

Composite Pattern Matching in Time Series

Asif Salekin,^{1*} Md. Mustafizur Rahman,¹ and Raihanul Islam¹

¹Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology
Dhaka-1000, Bangladesh

*asalekin@gmail.com

Abstract—For last few years many research have been taken place to recognize various meaningful patterns from time series data. These researches are based on recognizing basic time series patterns. Most of these works used template based, rule based and neural network based techniques to recognize basic patterns. But in time series there exist many composite patterns comprise of simple basic patterns. In this paper we propose two novel approaches of recognizing composite patterns from time series data. In our proposed approach we use combination of template based and rule based approaches and neural network and rule based approaches to recognize these composite patterns.

Index Terms—Pattern recognition, Rule based approach, Neural network, Template based approach, Composite pattern

I. INTRODUCTION

There is a wide range of applications in almost every domain where time series data is being generated. For example, daily fluctuations of the stock market, traces produced by a computer cluster, medical and biological experimental observations, readings obtained from sensor networks, position updates of moving objects in location-based services, etc, are all represented in time series. Consequently, there is an enormous interest in analysing (including query processing and mining) time series data, which has resulted in a large number of works on new methodologies for indexing, classifying, clustering, and summarizing time series data. Stock market price fluctuation also generates time series data. Technical analysts and traders claim certain stock market time series pattern and shapes lead to a profitable trade opportunity.

For last few years many research have been conducted to recognize meaningful patterns from time series data. We find that many composite patterns which frequently occurred in the time series can be formed by simply combining simple basic time series patterns. As far we are concerned that the current states of the art are not yet concerned about this emerging composite. In this paper we have worked with six basic patterns illustrated in Fig-1 and the composite patterns generated from these six basic patterns.

Scientists have been used template based approaches,

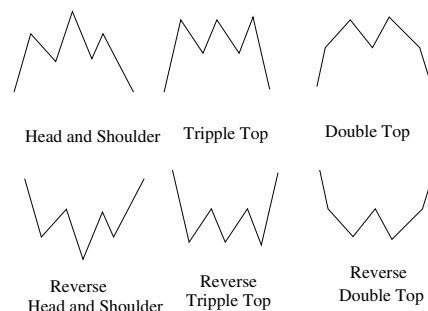


Fig. 1. Six basic patterns for stock data.

rule based approaches and neural network based approaches to recognize patterns from time series data. In this paper we have proposed two novel approaches to recognize composite pattern from time series data. We apply variable size sliding window on time series data and extract features from it. In our first approach we use Spearmans rank correlation coefficient to recognize preferred patterns and use rule sets to provide more accuracy in classification process. In second approach we use neural network as classifier of various preferred patterns in time series and also use rule sets to enhance the accuracy of pattern recognition.

II. PREVIOUS WORK

Several approaches have been developed to extracting meaningful patterns from time series. Two major approaches for pattern recognition in stock time series are template based and rule based approaches [1],[2]. In past few years many approach also focused on application of ANNs to stock market prediction [3],[4]. Recent research tends are using hybridize approaches for pattern recognition in stock time series. Tsaihet al. [7] used a combination of the rule-based technique and ANN to predict the direction of the stock and price 500 stock index futures on a daily basis. Shatkay et al. [5] suggested dividing time series data sequence into meaningful sub-sequence and Das et al. [6] introduced fixed length window to segment time series into subsequences and a time series was then represented by the primitive shape patterns that were formed. Chung et al. [8] introduced a Time series pattern matching based on Perceptually Important Point (PIP) identification. PIP algorithm is based on capturing the fluctuation of the sequence and

identifying the highly fluctuated points as PIPs. Zhang et al.[9] introduced a combination of template based and rule based approaches for pattern matching in stock time series. Their approach proposed a flexible online pattern-matching scheme based on fixed size sliding window, which is involved in the whole matching process, including both in feature point extraction and pattern matching. Zhang et al.[9] used PIP concept, introduced by Chung et al. [8] for feature point extraction from the time series data within a sliding window. Since the current states of the art are not yet concerned about this emerging composite patterns, in this paper we have proposed two novel approaches to recognize composite pattern from time series data.

III. COMPOSITE PATTERN MATCHING APPROACHES

In this section we will give a brief description about two proposed composite pattern matching approaches. As shown in Fig.1 each approach has three steps. Both of this approach is based on variable size sliding window. On the first step, we extract the feature points from time series data in a variable length sliding window. The feature point extraction method is based on finding Perceptually Important Points (PIP)[8] from time series data. On the next step, for our first approach we recognize composite patterns based on Spearman's Rank Correlation Coefficient. For our second approach we propose a pattern matching scheme relying on neural network where the inputs of the network are the normalized PIP points. On the third step for both of our approaches we use the rule sets to improve the ability of identifying patterns and distinguish them more effectively.

In both of our approaches we try to match basic pattern in a sliding window which has a fixed initial length and we will use the notation of primary window length for this fixed window length throughout the paper. However if we fail to find any basic pattern in a primary window, then we increase the window length incrementally to a threshold value. But if the length of window reaches to that threshold value and no preferred basic pattern is matched in that window then we move the sliding window to the next raw data where the length of the sliding window is set equal to primary window length again. However, if a match is found in that sliding window, then we move leftmost side of sliding window over the raw data depending on which basic pattern is matched and the length of the sliding window is set equal to primary window length. If any preferred basic pattern is matched for this new sliding window, a composite pattern is found. This composite pattern is constituted by these two basic matched patterns.

A. Feature extraction

A time series is a collection of observations of well-defined data items obtained through repeated measure-

ments over time. for example Stock time series is a curve where x-coordinate represents the trading days while the y-coordinate the closing prices. Time series sequence contains a large number of time points. So, it is costly and time consuming to analyze the patterns from the time series directly. A simple and efficient method is needed for representation or approximation of the time series sequence. Chung et al. [8] introduced Time series pattern matching based on Perceptually Important Point (PIP). In our approach PIP is used for feature point extraction. The PIP algorithm is based on finding the fluctuation of the sequence and takes these highly fluctuated points as PIPs. Initially, the first two PIPs are defined as the first and last point of input sequence (P) where input sequence (P) is actually the time series data within a sliding window(SW). The next PIP will be the point in P with maximum distance D to the first two PIPs. The fourth PIP will be the point in input sequence P with maximum distance D to its two adjacent PIPs. In this way we extract the PIPs from the input sequence P where the number of PIPs(N) is user defined and we will call this series as extracted series(SP). Chung et al. [8] applied different forms of maximum distance D and found that that it was efficient and effective to extract the feature points when D was perpendicular distance between the test point and the line connecting the two adjacent PIPs. Hence, perpendicular distance is used as D in PIP algorithm. The pseudo code of extracting Perceptually Important Point (PIP) is stated below:

```

1: procedure EXTRACTING PERCEPTUALLY IMPOR-
   TANT POINTS( $P[1 : n]$ )  $\triangleright$  Input sequence ( $P[1:n]$ ) of
   stock time series
2:    $SP[1] \leftarrow P[1]$   $\triangleright$  The first point of input
   sequence
3:    $SP[N] \leftarrow P[N]$   $\triangleright$  The last point of input
   sequence
4:   repeat
5:     Select  $P[i]$  with maximum perpendicular dis-
   tance
6:      $PD$  to the adjacent  $PIP$  points in  $P$ 
7:      $SP[j] \leftarrow P[i]$ 
8:   until  $j = N$ 
9: end procedure

```

B. Rule Sets For Basic Patterns

In Spearman's rank based basic pattern recognition approach, there is a loss of information when the data are converted to ranks. Also in our approach of basic pattern recognition using neural network has some error. So, we define a set of rules for each basic patterns. These rules describe the basic patterns more explicitly. Using rule-based method we can eliminate the error occurs in rank based and neural network based techniques, as this approach distinguished each pattern more accurately.

Such as in Head and Shoulders pattern amplitude difference between two shoulders will be below 15%. In our approach if a potential basic pattern is matched as output either from neural network base or rank base approach, we apply previously defined rules on extracted PIP series of that window. The predefined rules for six basic patterns are stated below.

- Head & Shoulders Pattern
 - $|SP[2] - SP[6]| < 15\%$
 - $|SP[3] - SP[5]| < 15\%$
 - $SP[4]$ is the top most point
 - $SP[2]$ and $SP[6]$ must be the second and third top point
 - $SP[1]$ and $SP[7]$ must be the lowest two points
- Head & Shoulders Pattern (Reversed)
 - $|SP[2] - SP[6]| < 15\%$
 - $|SP[3] - SP[5]| < 15\%$
 - $SP[4]$ is the lowest point
 - $SP[2]$ and $SP[6]$ must be the second and third lowest point.
 - $SP[1]$ and $SP[7]$ must be the highest two points
- Double Top Pattern(UP)
 - Difference between the top two points $< 15\%$
 - $SP[3]$ and $SP[5]$ are the top two points.
 - $SP[2]$ is higher than $SP[1]$
 - $SP[6]$ is higher than $SP[7]$
- Double Top Pattern(DOWN)
 - Difference between the top two points $< 15\%$
 - $SP[3]$ and $SP[5]$ are lowest two points.
 - $SP[2]$ is lower than $SP[1]$
 - $SP[6]$ is lower than $SP[7]$
- Triple Tops Pattern(UP)
 - $\text{Max}(|SP[2] - SP[4]|, |SP[2] - SP[6]|, |SP[4] - SP[6]|) < 15\%$
 - $|SP[3] - SP[5]| < 15\%$
 - $SP[2], SP[4], SP[6]$ must be three highest points
- Triple Tops Pattern(DOWN)
 - $\text{Max}(|SP[2] - SP[4]|, |SP[2] - SP[6]|, |SP[4] - SP[6]|) < 15\%$
 - $|SP[3] - SP[5]| < 15\%$
 - $SP[2], SP[4], SP[6]$ must be three lowest points

C. Pattern recognition based on Spearmans rank correlation coefficient

The Spearman rank correlation coefficient is a non-parametric technique which evaluate the degree of correlation between two variables. This technique works on the ranks of the time series data rather than the raw time series data. In our first approach we convert the extracted feature points $SP[1 : n]$ from each sliding window into rank values. In Table I rank conversion for each of the basic patterns is shown. Each of the six preferred basic

patterns in Fig-1 has predefined rank values. Converted rank values from each sliding window is then compared with the rank values of each preferred basic patterns. As in (1) we calculate the Spearmans Correlation Coefficient between rank values of preferred basic patterns and the converted rank values from sliding window. Here where d_i is the difference between ranks for each x_i, y_i data pair, and n is number of data pairs.

$$\gamma = 1 - \frac{6 \times \sum_{i=1}^n d_i^2}{n(n^2 - 1)} \quad (1)$$

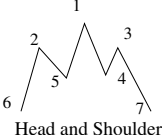
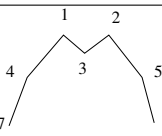
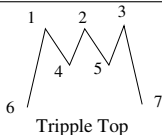
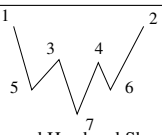
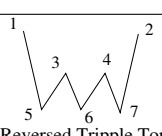
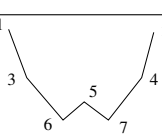
For a preferred basic pattern, If value of γ is more then a threshold value then rule sets for that preferred basic pattern is applied on $SP[1 : n]$ values of sliding window. If this $SP[1 : n]$ values passed the rule sets, that preferred basic pattern is considered to be matched in this sliding window.

D. Pattern recognition using a classification neural network

In various time series sequence data points vary extensively. For example in stock time series the variance of price of extracted PIPs will depend upon various parameters of real stock market. Hence we normalize the price of PIPs into a uniform interval $([0, 1])$ to eliminate the influences caused by this variance. Suppose extracted PIP series $SP[1 : N]$. $SP[i]$ is the highest price point and $SP[j]$ is the lowest price point in a stock time series. Then $SP_{height} = SP[i] - SP[j]$. We get the normalized extracted series $SN[1 : N]$, where for every $k = 1$ to N , $SN[k] = (SP[k] - SP[j]) / SP_{height}$. In our proposed approach we investigate the six (6) kinds of basic patterns of time series. Our approach is a process of classification using a three-layer feedforward neural network, whose inputs are extracted normalized series $SN[1:N]$ defined in section III-A. and outputs will be the predefined basic pattern shown in Fig-1.

A three-layer feedforward neural network is typically composed of one input layer, one output layer and one hidden layers. In the input layer, each neuron corresponds to a feature; while in the output layer, each neuron corresponds to a predefined basic pattern. Classification process starts with training the neural network with a group of training samples. Every training sample belongs to a certain predefined basic pattern. Then the testing samples are used to test the performance of the trained network. For a input feature vector the best output would be a output vector with all elements as zero, except one corresponding to which basic pattern the input sample belongs to. But, due to classification errors some sample input could not give the expected output. In our experiment, if any output neuron of network gives more than a threshold percent similarity, then that class of basic pattern is the potential match for input sample. In our approaches primary window length is W_0 . Both

TABLE I
RANK OF BASIC PATTERNS

Patterns Name	Pattern Figure	Position Order	Rank
Head and Shoulder	 Head and Shoulder	[6 2 4 1 5 3 7]	[6.5 2.5 4.5 1 4.5 2.5] 6.5]
Double Top	 Double Top	[6 4 1 3 2 5 7]	[6.5 4.5 1.5 3 1.5 4.5] 6.5]
Tripple Top	 Tripple Top	[6 1 4 2 5 3 7]	[6.5 2 4.5 2 4.5 2] 6.5]
Reversed Head and Shoulder	 Reversed Head and Shoulder	[1 5 3 7 4 6 2]	[1.5 5.5 3.5 7 3.5 5.5] 1.5]
Reversed Tripple Top	 Reversed Tripple Top	[1 5 3 6 4 7 2]	[1.5 6 3.5 6 3.5 6] 1.5]
Reversed Double Top	 Reversed Double Top	[1 3 6 5 7 4 2]	[1.5 3.5 6.5 5 6.5 3.5] 1.5]

of our Spearmans rank correlation coefficient approach and neural network based approach starts with applying $W = W_0$ length sliding window $SW[1 : W]$ on time series data $S[1 : n]$. Then we extract feature points $SP[1 : n]$.

For Spearmans rank correlation coefficient approach we extract rank values from sliding window SW . Then we compare the rank values extracted from each sliding window with the predefined rank values of each of the preferred basic patterns. If Spearmans rank correlation coefficient γ for any of the preferred basic pattern is more then threshold value then the corresponding input time series within the sliding window (SW) is the potential basic matched pattern.

For our neural network based approach we extract normalized feature points $SN[1 : N]$ from sliding window SW . Then we apply $SN[1 : N]$ as input vector of trained neural network. If any of the six (6) output neurons gives classification accuracy more than user defined threshold in percentage (i.e. 90 percent), then the corresponding input time series within the sliding window (SW) is the potential basic matched pattern.

After that, for both of our approaches we apply previously defined rules in section III-B on $SP[1 : N]$ of that sliding window for a potential basic pattern. If $SP[1 : N]$ can pass these rules, then we consider a predefined basic pattern is found and then we move leftmost side of window over the raw data depending on which basic preferred pattern is matched. If we find Head and Shoulder or Reverse Head and Shoulder or Tripple Top or Reverse Tripple Top pattern we move leftmost side of the window over the raw data to the third extracted point $SP[3]$ of $SP[1 : N]$ and the length of the window is set equal to primary window length. If we find a Double Top or Reverse Double Top pattern, we move leftmost side of window over the raw data to the fourth extracted point $SP[4]$ of $SP[1 : N]$ and the length of the window is set equal to primary window length. If any preferred basic pattern is matched for this new sliding window, a composite pattern is found. However if we fail to find any pattern in a primary window, then we increase the window length incrementally to a threshold value. But if the length of window reaches to that threshold value and no preferred pattern is found in that window then we move the window to the next raw data where the length of the window is set equal to primary window length W_0 . Then the same process will be again applied on the new window. This will continue until the end of the Stock time series S .

In Fig-2(a) for a window between the points a and b we find a Head and Shoulder pattern. Hence move leftmost side of window over the raw data to the third extracted point $SP[3]$ which is point c . For a window between the points c and d we find a Tripple Top pattern. These

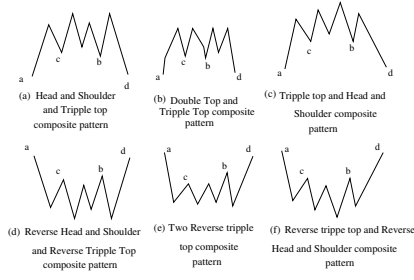


Fig. 2. Some example of composite patterns.

two patterns constitute a Head and Shoulder and Triple top composite pattern. Also in Fig-2(b) for a window between the points a and b we find a Double Top pattern. Hence move leftmost side of window over the raw data to the fourth extracted point $SP[4]$ which is point c . For a window between the points c and d we find a Triple Top pattern. These two patterns constitute a Double Top and Triple top composite pattern. The pseudo code of overall approach of composite pattern recognition is stated below:

```

1: procedure COMPOSITE PATTERN MATCHING ON
   SLIDING WINDOW( $S[1 : n]$ )  $\triangleright$  full time series data
2:   Set primary Window Width,  $W_0$ 
3:   Set Window Width,  $W = W_0$ 
4:   Threshold Window Size  $W_t$ 
5:   Set the number of feature points extracted from
6:   time series within sliding window,  $N$ 
7:   repeat
8:     Apply sliding windows on  $S[1 : n]$  to extract
9:     feature points  $SP[1 : N]$ 
10:    For neural network based approach extract
11:    normalized feature points  $SN[1 : N]$ , within
12:    sliding window,  $SW[1 : W]$ 
13:
14:    Apply normalized feature points  $SN[1 : N]$ 
15:    as inputs of neural network and check the 6
16:    outputs. If any of the output gives more than
17:    90% match, then this pattern class is
18:    considered as potential matched pattern
19:    ( $MP$ ).
20:
21:    For rank based approach extract
22:    rank values from feature points  $SP[1 : N]$ ,
23:    within sliding window,  $SW[1 : W]$ 
24:    If Spearmans rank correlation coefficient
25:    with any of the preferred basic pattern is
26:    more then threshold value then this basic
27:    pattern class is considered as potential
28:    matched pattern ( $MP$ ).
29:
30:    Check whether  $SP[1 : N]$  can pass the
31:    defined rules for MP pattern.
32:    if pattern match found then

```

```

33:      Move the window depending on which
34:      basic pattern is matched with window
35:      width  $W = W_0$ 
36:      if If any basic pattern matched for this
37:      window then
38:        A composite pattern is matched
39:      end if
40:    else if window width  $W < W_t$  then
41:      Increase the window width,
42:       $W = W + \text{next raw data}$ 
43:    else if window width  $W$  equal  $W_t$  then
44:      Move the window with
45:       $\text{step} = \text{next raw data} - SW[1]$ 
46:    end if
47:  until finish the time series  $S$ 
48: end procedure

```

IV. EXPERIMENTAL RESULT

In this section we present the experimental results that we conducted on our proposed approach. We have implemented our algorithm in MATLAB and run extensive simulation on a PC with Intel core i5 processor with clock speed 2.3GHz and 4 GB memory. We have uploaded our code, which is publicly available to access, at [12]. We have performed extensive experiments and compare our two proposed approaches. We run experiments on two real stock price dataset namely Dow Jones Industrial Average [10] (20904 data points) and General Electric Company (GE) [11] (12691 data points). According to the experiment on both datasets, window length of 15 to 20 data gives the best results for our both approaches.

A. Neural network and rule based approach

For training the neural network we partition the dataset of both [10] and [11] dataset into training and testing dataset. For dataset [10] the training dataset contains first 19000 points and rest of the data belongs to testing dataset. Similarly, we partition first 10000 points of dataset [11] as training dataset and rest of the data as testing dataset. We train neural network using various number of neurons in hidden layer and various number of iteration on training data and then apply our neural network and rule based approach with variable window length with initial window length of 15 data and increase window length by one raw data until window length reached the threshold Window Length of 20 data on test data. Number of composite patterns found from Dataset [11] for using various number of hidden layer and various number of iteration is shown in Fig-4. From Dataset [10] for training feed-forward neural network of 40 neurons in hidden layer and 4500 iteration on training data we find highest average of 25 patterns. And From Dataset [11] for training feed-forward neural network of 35 neurons in hidden

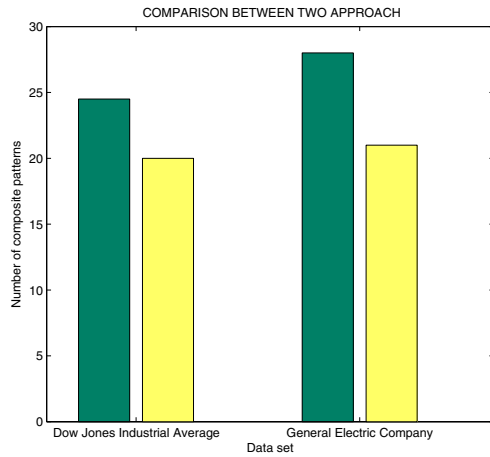


Fig. 3. Comparison of two approaches. On the left neural network based approach and on the right rank based approach

TABLE II
ERROR OF TWO APPROACHES(IN PERCENTAGE)

Stock Time Series	Neural network and Rule based	Rank based and Rule based
Dow Jones Industrial Average	6.5	13
General Electric Company	7.6	19.2

layer and 3500 iteration on training data we find highest average of 28 patterns.

B. Spearmans rank based and rule based approach

In this composite pattern recognition approach we extract seven (7) feature points from each sliding window. In this approach initial window length of sliding window is 15 data and increase window length by one raw data until window length reached the threshold Window Length of 20 data on test data. From test dataset of Dataset [10] using this Spearmans rank based and rule based approach we find 20 composite patterns and from test dataset of Dataset [11] using this Spearmans rank based and rule based approach we find 21 composite patterns

Fig-3 depicts comparison of the number of composite patterns found using our various approaches. Actual number of Composite pattern in [10] Dataset is 23 and actual number of Composite pattern in [11] Dataset is 26. Table II depicts the error of our proposed two approaches.

V. CONCLUSIONS

In this paper we introduce composite patterns and propose two hybrid composite pattern matching approach

based on variable length sliding window. Our hybrid approaches can efficiently recognize composite patterns. In

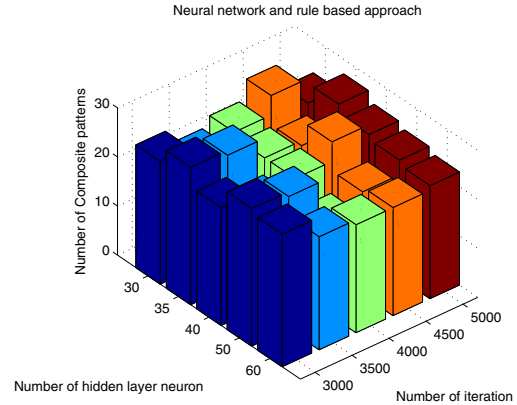


Fig. 4. Composite patterns found in Dataset of General Electric Company using neural network approach

this paper we work with basic six patterns and composite patterns originated from them. There are several more basic patterns and various composite structure can be generated from them. In future work can be done to recognize these several other patterns using these hybrid approaches.

REFERENCES

- [1] C. L. Osler, and P. H. K. Chang, "Head and Shoulders: Not Just a Flaky Patternl," Staff Report No.4, Federal Reserve Bank of New York.
- [2] F. Collopy, and J. S. Armstrong, "Rule-based forecasting:Development and validation of an expert systems approach to combining time series extrapolations ," *Management Science*, vol. 38, pp. 1394-1414, October 1992.
- [3] H. Ahmadi, "Testability of the arbitrage pricing theory by neural networks," in *Proceedings of the International Conference on Neural Networks, San Diego, CA, 1990*, pp. 385393.
- [4] J.H. Choi, M.K. Lee, M.W. Rhee, " Trading S& P 500 stock index futures using a neural network," in *Proceedings of the Annual International Conference on Arti-cial Intelligence Applications on Wall Street, New York, 1995*, pp. 63-72.
- [5] H. Shatkay, S. B. Zdonik, "Approximate queries and representations for large data sequences," in *Proceedings of International Conference on Data Engineering, Los Alamitos, CA: IEEE Computer Society Press, 1996*, pp. 536-545.
- [6] G. Das, K. I. Lin,H. Mannila , " Rule discovery from time series," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1998*, pp. 16-22.
- [7] R. Tsaih, Y. Hsu, C.C. Lai, "Forecasting S&P 500 stock index futures with a hybrid AI system," *Decision Support Systems.*, vol. 23, pp. 161-174, June. 1998.
- [8] T. C. Fu, F. L. Chung, R. Luk, and C. Ng, "Stock time series pattern matching: Template-based vs. rule-based approaches," *Engineering Applications of Artificial Intelligence.*, vol. 20(3), pp. 347-364, April 2007.
- [9] Z. Zhang, J. Jiang, X. Liu, R. Lau, H. Wang, R. Zhang, "A Real Time Hybrid Pattern Matching Scheme for Stock Time Series," in *Proc. ADC2010*, 2010.
- [10] <http://finance.yahoo.com/q/hp?s=DJI+Historical+Prices>; Last accessed : June 25, 2012, 1 am GMT.
- [11] <http://finance.yahoo.com/q/hp?s=GE+Historical+Prices>; Last accessed : June 25, 2012, 1 am GMT.
- [12] <https://gist.github.com/3207395>; Last accessed : June 25, 2012, 1 am GMT.