. "0" means that the next week's index is lower than the current value, while "1" means that the next week's index is higher. The total number of sample is 744 trading weeks, from January 1989 to December 1998. About 80% of the data (596 weeks) is used for training and 20% (148 weeks) for holdout. The holdout data is used to test results with the data that is not utilized to develop the model.

B. Experimental Results

This analysis compares the proposed method (*GAENSEMBLE*) with Bagging (using SVM equipped with RBF kernel as base classifier), AdaBoost, and Random Forest. Moreover, a standalone SVM also equipped with RBF kernel is used to investigate whether applying an ensemble approach leads to performance gains or not.

In order to obtain a better estimation of training accuracy rates, a 10-fold cross-validation (10-CV) method is applied to all methods. In 10-CV, the training dataset is partitioned into 10 subsets. Of these 10 subsets, 9 subsets are used as training data and a single subset is retained as the testing data. This cross-validation process is then repeated 10 times (the number of folds). The advantage of 10-CV is that all instances are used for both training and testing, and each instance is used for testing only once per fold. Thus, all training accuracy results presented in this section refer to the mean over the 10 different test sets.

Fig. 3 presents a chart comparing the training and testing accuracy (%) of each model. In a general analysis of this chart, it is possible to state that the proposed method has on average a higher accuracy rate than all the other methods both in the training and testing set. For the training data, the *GAENSEMBLE* outperforms Bagging by 2.79%, AdaBoost by 0.13%, Random Forest by 1.48%, and SVM by 2.32%. For the testing set, the proposed method also presented the highest accuracy rate, outperforming Bagging by 6.08%, AdaBoost by 4.06%, Random Forest by 4.93%, and SVM by 3.38%. Fig. 4 compares the accuracy standard deviation of the cross-validation process for each model. It is important to analyze this information, because it gives insights about the model stability. Thus, given that the proposed method has the smallest standard deviation, it can be said that it is also the most stable method.

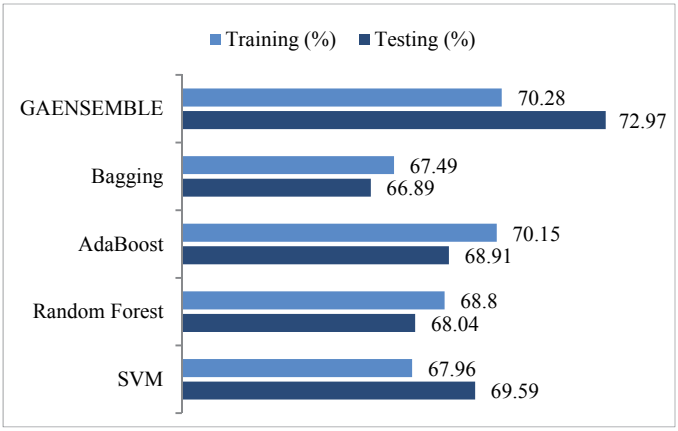

Fig. 3. Classification accuracies in the training and testing set

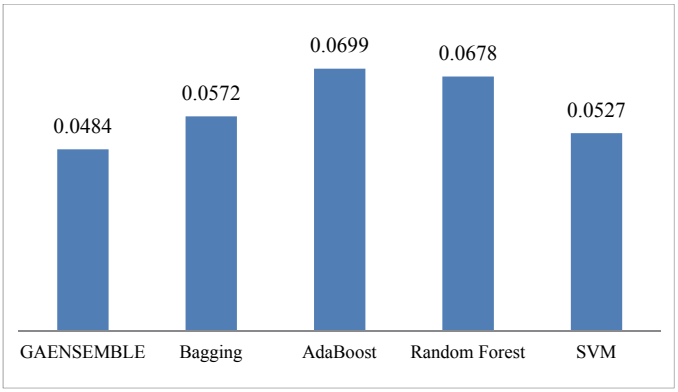

Fig. 4. Cross-Validation standard deviations

In addition, it is applied the McNemar's statistical test to further analyze the performance of each method on the testing set. This test is performed to examine whether differences of performances are statistically significant or not. This test is a nonparametric test for two related samples and may be used with nominal data [41]. Table 2 shows the results of the McNemar's test (p-values) when comparing the proposed method with the other methods. Adopting a significance level

of 0.05, a p-value less than 0.05 indicates that there is a significant difference among the two compared methods. As can be seen from Table 2, the accuracies on the testing set can be considered significantly different, which confirms that the *GAENSEMBLE* outperforms the other models.

TABLE II. RESULTS OF MCNEMAR'S TEST (P-VALUES)

	Bagging	**AdaBoost**	**Random Forest**	**SVM**
GAENSEMBLE	0.0116	0.0325	0.0297	0.0415

Finally, performing feature selection and parameter optimization simultaneously generates an enormous search space due to the large number of features combinations and the large range of parameters values. Because of this dimensionality, optimizing the ensemble system using the GA spent lots of time. Table 3 presents the time taken on the development notebook (an Asus N56VJ with a Intel Core i7-3630QM 2.4GHz and 8GB RAM running on Ubuntu 14.04 LTS 64bits), in seconds, to optimize each method in the training data. As expected, the *GAENSEMBLE* took longer than any other method.

TABLE III. TRAINING RUNTIMES

Method	**Runtime (s)**
GAENSEMBLE	15650.71
Bagging	52.18
AdaBoost	8664.91
Random Forest	257.46
SVM	35.33

V. CONCLUSION

This study proposed the use of machine learning algorithms, together with Technical Analysis tools, to forecast stock market movements. As a result, an Ensemble System based on Genetic Algorithm was designed to predict the weekly movement direction of Bovespa Index. In this method, a GA was applied to enhance the classification accuracy of the SVM Ensemble by performing feature selection and parameter optimization. Along with this, Technical Indicators were used to extract features as relevant as possible from historical data of Stock and Forex markets.

In order to prove the feasibility of the proposed method, experiments to compare it with well-known ensemble learning methods such as Bagging, Boosting and Random Forest were performed. As demonstrated in the experimental results, the proposed method outperforms the other ones in predicting whether the weekly price of Bovespa Index will rise or fall. Thus, this work concluded that the proposed method provides a promising alternative for stock market forecasting.

There are a number of further directions that can be investigated starting from this project. The first one is to explore new Technical Indicators or other types of financial data that might generate more informative features. Second, it is possible of performing a final tuning of the parameters of each SVM through a grid search on a subset of parameters generated depending on the final solution of the GA. Third, analyzing financial series with more historical data would make the proposed method learn more patterns from past data. Finally, in order to speed up the training process, it can be developed an algorithm that better utilize the power of parallelization provided by Genetic Algorithms and Ensemble Systems.

REFERENCES

[1] Y. Abu-Mostafa, A. Atiya. Introduction to financial forecasting. Applied Intelligence, v. 6, n. 3, p. 205-213, 1996.

[2] J. W. Hall. Adaptive selection of US stocks with neural nets. In: Deboeck GJ. (Ed.). Trading on the edge: neural, genetic, and fuzzy systems for chaotic financial markets. New York: Wiley, p. 45-65, 1994.

[3] J. J. Murphy. Technical Analysis of the Financial Markets. 1ª ed. New York: New York Institute of Finance, 1999.

[4] V. Setty, T. M. Rangaswamy, K. N. Subramanya. "A Review on Data Mining Applications to the Performance of Stock Marketing". International Journal of Computer Applications, v. 1, n. 3, p. 24-34, 2010.

[5] T. Kimoto, K. Asakawa, M. Yoda, M. Takeoka. "Stock market prediction system with modular neural network". Proceedings of the International Joint Conference on Neural Networks, San Diego, CA, 1990. p. 1-6.

[6] Y. Yoon, G. Swales. "Predicting stock price performance: a neural network approach". Proceedings of the 24th Annual Hawaii International Conference on System Sciences, Hawaii, 1991. p. 156-162.

[7] J. H. Choi, M. K. Lee, M. W. Rhee. "Trading S&P 500 stock index futures using a neural network". Proceedings of the Annual International Conference on Artificial Intelligence Applications on Wall Street, New York, NY, 1995. p. 63-72.

[8] R. C. Pinto. "Online Incremental One-Shot Learning of Temporal Sequences". 2011. 106 f. Thesis (Master) - Graduate Program in Computer Science, Institute of Informatics, Federal University Of Rio Grande do Sul, Porto Alegre, 2011.

[9] K. Kim. "Financial time series forecasting using support vector machines". Neurocomputing, v. 55, n. 2, p. 307-319, 2003.

[10] L. J. Cao, F. E. H. Tay. "Financial forecasting using support vector machines". Neural Computing And Applications, v. 10, v. 2, p. 184-192, 2001.

[11] F. E. H. Tay, L. J. Cao. "Improved financial time series forecasting by combining Support Vector Machines with self-organizing feature map". Intelligent Data Analysis, v. 5, n. 4, p. 339-354, 2001.

[12] L. I. Kuncheva. Combining Pattern Classifiers: Methods and Algorithms. [S.I.]: Wiley-Interscience, 2004.

[13] S. Lahmiri. Intelligent Ensemble Systems for Modeling NASDAQ Microstructure: A Comparative Study. Artificial Neural Networks in Pattern Recognition. Cham: Springer International Publishing, p. 240-251, 2014.

[14] Y. Ling, B. Yue, H. Zhang. "A New Wrapped Ensemble Approach for Financial Forecast". Journal of Intelligent Systems, v. 23, n. 1, p. 21-32, 2013.

[15] F. Prochnow. "Genetic Programing for Time Series Forecasting Applied to Financial Markets". 2013. 61 f. Final Paper (Bachelor) - Bachelor of Computer Science, Institute of Informatics, Federal University Of Rio Grande do Sul, Porto Alegre, 2013.

[16] J. Li. "FGP: A Genetic Programming Based Financial Forecasting Tool". 2000. 190f. Thesis (Doctor of Philosophy in Computer Science) - Department of Computer Science, University of Essex, Colchester, 2000.

[17] F. Allen, R. Karjalainen. "Using genetic algorithms to find technical trading rules". Journal of Financial Economics, v. 51, n. 2, p. 245-271, 1999.

[18] S. Sefiane, M. Benbouziane. "Portfolio Selection Using Genetic Algorithm". Journal of Applied Finance & Banking, v. 2, n. 4, p.143-154, 2012.

[19] P. Tan, M. Steinbach, V. Kumar. Introduction to Data Mining. Boston: Pearson Education, Inc., 2006.

[20] L. Breiman. “Bagging predictors”. Machine Learning, v. 24, n. 2, p. 123-140, aug. 1996.

[21] Y. Freund, R. Schapire. “Experiments with a new boosting algorithm”. Proceedings of the 13th International Conference on Machine Learning, Bari, Italy, 1996. p. 148-156.

[22] L. Breiman. “Random Forests”. Machine Learning, v. 45, n. 1, p. 5-32, oct. 2001.

[23] V. Vapnik. The Nature of Statistical Learning Theory. New York: Springer-verlag New York, 1995.

[24] N. Lima, A. Neto, J. Melo. “Creating an Ensemble of Diverse Support Vector Machines Using Adaboost”. Proceedings of the 2009 International Joint Conference on Neural Networks, Atlanta, Georgia, 2009. p. 2342-2346.

[25] I. Witten, E. Frank. Data Mining: Practical Machine Learning Tools and Techniques. 3ª ed. San Francisco: Morgan Kaufmann Publishers, 2005.

[26] J. H. Holland. Adaptation in Natural and Artificial Systems. Ann Arbor: The University Of Michigan Press, 1975.

[27] M. Melanie. An Introduction to Genetic Algorithms. Cambridge: Mit Press, 1998.

[28] D. Whitley, T. Starkweather, D. Fuquay. “Scheduling Problems and Traveling Salesmen: The Genetic Edge Recombination Operator”. Proceedings of the 3rd ICGA, Fairfax, VA, 1989. p. 133-140.

[29] M. F. Bramlette. “Initialization, Mutation and Selection Methods in Genetic Algorithms for Function Optimization”. Proceedings of the 4th ICGA, San Diego, CA, 1991. p. 100-107.

[30] Z. Michalewicz. “A Genetic Algorithms for the Linear Transportation Problem”. IEEE Trans. on Systems, Man, and Cybernetics, n. 21, v. 2, p. 445-452, 1991.

[31] A. Eiben, J. Smith. Introduction to Evolutionary Computing. Berlin Heidelberg New York: Springer-verlag, 2003.

[32] N. Lima, A. Neto, J. Melo. “Creating an Ensemble of Diverse Support Vector Machines Using Adaboost”. Proceedings of the 2009 International Joint Conference on Neural Networks, Atlanta, Georgia, 2009. p. 2342-2346.

[33] A. Canuto, D. Nascimento. “A Genetic-Based Approach to Features Selection for Ensembles Using a Hybrid and Adaptive Fitness Function”. Proceedings of the IJCNN, Brisbane, Australia, 2012. p. 1-8.

[34] D. Opitz. “Feature selection for ensembles”. Proceedings of the 16th National Conference on Artificial intelligence, Orlando, FL, 1999. p. 379-384

[35] C. Padilha, D. Neto, J. Melo. “Random Subspace Method and Genetic Algorithm Applied to a LS-SVM Ensemble”. Proceedings of the 22nd international conference on Artificial Neural Networks and Machine Learning, Lausanne, Switzerland, 2012. p. 164-171.

[36] S. Picek, D. Jakobovic, M. Golub. “On the Recombination Operator in the Real-Coded Genetic Algorithms”. Proceedings of the IEEE Congress on Evolutionary Computation, Cancún, Mexico, 2013. p. 3103-3110.

[37] Z. Michalewicz. Genetic Algorithms + Data Structures = Evolution Programs. 3ª ed. New York: Springer-verlag, 1996.

[38] BM&FBOVESPA: The New Exchange. Bovespa Index - Ibovespa. Available in: <http://www.bmfbovespa.com.br/indices/ResumoIndice.aspx?Indice=Ibovespa&Idioma=en-us>. Visited on: November 2014.

[39] Quandl. Available in: <https://www.quandl.com>. Visited on: November 2014.

[40] B. Rockefeller. Technical Analysis for Dummies. 2ª ed. Indianapolis: Wiley Publishing, Inc., 2011.

[41] B. Bostanci, E. Bostanci. “An Evaluation of Classification Algorithms Using Mc Nemar’s Test”. Proceedings of the 17th International Conference on Bio-Inspired Computing: Theories and Applications, 2012. p. 15-26.