Intraday Business Model Strategies on Forex Markets: Comparing the Performance of Price Pattern Recognition Methods

Nikitas Goumatianos[1], Ioannis Christou[2] and Peter Lindgren[3]

[1]*CTiF, Aalborg University, Aalborg, Denmark*
[2]*Athens Information Technology, 19Km Markopoulou Ave. PO Box 68, Paiania 19002, Greece*
[3]*Aarhus University, MBIT Research Group*
Full Professor Aarhus Universiy, Denmark

Received 12 March 2014; Accepted 26 May 2014
Publication 28 July 2014

Abstract

We compare the performance of four (4) different special-purpose pattern recognition methods designed for trading systems in forex markets. Each of these methods has been properly designed to exploit highly compute-intensive aggregation calculations of complex but efficient distributed SQL queries on the relational databases holding the market data. Specifically, we have implemented several methods for discovering hidden price patterns in forex markets time-series data that signal the short-term direction of price changes of the market. We encoded the time series forex data using (a) Binary Rule based Approximation (BRA), appropriate for candlestick-based pattern recognition methods, (b) a Symbolic Aggregate approximation (SAX) technique, (c) a modified SAX technique, called Volatility-Sized SAX algorithm (VSAX) and (d) a heuristic based on capturing local peaks and valleys in the time-series (called TP-SAX). Then, after processing the codified patterns (training and validating), we extracted those patterns whose accuracy exceeds a certain threshold, and stored them in a database. In the final stage, we use these discovered patterns in a trading system which only accepts as inputs only the estimated probabilities of future change in prices. In our experiments, we used

Journal of Multi Business Model Innovation and Technology, Vol. 2, 1–34.
doi: 10.13052/jmbmit2245-456X.211

time series data from forex market covering 10 paired-currencies (AUD/CHF, CAD/CHF, EUR/USD, etc.) with timeframes of 1 and 20 minutes. The results are quite impressive with significant net profits. Furthermore, we found that each pair currency produces patterns of very different accuracy and support level, and that there exist currencies for which no profitable strategy can be inferred using the proposed methods.

Keywords: Financial time series analysis, Data Mining, Pattern discovery, Forecasting prices, Intraday Patterns, SAX Patterns, Candlestick Patterns, Top/Bottom Patterns, important points, Association Rules, Trading Strategies, Forex Markets.

1 Introduction

1.1 Predictability of Price Patterns

The ability to predict the future prices of instruments (stocks, futures, options, etc), based on historical data, is the biggest challenge in investment industry, while from scientific perspective there is a dispute between scientists. The dispute goes as far back as at least 1953 (Kendall, 1953) when Kendall published his research, where he found that he could identify no predictable patterns in stock market prices, and as a consequence, he suggested that prices evolve randomly and in unpredictable ways regardless of past performance. Later, the Efficient Market Hypothesis (EMH) theory was formulated by Eugene Fama in 1970 (Fama, 1970), which states that security prices adjust rapidly to the arrival of new information and that security prices reflect all available information about security (stock). Therefore, if the EMH holds true, there is no way to outperform the market, and trading rules based on historical data are essentially useless. There are a lot of studies that support the theory of EMH and reject technical analysis. Most authors back then, concluded that the optimal strategy for the stock-market is the buy-hold strategy (Alexander, 1961; Jensen and Bennington, 1970; Fama, 1970). However, later studies showed just the opposite (Brock, Lakonishok and Lebaron (1992), Bessembinder and Chan (1995, 1998), Lo, Mamaysky and Wang (2000).

Technical analysis is based on the premise that history repeats itself. Therefore, analyzing the price time-series with the purpose of discovering useful patterns could be beneficial. Generally, in technical analysis, there are two types of price patterns: Charts formations (consisting of many consecutive data) and Candlestick Patterns usually consisting of only a few consecutive candlesticks. Both chart formations and candlestick patterns are divided into

two types: reversal patterns and continuation patterns. Both are geometrical representations of prices that can be visually distinguished on the price chart by the investor. Candlestick consists of two essential elements (see Figure 2): the rectangular part called "body" and two vertical line extensions called "shadow". The top and bottom of rectangle are determined by the open and close price of the time frame. When the "close value" is greater than "open value", the candlestick becomes white. When the "close value" is less than "open value", the candlestick becomes black. The thin vertical line above the body represents the "high value", and the line below represents the "low value". According to technical analysis, candlesticks, in order to be useful, should be combined with other forms of technical analysis such as technical indicators.

Regarding the ability of predicting future prices through pattern recognition methods, there are numerous studies which provide evidence that there exist patterns with predictive capability that can indicate the potential of getting profits. On the other hand, there are studies that reject this evidence and support that price patterns have no value or capability to produce trading profits. Most of these studies deal with popular candlestick patterns (with names such as "three-white-soldiers", "three-black-crows" etc.) and chart formations (with names such as Head and Shoulders, Double Tops/Bottoms, Triangles, Flag and Pennant, etc.). There are very few studies concentrating on hidden/discovering new patterns – beyond those are described in Technical analysis.

1.2 Previous Studies of Price Patterns

Many researchers have examined the ability to produce profits by using candlestick patterns at various markets. Caginalp and Laurent (1998) found that candlestick patterns of daily prices of S&P 500 stocks between 1992 and 1996, provided strong evidence and high degree of certainty in predicting future prices. In another study, Lee and Jo (1999) developed a chart analysis expert system for predicting stock prices. They reported that experiments on defined patterns (falling, rising, neutral, trend-continuation and trend-reversal) revealed that they were able to provide help to investors to get high returns from their stock investment. On the other hand, in a paper by Marshall et al (2000), the authors found candlestick technical analysis had no value on U.S. Dow Jones Industrial Average stocks during the period from 1992–2002. Two more recent studies, regarding candlestick patterns (Goumatianos et al, 2013a, 2013b), reported new methods for discovering

hidden profitable candlestick patterns using daily US stocks data. The first study presented a rule-based generator algorithm to create complex candlestick patterns so as to produce various types of stock price prediction patterns. The second study focused on hidden candlestick patterns on intraday stock US Data and presented a complete intraday trading management system using a stock selection algorithm for building long/short portfolios.

Regarding price patterns (chart formations), numerous methods have been proposed. Zhang et al. (2010) worked on pattern matching based on Spearman's rank correlation and sliding window, which is more effective, sensitive and constrainable compared to other pattern matching approaches such as Euclidean distance based, or the slope-based method. Another study (Walid et al, 2006) investigated the performance of twelve chart patterns in the EURO/Dollar (5-min mid-quotes) foreign exchange market, involving Monte Carlo simulation, and using identification methods for detecting local extreme. The authors found that some of the chart patterns (more than one half) have predictive power, but unfortunately, when used for creating & applying trading rules they seem unprofitable.

Another interesting study of evaluating pre-specified chart formations is the Template Grid method (TG), which involves the rank method for expressing the similarity between two patterns. Wang and Chan (2007), used this method for pattern recognition to predict stock prices by detecting so-called "bull flag" formations. More specifically, they used time-series data of Nasdaq Composite Index (NASDAQ) from March 1985 to March 2004 (4,785 trading days) and Taiwan Weighted Index (TWI) from January 1971 to March 2004 (9,284 trading days). A 10x10 Grid was used, with corresponding weights stored in the cells and 20-day fitting data (each of two successive days corresponding to one cell). They published one more similar recent study (Wang and Chan, 2009) which is an improvement of the same method applied to 7 US traded tech stocks. In that study, they analyzed a different chart formation, the so-called "rounding top and saucer" and found that the results have considerable forecasting power across them.

Regarding time-series representation, besides using numeric methods for analysis, there is a way of adopting symbolic representation (string of numbers or letters) of the data. One example of this method is the "Symbolic Aggregate approximation (SAX) which, after normalizing the time series data, divides them into equal time segments and then calculates the mean of the points of each segment. This method has been proposed by Lin and Keogh (2003) and it is based on Piecewise Aggregate Approximation (PAA) as described

by Keogh et al (2001). However, this representation seems that has one major disadvantage, namely that it sometimes misses important points or short trends.

1.3 Our Position/Contribution

We concentrated on trading strategies. A trading strategy is an algorithm, which takes inputs (mainly time series data), and produces signals for buy and sell an instrument. The profit/loss is calculating by summarization all transactions (trades) for the specific period the algorithm is applied. The objective of this research is to produce innovation by designing trading strategies using various pattern recognition methods in order to produce profits. Additionally, we hope to involve practitioners in critical thinking in applied research including the critique of various concepts, advanced portfolio management theories and synthesis. In parallel, it is expected to help the investors and fund managers to understand in depth various investment techniques (how to design trading strategies) that are efficient and engage them to improve themselves while understanding and minimizing risk.

Contrary to the EMH, our hypothesis is that when using computational intelligence (machine learning, data mining, pattern methods) one may be able to obtain access to high quality (processed) information that no one else has. The basic idea is that, although the same raw information is available to everyone, not everyone has the ability to analyze it successfully at all times, and so in certain times, there is opportunity to get profit while the market adjusts its prices.

Here, we concentrate on four types of pattern recognition methods that discover useful patterns whose appearance indicates with high probability, the imminent occurrence of an event such as a certain price increase etc. However, the discovery of such patterns alone may not be enough, as it is possible that even for highly accurate patterns, their predictions cannot be used to develop proper strategies for realizing actual profits. From the point of view of a portfolio manager, prediction accuracy of a pattern is useless when they cannot be combined in a full-fledged trading system that can produce good profits. We answer this question by designing and implementing such a full-fledged trading system incorporating the findings of the four pattern recognition algorithms and show that it produces very strong profits in a short amount of time thus justifying the purpose of the patterns found and the algorithms behind them.

2 Methods and Application

2.1 Data Setup

We used intraday data with a time frame period of 1 and 20 minute of totally 10 pair-currencies (AUD/CAD, AUD/CHF, AUD/JPY, CAD/CHF, CAD/JPY, CHF/JPY, EUR/CAD, EUR/GBP, EUR/USD, EUR/JPY) This data-set starts from Jan 2010 until the end of 2013 (31 December 2013) and contains approximately 11 million entries (for 1 min) and 0.53 million entries (for 20 min). The row data format for each entry is as follows:

Symbol, Date Time <range of 1 or 20 min>, Open Price, High Price, Low Price, Close Price

The data were divided into the following training and testing setups – similar to walk forward analysis:

1st setup: Training Patterns Data: [1/1/2010, 30/6/2012] Validating Patterns Data: [1/7/2012, 31/12/2012] Testing Strategy: [1/1/2013, 30/6/2013]

2nd setup: Training Patterns Data: [30/6/2010, 31/12/2012] Validating Patterns Data: [1/1/2013, 30/6/2013] Testing Strategy: [1/7/2013, 31/12/2013].

2.2 System Architecture

In Figure 1 we show a high-level diagram of the system architecture, consisting of four major subsystems: (a) Patterns Creation System (all type of patterns);

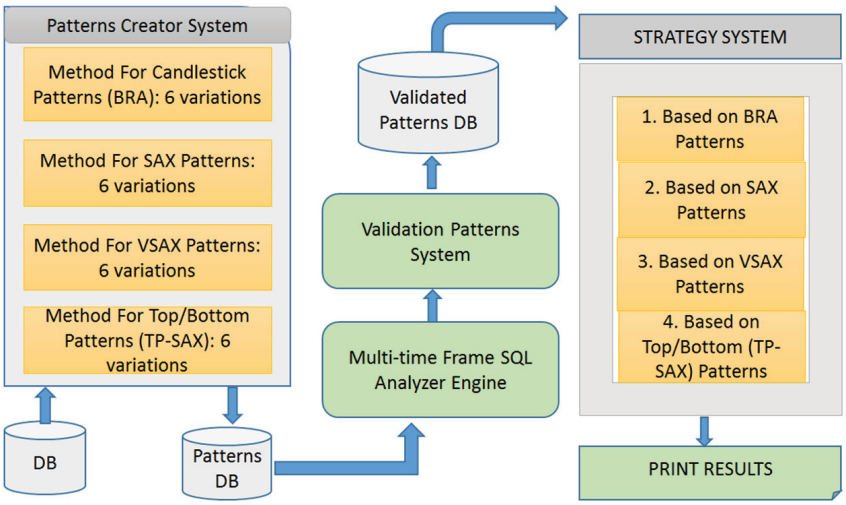


Figure 1 System architecture

(b) Multi-time frame SQL Analyzer Engine; (c) Validation Patterns System, and (d) Strategy System.

The first subsystem, the Pattern Creator System, uses four different methods for converting time series into proper binary/symbolic representations. Each method produces six different patterns by involving different in value parameters. So, each time, for each time frame, the system creates totally 24 *candidate* patterns (4 methods x 6 variations). The produced patterns are saved in the database in binary/symbolic representation. A multi-time frame SQL Analyzer processes the training patterns data by grouping and analyzing them for each predicate variable (totally used 10 variables, see Table 2). The above candidate patterns which satisfy certain constraints (rules), pass as true patterns, and are saved in the database as training valid data. The same process is repeated for the testing patterns data where those passed the rules called testing valid data. Then, the validation system which compares the training and testing patterns data according to specific rules creates a new database called "validated patterns DB". These final patterns are later used by the strategy system to create trades.

2.3 Patterns: Design/Methodology/Approach

The norm in real-world time series data is to exhibit high degrees of fluctuation which are sometimes considered as noise, but unless done very carefully, the filtering out of the "noise" may also cause significant loss of information if the "noise" turns out to carry hidden information after all. To develop a method of discovering patterns in time series, one needs to develop a method for presenting the data. The question is then, how we can determine the important points and how the "smoothing" of data for presentation and decision support purposes should be done. As we previously mentioned, in this research, we developed four different methods using six variations for each method. In the next sections, we analyze each one.

2.3.1 Candlestick Patterns (Binary Rule Based Approximation – BRA)

The method of transforming candlestick patterns is called Binary Rule-based Approximation (BRA), because it involves rules to express relations between candlesticks. It is implemented in two stages by using the Specification Rule Engine (SRE) and next the Knowledge Representation Module (KRM). The SRE contains rule-based expressions that record two (2) types of information: Bits referred to candlestick itself and bits referred to exact position

(relationship) among two or more candlesticks. The KRM uses the SRE to codify the candlestick patterns in binary format (series of strings of '0' or '1' - each bit is one rule) for each instrument and for each time frame.

For illustration purposes, we present only a sample of bit-codification rules for each category:

(i) *Rules refer to one candlestick itself, see Figure 2*

(ii) *Rules refer to the relationship between candlesticks, see Figure 3*

As shown in Figure 4, the KRM uses a binary translation system which according to SRE converts candlesticks into a binary product and then it stores it in database.

Actually, the system produces six binary products:

- Limited info (13 rule-bits) patterns of two consecutive candlesticks.
- Extended info (28 rule-bits) patterns of two consecutive candlesticks.
- Limited info (26 rule-bits) patterns of three consecutive candlesticks.
- Extended info (44 rule-bits) patterns of three consecutive candlesticks.
- Limited info (33 rule-bits) patterns of four consecutive candlesticks.
- Extended info (48 rule-bits) patterns of four consecutive candlesticks.

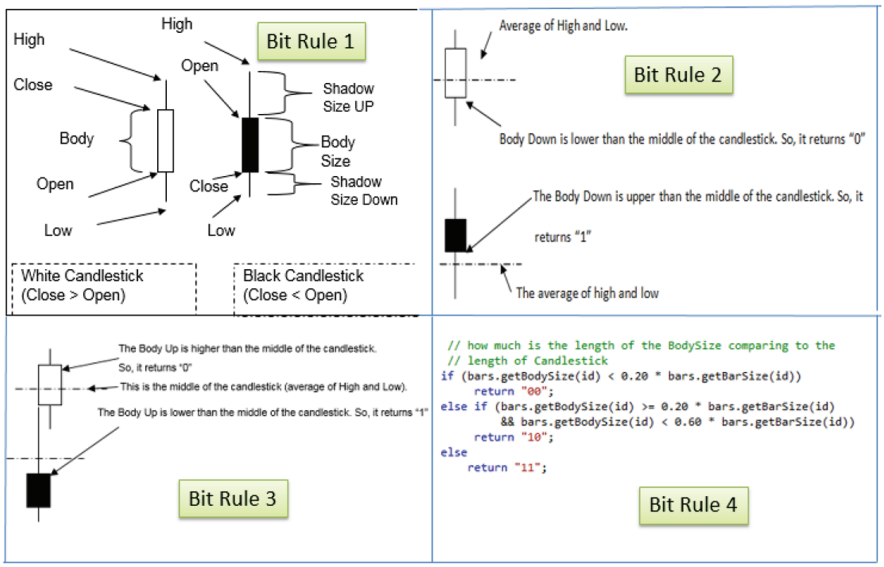


Figure 2 Rules for one candlestick

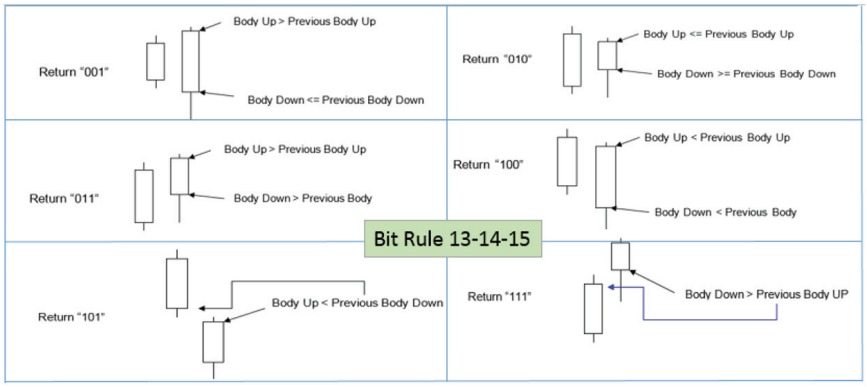


Figure 3 Relationship rules

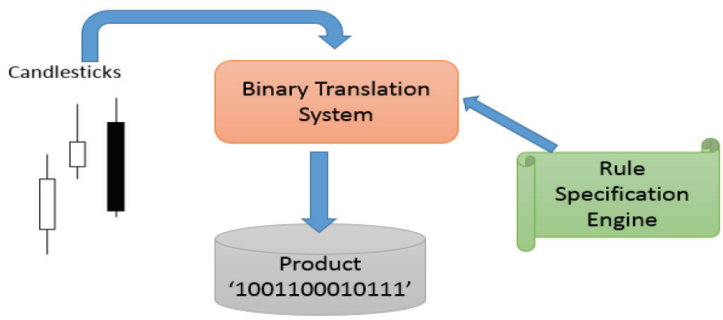


Figure 4 Structure of BRA algorithm

2.3.2 SAX patterns

This algorithm, as described by Lin and Keogh (2003), first normalizes the time series data using the following equation:

$$x_i' = \frac{x_i - \mu_x}{\sigma_x} \tag{1}$$

where x_i is the current point of original time series data, μ_x is the mean of time series data of window size n, σ_x is the standard deviation of the data in the time-series.

Then, the normalized data of size n are divided in w equal size segments and each segment is represented by the arithmetic mean of points w/n, as it is shown in Figure 5 (small black circles represent the mean of each segment). In this example, the normalized data of size n are divided into 5 segments (see

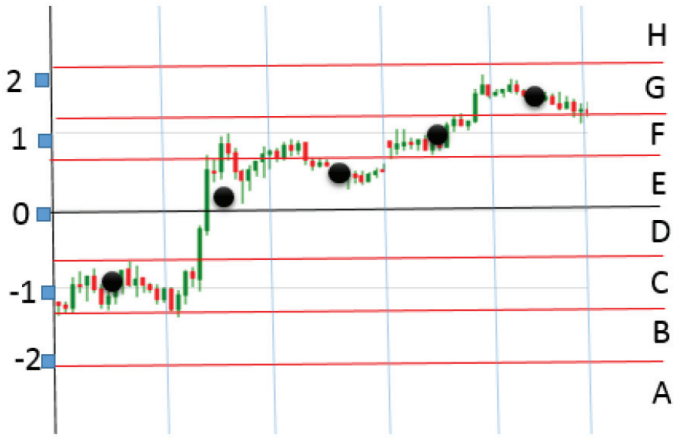


Figure 5 SAX representation

vertical blue lines). The horizontal red lines divide the chart into parallel zones where each zone is represented by a letter (A, B, .., H). It should be noted that the intervals, which define the zones, are not equi-probable defined by normal distribution curve, as it is originally described by SAX algorithm (it means that a randomly selected pattern has the same probability to belong to any zone – each zone has the same number of the patterns), but fixed values. The pattern symbolic representation of Figure 5 (starting from the most recent) is 'GFEEC'

Here, we run six variations of SAX algorithms and used eight zones (as depicted in Figure 5) defined by seven internals {-1.8. -0.9, -0.4, 0, 0.4, 0.9, 1.8}, as follows:

- SAX where window size (n) = 18, word size = 6 and pattern symbol length = 3
- SAX where window size (n) = 36, word size = 12 and pattern symbol length = 3
- SAX where window size (n) = 24, word size = 6 and pattern symbol length = 4
- SAX where window size (n) = 48, word size = 12 and pattern symbol length = 4
- SAX where window size (n) = 30, word size = 6 and pattern symbol length = 5
- SAX where window size (n) = 60, word size = 12 and pattern symbol length = 5

2.3.3 Volatility-Sized SAX (VSAX) patterns

This algorithm is similar to SAX algorithm, with one major difference. The word size (segment) is not fixed length; it reduces or expands in size dynamically depending on the current volatility in order to capture better the changes and trends. Therefore, the window size (n) is variable depending on the volatility. The VSAX algorithm requires four input parameters to be calculated:

- The word length *w*, which corresponds to a letter
- The length of the pattern (how many words)
- The coefficient check point *c*,
- The interval points for defining the zone spacers.

Here, the internals are: {-1.8. -0.9, -0.4, 0, 0.4, 0.9, 1.8}, while the ratio check point by default is ½. The coefficient check point is used to determine the minimum word size required for the mean calculation. For our current research we used c=½. The minimum word size is equal to wc. Figure 6 depicts the implementation of SAX and VSAX algorithms having eight zones break (A, B, . . . , H).

Let's suppose the length of the pattern is 3, the vertical red lines 0, 1, and 4 represent the segments w (0->1) and 2w (1->4). In the SAX algorithm, we take the moving average of each segment, and translate them into letters.

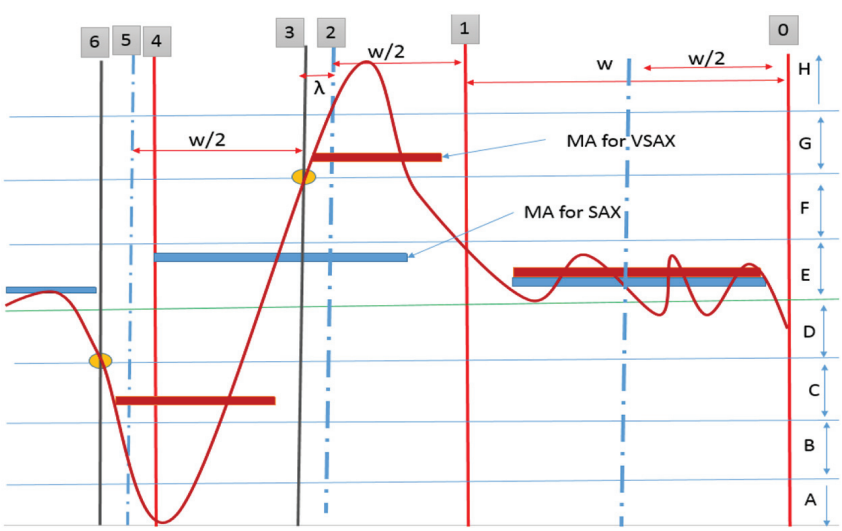


Figure 6 Volatility-Sized SAX representation

Here, the string code for SAX is 'EEE' (see thin blue rectangles). Note that in the second segment (1->2) there is a high volatility (great up and then down change). This volatility is not recorded because it happens in the same segment that two opposite changes exist.

On the other hand, our algorithm, VSAX, is designed to capture volatility while supporting the smoothing of the data. Starting from '0' point (most recent), it calculates the moving average (MA) of the w/2 segment and finds the zone in which it belongs. In the example, it is in 'E' parallel zone. Then, it continues (increasing the length of the segment) until reaching the full length (=w, point '1') as long as each next point remains in the same zone. Here the first segment is w, the same as in SAX. For the second segment (1->4), VSAX calculates the MA of next w/2 (1–2), found to belong in zone 'G'. Then, it continues until the point '3'. Because point '3' belongs to different zone ('F'), VSAX stops to w/2 + λ segment, and therefore the second code letter is 'G'. From point '3', it repeats the same procedure and stops at point '6' because it changes zone. So, the code of the pattern, according to VSAX method, is 'EGC', instead of 'EEE' produced by SAX algorithm.

What have been done is to modify the SAX algorithm in such way that each segment has length between $w/2$ and w depending on volatility changes. The window length (n) of SAX is: (number of segments) $\times$ (w), whereas in VSAX is dynamically adjusted according to:

(Number of segments) $\times$ *(w/2) <= VSAX window <= (Number of segments)* $\times$ *(w)*

2.3.4 Top/Bottom SAX (TB-SAX) patterns

This algorithm has significant differences from SAX algorithm. The points for the symbolic translation are not moving averages from equal segments. The points are now "local top/bottoms" of the typical prices of the instrument:

Typical Price = (High + Low + Close) / 3

The word size (segment) is not fixed length, it is defined as the horizontal distance (expressing in the number of candlesticks) between two (2) local top-bottoms. The TB-SAX algorithm requires four input parameters for its execution:

- The maximum window size (how many periods to look back)
- The pips threshold value for distinguishing the local top/bottoms
- The number of Top/Bottom points (to determine),
- The interval points for defining the zone spacers. Here, the internals are: {-1.8. -0.9, -0.4, 0, 0.4, 0.9, 1.8}.

The algorithm starts from the most recent point and uses the original raw time-series data. A point is considered as a local top/bottom if the typical price changes more than the pips threshold value. If this change continues in the same direction, then, the previous local point is replaced by the newly found point. The algorithm stops when the required number of top/bottom points has been found or it reaches the maximum permitted window size. For example, if the top/bottoms are formed within 110 candlesticks and maximum window size is 100, then the pattern is rejected and not saved in database. In Figure 7, we show the local top/bottoms starting from the most recent point. The points 0, 1, 2, 3 and 4 are top/bottoms because the change in pips from previous top/bottom point was more than the pips threshold value (16 pips). Note that the point 5 is not local bottom because the typical price change is less than 16 pips. In this step, the algorithm stores in an array of integers all top/bottom indices {i0, i1, i2, i3, i4} and window size.

The algorithm then normalizes only the data of the window size according to (1). The zones that the points belong to, form the pattern symbolic code (from which point, we essentially follow the same process as in SAX algorithm). Here, in Figure 8, the symbol code is 'EBFCF'.

We implemented six variations of this method as shown in Table 1. The Max Length Formation sets a limit of how many bar items can be used to form the specific number of Top/Bottoms. The selected variations of Max Length Formations (60, 120, 160, 180, 240) came out by experimentally way and more specifically to be able to form patterns from 60% to 80% times of all points contained in the sample training data. For example for the "Set 1" if

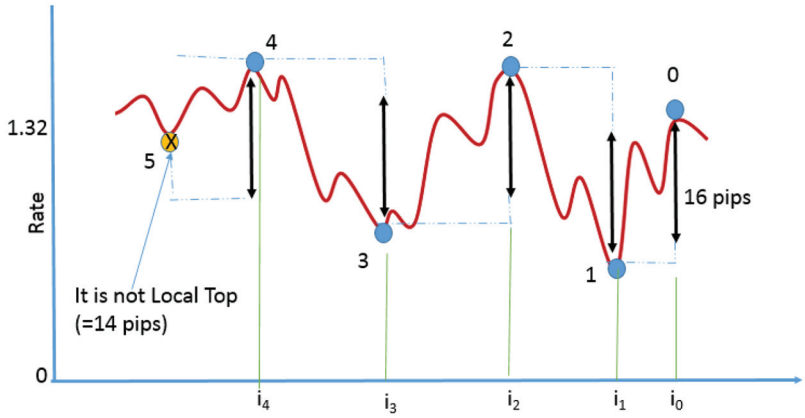


Figure 7 TB-SAX: find indices of top/bottom points (original data)

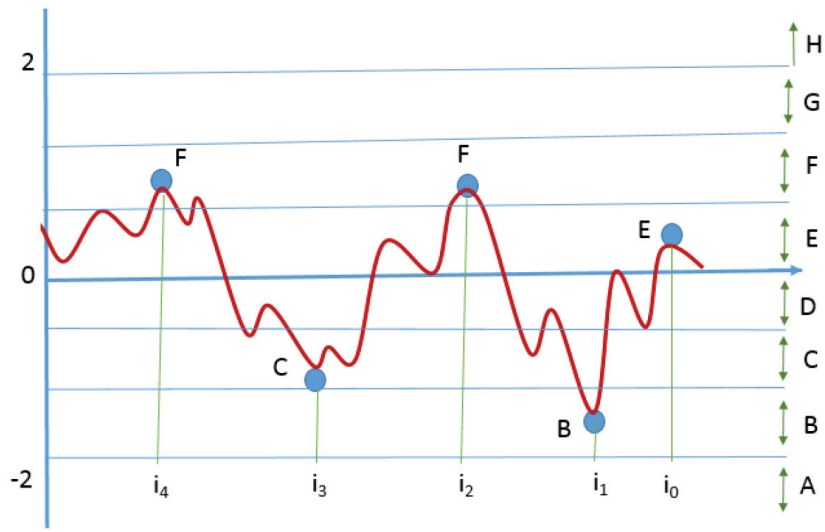


Figure 8 TB-SAX representation (normalized data)

Table 1 Variations (sets) of TB-SAX per time frame

	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6
Number of Top/Bottoms	3	3	4	4	5	5
Max length formation	60	120	120	160	180	240
Threshold pips value (1 min)	5	8	5	8	5	8
Threshold pips value (20 min)	10	18	10	18	10	18

we select a length less than 60, the number of created patterns (TB-SAX) are less than 60% of the number of total training rows data (bar items of specific time frame).

2.4 Prediction Types

A price pattern is presented by its attributes (*attr*) and its predicate variables (*pred*): P_i = *(attr,* $\{$*(pred, result)*$\}$*)*. In more detail, *attr = (tf, ptnCode, pipsRange)* and *pred* = (*tf, i, c, p*), where

- *tf* = time frame (1min, 20 min),
- *ptnCode* is its actual pattern code,
- *pipsRange* is the difference between highest – lowest price value of the pattern expressed in pips and refered to specific periods back (here it is 20),

- i is the interval periods ahead the pattern (e.g. if i=1and tf=5, presents within next one period - next 5 min),
- c is the predicate type which takes two values: greater than (>) and less than (<),
- p is the value in pips. For example the presentation: {*pred, result*} = {*(20, 2, >, 8), true*} means that prediction of the "pattern time frame 20 min, within next 2 periods ($2 \times 20 = 40$ *min*), highest price is greater than (>) 8 pip" is true.

We used ten predicates for each price patterns, as shown in Table 1. The values (in pips) of predicate variables were determined empirically by running the search algorithm many times in a sample of data and getting as best values those that can find patterns with accuracy in prediction above 60%, and have minimal support of 1%. If we decrease the predictive threshold values (pips), then we can find more patterns (increase support level) with even better accuracy. However, in such a case we cannot detect patterns that are truly useful (with sufficient forecasting power).

2.5 Analyzing Price Patterns: Multi-Timeframe SQL Analyzer/Validator

As it referred in section 2.5, a price pattern is presented by its attribute and its predicate variables: $P_i = (attr_i, \{(pred_i, result_i)\})$. To explain better how the Multi-Timeframe SQL Analyzer/Validator works, we will give some extra definitions.

Definition 1: Multi-Predicate Instance (M-PRED) is the set of supported predicate variables (see Table 2) which characterize each price

Table 2 Predictive variables and values (in pips) per time frame

	Time Frames	
Predictive Variables (All Changes in Pips)	01 Min	20 Min
Next period lowest price less than	–2	–6
Next period highest price greater than	2	6
Within next 2 periods lowest price less than	–3	–8
Within next 2 periods highest price greater than	3	8
Within next 5 periods lowest price less than	–5	–12
Within next 5 periods highest price greater than	5	12
Within next 10 periods lowest price less than	–10	–25
Within next 10 periods highest price greater than	10	25
Within next 20 periods lowest price less than	–15	–35
Within next 20 periods highest price greater than	15	35

pattern and denoted in the following form: M-PRED$_i$(P$_i$) = $\{\{\text{pred}_1, \text{r}_1\}, \{\text{pred}_2, \text{r}_2\}\ldots\{\text{pred}_{10}, \text{r}_{10}\}\}$, where $\text{r}_1, \text{r}_2\ldots\text{r}_{10}$ are results for each corresponding $pred_k$. A price pattern can be written as:

P$_i$ = (attr, M-PRED$_i$).

Definition 2: Predictive Accuracy (%) (*PA*) of *k* predicate variable ($pred_k$) (referred to a pattern (p_i)is calculated as following:

*Predictive Accuracy (PA) = 100 * (number of* P_i *having* r_k *= true)/(Total number of* P_i *appeared)*, where r_k is result (true/false) of kth predicate variable ($pred_k$).

Definition 3: Valid Predictive Accuracy (%) (VPA) of kth predicate variable ($pred_k$): A predicate variable ($pred_k$) has result (r_k) = true as long as *PA* is greater or equal to 60%. $Valid\ pred_k : \{r_k = true \Leftrightarrow PA \geq 60\}$

Definition 4: Forecasting Power (*FP*) of a pattern (p_i): We consider that a pattern has "Forecasting Power" as long as there is at least one $pred_k$ whose result. r_k *is true* For a pattern (P_i) and its Multi-Predicate Instance ($M - PRED_m$) *FP* is defined as: $FP(P_i) = true \Leftrightarrow \exists(pred_k, r_k)$: $r_k = true$

Definition 5: Single Algorithm Support: By default we denote the "Single Algorithm Support of Time Frame *tf*" as *S-Sup* and it is defined as: 100×*(total occurrences of all type of patterns having forecasting power) / (total item bars of time frame of the specific* time range)%.

The above definitions are required for understanding the implementation. First, we develop SQL scripts running in Microsoft SQL Server 2012 by grouping properly the data per predicting variable, per frame, per grid and per training/testing sample. All grouping SQL scripts are based on the prediction accuracy of a given pattern that is of course defined as the number of times that price climbed above x pips within next y periods divided by the total number of times a match for the formation has appeared.

Additionally, we involved a factor called "pattern's pips range" which represents the difference between the highest and lowest value of a given time range expressed in pips value. This factor has proved to be very important in estimating the forecasting power of a pattern. We briefly mention here, that as an example, for patterns of timeframe 1 min, only those patterns which have pips range greater than 25 pips have forecasting power (FP), while for patterns of timeframe 20 min we used 35–40 pips range.

In the final stage, the Patterns Validation System examines all training and testing sets to check if they satisfy the following rules:

- In Training Data Set: Prediction accuracy exceeds 60% , then in Testing Set Data: Prediction accuracy must exceed 60%
- In order for a pattern to be valid, it should pass the above test for at least one predictive variable.
- The occurrences of each pattern for a training data set should be more than 20 times, while in testing more than 10 times.

3 Strategy Design

Since the support level is significantly high, we could design a proper strategy to exploit all these predictions. From our point of view, we have two options: to develop a complete trading system based only on patterns predictive information or to use it as secondary system such as to develop a trading system based on technical indicators and use the patterns' predictive information as a filter system. We decided not to involve any technical indicator and exploit properly all patterns that appear to have significant forecasting power and have passed the validation process. To design the strategy we followed a high level framework by repeating the process of analyzing data, design and testing the system until setting goals succeed (have net profits, t-test statistics > 1.6, etc). In Figure 9, we show the framework which helped us to achieve the goals of having high performance.

The main idea for the strategy is to take positions (long or short) based on the prediction accuracy of the current pattern that appears. We constructed four setups for testing strategy. Each strategy setup contained six variations of each pattern method (BRA, SAX, VSAX, TB-SAX). The reason was to have better support for decision trading and compare the performance of each method (on basis 6 variations each). Moreover, the decision of using all variations of each method came out during the initial design of the system as Figure 9.

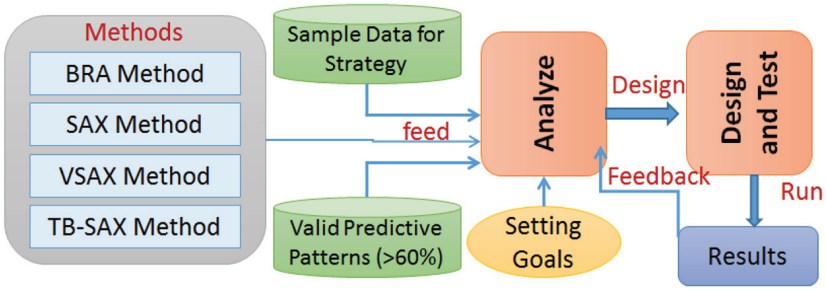


Figure 9 High level diagram of designing the strategy

To design the algorithm properly we have to face the following problems to resolve:

- How to avoid losses in periods that we have no predictive patterns and have opened a position (see Figure 10).
- The time we decide to close an open position.
- Which predicate variable should take into account more.
- How to face conflicts in case of appearing multi-patterns with different in direction predictions.

For the first, to avoid losses, we employed a "Trailing Stop-Loss Functions" – one for long positions and one for short positions. This additionally helps to maintain our profits. The stop loss is expressed in pips (constant value). When the price goes to the favorite direction, it drags the stop along with it, but when the price stops going to this favorite direction, the stop-loss price remains at the level it was dragged to. For example, let's suppose we open a long position at 1.3240 with 20 pips trailing stop (now it is placed at 1.3220). If the price goes up at 1.3270 (30 pips more), the stop will be moved at 1.3250. So, if the price goes down, the system will automatically close the position at 1.3250 with a gain of 10 pips.

To explain how the trading decision is made, we give some more definitions.

Definition 6: Trading decision variable. It has one of four predefined values: Trading Decision (D) = {NOT TRADE, CONFLICT, ENTER LONG, ENTER SHORT}.

Defenition 7: Single Trend Behavior (STB). Each Multi-Predicate Instance (*M-PRED*) of a pattern is characterized by a Trading Decision variable: {ENTER LONG, ENTER SHORT}.

If $MAX((r_1, r_2, r_3, \ldots, r_{10}) = MAX(r_2, r_4, r_6, r_8, r_{10})$ then $STB = ENTER\ LONG$

If $MAX((r_1, r_2, r_3, \ldots, r_{10}) = MAX(r_1, r_3, r_5, r_7, r_9)$ then $STB = ENTER\ SHORT$

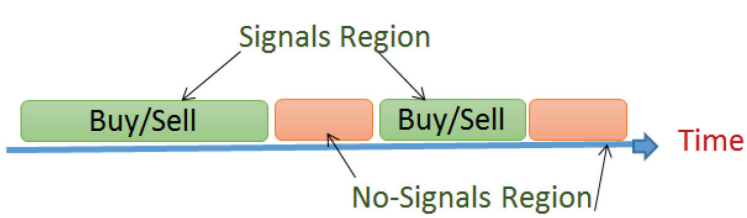


Figure 10 Time regions of producing trading signals

Definition 8: Agregate Trend Behavior (*ATB*). It takes into account all predicate variables and it is calculated as follows:

ATB = $(r_{2+}r_{4+}r_{6+}r_{8+}r_{10}) \circledcirc (r_{1+}r_{3+}r_{5+}r_{7+}r_{9})$, where r_1, $r_{2...}$ r_{10} are the probabilities % for each corresponding $pred_k$ (see Table 1). Given a threshold h_o>0, the *ATB* is Bulish if *ATB*> h_o , Bearish if *ATB* < -h_o, No Trend if *ATB* ε [-h_o, h_o].

Defenition 9: Pattern Total Trend Behavior (*PTTB*). A pattern is characterized Bulish if *STB* = *ATB* = *ENTER LONG*. Bearish if *STB* = *ATB* = *ENTER SHORT*. Otherwise is *NO TREND*.

Defenition 10: Strategy Decision Making (*D*). At specific time, given a set of patterns (formed by different methods) which correspong to a set of Multi-Predicate Instances { *M-PRED*}, the strategy decision is made by the following conditions:

***D = ENTER LONG*:** if Each *M-PRED (PTTB = ENTER LONG or NOT TRADE)* and at least one *M-PRED* is *ENTER LONG*

***D = ENTER SHORT*:** if Each *M-PRED (PTTB = ENTER SHOR or NOT TRADE)* and at least one *M-PRED* is *ENTER SHORT*

***D = NOT TRADE*:** if Each *M-PRED (PTTB = NOT TRADE).*

***D = CONFLICT*:** if at least one *M-PRED is ENTER LONG* and least one *M-PRED* is *ENTER SHORT*.

Therefore, the strategy opens a position if Strategy Decision Making (D) is either ENTER LONG or ENTER SHORT (definition 10). An opened position is closed as soon as the D produces opposite signal or the trailing stop is activated to protect losses or profits. For example, if there is an open long position, this remains open as long as D = NOT TRADE or D = ENTER LONG and trailing stop is not activated. When D = ENTER SHORT, the system closes the long position and simultaneously opens a short position.

Regarding the Money Management System we employed a variable position sizing of using 10% of required margin for current balance. It means that we re-invest profits for each additional trade. The margin level used for trading was 1/100. If we employed more position sizing we could get better results, but the risk of ruin would increase as well.

Regarding the sample data used for testing the strategy, we used two setups of different periods as described in section 2.1: 1st sample testing data: [1/1/2013, 30/6/2013] and 2nd is [1/7/2013, 31/12/2013]. Actually, for testing we used two different sets of data: One for testing/validation the patterns and one different (unknown for the system) for running the strategy. The strategy

was applied to EUR/USD pair currency which had better performance mainly due to higher patterns support.

4 Empirical Results

4.1 Comparing the Predictability of Pattern Mining Methods

After applying patterns validation process, we calculated the total support level for each time frame and for each method (including all variations – 6 for each method).

Figure 11, shows the support level (%) for 1 min time frame patterns, while in Figure 10 is for 20 min.

In Figure 11 & 12, we display the effective support per method (includes its 6 variations). For each time unit (bar item) we have six (6) patterns for each method. The number of valid patterns ranges from 0 to 6 (all 6 variations patterns have prediction accuracy greater than 60%). The effective support is calculating as the following:

Effective Support of Method= Count {if the bar item has at least one predictive valid pattern without any conflicts return 1 otherwise return 0} / total number of bar items.

In other words each bar item it counts 1 if any pattern is valid without conflict. Conflict we have if one pattern is bullish and one other is bearish. For more details, see the previous section 3.

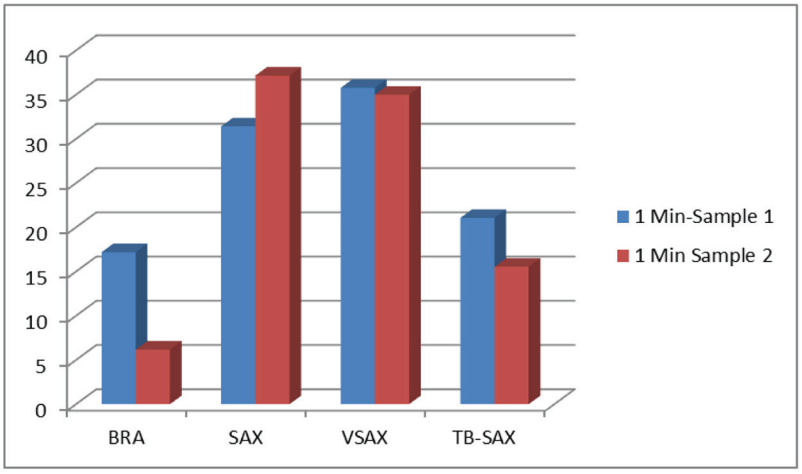


Figure 11 Total support (%) per method for 1 min time frame

Table 3 Validated patterns & their occurrences

	Total Occurrences - Data Sample 1		Total Occurrences - Data Sample 2	
Pattern Type	Time Frame 1 Min	Time Frame 20 Min	Time Frame 1 Min	Time Frame 20 Min
Limited Info Candlesticks Of 2	18,618	7,198	9,696	5,775
Extended Info Candlesticks Of 2	23,559	2,440	13,740	2,252
Limited Info Candlesticks Of 3	26,005	1,863	13,964	2,020
Extended Info Candlesticks Of 3	3,465	0	4,092	0
Limited Info Candlesticks Of 4	11,046	55	7,545	58
Extended Info Candlesticks Of 4	1,602	0	1,864	0
Total	84,295	11,556	50,901	10,105
SAX Of 3 Word 6	5,599	552	1,715	364
SAX Of 3 Word 12	6,081	472	3,596	197
SAXOf 4 Word 6	10,253	1,991	7,025	1,206
SAX Of 4 Word 12	3,918	3,005	7,415	1,731
SAX Of 5 Word6	5,597	2,138	6,340	1,519
SAX Of 5 Word 12	3,944	1,947	7,229	857
Total	35,392	10,105	33,320	5,874

(Continued)

Table 3 Continued

	Total Occurrences - Data Sample 1		Total Occurrences - Data Sample 2	
Pattern Type	Time Frame 1 Min	Time Frame 20 Min	Time Frame 1 Min	Time Frame 20 Min
VSAX Of 3 Word 6	8,757	3,987	5,389	2,799
VSAX Of 3 Word 12	9,150	2,532	7,103	1,782
VSAX Of 4 Word 6	10,774	3,097	6,347	1,894
VSAX Of 4 Word 12	12,195	2,491	9,281	1,866
VSAX Of 5 Word 6	3,517	849	2,362	476
VSAX Of 5 Word 12	5,883	1,148	4,683	989
	50,276	14,104	35,165	9,806
TB-SAX Of 3 - (5/10 pips for 1-20 min)	9,812	2,727	10,490	3,025
TB-SAX Of 3 - (8/18 pips for 1-20 min)	8,687	2,197	7,626	1,903
TB-SAX Of 4 - (5/10 pips for 1-20 min)	13,101	3,054	8,194	2,488
TB-SAX Of 4 - (8/18 pips for 1-20 min)	9,832	2,365	7,221	2,005
TB-SAX Of 5 - (5/10 pips for 1-20 min)	11,112	3,551	8,065	2,172
TB-SAX Of 5 - (8/18 pips for 1-20 min)	9,428	1,948	8,961	1,700
Total	61,972	15,842	50,557	13,293

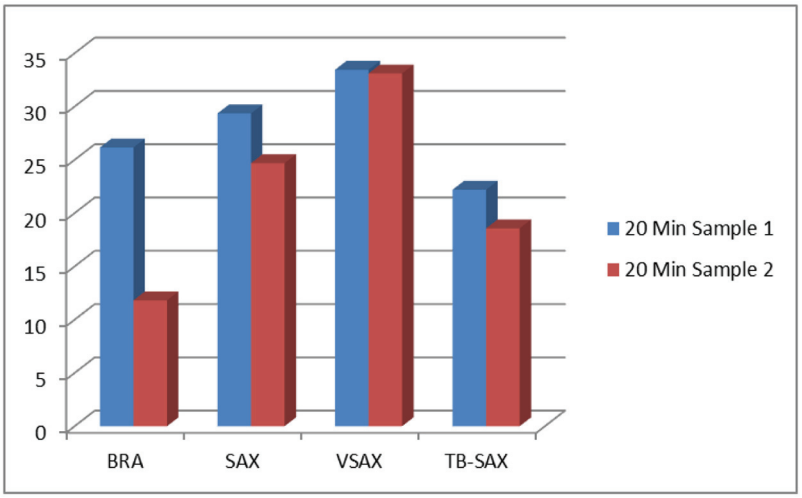


Figure 12 Total support per method for 20 min time frame

4.2 Strategy Performance

We run the strategy for each method, for each time frame and for each sample data. Below the Table 4 presents the trading results of 1 min time frame and data sample 1. We can distinguish that BRA patterns has superior results over the other methods, while SAX has the worst results. Additionally, for SAX, t-test statistics is low (1.52) less than 1.6 – and the percentage of wining trades

Table 4 Running strategy for 1 min time frame of EUR/USD – data sample 1

	1 Min Time Frame - Sample 1			
	BRA	SAX	VSAX	TB-SAX
Starting Capital	10,000.00	10,000.00	10,000.00	10,000.00
From Date	2-Jan-13	2-Jan-13	2-Jan-13	2-Jan-13
Until Date	30-Jun-13	30-Jun-13	30-Jun-13	30-Jun-13
Net Profit Total ($)	74,030.18	11,530.75	36,257.57	23,036.07
Net Percent (%) Total	740.30	115.31	362.01	230.36
Number of Trades	5,275	2,993	2,197	3,389
Percent (%) of winning Trades	52.83	46.37	50.71	51.87
Max Runup Amount ($)	3,247.46	1,770.31	2,255.88	2,132.00
Max Draw Down Amount ($)	1,080.36	293.29	599.63	435.27
Average Net Profit per Trade ($)	14.03	3.85	16.48	6.80
Ratio Average Win to Avg Loss	1.02	1.26	1.17	1.10
Profit Factor	1.14	1.10	1.20	1.10
t-test Statistics	3.29	1.52	2.87	1.96

is less than 50%. For the above reasons we can accept the performance of all methods except SAX.

Next, Table 5 presents the trading results of 1 min time frame and data sample 2. We can distinguish that TB-SAX strategy has the worst results and its t-test statistics is below 1.6 (1.50). In conclusion, comparing results between sample data 1 and 2 for 1 min time frame, only BRA and VSAX have significant and more consistent results.

The time frame 20 min, has on average good results for all methods, as it is seen in Table 6. VSAX and TB-VSAX seem to have superior results.

Table 5 Running strategy for 1 min time frame of EUR/USD – data sample 2

	1 Min Time Frame - Sample 2			
	BRA	SAX	VSAX	TB-SAX
Starting Capital	10,000.00	10,000.00	10,000.00	10,000.00
From Date	30-Jun-13	30-Jun-13	30-Jun-13	30-Jun-13
Until Date	31-Dec-13	31-Dec-13	31-Dec-13	31-Dec-13
Net Profit Total ($)	37,073.90	13,519.71	46,042.28	8,877.45
Net Percent (%) Total	370.74	135.20	460.42	88.77
Number of Trades	4,154	4,981	5,117	2,132
Percent (%) of winning Trades	56.28	59.71	58.10	53.99
Max Runup Amount ($)	2,212.80	1,341.53	6,249.00	742.43
Max Draw Down Amount ($)	574.41	321.88	852.79	22.43
Average Net Profit per Trade	8.92	2.71	9.00	4.36
Ratio Average Win to Avg Loss	0.87	0.72	0.81	0.93
Profit Factor	1.11	1.07	1.12	1.10
t-test Statistics	2.55	1.63	2.25	1.50

Table 6 Running strategy for 20 min time frame of EUR/USD – data sample 1

	20 Min Time Frame - Sample 1			
	BRA	SAX	VSAX	TB-SAX
Starting Capital	10,000.00	10,000.00	10,000.00	10,000.00
From Date	2-Jan-13	2-Jan-13	2-Jan-13	2-Jan-13
Until Date	30-Jun-13	30-Jun-13	30-Jun-13	30-Jun-13
Net Profit Total ($)	21,822.76	19,572.02	40,808.95	40,842.76
Net Percent (%) Total	218.73	195.72	408.09	408.43
Number of Trades	465	258	513	306
Percent (%) of winning Trades	57.42	56.59	61.79	57.52
Max Runup Amount ($)	2,319.95	3,222.99	2,455.76	6,858.03
Max Draw Down Amount ($)	1,414.55	1,368.06	1,554.32	2,033.58
Average Net Profit per Trade	46.93	75.86	79.55	133.47
Ratio Average Win to Avg Loss	0.91	1.02	0.89	1.00
Profit Factor	1.28	1.33	1.43	1.35
t-test Statistics	1.67	1.77	2.97	1.96

Table 7 Running strategy for 20 min time frame of EUR/USD – data sample 2

	20 Min Time Frame - Sample 2			
	BRA	SAX	VSAX	TB-SAX
Starting Capital	10,000.00	10,000.00	10,000.00	10,000.00
From Date	30-Jun-13	30-Jun-13	30-Jun-13	30-Jun-13
Until Date	31-Dec-13	31-Dec-13	31-Dec-13	31-Dec-13
Net Profit Total ($)	34,854.55	11,300.08	24,404.31	15,247.44
Net Percent (%) Total	348.55	113.00	244.04	152.07
Number of Trades	227	245	439	220
Percent (%) of winning Trades	55.07	59.18	63.78	56.82
Max Runup Amount ($)	4,904.87	2,868.32	3,630.94	3,136.80
Max Draw Down Amount ($)	1,890.83	965.00	1,281.05	852.53
Average Net Profit per Trade	153.54	46.12	55.59	69.12
Ratio Average Win to Avg Loss	1.16	0.85	0.73	1.10
Profit Factor	1.43	1.23	1.28	1.42
t-test Statistics	2.03	1.29	1.94	2.01

In Table 7, SAX has low performance and low t-test statistics (1.29) at data sample 2. Although SAX brings positive net profit, its low t-test statistics warn us that these results maybe are not consistent.

The profit/loss curve of each method is depicted in Figure 13 and Figure 14. The vertical axis represents the account balance which starts at USD $10,000 initial investment. The horizontial axis is bar item index which corresponds to time. For I min (Figure 11), the point index value from zero to 180,000

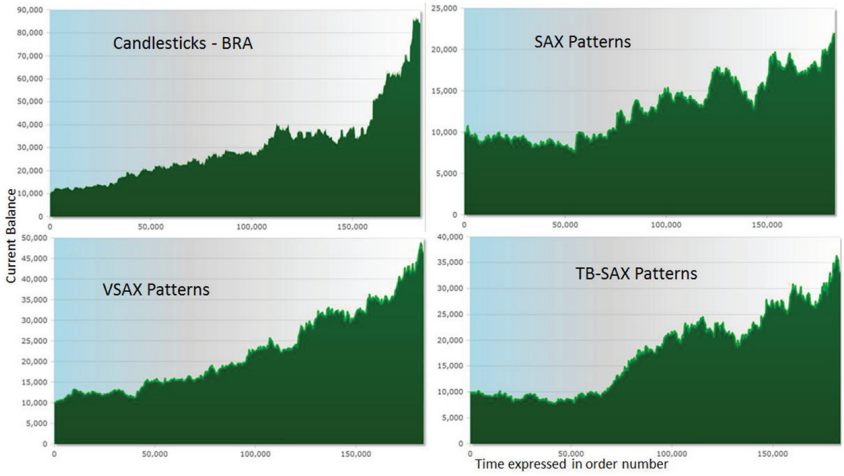


Figure 13 Profits curve for 1 min EUR/USD – sample 1

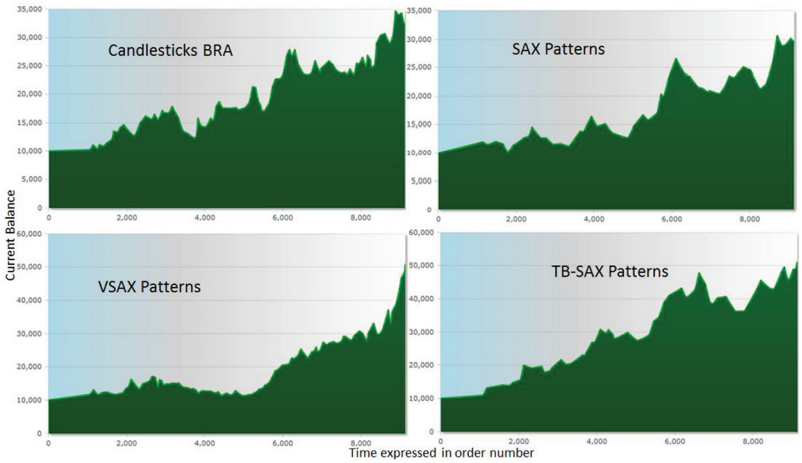


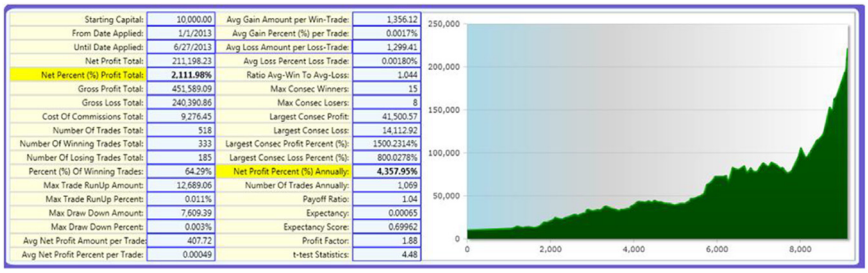
Figure 14 Profits curve for 20 min EUR/USD – sample 1

corresponds to six (6) month data, while in Figure 14 corresponds to the same time range of data.

4.3 Combining Different Pattern Methods

In Figure 15, we show the trading positions. The blue arrows correspond to long positions only. The red arrows correspond to short positions only. When we see two arrows in the same point (candlestick), it means that one position

Figure 15 Graphical display of trading signals on 20 min EUR/USD – sample 1



Starting Capital:	10,000.00	Avg Gain Amount per Win-Trade:	1,356.12
From Date Applied:	1/1/2013	Avg Gain Percent (%) per Trade:	0.0017%
Until Date Applied:	6/27/2013	Avg Loss Amount per Loss-Trade:	1,299.41
Net Profit Total:	211,198.23	Avg Loss Percent Loss Trade:	0.00180%
Net Percent (%) Profit Total:	**2,111.98%**	Ratio Avg-Win To Avg-Loss:	1.044
Gross Profit Total:	451,589.09	Max Consec Winners:	15
Gross Loss Total:	240,390.86	Max Consec Losers:	8
Cost Of Commissions Total:	9,276.45	Largest Consec Profit:	41,500.57
Number Of Trades Total:	518	Largest Consec Loss:	14,112.92
Number Of Winning Trades Total:	333	Largest Consec Profit Percent (%):	1500.2314%
Number Of Losing Trades Total:	185	Largest Consec Loss Percent (%):	800.0278%
Percent (%) Of Winning Trades:	64.29%	Net Profit Percent (%) Annually:	**4,357.95%**
Max Trade RunUp Amount:	12,689.06	Number Of Trades Annually:	1,069
Max Trade RunUp Percent:	0.011%	Payoff Ratio:	1.04
Max Draw Down Amount:	7,609.39	Expectancy:	0.00065
Max Draw Down Percent:	0.003%	Expectancy Score:	0.69962
Avg Net Profit Amount per Trade:	407.72	Profit Factor:	1.88
Avg Net Profit Percent per Trade:	0.00049	t-test Statistics:	4.48

Figure 16 Combined patterns strategy: results for 20 min EUR/USD – sample 1

is closed and the same time it opens new position at the opposite direction (long or short).

We run the strategies by combining more than one pattern algorithms. We found that the combination of Variable-SAX patterns and Top/Bottom patterns increase the perormance of the system producing amassing profits. Figure 16, shows these results:

Actually, the strategy was the same. The only thing changed was the input parameters for selecting the type of the patterns. The trading decision comes out of examining 6 pattern variations from VSAX and 6 pattern variations from TB-SAX. Here, we may have more conflicts during the examinations.

5 Discussion

5.1 Limitations – Spread Effects

We can see that the student t-statistic is very high (well above 1.6), providing strong evidence that our trading system produces very profitable trades that cannot be attributed to chance. But for real time trading we should expect less profit. We ignored any slippage that could affect negatively our results (less profitable), but the cost of spreading (commission) is about the average of current market.

The effects of spread (difference in price between ask and bid) is very important and affects the performance of the system. In Table 8, we show how the performance of combined Patterns Strategy (see Figure 14) decreases with increasing spread. This is due to higher transaction costs and the high number of produced trades. If we apply the effects of spread in previous methods (see Tables 4–7), we can still have profits until 1.0 -1.5 spread. For spreads above 1.5 we have no profits. Moreover, the negative effect is greater in smaller time frames such as 1 min.

Table 8 The effect on spread in trading performance (EUR/USD spread)

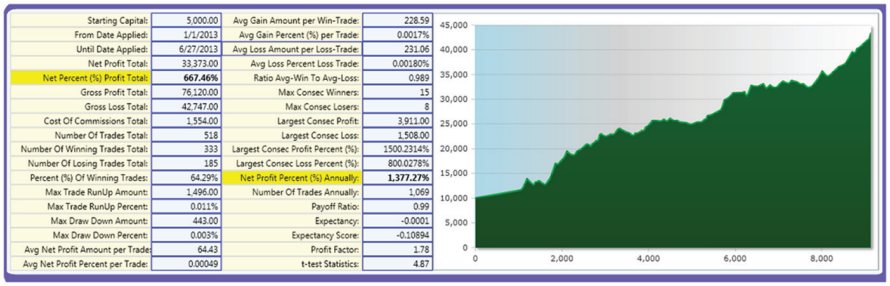
	Spread=0.5	Spread=0.8	Spread=1.3
Starting Capital:	10,000.00	10,000.00	10,000.00
From Date Applied:	1/1/2013	1/1/2013	1/1/2013
Until Date Applied:	6/27/2013	6/27/2013	6/27/2013
Net Profit Total:	189,541.25	160,959.22	109,166.49
Net Percent(%) Profit Total:	**1,895.41%**	**1,609.59%**	**1,091.66%**
Gross Profit Total:	416,653.75	369,753.54	281,657.21
Gross Loss Total:	227,112.50	208,794.32	172,490.72
Cost Of Commissions Total:	14,414.95	20,792.19	30,810.80
Number Of Trades Total:	518	518	518
Number Of Winning Trades Total:	331	328	322
Number Of Losing Trades Total:	187	190	196
Percent(%) Of Winning Trades:	63.9%	63.32%	62.16%
Max Trade RunUp Amount:	11,509.87	9,942.87	7,065.09
Max Trade RunUp Percent:	0.011%	0.011%	0.011%
Max Draw Down Amount	6,930.98	6,024.61	4,342.70

We discussed the performance of the EUR/USD. But what about the support level for the other instruments? In Table 9, we show the number of different patterns identified and the total occurrences of the valid patterns for all methods regarding 1 min time frame. There are more than 100,000 total occurrences in four of ten instruments. Larger occurrences mean greater support which can be used by the trading system to produce more profits.

We should note that patterns, which have forecasting power in one instrument, may not have acceptable accuracy (less than 60%) in another instrument. Moreover, some instruments produce very low in support predictive patterns (EUR/GBP, CAD/JPY) and therefore, for such instruments, one cannot develop profitable strategies using only patterns information.

Table 9 Valid pattern occurrences per instrument

Instrument	Number of Patterns	Total Occurrences
AUD/CAD	200	19,178
AUD/CHF	849	105,258
AUD/JPY	576	62,679
CAD/CHF	739	88,810
CAD/JPY	144	10,622
CHF/JPY	668	76,131
EUR/CAD	1,247	144,073
EUR/GBP	5	272
EUR/JPY	1,268	125,815
EUR/USD	1,931	287,189



Starting Capital:	5,000.00	Avg Gain Amount per Win-Trade:	228.59
From Date Applied:	1/1/2013	Avg Gain Percent (%) per Trade:	0.0017%
Until Date Applied:	6/27/2013	Avg Loss Amount per Loss-Trade:	231.06
Net Profit Total:	33,373.00	Avg Loss Percent Loss Trade:	0.00180%
Net Percent (%) Profit Total:	**667.46%**	Ratio Avg-Win To Avg-Loss:	0.989
Gross Profit Total:	76,120.00	Max Consec Winners:	15
Gross Loss Total:	42,747.00	Max Consec Losers:	8
Cost Of Commissions Total:	1,554.00	Largest Consec Profit:	3,911.00
Number Of Trades Total:	518	Largest Consec Loss:	1,508.00
Number Of Winning Trades Total:	333	Largest Consec Profit Percent (%):	1500.2314%
Number Of Losing Trades Total:	185	Largest Consec Loss Percent (%):	800.0278%
Percent (%) Of Winning Trades:	64.29%	Net Profit Percent (%) Annually:	**1,377.27%**
Max Trade RunUp Amount:	1,496.00	Number Of Trades Annually:	1,069
Max Trade RunUp Percent:	0.011%	Payoff Ratio:	0.99
Max Draw Down Amount:	443.00	Expectancy:	-0.0001
Max Draw Down Percent:	0.003%	Expectancy Score:	-0.10894
Avg Net Profit Amount per Trade:	64.43	Profit Factor:	1.78
Avg Net Profit Percent per Trade:	0.00049	t-test Statistics:	4.87

Figure 17 Fixed contract: results for 20 min EUR/USD – sample 1

5.2 Money Management Implications

An important reason for the high performance of the strategy tested, is the money management techniques involved. We used variable position (sizing system) where profits are re-invested. For the starting capital USD$10,000 the exposure is always 10% of current capital with margin level 1:100. It means that if we have USD$10,000, we can invest USD$1000 margin required – therefore 1 lots (1 lot = USD$100,000). If the capital becomes USD$100,000 then, we can trade 10 lots (maximum). If we trade with fixed contract – 1 lot (USD$100.000) for all trades then the performance decreases to ¼. Figure 17 shows the results of running again the system of combined patterns strategy (Figure 16). We can see the net profit from 211,198 drops to 33,373.

6 Conclusion and Future Work

We discovered price patterns with high support levels able to support strategies, without any assistance from technical indicators, which produce very significant profits. But these strategies cannot be applied to all instruments. Issues such as high spread and low volatility of the markets can negatively affect the profitability of our proposed system. The EUR/USD instrument, due to its very high volatility and very low spread can therefore be used to materialize very significant profits. The lower volatility decreases the predictive pattern support level, while the higher spread decreases the profits due to the transaction costs. For those instruments with high spread value and middle volatility, these methods can still be useful as additional support tool to filter trades from a strategy.

We also compared different methods of pattern discovery. The new modified SAX type algorithms (VSAX & TB-SAX) brought superior trading results compared to the base SAX method. Also, the candlestick patterns (BRA) brought great profits in small time frame (1 min). The combination of different methods (Figure 16) seems to increase profits even more. Obviously, this result does not mean that by combining more and more pattern methods we will keep getting better results. The main problem is the production a lot of different patterns (one pattern produced by each method and each variation) at specific time which denote opposite predictions. An interesting research direction, could involve Genetic Algorithms to select the best methods at each time, with best variations that constantly produce great profits.

The pattern length (the numbers or letters present the pattern) was from 2–5, enough to create useful support level to be exploited by trading systems. If we used greater pattern length, then, we should use other technique for ranking the similarity between patterns, maybe calculating the differences of the zones (distance) for each point. This direction of this type of research may bring some useful results, especially for SAX and VSAX algorithm. As for TB-SAX, it seems difficult to bring better results because already includes a lot of bar items (100–200).

We have presented useful insights towards pattern discovery methods and techniques to design trading strategies and we hope to involve other researchers to find further improvements to other type of instruments such as stocks, futures, etc.

References

[1] Alexander, S.S. 1961. Price movements in speculative markets: trends or random walks, Ind. Mgmt. Rev., 2:7–26.

[2] Bessembinder H. and K. Chan, 1995. The profitability of technical trading rules in the Asian stock markets, Pacific Basin Finance Journal, 3(2–3):257–284.

[3] Brock, W., Lakonishok, J., Lebaron, B., 1992. Simple technical trading rules and the stochastic properties of stock returns, Journal of Finance, 47:1731–1764.

[4] Caginalp G., Laurent, H. 1998. The predictive power of price patterns, Applied Mathematical Finance 5:181–205.

[5] Goumatianos N., Christou I.T., Lindgren P., 2013. Useful pattern mining on time series: Applications in the stock market, Proceedings of the

2nd International Conference on Pattern Recognition Applications and Methods (ICPRAM 2013), Barcelona, Spain.

[6] Goumatianos N., Christou I.T., Lindgren P., 2013. Stock Selection System: Building Long/Short Portfolios Using Intraday Patterns, Elsevier Procedia Economics and Finance, 2:296–307 (Proc. Intl. Conf. Appl. Econ. 2013).

[7] E. Keogh, K. Chakrabarti, M. Pazzani and S. Mehrotra, Dimensionality reduction for fast similarity search in large time series databases, Knowledge and Information Systems, vol.3, no.3, pp. 263–286, 2001.

[8] Fama, E.F. 1970. Efficient capital markets: A review of theory and empirical work, Journal of Finance, 25:383–417.

[9] J. Lin, E. Keogh, S. Lonardi and B. Chiu, A symbolic representation of time series, with implications for streaming algorithms, Proc. of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, pp. 2–11, 2003.

[10] Jensen M.C., Bennington, G.A., 1970. Random walks and technical theories: Some additional evidence, Journal of Finance, 25(2): 469–482.

[11] Kendall, Maurice (1953). "The Analysis of Economic Time Series", Journal of the Royal Statistical Society, Series A, 96, pp. 11–25.

[12] Lee K.H., Jo, G. S., 1999. Expert system for predicting stock market timing using a candlestick chart, Expert Systems with Applications, 16:357–364.

[13] Lo, A.W., Mamaysky, H., Wang, J., 2000. Foundations of technical analysis: computational algorithms, statistical inference, and empirical implementation, Journal of Finance, 55:1705 – 1770.

[14] Marshall, B.R., Young, M.R., Rose, L.C., 2006. Candlestick technical trading strategies: can they create value for investors? Journal of Banking & Finance, 30:2303–2323.

[15] Wang, J.-L., Chan, S.-H., 2007. Stock market trading rule discovery using pattern recognition and technical analysis, Expert Systems with Applications, 33:304–315.

[16] Wang J.-L., Chan S.-H., 2009. Trading rule discovery in the US stock market: An empirical study, Expert Systems with Applications, 36: 5450–5455.

[17] Walid, B., Van Oppens, H., 2006. The performance analysis of chart patterns: Monte-Carlo simulation and evidence from the euro/dollar foreign exchange market, Empirical Economics 30:947–971.

[18] Zhang Z., Jiang J., Liu X., Lau R., Wang H., Zhang R., 2010. A real-time hybrid pattern matching scheme for stock time series. Proc. 21st Australasian Conf. on Database Technologies 104:161–170.
[19] Anderson, J.C., 6 Narus J.A. (1999) Business Marketing Management Understanding, creating and delivering value. Upper Saddle River, NJ Prentice Hall.

Biographies

Nikitas Goumatianos holds a M.Sc. in Computer Science from Staffordshire University (UK) (2006), an MBA from Naval Postgraduate School, Montey, CA, USA, (2008) and he is currently candidate Ph.D. student at Aalborg University, Denmark.

He is former Air Force officer (pilot and now retired colonel). He has worked 10 years as software analyst/developer and later as program manager in Air Force computer center dealing with the development of Command, Control and Information Systems. The last 4 years (2008–2012) worked as Contracting Officer in General Directorate for Defense and Investments Armaments. His interest is the research and the development of financial forecasting trading programs involving machine learning and data mining techniques.

Ioannis T. Christou holds a Dipl. Ing. Degree in Electrical Engineering from the National Technical University of Athens, Greece (1991), an M.Sc.

(1993) and Ph.D. (1996), both in Computer Sciences from the University of Wisconsin at Madison (Madison, WI, USA), and an MBA from NTUA and Athens University of Economics and Business (2006).

He has been with Delta Technology Inc. as a Senior Developer, with Intracom S.A. Development Programmes Dept. as an Area Leader in Data & Knowledge Engineering, and with Lucent Technologies Bell Labs as a Member of Technical Staff. He has also been an adjunct assistant professor with the University of Patras, Dept. of Computer Engineering & Informatics, Greece, and an adjunct professor at Carnegie-Mellon University, Pittsburgh, PA, USA. He is currently an associate professor at Athens Information Technology, Paiania, Greece, and is the CTO and co-founder of IntelPrize, a Big-Data start-up company, and has published more than 60 articles in scientific journals and peer-reviewed conferences. His current research interests include Analytics & Computational Intelligence, Optimization, Data Mining, and Parallel & Distributed Computing.

Dr. Christou is a member of the IEEE, ACM, INFORMS and of the Technical Chamber of Greece.

Peter Lindgren is Full Professor of Multi Business Innovation and Technology at Aarhus University, Denmark. He holds B.Sc in Business Administration, M.Sc in Foreign Trade and Ph.D. in Network-based High Speed Innovation. He has (co-)authored numerous articles and several books on subjects such as product development in metwork, electronic product development, new global business development, innovation management and leadership, and high speed innovation. His current research interest is in new global business models, i.e. the typology and generic types of business models and how to innovate them.