The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.52 e-ISSN: 2251-7545

Stale Model Detection for Algorithmic Trading

Objective measurements for machine learning based trading models evaluation

Carlos Daniel Nocito
Department of Electrical and Computer Engineering
University of Miami
Miami, Fl
c.nocito@umiami.edu

Miroslav Kubat
Department of Electrical and Computer Engineering
University of Miami
Miami, Fl
mkubat@miami.edu

***Abstract*—The use of data mining and adaptive learning is a very controversial issue among the algorithmic trading community in the financial world. The reason for the mistrust in the techniques arrives from some very well-known problems: overfitting to training data, concept drift, and insufficient support for the derived models. In this paper, we present a new element to the use of some classic data mining and adaptive learning techniques to algorithmic trading: a set of objective distance measurements that track the similarity between the prediction model and the actual system. We use historical market data to develop and test an algorithm, and investigate the correlation between prediction accuracy of the models and the distance measurements. We find that this tracking could allow investors to discard stale models earlier, thus decreasing losses.**

Keywords—data mining; machine learning; algorithmic trading; decision trees; accuracy tracking; Jensen-Shannon; Kullback-Leibler;

I. Introduction

Algorithmic trading refers to the use of electronic platforms for executing investment strategies or algorithms in financial markets. It is widely used by investment banks, pension funds, mutual funds, and other investor driven institutions. As much as 80% of the volume in American exchanges originated from such trading platformsin 2008 [1]. A significant percentage of the algorithmic trading volume is originated by "black boxes", which are sophisticated event driven systems that automatically manage portfolios based on mathematical models.

Financial market scholars have investigated the behavior and nature of the system at hand for many years. From an engineer's perspective, financial markets are non-linear, time variant, multiple input/single output (MISO) systems. Figure 1 shows the price evolution of SPY, which is an instrument that tracks the S&P 500 index. As we can see in the figure, there are well defined trends in the evolution of the price of that symbol. In the most simplistic case, the task of a black box algorithm is to maximize the returns on the initial capital by buying such instruments at local minima and selling them at local maxima [2]. Because of the competitive nature of the markets, the actual algorithms ran in black boxes are usually kept a secret from the general public.Nonetheless, because of the above mentioned nature of the signal, it makes sense to think that data mining or machine learning algorithms could produce adequate mappings from the input domain (historical price data, market news, etc.) to the predicted co-domain (future price of the instrument, or overall direction of the price).

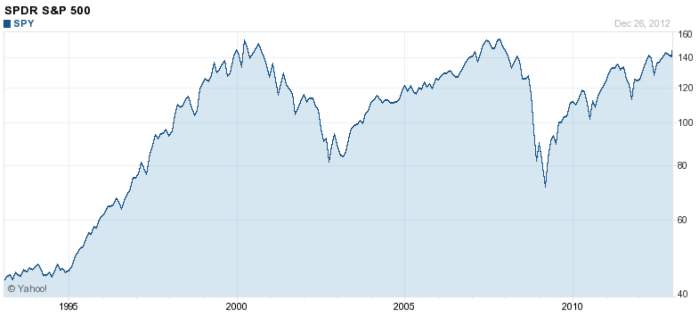


Figure 1: Evolution of the price of SPY

In our ongoing work, we treat market direction prediction as a classification task. The feature space is composed of historical market data, which is a collection of past prices of the symbols we are trying to track, and commonly used derived indicators, such as simple moving averages, exponential moving averages, volume of shares traded per time unit, and the deltas there of, among others. Further more, it is well known that there are strong correlations between some symbols [3], so historical market data and indicators for multiple symbols are used for the classification task. Finally, because of the affinities of the authors to the techniques, decision trees are used to find a mapping from the feature space to the desired class labels ("buy", "sell", "hold", or "trade", "don't trade", for example).

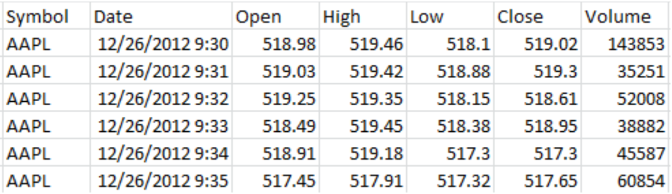
In this paper, we investigate the use of some objective measures of the difference between the probability disbributions observed in the output of the real system (the financial markets) and our classifier. In section II we provide a brief description of how we developed the classifier used for the different experiments. In section III we define the three distance measurements considered, namely geometric distance, Kullback-Leibler diveregence andJensen-Shannon divergence.

In section IV we evaluate the results obtained with the different methods when compared to the realative return of the algorithm over the given widows, and examinethe correlation between the different methods and the classifier's prediction accuracy, which is the objective performance measurement we chose for this work. Finally, in section V we present our conclusions.

II. Model Development

A. Data Pre-Processing

Market data feeds are multivariate in nature, because the number and frequency of trades varies constantly. For this reason, it is a very common practice to summarize the data into minute bars. Figure 2 shows 5 sample bars of historical market data obtained from the commercial service End of Day Data (www.eoddata.com).



Symbol	Date	Open	High	Low	Close	Volume
AAPL	12/26/2012 9:30	518.98	519.46	518.1	519.02	143853
AAPL	12/26/2012 9:31	519.03	519.42	518.88	519.3	35251
AAPL	12/26/2012 9:32	519.25	519.35	518.15	518.61	52008
AAPL	12/26/2012 9:33	518.49	519.45	518.38	518.95	38882
AAPL	12/26/2012 9:34	518.91	519.18	517.3	517.3	45587
AAPL	12/26/2012 9:35	517.45	517.91	517.32	517.65	60854

Figure 2: 5 bars of historical market data

The symbol column contains the name of the symbol for the given row. The Date column specifies the time period described by the row, for example 12/26/2012 9:30:00 to 12/26/2012 9:30:59 in the first row of the figure. The Open column contains the price of the symbol at the beginning of the time interval, as dicated by the last sucessful trade. The High and Low columns represent the highest and lowest prices at which the symbol was traded during the time interval. The Close column represents the price of the symbol at the last moment of the time interval, again as dictated by the last sucesful trade. Finally, the Volume column represents the number of shares traded during the described time interval. The actual training set contains additional columns in three main categories:

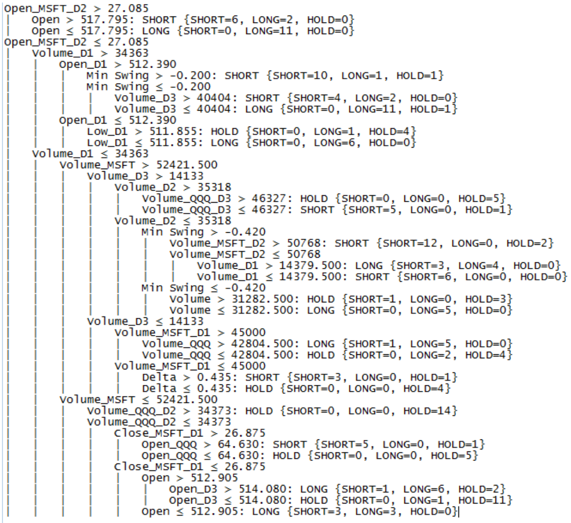
1. 1, 2 and 3 time interval deltas, so temporal information can be captured by the decision tree. In particular, these deltas are captured as percentage increase or decrease over the current time interval
2. Same information for two related symbols. In this case, we aim to predict the price movement of AAPL (Apple Inc.). We include information about QQQ (PowerShares QQQ, a stock basket that tracks theNASDAQ 100 index) and MSFT (Microsoft Corporation, a direct competitor of Apple Inc.)
3. Because we are using decision tress for the classifier, a supervised learning algorithm, we create a class label as follows:

$$L(x_i) = \begin{cases} abs(close_{i+1} - open_{i+1}) \leq open_{i+1} \cdot 1.001, "don't" \\ close_{i+1} - open_{i+1} > open_{i+1} \cdot 1.001, "long" \\ close_{i+1} - open_{i+1} < open_{i+1} \cdot 1.001, "short" \end{cases} \quad (1)$$

wherei indicates the time interval to be populated. Clearly this is a very arbitrary class label, but the idea is to trade only when there is an immediate opportunity to make a 0.1% or better profit on each trade.

B. Classifier Training

There are many papers on the use of data mining and decision trees for stock prices prediction. Nair et al. [4], Wang et al [5] and Ochotorena et al [6] also apply machine learning to the development of stock price predictors. In our work for this paper, several C4.5 decision trees were developed and compared in terms of accuracy (as defined by percentage agreement between the classifier and the ground truth in evaluation data), precision, recall and profitability of the algorithm over a 24 hour period. A simple search algorithm was used to optimize the training parameters, and the resulting trees were evaluated using 5-fold testing. Figure 3 shows the model classifier used for these experiments and Table Ishows the key performance metricsobtained by the model. The overall accuracy of the model is 42.49%, which is 9.16% higher than random (33.33% in this 3 label class-balanced system). It is important to note that even though the training and evaluation systems are class balanced, the experiments conducted in Section IV are conducted on the unprocessed unbalanced data.



```
Open_MSFT_D2 > 27.085
|   Open > 517.795: SHORT {SHORT=6, LONG=2, HOLD=0}
|   Open ≤ 517.795: LONG {SHORT=0, LONG=11, HOLD=0}
Open_MSFT_D2 ≤ 27.085
|   Volume_D1 > 34363
|   |   Open_D1 > 512.390
|   |   |   Min Swing > -0.200: SHORT {SHORT=10, LONG=1, HOLD=1}
|   |   |   Min Swing ≤ -0.200
|   |   |   |   Volume_D3 > 40404: SHORT {SHORT=4, LONG=2, HOLD=0}
|   |   |   |   Volume_D3 ≤ 40404: LONG {SHORT=0, LONG=11, HOLD=1}
|   |   Open_D1 ≤ 512.390
|   |   |   Low_D1 > 511.855: HOLD {SHORT=0, LONG=1, HOLD=4}
|   |   |   Low_D1 ≤ 511.855: LONG {SHORT=0, LONG=6, HOLD=0}
|   Volume_D1 ≤ 34363
|   |   Volume_MSFT > 52421.500
|   |   |   Volume_D3 > 14133
|   |   |   |   Volume_D2 > 35318
|   |   |   |   |   Volume_QQQ_D3 > 46327: HOLD {SHORT=0, LONG=0, HOLD=5}
|   |   |   |   |   Volume_QQQ_D3 ≤ 46327: SHORT {SHORT=5, LONG=0, HOLD=1}
|   |   |   |   Volume_D2 ≤ 35318
|   |   |   |   |   Min Swing > -0.420
|   |   |   |   |   |   Volume_MSFT_D2 > 50768: SHORT {SHORT=12, LONG=0, HOLD=2}
|   |   |   |   |   |   Volume_MSFT_D2 ≤ 50768
|   |   |   |   |   |   |   Volume_D1 > 14379.500: LONG {SHORT=3, LONG=4, HOLD=0}
|   |   |   |   |   |   |   Volume_D1 ≤ 14379.500: SHORT {SHORT=6, LONG=0, HOLD=0}
|   |   |   |   |   Min Swing ≤ -0.420
|   |   |   |   |   |   Volume > 31282.500: HOLD {SHORT=1, LONG=0, HOLD=3}
|   |   |   |   |   |   Volume ≤ 31282.500: LONG {SHORT=0, LONG=5, HOLD=0}
|   |   |   Volume_D3 ≤ 14133
|   |   |   |   Volume_MSFT_D1 > 45000
|   |   |   |   |   Volume_QQQ > 42804.500: LONG {SHORT=1, LONG=5, HOLD=0}
|   |   |   |   |   Volume_QQQ ≤ 42804.500: HOLD {SHORT=0, LONG=2, HOLD=4}
|   |   |   |   Volume_MSFT_D1 ≤ 45000
|   |   |   |   |   Delta > 0.435: SHORT {SHORT=3, LONG=0, HOLD=1}
|   |   |   |   |   Delta ≤ 0.435: HOLD {SHORT=0, LONG=0, HOLD=4}
|   |   Volume_MSFT ≤ 52421.500
|   |   |   Volume_QQQ_D2 > 34373: HOLD {SHORT=0, LONG=0, HOLD=14}
|   |   |   Volume_QQQ_D2 ≤ 34373
|   |   |   |   Close_MSFT_D1 > 26.875
|   |   |   |   |   Open_QQQ > 64.630: SHORT {SHORT=5, LONG=0, HOLD=1}
|   |   |   |   |   Open_QQQ ≤ 64.630: HOLD {SHORT=0, LONG=0, HOLD=5}
|   |   |   |   Close_MSFT_D1 ≤ 26.875
|   |   |   |   |   Open > 512.905
|   |   |   |   |   |   Open_D3 > 514.080: LONG {SHORT=1, LONG=6, HOLD=2}
|   |   |   |   |   |   Open_D3 ≤ 514.080: HOLD {SHORT=0, LONG=1, HOLD=11}
|   |   |   |   |   Open ≤ 512.905: LONG {SHORT=3, LONG=3, HOLD=0}
```

Figure 3: Developed classifier

Table I. Performance of classifier used for the experiments.

Class Label	Precision	Recall
Short	42.37%	41.67%
Long	43.55%	45.00%
Don't Trade	41.38%	40.68%

Throughout the paper we try to keep the contents and attention away from the specifics of the equities trading domain, but it is worth mentioning that the main objective indicators of performance in this domain are derived from the profit or loss attained by the system, how it outperforms simply buying the stock and holding it for the same time period, and the maximum withdraw, which is a measure of the lowest decline of portfolio value in the period evaluated. Table II shows this metrics for our classifier.

Table II. Financial performance of the classifier used for the experiments.

Metric	Value
Buy and hold return (baseline)	-1.17%
System profit (+) or loss (-)	5.75%
Maximum withdraw	0.73%

III. Distance Measurements

The overall theme of our research is to develop tools that close the existing gap between machine learning and algorithmic trading. In the previous section we introduced a classifier that is very representative of the framework we use to develop our machine learning based trading algorithms, but as mentioned in the introduction, issues such as overfitting, concept drift and lack of proper support generate reluctance in the financial community to adopt such classifiers. In this section, we evaluate the use of three different measurements to evaluate the relevance of the model in use on an ongoing basis (whether or not the model became "stale").

A. *Geometric distance between the probability distributions*

We purposely chose a 3 class label classifier over a binary class label classifier, so that the concepts make more intuitive sense. The time series we are working with are of the generic form $x_0, x_1, \ldots, x_N$, where N is the number of time intervals. We apply a transformation $l_{i+1} = L(x_i)$ to a sequence of feature vectors X as shown in (1),so we can add a class label that can be used for the supervised learning process. We use machine learning to derive a predictor $\hat{F}(x_i)$which generates a predicted value of l, $\hat{l}$. The possible values of l and $\hat{l}$ are $\langle$"long","short","don't trade"$\rangle$. If we additionally define a moving window T we can approximate the probability distributions of l and $\hat{l}$ as:

$$\varphi_l(y) = \frac{count(y\ in\ observed\ values\ of\ \ l\ in\ T)}{|T|} \qquad (2)$$

$$\varphi_{\hat{l}}(y) = \frac{count(y\ in\ preficted\ values\ of\ \hat{l}\ in\ T)}{|T|} \qquad (3)$$

Equation (2) represents the PDF of the actual system, and (3) represents the PDF of the predicted system. The objective of this work is to find distance or divergence measurements between those functions that correlate with the performance of our classifier. The first and simplest distance to be considered will be the geometric distance defined as:

$$d^T(\varphi_l, \varphi_{\hat{l}}) = \sqrt{\sum_{j\ outcomes}\left(\varphi_l\left(y_j\right) - \varphi_{\hat{l}}\left(y_j\right)\right)^2} \qquad (4)$$

which is nothing more than the canonical norm in a subspace generated by the probabilities of the different outcomes.

B. *Kullback-Leibler divergence*

As it is known, the problem of tracking the probability distribution of a multivariate data streamsis very common in communication systems.Kuncheva [7] justifies Kullback-Leibler as a valid measure to detect deviance from probabilistic distributions in multivariate systems. The Kullback-Leibler divergence, also known as information gain, is commonly used in information theory to quantify the information lost when approximating a probability distribution P with a probability distribution Q. It is also very commonly used in machine learning and data mining to select the most relevant features and the optimal splits for numerical attributes in supervised learning. For discrete probability distribution functions, like $\varphi_l(x)$and $\varphi_{\hat{l}}(x_j)$, it is defined as:

$$D_{KL}^T(P||Q) = \sum_i ln\left(\frac{P(i)}{Q(i)}\right)P(i) \qquad (5)$$

Even though information gain was used to create the classifier, the application here is different: we are measuring the divergence between the observed distribution $\varphi_l(y)$ and the predicted distribution $\varphi_{\hat{l}}(y)$ over the sliding window T, in order to measure the degradation of the model during different windows.

C. *Jensen-Shannon divergence*

A refined version of the Kullback-Leibler divergence is the Jensen-Shannon divergence. The Jensen-Shannon divergence measure has the great advantage of being bounded between 0 and ln(2). It is defined as:

$$D_{SJ}^T(P||Q) = \frac{1}{2}D_{KL}^T(P||M) + \frac{1}{2}D_{KL}^T(Q||M) \qquad (6)$$

where

$$M = \frac{1}{2}(P + Q) \quad (7)$$

In this work we investigate the correlation between these three different metrics and both the classification accuracy and the return of the trading algorithm.

IV. Evaluation of Performance

We choose as the first baseline of comparison the relative return of the algorithm over a given window T, because it is the basis of most evaluation criteria in this specific domain. We normalized the return to 0 for the biggest loss and 1 for the maximum profit obtained in any of the studied windows. Figure 4 illustrates the correlation between the window accuracies and the relative returns.

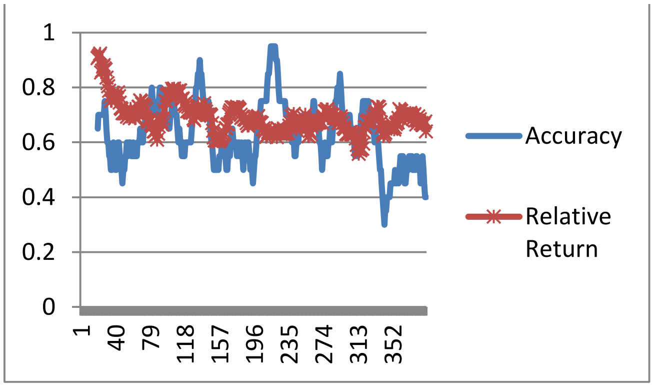


Figure 4: Window accuracies and relative return

As seen in figure 4, there seems to be very little correlation between the predictor's accuracy and the relative rate of return, which is to be expected among other reasons because we quantified the outcomes in 3 class labels, which doesn't account for the magnitude of the return on the winning trades or the losses on the losing trades. In essence, it seems like decisions based on this indicator will lack statistical support, may be unrelated to the actual performance of the classifier, and provide very little insight about corrective actions. Figures 5, 6 and 7 show the accuracy compared to the geometric distance, the Kullback-Leibler divergence and the Jensen-Shannon divergence respectively.

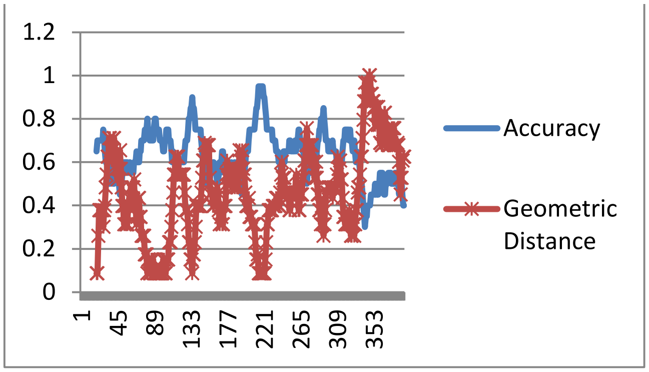


Figure 5: Window accuracies and geometric distances

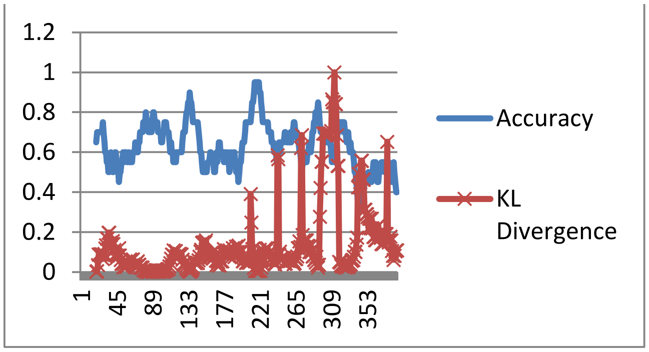


Figure 6: Window accuracies and Kullback-Leibler divergences

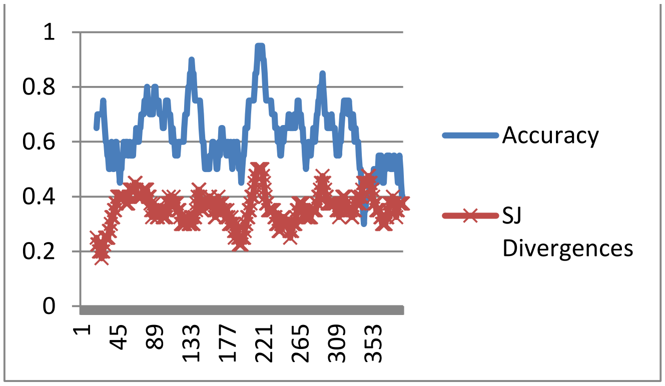


Figure 7: Window accuracies and Shannon-Jensen divergences

As seen in the figures, there are much stronger reverse correlations between the 3 metrics studied in this work and the predictor's accuracy. Table III summarizes the results, and clearly shows that the best candidate as per the experiments conducted is the geometric distance. A large negative correlation indicates that a degradation in the model fit to the observed system, as measured by that metric, correlates to decreased accuracy. A correlation of close to 0, like the one seen in the relative return, indicates that the prediction accuracy of the model is irrelevant to the return rate, so it provides no analytical support to any decisions made about the model.

Table III. Comparison of correlations

Metric	Correlation
Relative Return	0.0061
Geometric distance	-0.8511
Kullback-Leibler	-0.3200
Jensen-Shannon	-0.3434

V. Conclusion

Underlying the challenges mentioned in Section I to the application of machine learning and data mining techniques to the generation of trading algorithms, resides a fear to mistaking a good model with a “lucky” model, which may at anytime deviate very significantly from the expected behavior and thus generate unexpected losses. In Figure 4 we see that if only relative return is used as an indicator of performance, the extremely low correlation translates into a useless or at best much lagged track of performance in terms of the classifier’s behavior.

We present in this work three alternative metrics that can be used for tracking the models fit to the observed system. The geometric distance shows a very strong reverse correlation to the accuracy of the classifier, and seems to be the best indicator of performance. Both the Kullback-Leibler and Jensen-Shannon divergences show observable reverse correlation to the classifiers accuracy as well, the second being only marginally better due to the better numerical properties.

The overall theme of our work is to adapt machine learning and data mining techniques to the algorithmic trading domain. By using more rigorous evaluation of the derived models and how they match the tracked system, as it was done in this paper, investors can use such algorithms with an aggregated level of confidence, and can make more informed decisions about the real risk they are taking. This in term could lead to better portfolio management.

References

[1] Wikipedia contributors, "Algorithmic trading," *Wikipedia, The Free Encyclopedia,* http://en.wikipedia.org/w/index.php?title=Algorithmic_trading&oldid=530005242 (accessed December 27, 2012).

[2] Luis Torgo, “Predicting Stock Market Returns,” in Data Mining with R, 1st ed. Boca Raton, Florida: CRC Press, 2011, ch. III, pp. 95–163

[3] Wikipedia contributors, "Pairs trade," *Wikipedia, The Free Encyclopedia,* http://en.wikipedia.org/w/index.php?title=Pairs_trade&oldid=526503684 (accessed December 27, 2012).

[4] Nair, B.B.; Dharini, N.M.; Mohandas, V.P.; , "A Stock Market Trend Prediction System Using a Hybrid Decision Tree-Neuro-Fuzzy System," *Advances in Recent Technologies in Communication and Computing (ARTCom), 2010 International Conference on* , vol., no., pp.381-385, 16-17 Oct. 2010 doi: 10.1109/ARTCom.2010.75

[5] Huacheng Wang; Yanxia Jiang; Hui Wang; , "Stock return prediction based on Bagging-decision tree," *Grey Systems and Intelligent Services, 2009. GSIS 2009. IEEE International Conference on* , vol., no., pp.1575-1580, 10-12 Nov. 2009

[6] Ochotorena, C.N.; Yap, C.A.; Dadios, E.; Sybingco, E.; , "Robust stock trading using fuzzy decision trees," *Computational Intelligence for Financial Engineering & Economics (CIFEr), 2012 IEEE Conference on* , vol., no., pp.1-8, 29-30 March 2012

[7] Kuncheva, L.; , "Change Detection in Streaming Multivariate Data Using Likelihood Detectors," *Knowledge and Data Engineering, IEEE Transactions on* , vol.PP, no.99, pp.1, 0 doi: 10.1109/TKDE.2011.226