

Prediction of assets behavior in financial series using machine learning algorithms

Esteves da Silva, Diego
Dept. Computer Engineering
IME - Military Institute of Engineering
Rio de Janeiro, Brazil

Duarte, Julio Cesar
Dept. Computer Engineering
IME - Military Institute of Engineering
Rio de Janeiro, Brazil

Abstract—The prediction of financial assets using either classification or regression models, is a challenge that has been growing in the recent years, despite the large number of publications of forecasting models for this task. Basically, the non-linear tendency of the series and the unexpected behavior of assets (compared to forecasts generated in studies of fundamental analysis or technical analysis) make this problem very hard to solve. In this work, we present for this task some modeling techniques using Support Vector Machines (SVM) and a comparative performance analysis against other basic machine learning approaches, such as Logistic Regression and Naive Bayes. We use an evaluation set based on company stocks of the BVM&F, the official stock market in Brazil, the third largest in the world. We show good prediction results, and we conclude that it is not possible to find a single model that generates good results for every asset. We also present how to evaluate such parameters for each model. The generated model can also provide additional information to other approaches, such as regression models

Keywords—Forecasting; Stock Market; Machine Learning; Financial Series

I. INTRODUCTION

The prediction of financial market assets is an issue that concerns both investors and researchers. In recent years, it has been studied using different machine learning approaches, as show in [12]. Despite the large amount of research, the prediction of the behavior of an asset in the real world, either with classification or regression models, is still a difficult task to accomplish [13].

The main difficulty on making good predictions is due to both the non-linear characteristic of financial time series and the great amount of uncertainty and noise found in financial market data [14], [15], [16]. For this reason, we argue that classical statistical models are not good to make this kind of prediction. This type of time series requires the use of algorithms with a greater ability to generalize, such as Support Vector Machine (SVM) and Artificial Neural Networks (ANN).

This work focuses in solving the asset prediction problem in the financial market, addressing the problem as a classification task and modeling it using supervised techniques such as Support Vector Machine, Logistic Regression and Naïve Bayes together with interday data to generate its classifiers.

The main purpose is to serve as decision model for the investor and can still be used as an entry into new models, particularly regression, in order to reduce its error.

The choice of the Support Vector Machine (SVM) algorithm was made in order to present results comparable and often superior to those achieved by other machine learning algorithms such as Artificial Neural Networks [30],[31].

We also use the Logistic Regression (LR) (with gradient descent method to choose the best parameters) and Naive Bayes algorithms (NB) to serve as a comparison in our study [3]. A baseline (BLS) has been implemented to verify the distance between the probabilities of success of an investor without any market knowledge to the accuracy found with some parameters applied to algorithms. We also use the open source Framework FAMA [6] for development and implementation of algorithms.

Despite good results in recent studies [12], the challenge of finding models with good generalization ability with actual data is still open. We conclude with the results of the experiments that we can find models with good amount of hits if the parameters are set correctly, and verify that, despite the good generalization characteristic proposed in algorithms in machine learning algorithms, it is not possible to apply a sole model for all stock assets.

II. RELATED WORK IN ASSETS' PREDICTIONS

In recent years several techniques for regression and classification financial assets have been explored, from classical statistical methods to more complex algorithms for machine learning, such as Artificial Neural Networks [19],[20], Logistic Regression [17],[18], PLSR [21] and more recently Support Vector Machine [22],[23],[24]. Reference works in the area prove that these soft computing techniques are well accepted for the study and evaluation of financial series.

The main difference between most of these works is the output information, while some of them provide the action to be taken by the user (classification problem), some others focus on the minimum and maximum stock values achieved during the day (regression problem).

Whatever the choice, we still got the same problem: the accuracy's loss in new periods and scenarios.

Our novel contributions are the analysis (interday stock parameters and some others variables in this problem) of how this models (focusing in the classifiers generated by SVM) behave with new data scenarios and the adjustments needed to minimize the loss percentage rate expected between training and tests scenarios. In order to achieve these results, we define a hybrid model using a sliding cross validation environment, where the model is re-trained after a defined period

III. MACHINE LEARNING TECHNIQUES AND MODELING

In this section we detail the machine learning algorithms used in this work, as well, as their important parameters. For the training task, we set our target variable (y) as “+1” if the day’s closing value of the asset is grater them the day before’s, which means that today, the asset closing value grew when compared to the closing value of the day before, also, if the value of the subtraction (day closing value – day-1 closing value) is negative, then we set the output target variable as “-1”, thus creating a binary classification problem.

A. Tendency Keeper (TK)

Tendency Keeper is actually not a machine learning algorithm and it was used as our baseline system (BLS) for comparisons. The TK approach performs next day classification only considering the closing value in the day (cvd) subtracted the closing value of the previous day (cvd-1). If we have a negative value (cvd < cvd-1) then the output value for the next day is classified as the negative class (“-1”), on the other way around, if we have a zero or positive value (cvd >= cvd-1), then the TK sets the positive class to the target variable (“+1”).

Here, we expect an accuracy rate close to 50%, similar to a simple guess, which is the probability of success of an investor without any knowledge of the financial market.

We use this one as a lower limit to compare the quality of classifiers.

B. Naïve Bayes (NB)

The Naive Bayes algorithm is a probabilistic model based on Bayes rule. Because of its simplicity, this algorithm is widely used in Machine Learning, for both discrete and continuous X. It is a naive approach because it considers the attributes to be conditionally independent, i.e., a given event does not imply another one.

In other words the attributes X_1, \dots, X_n are all conditionally independent given Y and the data has a normal symmetrical distribution. Despite this naive and simplistic premise, it reports good performances in several classification tasks [7].

We can represent Bayes rule as:

$$P(Y = y_i | X = x_k) = \frac{P(X=x_k|Y=y_i) P(Y=y_i)}{\sum_j P(X=x_k|Y=y_j) P(Y=y_j)} \quad (1)$$

Where:

y_i denotes the target value of the i^{th} example

x_k denotes the k^{th} attribute value for an example x

Our implementation assumes that for each attribute x_k in ith example x, we calculate standard-deviation (σ) and the average (μ) for each class, in our study case, “+1” and “-1”.

After this we compare the result of each formula (as defined in Equation 1) for each class. The algorithm evaluate the class (“+1” or “-1”) using the higher output value of the calculation for each class

$$\frac{1}{(\sigma(C) * \sqrt{2\pi})} * e^{-z} \quad (2)$$

Where z is defined by:

$$\frac{((x_k) - \mu(C))^2}{2 * \sigma(C)^2} \quad (3)$$

C. Logistic Regression

Logistic Regression is a function approximation algorithm that uses training data to directly estimate P(Y|X). It is an approach to learn functions of the form: $X \rightarrow Y$.

Roughly, it gives the probability of $y = 1$ as a set of discrete or continuous variables in a vector X. In this implementation we use a gradient descendent function to find the best set of parameters.

As all others algorithms, our output value is labeled as “+1” or “-1”. We can denote an example by \bar{x} and the value of k^{th} feature as x_k . It also defines an additional feature, $x_0 = 1$ (bias feature). The probability of an example being a positive value is given by:

$$p(y = +1 | \bar{x}) = g(\sum_{k=0}^n w_k x_k) \quad (4)$$

$$\text{Where } g(z) = \frac{1}{1+e^{-z}} \quad (5)$$

and $w_k, k \in \{0, \dots, n\}$ denote the weight for the k^{th} feature. In training we get weight vector (W) defined by the gradient descendent method.

D. Support Vector Machine (SVM)

Support Vector Machine (SVM) is nowadays the most promising machine learning algorithms, is based by statistical learning theory [4], developed by study cases started with [5] and establishes a series of principles to be followed in obtaining classifiers with good generalization.

Basically, the algorithm defines some key points to be the support vectors, at first defined by the biggest distance between the linear classifier and the closest class’s examples (labeled as +1 and -1 values). In other words, it defines a margin which is the width that the boundary could be increased by, before achieves a data example. In the experiments sections, we will explain more directly how its variations (kernels and theirs parameters) impacted the prediction result.

We used the library Libsvm [1], with was integrated in FAMA framework to implement our experiments.

Some solutions for solving classifications problems requires that input data must be linearly separated, but we know that is not always possible. To solve this issue, Vapnik proposed a mathematical method to transform low-dimensional data into a high-dimensional projection (using kernel functions), which is easier to separate input data linearly.

The resulting hyperplane is defined maximizing the distance between the “nearest” vectors of different classes. The thinking is that a bigger margin directly implies in the best capacity of generalization. The Figure 1 shows this idea.

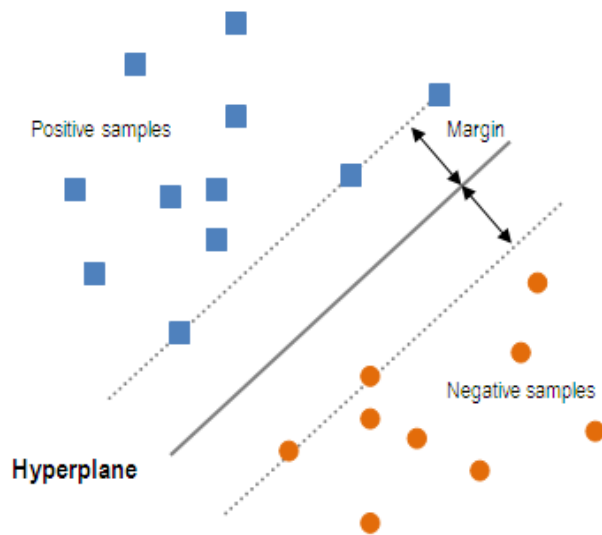


Fig. 1. The simplest kind of SVM: samples and key-points as support vectors (on the dashed line) as maximum margin linear classifier.

The Kernels are functions that help the transformation from low-dimensional feature space into a higher-dimensional feature space, which is a necessary condition for separating the input data values properly.

There are some kernels implementations and we describe the kernels type used into our model analysis. Despite the fact that the LIBSVM library supports several formulations for classification, regression and distribution estimation, we focus our work on the classification models: C-Support Vector Classification [9] and nu-Support Vector Classification [10].

As kernel types, there are many kernels functions, but the common are: linear, polynomial, sigmoid and radial base function.

- Linear: $K(x_i, x_j) = x_i^T x_j$
- Polynomial: $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0$
- Sigmoid: $K(x_i, x_j) = \exp(-\gamma |x_i - x_j|^2), \gamma > 0$
- Radial Base Function: $K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$

where K is called the kernel function and x_i are the training vectors. We test all these in our experiments. The results are showed and detailed in section 4

As we can see, depending on the kernel choice, some parameters (γ, r, d) have to be set.

IV. EXPERIMENTS AND RESULTS

A. Database

We use data from the Bovespa website [8], available through interface files, to create a financial series. The preprocessing task produced a new database contains 107(187-80) records with information between 01-Jan-2006 and 31-Dec-2006-10, since we need to disregard the first 80 values (window’s maximum size) to set the discrepancy values into our generated dataset. In the original dataset, we have 187 days, in order to compute the difference of the 80 day before’s, we need to discard 80 first days.

The preprocessing task must consider some factors such as: outliers’ removal from the sample, attributes’ selection and scaling of the values

B. Input Attributes

From our database, we selected the attributes, according to technical analysis, that are most relevant to the final value of an stock [25]: opening price (open), closing price (close), day maximum (max), day minimum (min) and volume (vol). We labeled these values as “base attributes”.

Moreover we used the series discrepancy (3, 5, 10, 15, 20, 40, 60 or 80 days). The results show the direct impact in model sensibility when we change the window, overfitting the model when using bigger window values.

A good pattern that was found is the combination between these “base attributes” and the series discrepancy values. We realize that when used together, this combination can be good for accuracy since we have a low discrepancy (most of the time when this value is equal 3 or 5). This pattern was valid only for values less than or equal to 5.

As example, from original database values, we produce the following input structure, using a discrepancy value of 3, showed in Table I.

TABLE I. AN SIMPLE EXAMPLE WITH THE FIRST SIX DAY’S VALUES AS INPUT MATRIX FOR 3 DAY’S DISCREPANCY VALUE.

PETR4								
Open	Close	Min	Max	Vol (*10 ⁹)	d-1	d-2	d-3	y
39,1	39,4	39,1	39,7	15,98	-0,28	0,69	2,19	+1
39,5	40,8	39,5	40,8	24,15	1,40	-0,28	0,69	+1
40,6	40,9	40,3	41,3	18,07	0,10	1,40	-0,28	+1
44,1	43,7	43,1	44,4	27,51	2,80	0,10	1,40	-1
43,4	42,8	42,5	43,9	23,86	-0,90	2,80	0,10	+1
42,9	43,4	42,9	43,8	38,53	0,60	-0,90	2,80	-1

C. Outliers: split and inplit

We must consider two relevant aspects in financial series: split and inplit. Both are techniques used as strategy aiming asset price increase.

“Split” is a strategy that companies use in order to improve the liquidity of an asset. The Split technique occurs when the

stock price is too expensive, difficulting the financial transaction, often caused by memory investor [12].

“Inplit” is the inverse operation of the “split”. The inplit is used to enhance the liquidity of the asset when their price is far below the market and reducing the volatility of the asset (when the asset’s value is too low any variation represents a large variation of percentage).

In this work, we must consider that operations (both split and inplit) as a noise in the financial series, since it generates a large variation in the price of the asset, although they do not cause any impact in the investment portfolio’s value

D. Scaling

In machine learning techniques, it is advisable to put all input values into a range of [-1, +1] or [0, +1]. This increase the performance of algorithms (by avoid numerical difficulties during the calculation, for instance kernel functions usually depend on the inner products of feature vectors) and doesn’t privilege some (greater) numeric values. Warren S. Sarle [13] explains the importance of scaling in your research.

Obviously, the same scaling method must be used in both training and testing dataset. For example, if the x attribute of training was scaled from [-100, +100] to [-1, +1] and the same attribute in the test data lies in the range [-120, +80] then the result test dataset must be scaled to [-1.20, + 0.8]

After the data preprocessing task, we perform some analysis of the SVM parameters in order to find a model that presents a good performance with new data.

V. EXPERIMENTS

Among the choice of attributes, we analyze the impact of some parameters and their variations into model’s accuracy. These parameters are described in Table II.

TABLE II. ALGORITHMS PARAMETERS FOR MODEL TRAINING.

Parameter	Observations
svdtype	0 = c-SVC, 1 = v-SVC
kernel type	0 = Linear, 1 = Polynomial, 2 = RBF, 3 = Sigmoid
degree	Set degree in kernel function (polynomial, RBF and Sigmoid)
gamma	Set gamma value in kernel function (polynomial and sigmoid)
coef0	Set coef0 in kernel function
C	Set c parameter for C-SVC
V	Set v parameter for v-SVC

Furthermore, we analyze some specifics details in modeling task as training period, input attributes and cross validation method compared with our sliding cross validation implementation.

One problem yet to be solved is finding a good training period. Given the characteristics of financial time series, we can not train with a very large (subject to underfitting) or a very small dataset (subject to overfitting) [9]. The next Tables (III, IV and V) shows this behavior. We argue the oldest values are less important for closing value day then closer values. When we work with a bigger training dataset, the output model cannot find a generic good model for prediction.

TABLE III. SVM – ACCURACY OF SVM IN DIFFERENT SIZES OF TRAINNING BASES

Stock	Period				
	Window	5 months	10 months	15 months	20 months
ALLL11	80	90%	84%	70%	65%
PETR4	80	89%	80%	62%	61%
ELET6	80	92%	82%	71%	65%
CSNA3	80	95%	82%	72%	58%

^{a1} (SVM Type = nu-SVC, Kernel Type = Linear, nu parameter value = 0.5)

TABLE IV. NB - ACCURACY OF SVM IN DIFFERENT SIZES OF TRAINNING BASES

Stock	Period				
	Window	5 months	10 months	15 months	20 months
ALLL11	80	83%	74%	56%	54%
PETR4	80	85%	83%	76%	58%
ELET6	80	84%	73%	60%	48%
CSNA3	80	91%	77%	76%	58%

TABLE V. LR - ACCURACY OF SVM IN DIFFERENT SIZES OF TRAINNING BASES

Stock	Period				
	Window	5 months	10 months	15 months	20 months
ALLL11	80	55%	49%	49%	48%
PETR4	80	45%	48%	47%	47%
ELET6	80	48%	45%	49%	49%
CSNA3	80	56%	50%	45%	45%

From technical analysis, we used basic values as: opening value, closing value, highest and lower value in the day and also volume. Dow’s theory argues that these variables can be used to predict the market movement [26]. We cannot take this statement into our model for all assets. Basically we found a pattern in some assets, when we use a small discrepancy (3, 5) we can see little improvement in accuracy but, when this value grows, in most cases, this improvement is lost. Furthermore, there are some assets where this affirmative is not true, such as ELET6 asset.

In Table VI and Table VII, we show the accuracy of SVM and NB training considering discrepancy values (D) and considering both base attributes plus discrepancy values (BA+D). For ALLL11 and also CSNA3 we cannot see any (real) improvement into hit rate when se use both input groups. For PETR4 in some window size values we have a considered gain. For ELET6 the combination of base attributes and discrepancy values helps in the prediction task for any window size value when we compare SVM results. In Table VIII, we show the results of the cross validation training with 80% of data and testing with 20%. Table IX show the cross-validation results for NB.

TABLE VI. ACCURACY OF SVM MODEL IN TRAINING DATABASE FOR DIFFERENT WINDOW SIZE AND DIFFERENT INPUT PARAMETERS

Window	ALLI1		PETR4		CSNA3		ELET6	
	D	BA+ D	D	BA+ D	D	BA+ D	D	BA+ D
3	54%	53%	54%	56%	58%	47%	54%	61%
5	57%	57%	55%	52%	48%	53%	54%	57%
10	59%	42%	63%	46%	58%	56%	53%	59%
15	58%	60%	61%	50%	66%	62%	58%	69%
20	63%	57%	68%	52%	71%	57%	68%	72%
40	76%	75%	79%	86%	83%	84%	72%	81%
60	85%	83%	86%	84%	87%	87%	88%	89%
80	90%	90%	89%	89%	94%	95%	90%	92%

5 months series length (SVM Type = nu-SVC, Kernel Type = Linear, nu parameter value = 0.5) (SVM Type = nu-SVC, Kernel Type = Linear, nu parameter value = 0.5)

TABLE VII. ACCURACY OF NB MODEL IN TRAINING DATABASE FOR DIFFERENT WINDOW SIZE AND DIFFERENT INPUT PARAMETERS

Window	ALLI1		PETR4		CSNA3		ELET6	
	D	BA+ D	D	BA+ D	D	BA+ D	D	BA+ D
3	63%	62%	55%	64%	57%	59%	54%	65%
5	62%	61%	62%	68%	61%	62%	54%	64%
10	60%	61%	64%	66%	68%	70%	57%	59%
15	66%	66%	69%	67%	74%	72%	65%	64%
20	69%	68%	72%	70%	75%	71%	64%	60%
40	71%	72%	76%	77%	81%	83%	66%	61%
60	77%	77%	82%	80%	86%	85%	78%	76%
80	83%	83%	86%	85%	91%	91%	85%	84%

5 months series length

Bezzera da Silva et. Al. [11] studies the correlation between stocks in Bovespa with market graph by Power Law.

Despite the difference in the research focus, we can see the relation between stocks and if we can cluster these different groups, specifics model rules can be applied to get best model as possible.

Analyzing the variations of kernels, we can see the best values for accuracy by using the Polynomial Kernel and window variable value set to 3 and 5 (for ELET6). This test is carried out with the nu value of 0.5 [2].

TABLE VIII. COMPARING SVM CROSS-VALIDATION (80/20) AND TRAINING ACCURACY (%)

Window	ALLI1		PETR4		CSNA3		ELET6	
	Train	80/20	Train	80/20	Train	80/20	Train	80/20
3	63	64	63	64	54	59	55	64
5	82	68	81	68	80	64	80	59
10	89	64	94	64	94	59	95	59
15	92	64	96	50	95	59	95	59
20	96	64	100	50	97	59	100	59
40	100	64	100	59	100	59	100	59
60	100	64	100	59	100	59	100	59
80	100	68	100	59	100	59	100	59

5 months series length (SVM Type = nu-SVC, Kernel Type = Linear, nu parameter value = 0.5) (SVM Type = nu-SVC, Kernel Type = Linear, nu parameter value = 0.5)

TABLE IX. COMPARING NB CROSS-VALIDATION (80/20) AND TRAINING ACCURACY (%)

Window	ALLI1		PETR4		CSNA3		ELET6	
	Train	80/20	Train	80/20	Train	80/20	Train	80/20
3	63	59	55	64	57	45	54	40
5	62	36	62	68	61	40	54	36
10	60	54	64	64	68	60	57	40
15	66	54	69	50	74	60	65	59
20	69	54	72	68	75	54	64	40
40	71	54	76	59	81	45	66	27
60	77	63	82	63	86	27	78	13
80	83	54	86	59	91	54	85	22

In financial series models, it is important to retrain the model after a certain period in attempt to get the actual tendency. We recognize the importance of historical data, of course, but argue that only a specific time period is really important in order to make a correct prediction. We can prove this affirmative comparing standard cross-validation method with 80% of total data used to train and get the classifier and 20% to test the accuracy of the model and sliding cross-validation method. This one was created using the same parameters used with traditional cross-validation but being retrained before predict next day tendency.

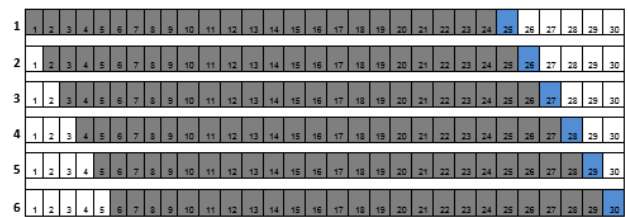


Fig. 2. Sliding Cross-Validation example with 30 days dataset. Train days = 24. Test days = 6

In our tests we have used 80 days to train and predict next day tendency (high or low). With “sliding” validation, we get the average of the output values to calculate model’s accuracy. This parameter is still open in our study and presents best performance with SVM approach.

Table X shows the difference in accuracy between standard cross-validation (C1) and sliding cross-validation (C2) which strongly indicates the need of retraining the model with 80-days prior.

TABLE X. COMPARISON BETWEEN CROSS-VALIDATION AND SLIDING CROSS-VALIDATION METHODS

Window	ALLI1		PETR4		CSNA3		ELET6	
	C1	C2	C1	C2	C1	C2	C1	C2
3	64%	65%	64%	71%	59%	64%	64%	68%
5	68%	71%	68%	71%	64%	65%	59%	68%
10	64%	65%	64%	64%	59%	64%	59%	76%
15	64%	62%	50%	64%	59%	64%	59%	68%
20	64%	65%	50%	64%	59%	64%	59%	68%
40	64%	71%	59%	64%	59%	64%	59%	68%
60	64%	76%	59%	64%	59%	64%	59%	68%
80	68%	65%	59%	64%	59%	64%	59%	68%

As we can see our method sliding cross-validation performs best when compared with cross-validation method. This shows the need to retrain the model after a certain time period, likely looking for the tendency period

VI. CONCLUSION AND FUTURE WORK

As expected, the SVM algorithm had a better accuracy than the other algorithms studied in this work and it also presented good generalization abilities. It can be noticed that the parameter adjustment using kernel functions and the defined margin, especially regarding the implementation, directly impacted the outcome of the model. Despite all difficulties found in the financial time series, such as noise and uncertainties, after adjustment of the data, we obtained good results to serve as a basis for decision making. Another important factor is the period considered for training the model, which does not produce good results in cross validation when it is too small or too large. The approach to validation of the model followed the method of the experiment. The retraining presented with our sliding cross validation method provides the best results compared with cross validation method. It highlighted the need to retrain the model after a certain period.

In attempt to do better predictions, some factors will be considered in future works. The **moving averages** (simple, weighted, exponential and others), are often used in technical analysis as input parameters in the model to indicate an uptrend or downtrend (through lines of support and resistance). It can be a good factor to the TABLE model once we have difference in the behavior of the predictor variable in a downward trend and upward trend.

We also can look for the relation between stocks. Recent works use graphs to **group stocks** through its correlation [11]. This grouping can bring benefits to the data analysis, since it is expected that correlated assets by similarity in behavior.

We argue that the last days have more influence on the price's behavior of the stock and it can be proved that by sliding validation method that considers only 80-days to train and produce a better model. Next steps can consider analyze of the variation of this variable in accuracy results as well as **calculate the input variables by weights**. We also will consider a **hybrid model** created by the analysis of confusion matrix. Finally the prediction target can be reformulated to transform the problem to multiclass. A **sensibility factor** can separate the samples into 3 classes as "high negative variation", "neutral variation" or "high positive variation" by calculates the variation of price. This can put the focus on more specific situations.

On the other hand, recent works have focused in **semantic observation** [27],[28],[29]. Rules can be extracted from the database and applied to the model in an attempt to find a pattern that minimizes the prediction error.

REFERENCES

- [1] Chang, C. C., & Lin, C. J. (2007). LIBSVM: a Library for Support Vector Machines. Available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [2] Hsu, C. W., Chang, C. C., & Lin, C. J., (2003). A Practical Guide to Support Vector Classification. Available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
- [3] Tom M. Mitchell. Machine Learning. McGraw-Hill, New York, 1997.
- [4] V. N. Vapnik. The nature of Statistical learning theory. Springer-Verlag, New York, 1995.
- [5] V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. Theory of Probability and its Applications, 16(2):283–305, 1971
- [6] FAMA: Framework para Aprendizizado de Máquina. Available at <https://code.google.com/p/fama>
- [7] A. McCallum and K. Nigam, A comparison of event models for Naive bayes text classification, In Proc. of the AAAI-98 Workshop on learning for text categorization, pp. 41-48
- [8] Bovespa Historical Data Series available at: <http://www.bmfbovespa.com.br/shared/iframe.aspx?idioma=ptbr&url=http://www.mfbovespa.com.br/pt-br/cotacoes/historicas/FormSeriesHistoricas.asp>
- [9] Cortes and V. Vapnik. Support-vector network. Machine Learning, 20:273{297,1995.
- [10] Scholkopf, A. Smola, R. C. Williamson, and P. L. Bartlett. New support vector algorithms. Neural Computation, 12:1207{1245, 2000.
- [11] Bezerra da Silva, Eduardo, de Lima, Leonardo Silva, Attias, Thiago. "Uma Aplicação da teoria de grafos em redes complexas: estudo de caso do mercado de ações da Bovespa", Rio de Janeiro, CEFET, 2009.
- [12] Atsalakis, G. S.; Valavanis, K. P.. Surveying Stock Market Forecasting Techniques – Part II: Soft Computing Methods. Expert Systems with Applications, 36, 3, 5932-5941, 2009.
- [13] Leonardo C. Martinez, Diego N. da Hora, J. R. M Palotti, Gisele L. Pappa, and W. Meira Jr. From na artificial neural network to a stock market day-trading system: A case study on the bm&f bovespa. The 2009 International Joint Conference on Neural Networks, 2009.
- [14] Yaser S. Abu-Mostafa and Amir F. Atiya. Introduction to financial forecasting. Appl Intell., 6(3):205-213, 1996.
- [15] R. Lawrence. Using Neural Networks to Forecast Stock Market Prices. University Of Manitoba, 1997.
- [16] Bruce Vanstone and Gavin Finnie. An empirical methodology for developing stockmarket trading systems using artificial neural networks. Expert Syst. Appl., 36(3):6668-6680, 2009.
- [17] Arun U., Gautam B. and Avijan D. Forecasting Stock Performance in Indian Market using Multinomial Logistic Regression. Journal of Business Studies Quarterly, 2012.
- [18] Dutta, A. et al. (2008). "Classification and Prediction of Stock Performance using Logistic Regression: An Empirical Examination from Indian Stock Market: Redefining Business Horizons". McMillan Advanced Research Series. p.46-62.
- [19] Ajith Abraham, Baikunth Nath and P. K. Mahanti. Hybrid intelligent systems for stock market analysis. In International Conference on Computational Science (2), pages 337-345, 2001.
- [20] Pei-Chann Chang, Chen-Hao Liu, Jun-Lin Lin, Chin-Yuan Fan, and Celeste S. P. Ng. A neural network with a case based dynamic window for stock trading prediction. Expert Syst. Appl, 36(3):6889-6898, 2009.
- [21] Leandro G. M. Alvim; Ruy L. Milidiú; Cícero N. dos Santos. Daily volume forecasting using high-frequency predictors. In IASTED International Conference on Artificial Intelligence and Applications, AIA 2010, Innsbruck, Austria, 2 2010. IASTED.
- [22] Rohit Choudhry and Kumkum Garg. A hybrid machine learning system for stock market forecasting. In WASET -Proceedings of World Academy of Science, Engineering and Technology ISSN: , Commenced: 2005, January, volume 39, Copenhagen, DK, 2005
- [23] Wei Huang, Yoshiteru Nakamori, and Shou-Yang Wang. Forecasting stock market movement direction with support vector machine. Comput. Oper. Res., 32(10):2513–2522, 2005
- [24] Ling-Bing Tang, Ling-Xiao Tang, and Huan-Ye Sheng. Forecasting volatility based on wavelet support vector machine. Expert Syst. Appl., 36(2):2901–2909, 2009.
- [25] Lemos, Flávio; Cardoso, Flávio; Cardoso, Celso. "Análise Técnica Clássica - Com as Mais Recentes Estratégias da Expo Trader Brasil". Editora Saraiva, 2010.
- [26] Robert Rhea. "The Dow Theory (Fraser Publishing Library)". Barron's The National Financial Weekly; 8th Printing edition, 1932.

- [27] Kumar, B. Rama; Kumar, S. Bangari; Prasad S. Chandragiri. "Financial News Classification Using SVM". *International Journal of Scientific and Research Publications*, Volume 2, Issue 3, March 2012
- [28] Alencar, P. Franco; Ávila, M. Guilherme; "Mineração Web - Predição do comportamento de ativos na Bovespa", Brasília, Universidade de Brasília, 2011.
- [29] Lopes, Jefferson P. Thomas; Hiratani, Gabriel Koji Lemos; Barth, Fabricio J.;Rodrigues Jr., Orlando; Pinto, Juliana Maracci. "Mineração de Opiniões aplicada à Análise de Investimentos", São Paulo, Centro Universitário SENAC, 2008.
- [30] Joachims, T. (2002). *Learning to Classify Text using Support Vector Machines: Methods, Theory, e Algorithms*. Kluwer International Series in Engineering and Computer Science. Kluwer Academic Publishers, Norwell, MA
- [31] Noble, W. S. (2004). *Kernel Methods in Computational Biology*, chapter Support vector machine applications in computational biology, pages 71–92. MIT Press, Cambridge, MA