Knowl Inf Syst (2017) 53:767–804
DOI 10.1007/s10115-017-1052-2

REGULAR PAPER

An algorithmic framework for frequent intraday pattern recognition and exploitation in forex market

Nikitas Goumatianos[1,2] · **Ioannis T. Christou**[1] · **Peter Lindgren**[3] · **Ramjee Prasad**[2]

Received: 24 July 2015 / Revised: 10 March 2017 / Accepted: 5 April 2017 /
Published online: 22 April 2017

Abstract We present a knowledge discovery-based framework that is capable of discovering, analyzing and exploiting new intraday price patterns in forex markets, beyond the well-known chart formations of technical analysis. We present a novel pattern recognition algorithm for Pattern Matching, that we successfully used to construct more than 16,000 new intraday price patterns. After processing and analysis, we extracted 3518 chart formations that are capable of predicting the short-term direction of prices. In our experiments, we used forex time series from 8 paired-currencies in various time frames. The system computes the probabilities of events such as “within next 5 periods, price will increase more than 20 pips”. Results show that the system is capable of finding patterns whose output signals (tested on unseen data) have predictive accuracy which varies between 60 and 85% depending on the type of pattern. We test the usefulness of the discovered patterns, via implementation of an expert system using a straightforward strategy based on the direction and the accuracy of the pattern predictions. We compare our method against three standard trading techniques plus a “random trader,” and we also test against the results presented in two recently published studies. Our framework performs very well against all systems we directly compare , and also, against all other published results.

✉ Ioannis T. Christou
ichr@ait.edu.gr

Nikitas Goumatianos
larnikitas@gmail.com

Peter Lindgren
peterli@huh.au.dk

Ramjee Prasad
prasad@es.aau.dk

[1] Athens Information Technology, 44 Kifisias Ave., 15125 Marousi, Greece

[2] CTiF, Aalborg University, 9220 Aalborg, Denmark

[3] Aarhus University, Norde Runggate 1, 8000 Aarhus C, Denmark

Keywords Data mining · Pattern recognition · Hidden intraday patterns · Template grid method · Forex

1 Introduction

The ability to predict the future prices of instruments (stocks, futures, options, etc.), based on historical data, is a major challenge in the investment industry. Generally, in technical analysis, there are two types of price patterns: charts formations (consisting of many consecutive data) and candlestick patterns consisting of 2–4 consecutive candlesticks. Some candlestick patterns are so well known that are given names such as "Engulfing," "Harami," "Doji" [5]. Chart formations on the other hand are usually divided into two types: reversal patterns and continuation patterns. Reversal patterns depict an important change in price trend. Continuation patterns on the other hand confirm that current trend will be maintained. In this paper, we focus on knowledge extraction techniques for discovering hitherto unknown chart formations (price patterns) that carry information about future prices.

1.1 Previous studies

Regarding candlestick patterns, there are several studies that provide evidence that these types of patterns have predictive capability. Caginalp and Laurent [6] found that candlestick patterns of daily prices of S&P 500 stocks between 1992 and 1996 provide strong evidence and high degree of certainty in predicting future prices. Lee and Jo [19] developed a chart analysis expert system for stock price forecasting. 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 contrary, Marshall et al. [25] found that candlestick technical analysis has no value on U.S. Dow Jones Industrial Average stocks during the period from 1992–2002. Two more recent studies, regarding candlestick patterns [10,11], refer to methods used to discover hidden candlestick patterns using daily US stock data. The first study presents a rule-based generator algorithm to create complex candlestick patterns so as to produce various types of stock price prediction patterns. The second study refers to hidden candlestick patterns on intraday stock US data and presents 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 used. Zhang et al. [41] describe a Pattern Matching Technique 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 [38] investigated the performance of 12 chart patterns in the EUR/USD (5-min mid-quotes) foreign exchange market, involving Monte Carlo simulation, and using identification methods for detecting local extremes. They found that some of the chart patterns (more than one half) show power to predict future trends, but when applying trading rules they seem unprofitable.

Some researchers such as Toshiniwak and Joshi [34] and Keogh and Pazzani [17] focused on the accuracy of matching in the pattern search method by using the Euclidean distance to measure the similarity between the query and candidate sequence. If the Euclidean distance between two time sequences of length n is less than a threshold value, then, the two sequences are considered to be same. Others such as Kong et al. [18] used association rule mining to discover multi-temporal patterns with four different temporal predicates, namely "before," "during," "equal" and "overlap." They discovered chains of relationships between different indices of two stock markets (Chinese mainland and Hong Kong).

To improve the forecasting accuracy of stock prices, many researchers developed hybrid forecasting models, such as Cheng et al. [7], who proposed a framework consisting of four novel data mining methods including "Cumulative Probability Distribution Approach" (CPDA), "Minimize the Entropy Principle Approach" (MEPA), Rough Sets Theory (RST) and Genetic Algorithms (GA) that uses a preprocessing method for data transformation and selection of essential technical indicators.

Other researchers focused on portfolio management, such as Xu and Cheung [39] who proposed an Adaptive Supervised Learning Decision (ASLD) trading system for the forex market which learns the best past investment decisions directly. Later, Huang et al. [13] proposed an extension of "Adaptive Supervised Learning Decision" (EASLD) trading system to enhance portfolio management involving 6 forex rates. Their results showed nearly 158% profit with a small downside risk. Raudys [32] analyzed the influences of sample size and input dimensionality on the accuracy of determining portfolio weights of automated trading systems and reduced these effects by performing a clustering of multiple time series and splitting them into a small number of blocks.

Kao and He [16] developed an optimization approach based on agent technology to integrate multiple trading strategies across asset classes not just for back-testing optimization but for live real-time trading; their results (though not tested in live systems) were quite promising. Their work is related to ours in that our developed strategy also relies on the combination and integration of different patterns developed for different time frames and grid dimensions firing simultaneously producing trading signals that we fuse into a single final decision. Some years earlier, a decision support system allowing for traditional stock market investment evaluation and analysis including model construction, evaluation and assessment was developed by Poh [31], allowing for strategy evaluation and sensitivity analyses.

In a series of papers, Leigh et al. [20–22], Bo et al. [3] and Wang and Chan [36] used classical template matching from image recognition [8] to develop a Template Grid method (TG) that was customized to predict stock prices by detecting the so-called "bull flag" formation pattern. More specifically, the study of Wang and Chan [36] used time series data of Nasdaq Composite Index (NASDAQ) from 04/03/1985 to 03/20/2004 (4785 trading days) and Taiwan Weighted Index (TWI) from 06/01/1971 to 03/24/2004 (9284 trading days). A 10×10 grid was used, with corresponding weights stored in the cells and 20-day fitting data (compressing two successive days into a single column of the grid). One more similar study by Wang and Chan [36] used an improvement of the same method applied to 7 US-traded tech stocks. In that study, they analyzed a different chart formation known in the technical analysis community, the so-called rounding top and saucer pattern and found that the results have considerable forecasting power across them. The common characteristic of these studies is that the algorithm/method is only capable of evaluating pre-specified chart formations which are already popular in technical analysis.

Wu et al. [38] merged technical analysis techniques and signals from well-known technical indicators with sentiment analysis from external sources to obtain a trading system that can obtain some profit margins on the Taiwan stock market.

Finally, it is worth mentioning some very recent developments specific to the forex markets. Theofilatos et al. [33] used Random Forests (ensembles of random decision trees) and assumed a total fixed transaction cost of 1 pip per transaction; their essentially "always-in-the-market" strategy, applied on a time frame of 1 day, resulted in a net profit percentage of 7.28% on the EUR/USD instrument for the testing dataset from March 3, 2009 to Dec. 30, 2010. However, the study does not mention any indicators of the risk associated with the devised strategy. Haeri et al. [12] used a combination of Hidden Markov Models (HMM) and CART to forecast the EUR/JPY instrument in the daily time frame, but the accuracy they report (53%) is not immediately useful for any trading purposes. Petrov and Tribelsky [30] applied the Tokens algorithm for studying dynamical systems in phase space, in order to predict the daily EUR/USD rates between 2010 and 2014; they found their system profitable under the assumption of zero transaction costs, but when transaction costs (spreads) were taken into account, they found their system was consistently losing money. Ozturk [28] presents a heuristic-based trading system using technical indicator rules and reports profitable trades on the 1-min time frame on the EUR/USD and GBP/USD instruments. And in the spirit of Wu et al. [26,38] use text mining of news headlines to predict the direction of EUR/USD on the 1-h time frame; in fact, several years ago, Zhang and Zhou [40] explored and compared different data mining techniques in financial applications, and especially, for forecasting forex spot rates, they presented various techniques, including genetic algorithms to find trading rules, neural networks and various text mining algorithms for analyzing the market news headlines. Because these latter two studies [28] and [26] use comparable datasets to our own, we directly compare the performance of our algorithm against both of them in the experimental results section.

1.2 Position

Analyzing historical data is at the core of most investing activities. It also lies at the heart of most major methods for estimating and/or validating market risks and expected returns. However, the efficient and effective analysis of such massive amounts of data that are continuously generated from every market in the world is no trivial task; how to extract actionable knowledge from these data is a very challenging problem. The main question of this research is the following: Are there any (frequent) price patterns other than the known popular chart formations ("Bull Flag" etc.) that can help predict the future direction of prices? If so, how many are there, what are their shapes and frequency of appearance? And perhaps most importantly, once we have discovered such patterns, can we exploit them to produce trading net profits?

The rest of this paper describes the algorithmic framework and system we designed to answer the above research questions.

1.3 Our contribution

We summarize our main contributions as follows:

- Our contribution to technical analysis is that we discovered in total 3518 new chart formations for all instruments (currency pairs) across various time frames (1, 20, 60 min) for predicting the future direction of prices. Among them, more than 200 new chart formations have similar or greater frequency of appearance and prediction accuracy with those coming from technical analysis and can visually be detected.
- We perform pattern recognition and evaluation via a special purpose algorithm we developed for determining "Template Grids" (each grid corresponding to a different prototype pattern) of mixed dimensions and fitting the time-series segments in those grids. Performing template matching on grids is mentioned in Duda and Hart [8] for image recognition applications; for time-series pattern recognition, it appears in the context of speech recognition in Ney et al. [27]. Applying the idea in financial time series was proposed in a case-study short paper Leigh et al. [20] who used a fixed predetermined template grid of size 10 × 10 and searched for the "bull flag" pattern. The original method of Leigh et al.

[20] on which our method is loosely based on, therefore is not capable of determining frequent patterns that are not known a priori (in fact, the grid size they used is relatively small and would not be able to capture most known complex price patterns, even though they are known a priori). Contrary to this setting, in our research, the focus is to mine for useful prototype patterns which then determine the content of our Template Grids; this mining occurs in four search spaces of varying dimensions, namely 10×10, 15×10, 20×15 and 25×15, despite the additional computational required power, so as to capture more complex patterns and price movement details. To improve accuracy, our method takes into account some attributes including pips range, price-level filters, a range of predicate variables (to be defined later) and resolves conflicts arising when using simultaneously different size template grids.

- To our knowledge, our experimental setup is orders of magnitude larger than any other that have been reported in the literature; for example, Chang et al. [7] present a framework for associative classification rule mining using as training data, the day trading data between Jan. 1, 2005 and Dec. 31, 2007 (3 years) from the top 10 stocks listed in the S&P 500 index; their test dataset is the day trading data of the same stocks for the year 2008. In comparison, our dataset is 24.6 million rows, which is about 1000 times larger than theirs.
- Most previous studies involved the popular chart formations with the assistance of technical indicators to produce trading signals (Maginn et al. [24]; Wu et al. [38]), because the technical analysis' chart formations do not appear in sufficient frequency to produce trading signals. Here, the *great number of occurrences of the newly discovered patterns* and the proper utilization of them is sufficient to produce trading signals at all times *without* the assistance of any technical indicator.
- By the utilization of *diversity* and *complexity* behavior of price patterns we were able to produce a complete "*Leading*" type Trading Strategy giving signals before the new trend or reversal occurs, whereas almost every technical indicator is a lagging indicator and as such gives signals "after the fact."

2 Methods and application

We begin our discussion with a high-level description of our overall framework architecture, and then proceed to discuss each component's role and functionality within our framework.

2.1 System description

In Fig. 1 we show a high-level diagram of the entire system architecture, consisting of five major subsystems: (a) Patterns Creation System (prototype patterns); (b) Pattern Matching System; (c) Multi-time-frame SQL Rule Analyzer, (d) Validation System and e) Trading Strategy.

The "Patterns Creation System" lies at the heart of the whole system. It is responsible for producing template grid patterns, called prototype patterns, of various dimensions (10×10, 15×10, 20×15, 25×15) obeying specific rules for confirming the diversity of the patterns (to be explained shortly). Previous studies have used a template grid (TG) of size 10×10 to adjust a specific chart formation such as saucer, bull/bear flag and then tried to find out how many times the chart formation appears. Discovering new patterns which have not been published/known before is a challenging data mining issue to resolve (clearly, exhaustive search is not an option, as in a $M \times N$ grid there are in total M^N different patterns to be investigated.) The Pattern Matching System employs a fast algorithm for searching quickly

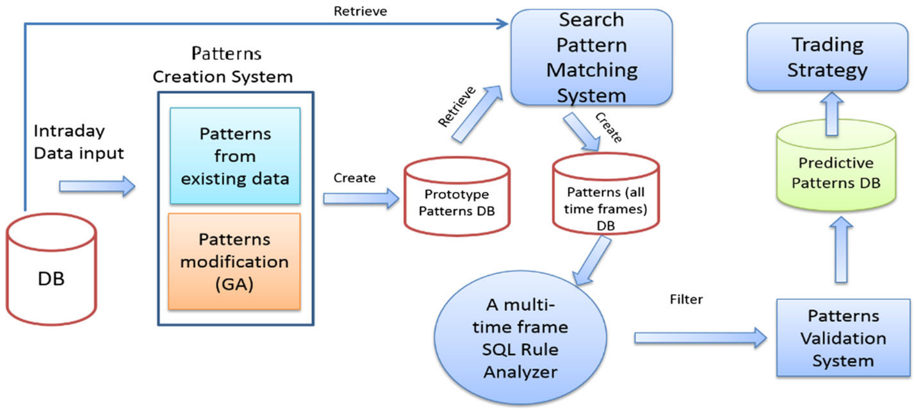

Retrieve
Intraday Data input
Patterns Creation System
Patterns from existing data
Patterns modification (GA)
DB
Create
Prototype Patterns DB
Retrieve
Search Pattern Matching System
Create
Patterns (all time frames) DB
A multi-time frame SQL Rule Analyzer
Filter
Patterns Validation System
Predictive Patterns DB
Trading Strategy

Fig. 1 The system architecture

Table 1 Notation used in the paper

Symbol	Meaning
TG	Template Grid: a two-dimensional grid of dimensions ($M \times N$)
tf	time-frame length
P_i	time-series segment representing close-price movement in as many periods as columns in its TG
sim(P,p)	the similarity between a prototype pattern P and a pattern p
F_i	number of occurrences of prototype pattern $Prot_i$
attr	a tuple containing the parameters characterizing a pattern
pred	a tuple containing values about forecasting properties of the pattern
$PA_k(Prot_i)$	Prediction Accuracy of kth predicate variable of prototype pattern $Prot_i$
TD	Trading Decision Variable (taking values from a finite set)
TB	Trend Behavior of a pattern (takes value from a finite set)
$FP(Prot_i)$	Forecasting power of prototype pattern $Prot_i$

the entire database to match patterns with a given array of prototype patterns. The matched patterns are stored in a relational database for further analysis using the Multi-time-frame SQL Rule Analyzer and subsequent validation by the validation sub-system. The predictive database is used by the Trading Strategy sub-system which runs on out-of-sample data. We discuss each sub-system in detail in the next subsections.

2.2 Patterns: design/methodology

Before we continue, for ease of reading the rest of the paper, we present a short table (Table 1) with the notations used in the rest of the text.

2.2.1 Template grid and prototype patterns creation system

In order to discuss in detail our system, we first provide a few definitions that relate to how we represent a given price pattern in proper codes that can be processed and further compared

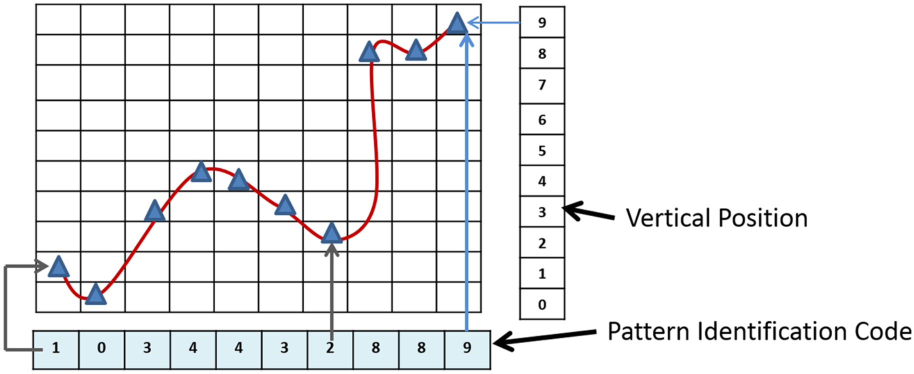


Fig. 2 Coding patterns from chart formations. The vertical positions of the price line are essentially indices starting at zero. The produced one-dimensional array of values [1, 0, 3, 4, 4, 3, 2, 8, 8, 9], which represents the data points (price value) positions of the line during the time, is the Pattern Identification Code (PIC)

with other patterns, what we mean by "Template Grid" and what we mean by "prototype pattern." We need those definitions together with a definition of a pattern similarity metric, so as to be able to define whether a new chart formation can be classified as belonging to a particular "prototype pattern" class.

Definition 1 (*Template Grid*) We define a Template Grid (TG for short) to be a two-dimensional table of dimensions $M \times N$ used for capturing chart formations in which each column corresponds to a specific time t (the latest column is the most recent time t_1, the previous columns being $t_1 - tf, t_1 - 2tf, \ldots t_1 - (N-1)tf$ where tf denotes the time-frame length). The vertical column is used to present the position of the point of chart formation for a specific time, as the example in Fig. 2 shows. The lowest point in a column represents the lowest price value of the specific time window, while the highest point represents the highest price value for the same time window. Obviously, the length of the time window equals the horizontal dimension of the grid times the time-frame length. We dynamically assign to each grid cell a weight that we then use to estimate the similarity between two chart formations according to a formula we discuss below.

Definition 2 (*Pattern Identification Code (PIC)*) Every possible chart formation is encoded as a one-dimensional array which contains the positions of the shape (chart) in a given TG; this array becomes the PIC of the pattern (again, see the example in Fig. 2).

Definition 3 (*Candidate Prototype Pattern*) We define a Candidate Prototype Pattern with respect to a particular time-series window and TG, to be a chart formation of length equal to the width of the TG that occurs at least a certain number of times in the overall time series.

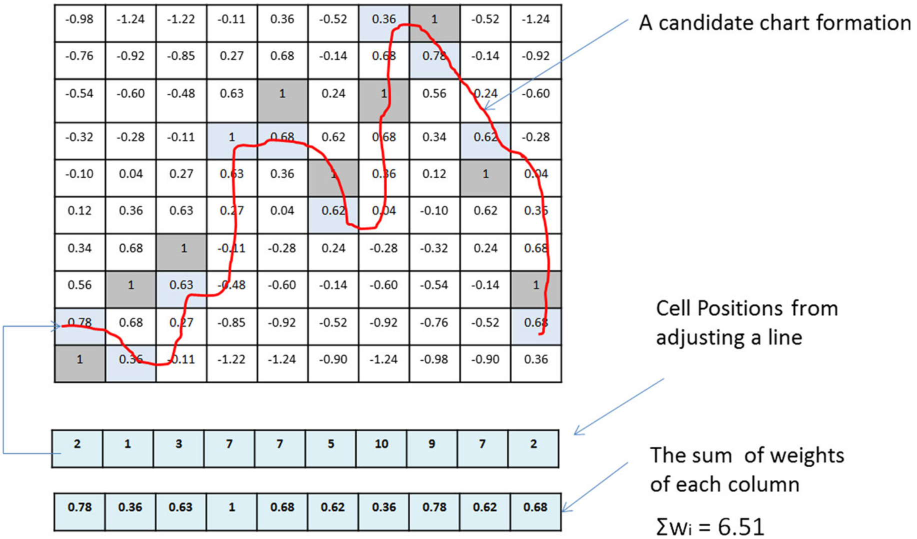
Definition 4 (*Prototype Pattern*) A Prototype Pattern is a Candidate Prototype Pattern selected for discovering similar price patterns appearing in different time periods.

The next thing we need is to define what we mean by "similar" price patterns. To define pattern similarity, we first introduce weights to each cell of a TG applying for a particular prototype pattern, by applying the following method: for each column c of the TG for a particular prototype pattern, the weights of this column are calculated according to the formula $w_{j,c} = 1 - |p - j|D$, $j = 1 \ldots M$ where $D = \frac{2M}{[(M-p)(M-p+1)+(p-1)p]}$ and p is the position of the prototype pattern in the grid column (see Fig. 2). In general, the position of a cell in a grid column that will be assigned the value 1 at any given time, given a price value p is computed as $pos = [M(p - L)/(H - L)]$ where L, H represent the lowest and highest (close) prices during the selected period and Mrepresents the vertical dimension of the grid (number of rows). It is easy to see from the definition of the value D that the sum of all weights along any given column is exactly zero, as in the method of Wang and Chan [36]. Given the TG weights for a prototype pattern, we define the similarity (or rank) of a new pattern (chart formation) to be the average over all grid columns of the weight of the TG cell in which the new pattern sits, multiplied by 100, as follows:

Definition 5 (*Pattern Similarity*) For a given TG and prototype pattern$Prot_i$, the *similarity* of a pattern p_m is defined as $sim(Prot_i, p_m) = 100 \sum_{i=1}^{N} w_{r_i,i}/N$ where r_i is the position of the pattern p_m in the ith column, and N is the horizontal dimension of the TG (number of columns).

The similarity measure attempts to capture the similarity of a candidate chart formation (pattern) to that of the prototype's chart formation, and it is a direct application of the standard template matching methods in image processing [8]. It is easy to see that given the previous definitions, the similarity of a pattern given a prototype pattern will always be less than or equal to 100, with equality achieved only when the prototype pattern, and the pattern itself match their positions *exactly* in all the given TG columns. In our research we consider that a price pattern (P_m) belongs to a prototype pattern $(Prot_i)$ if *sim* $(Prot_i, P_m) \geq 60$.

It is worth noting that restricting each grid column to have only one cell value equal to "1," not in three (3) cells as in Wang and Chan [36], proved to be a significant improvement, because it allows to make more accurate comparisons about the similarity of two patterns. To see why, consider, in Fig. 3, a line that passes in the neighboring cells of those having value equal to "1." In our methodology the similarity is 65.1%, while with the previous methodology the similarity would be exactly 100%.



-0.98	-1.24	-1.22	-0.11	0.36	-0.52	0.36	1	-0.52	-1.24
-0.76	-0.92	-0.85	0.27	0.68	-0.14	0.68	0.78	-0.14	-0.92
-0.54	-0.60	-0.48	0.63	1	0.24	1	0.56	0.24	-0.60
-0.32	-0.28	-0.11	1	0.68	0.62	0.68	0.34	0.62	-0.28
-0.10	0.04	0.27	0.63	0.36	1	0.36	0.12	1	0.04
0.12	0.36	0.63	0.27	0.04	0.62	0.04	-0.10	0.62	0.36
0.34	0.68	1	-0.11	-0.28	0.24	-0.28	-0.32	0.24	0.68
0.56	1	0.63	-0.48	-0.60	-0.14	-0.60	-0.54	-0.14	1
0.78	0.68	0.27	-0.85	-0.92	-0.52	-0.92	-0.76	-0.52	0.68
1	0.36	-0.11	-1.22	-1.24	-0.90	-1.24	-0.98	-0.90	0.36

2	**1**	**3**	**7**	**7**	**5**	**10**	**9**	**7**	**2**

0.78	**0.36**	**0.63**	**1**	**0.68**	**0.62**	**0.36**	**0.78**	**0.62**	**0.68**

A candidate chart formation
Cell Positions from adjusting a line
The sum of weights of each column
ΣWi = 6.51

Fig. 3 Calculating the similarity of a pattern (shown in *red*) with respect to an existing prototype pattern embedded in the TG (color figure online)

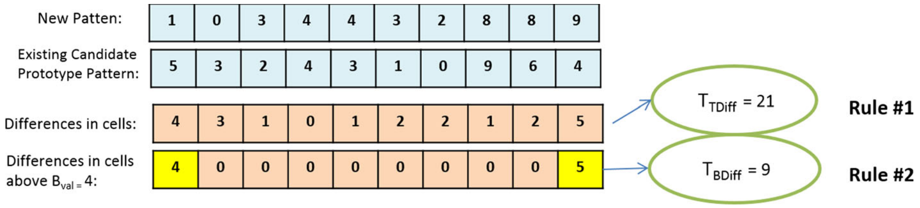
Given the above definitions, we must now determine which candidate prototype patterns ("shapes") should be selected as prototype patterns. As mentioned before, constructing all possible codes is not possible because it creates countless combinations which end up with many, useless or very similar prototype patterns. Instead, we create the PICs of the Candidate Prototype Patterns by scanning the historical data of the first one third interval of the total time series, utilizing a shifting time window of length equal to the width of our TG, and, in order to introduce some more variety, by involving Genetic Algorithms; still, the issue of creating similar candidate prototype patterns remains. To avoid this problem we designed a "*Prototype Pattern Similarity Avoidance*" technique, so that any specific discovered pattern has low probability (<0.2) to belong to more than one candidate prototype pattern, via two rules:

- Rule #1: The sum of differences between the cells of PIC between the new pattern and each prototype pattern is greater than or equal to a certain threshold T_{TDiff}"Threshold for Total Differences"
- Rule #2: The sum of differences between the cells of PIC between the new pattern and each prototype pattern that have difference greater than or equal to B_{val} "Big Value" is greater or equal to T_{BDiff}"Threshold for Big Differences"

Notice that Rule #1 gives only the general picture (total sum) of pattern differences. It cannot determine whether significant individual differences exist in any enough columns in the grid; this is why Rule #2 is needed.

The technique is applied during the prototype pattern discovery process, by requiring that in order for a candidate prototype pattern to be selected as a member of the discovered prototype patterns, it must obey both the above two rules, as shown in the example of Fig. 4 which demonstrates the rules used to create a new pattern with an existing candidate prototype pattern for the 10×10 grid (obviously, these patterns obey the rules, are therefore considered as different, so the new pattern is characterized as new prototype pattern and can be stored in the database). Selecting "optimal" values for the variables T_{TDiff}, B_{val} and T_{BDiff} is far from trivial because of the required computational power; the SQL server alone runs more than 600 database procedures and views for 24.6 million rows of data. We resorted to a systematic methodology by setting initially very small values for the threshold variables, use them to produce a set of prototype patterns via the process shown in Fig. 5, and explained further below, and examine the results to see if at least 70% of the entire set of chart formations in the time series has a similarity greater than or equal to 60% for at least one of the produced prototype patterns, and if not, increase one of the variables at a time by one and repeat the process. This scheme does not guarantee convergence to a global optimum, but it always results in a set of values that produce very acceptable behavior, in terms of the time-series coverage, similarity and frequency of occurrence of the produced prototype patterns. We do not formulate this process as a multi-objective decision problem . Via our methodology, we determined that setting the threshold parameters mentioned above to values $T_{TDiff} = 20$ (rule #1) and $B_{val} = 4$, $T_{BDiff} = 8$ (rule #2) results in a scheme that produces prototype patterns that are sufficiently different from each other, so that any given chart formation in the data has less than 20% probability of belonging to more than one prototype pattern. The resulting patterns could then be exploited by our trading system (see Sect. 2.3) to produce positive net profits (our objective problem).

As already mentioned, we employ a Genetic Algorithm (GA) to help with the creation of more prototype patterns. Figure 5 visualizes the process of creating new prototype patterns. In the initial step, the prototype creator captures random patterns from historical data. The GA creates an initial population from historical data whose size varies depending on the



New Patten:	1	0	3	4	4	3	2	8	8	9
Existing Candidate Prototype Pattern:	5	3	2	4	3	1	0	9	6	4
Differences in cells:	4	3	1	0	1	2	2	1	2	5
Differences in cells above B_{val} = 4:	4	0	0	0	0	0	0	0	0	5

Fig. 4 Comparing two prototype patterns (10 × 10) about their similarity/diversity

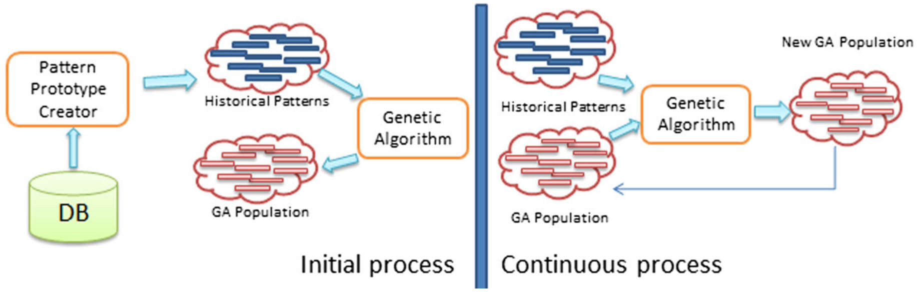


Fig. 5 The process of creating prototype patterns

dimensions of the grid (100 for the 10 × 10, 200 for the 15×10, 500 for the 20×15 and 1000 individuals for the 25×15 grids), and then proceeds for up to 600 generations creating new populations of prototype patterns using the patterns represented in the current population. The GA runs in parallel with the process of creating patterns from the historical database so that in each of its iterations, in the combination phase of selecting parents to produce offspring, the GA combines a pair of one prototype pattern from historically collected patterns and one from the existing GA population.

The chromosomes representing prototype patterns are encoded naturally as fixed-length integer arrays. We use standard 1-point crossover as the main recombination operator for the next population generation. In the mutation process, chromosomes are randomly selected for mutation, in which case, two of their genes (randomly selected) change their values to another value in the range $[1, M]$ where M is the grid-height. The selection process involves the rules described in Fig. 4. Each candidate pattern is compared to each of the existing prototype patterns in the database. If it is different according to the rules, then, it is added to the existing population; otherwise, it is discarded from the population. The above process is repeated not only for each different grid, but also for each selected time frame (1, 20, and 60 min). In our experiments, a total of 16,333 different prototype patterns were created, 20% of which were created by the GA process we ran for each grid and time frame.

2.2.2 Prediction types/analyzing price patterns

Intuitively, we hypothesized that by enabling simultaneously many different time windows for predictions we should increase the chance of having greater support level for our patterns; the results reported in Sect. 3 validate this hypothesis. Before we continue further we need to provide a few more definitions.

Definition 6 (*Pattern attributes/predicate variables*) We represent any pattern (chart formation) P_i by a tuple representing its attributes (*attr*) and a tuple representing its predicate

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

Predicate variable (k)	Intervals: next periods (i)	Predictive type (ψ)	Change price in pips (v) for time frames		
			01 min	20 min	60 min
1	Within next (1) period	Lowest price is less than	−2	−5	−10
2	Within next (1) period	Highest price is greater than	2	5	10
3	Within next (5) periods	Lowest price is less than	−5	−12	−20
4	Within next (5) periods	Highest price is greater than	5	12	20
5	Within next (10) periods	Lowest price is less than	−10	−25	−35
6	Within next (10) periods	Highest price is greater than	10	25	35
7	Within next (20) periods	Lowest price is less than	−15	−35	−45
8	Within next (20) periods	Highest price is greater than	15	35	45
9	Within next (50) periods	Lowest price is less than	−30	−50	−80
10	Within next (50) periods	Highest price is greater than	30	50	80

variables (*pred*): $P_i = (attr, \{pred, result\})$. In more detail, $attr = (tf, prot, ptnCode, rank, pipsRange, occurrences)$, $pred = (tf, i, \psi, p)$ and $result \in \{TRUE, FALSE\}$ where *tf* = time frame (e.g., 5 min), *ptnCode* is its actual PIC, *prot* is the prototype pattern to which the pattern belongs, *rank* is the similarity between the pattern and its prototype pattern, *pipsRange* is the difference between highest–lowest price value of the pattern expressed in pips, *occurrences* is the number of times the prototype pattern to which the pattern belongs appears in the entire time series (the pattern P_i itself counting as one of the occurrences), i is the time-window length (in periods) we look ahead for the pattern price changes, ψ is a binary predicate type which can take one of two values: greater than ($>$) or less than ($<$), and p is the value in pips.

For example, the expression: {*attr*, *pred*, *result*} = { (20, [2, 1, …, 10], *prot*, 10, 50), (20, 5, >, 12), *TRUE*} means that prediction of the pattern with PIC = [2,1,…10] "*pattern time frame 20 min, within next 5 periods (100 min), highest price is greater than* ($>$) *12 pips*" is *TRUE* (highest price within 5 periods from the end of this time-series segment was indeed above 12 pips). As mentioned above, we define the occurrences (F_i) of prototype pattern ($Prot_i$) to be the total number of price patterns (P_m) in the time series belonging to $Prot_i$.

Definition 7 We define the Multi-Predicate Instance $M\text{-}PRED_i\,(Prot_i)$ of a prototype pattern to be the set of supported predicate variables (defined in Table 2) which characterize each price pattern and denoted in the following form:

$$M\text{-}PRED_i\,(Prot_i) = \left\{\left\{pred_1, r_1\right\}, \left\{pred_2, r_2\right\}, \ldots \left\{pred_k, r_k\right\} \ldots \left\{pred_{10}, r_{10}\right\}\right\},$$

where $r_1, r_2 \ldots r_{10}$ are results for each corresponding $pred_k$

Definition 8 We define the Prediction Accuracy (%) (PA_k) of *k*th predicate variable ($pred_k$) referred to a prototype pattern ($Prot_i$) to be the quantity $PA_k = 100 *$

$(number\ of\ P_m\ having\ r_k = TRUE)/(Total\,number\,of\ P_m)$, or simply $100 \times VF_i/F_i$, where r_k is result (true/false) of the kth predicate variable ($pred_k$), and the Valid Occurrences (VF_i) of a prototype pattern ($Prot_i$) are defined as the number of chart formations $(P_m) having\ r_k = true$ for at least one k. We define the Prediction Accuracy (%) (PA) (alternatively, pattern accuracy) of a prototype pattern to be the maximum of prediction accuracy of all predicate variables $(r_1, r_2, \ldots r_k, \ldots r_{10})$, or $\max(PA_1, PA_2, \ldots PA_k, \ldots PA_{10})$.

Definition 9 The Validated Prediction Accuracy of kth predicate variable ($pred_k$) is defined to be the Prediction Accuracy of the kth predicate variable ($pred_k$) as long as the predicate has result $(r_k) = TRUE$ and $PA_k \geq 60\%$.

Finally, we define what we mean by "pattern forecasting power" and "pattern support."

Definition 10 Forecasting Power (FP) of a prototype pattern ($Prot_i$): We say that a prototype pattern has "Forecasting Power" as long as there is at least one $pred_k$ whose result $r_k = TRUE$. For a pair of prototype pattern ($Prot_i$) and its Multi-Predicate Instance ($M\text{-}PRED_m$), FP is defined as $FP(Prot_i) = TRUE \Leftrightarrow \exists(pred_k, r_k) : r_k = TRUE$. For specific currency pair and time frame we define pattern support as the following quantity: 100×(number of time series rows which form at least one pattern of any dimension of the template grid which match (similarity $> 60\%$) with a prototype pattern having forecasting power) / (total number of time series rows in a format of the given time frame)%.

Table 2 shows the 10 predicates used in our research. The values (in pips) of predicate variables were determined empirically also by running the search algorithm many times in a sample of data and getting as best values those that can find patterns with pattern accuracy above 60% and have minimal support of 1%.

In words, the prediction accuracy of a given chart formation is 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 introduce a factor called "*pattern's pips range*" which represents the difference between the highest and lowest value of a grid pattern expressed in pips value. This factor proved to be very important in estimating the forecasting power of a pattern, but we defer detailed discussion until Sect. 5.1. We briefly mention here that as an example, for patterns of time frame 1 min, only those patterns which have pips range greater than 20 pips have forecasting power (FP). This threshold value (20 pips) depends on the time-frame length and it grows with larger time frames. This observation has been validated in all our runs.

Then, the Patterns Validation System examines all training and testing sets by applying threefold cross-validation (defined in Sect. 3.3) which additionally determines the prediction accuracy of the algorithm as the average of all combinations.

2.3 Strategy design

The final step is to test whether the discovered prototype patterns are actually useful in trading decision support systems. Toward this end, we designed a straightforward strategy based only on patterns, interpreting their predictions as signals, and evaluated the results of running this strategy on a large number of different setups.

To avoid losses, we employ a "Trailing Stop-Loss Function"—one for long positions and one for short positions. This additionally helps maintain profits. The stop loss is expressed in pips (constant value). When the price goes in the "right" direction, it drags the stop along with it, but when the price reverses direction, the stop-loss price remains at the level it was dragged to. At this point, we are ready to explain in full detail in the following section, how we turn the predictions made by our validated prototype patterns into buy/sell decisions.

2.3.1 Strategy for exploiting patterns

To prove the usefulness of the discovered prototype patterns, but for reasons of completeness as well, we devised a straightforward patterns-based strategy, which we call "*Grid Patterns Strategy*," or more simply, "*Patterns Strategy*" that uses the output of the validated prototype patterns' predictions as signals for trading decisions. To understand exactly how we use the signals provided by the patterns in our decision making process for determining entry/exit points, we give a few more definitions.

We define a Trading Decision variable (*TD*) to be a discrete variable with domain of definition the set $\{NOT\ TRADE,\ CONFLICT,\ ENTER\ LONG,\ ENTER\ SHORT\}$. We define "Trend Behavior (*TB*)" of a pattern to be "Bearish" or "Bullish" or "NoTrend," according to the value of the quantity $Trend(TB) = (r_2 + r_4 + r_6 + r_8 + r_{10}) - (r_1 + r_3 + r_5 + r_7 + r_9)$, where $r_1, r_2 \ldots r_{10}$ are probabilities % for each of the corresponding $pred_k$ defined in Table 2. Given a threshold $h_o > 0$, a Trend Behavior is Bullish if $TB > h_o$, Bearish if $TB < -h_o$, NoTrend if $TB \in [-h_o, h_o]$. Now, the Trading Decision variable that a *prototype pattern generates* at any point in time may take values from the restricted set: $\{NOT\ TRADE,\ ENTER\ LONG,\ ENTER\ SHORT\}$, according to the following rules:

$$\begin{aligned}
&TD = ENTER\,LONG \Leftrightarrow MAX\,(r_1, r_2 \ldots r_{10}) = MAX\,(r_2, r_4, r_6, r_8, r_{10})\ \text{, and}\\
&TB = \text{Bulish: } (\textit{price will increase above a specific level})\ .\\
&TD = ENTER\,SHORT \Leftrightarrow MAX\,(r_1, r_2 \ldots r_{10}) = MAX\,(r_1, r_3, r_5, r_7, r_9)\ \text{, and}\\
&TB = \text{Bearish: } (\textit{price will decrease below a specific level})\ .\\
&TD = NOT\ TRADE \Leftrightarrow TB = \text{NoTrend}
\end{aligned}$$

The above simple rules essentially specify when we consider the prediction of a single prototype pattern as an "*ENTER LONG*," "*ENTER SHORT*" or "*NOT TRADE*" signal. Of course, at any given time, we have to consider the predictions of all relevant prototype patterns in our database. In our devised strategy, the rules that produce the final decision of the system are the following:

$$\begin{aligned}
&TD = ENTER\,LONG \Leftrightarrow Each\ M\text{–}PRED\\
&\quad\ \text{is ENTER LONG or NOT TRADE}\textit{AND At least one}\ \text{M–PRED}\\
&\quad\ \text{is ENTER LONG .}\\
&TD = ENTER\,SHORT \Leftrightarrow Each\ M\text{–}PRED\\
&\quad\ \text{is ENTER SHORT or NOT TRADE}\textit{AND At least one}\ \text{M–PRED}\\
&\quad\ \text{is ENTER SHORT.}\\
&TD = NOT\,TRADE \Leftrightarrow Each\,M\text{-}PRED\ \text{is NOT TRADE}\\
&TD = CONFLICT \Leftrightarrow At\,least\,one\ M\text{–}PRED\\
&\quad\ \text{is ENTER LONG AND least } \textit{one M–PRED}\\
&\quad\ \text{is ENTER SHORT.}
\end{aligned}$$

The rules above, together completely specify our strategy, as they always decide when to open and close a trading position (long or short). Additionally, to increase the performance of the strategy, we have devised *two important filters*: the *pips range* and the *price-level-bands*

```
Input:
ThresholdBands;
For each time series data (row) starting from the oldest in time
  Create Pattern Identification Codes for the current row (for each TG dimension);
  Search above if they exist in the predictive database while having pattern similarity > 60;
  If found more than one prototype patterns for a specific grid dimension, then choose the prototype pattern
with the maximum similarity value
  The prototype pattern can be accepted if the following are true (according to the two filters described in the
text):
        (a)   Current Pips Range (CPR) >= Min Pips Range of prototype
        (b)   Current Price Level (PL) > Average Price Level of Prototype Pattern - ThresholdBands
        (c)   Current Price Level  (PL)< Average Price Level of Prototype Pattern + ThresholdBands
  If any accepted prototype pattern exists, make appropriate prediction for long or short position;
  If prediction is "ENTER LONG" then open long position (if long is not open) and close existing short
position (if short is open)
  Else If prediction is "ENTER SHORT" then open short position (if short is not open) and close existing long
position (if long is open)
  Else do nothing;
  Check for stop loss position;
End For
```

Fig. 6 Strategy trading algorithm

filter. The pips-range filter specifies that the current chart formation's pips range is greater than a specific level value which depends on the time frame. To describe the price-level-bands filter, we define the Price Level (*PL*) of *x* periods to be the position of the current price relative to the highest and lowest prices of specific periods ago:

$$PL(x) = 100 \cdot \frac{current\,price - lowest\,price\;of \times periods\;ago}{highest\,price\;of \times periods\;ago - lowest\,price\;of \times periods\;ago}$$

The class of a prototype pattern has a numerical property called Average Price Level (*APL*) which is the Average *PL* of all matching data points of a specific time interval. The price-level-bands filter employed a user-defined quantity *ThresholdBand*, and simply checks whether $PL(100) \in [APL\text{-}ThresholdBan,\ APL+ThresholdBand]$.

Figure 6 formalizes our strategy in pseudocode; the algorithm opens a position if the rules above produce a decision of value either *ENTER LONG* or *ENTER SHORT*. An opened position is closed as soon as the rules produce opposite signals, or the trailing stop is activated to protect losses or profits.

Figure 7 shows the trading algorithm before the final trading decision, illustrating the two filters in action.

3 Empirical results

We implemented the entire system in the Microsoft .NET framework (.NET 4.5, WPF C#) and runtime using MS SQL Server 2012. The architecture of the system and the algorithmic implementations take advantage of both shared-memory and distributed memory parallel programming, exploiting to the fullest extent possible all the advantages of the many cores found in today's off-the-self hardware as well as the fast cluster interconnects for modern data centers, thus allowing the system to scale to very large data sizes as our experimental results show. In our architecture, one of the available computing nodes takes on the role of a *proxy scheduler* that is responsible for assigning tasks to the registered computers in the cluster. A total of 4 computers were used in our experiments, equipped with Intel core i7 3770

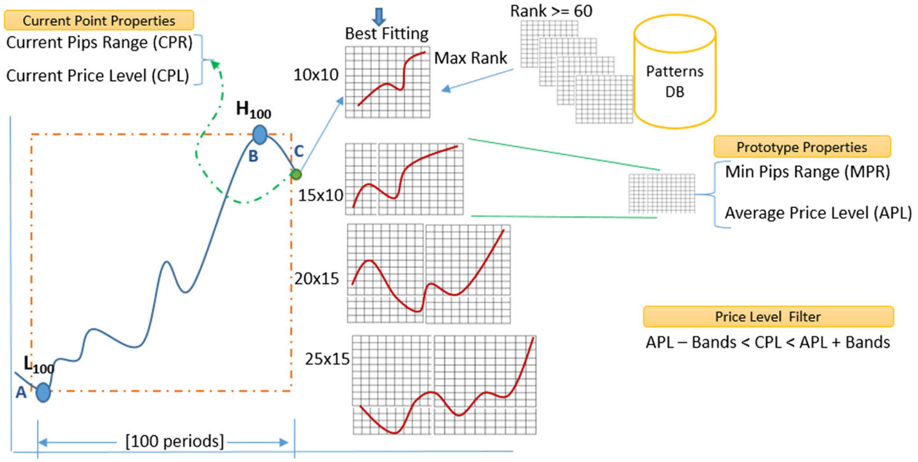


Fig. 7 Filters illustration. In the series time curve, the point "*A*" is the lowest price (L_{100}) over 100 periods ago, the point "*B*" is the highest price (H_{100}) of the same period and the point "*C*" is the current point where we must make trading decision. For the point "*C*," we calculate the current PIC codes, one for each grid size and the current pips-range and PL values, namely CPR and CPL. Then, we query the patterns database, to find the matching prototype patterns with forecasting power. If there are more than one for a specific grid size, as best fitting (shown in the *top* of the diagram), we take the prototype pattern with max similarity to the current chart formation. Then, we apply the price-level-bands and pips-range filters. After that we will have from 0 to 4 prototype patterns that passed the criteria

and 3820 CPUs, 16 or 32 GB RAM, and multiple SSD 256 GB HDDs, running MS Windows 7 64-bit. In our configuration, each client-application (running on a different machine) was assigned to run one of the four (4) Template Grids (10×10, 15×10, $20 \times 15 and 25 \times 15$). The average wall-clock time of creating the grid prototype patterns including optimizations was about 120 min. The average wall-clock time for discovering and storing the successful patterns including all training and testing periods as well as cross-validation tests took about 24 min.

3.1 Technical trading systems

Regarding the evaluation of the strategy performance we mainly employ the *Student t test statistic*, a very well-known indicator that is commonly used to determine if a trading system is likely to be profitable in the future. Here, it is applied on the average trades net profit and determines if the average net profit is significantly greater than zero at a specified confidence level. We also report results from other more specialized indicators such as the Sharpe and the Sortino ratio that attempt to measure exposure to risk. We compare our strategy with other trading strategies by building three different trading systems based on well-known strategies that are trained and tested on the same periods of time as our proposed pattern strategy and involving the same trailing stop-loss functionality.

Trading System No1 Moving Average (MA). We develop a trading system using the moving average of close prices. The trading signals are produced by the crossover between MA and close prices. When the close price crosses over the MA the system produces an "*ENTER LONG*" signal and in the opposite (crossing below) the signal is "*ENTER SHORT*." For opening long position, first, we close any open short position (not closed by stop/loss function), while for opening short position (sell short), we first close any open long position (not closed by stop/loss function).

Trading System No2 Price Oscillator (PO) [35]. It is the difference between two moving averages, one for shorter period and one for longer period. The buy/sell signals are produced by the crossover between PO and its MA, similarly to trading system No1.

Trading System No3 Direction Movement Index (DMI) [35]. This system follows a set of simple rules: take long position and liquidate short position whenever the +DI indicator crosses above the −DI indicator. Liquidate the long position and establish a short position, when the −DI crosses above the +DI.

3.2 Random generator for trades

Additionally, we compare our pattern strategy to a system that opens and closes positions based on *generally random signals*. For this purpose, we developed a system that generates random signals that are in turn used by a proper (not random) strategy. Figure 8 shows the random signals generator algorithm, where the number of generated signals is different between executions. We should notice that there may be time regions of no trade when no signal is produced by the random signal generator. In such cases, after a time threshold is reached, we simply liquidate the trade. In summary, a current opened trade is liquidated when either we have no signal for some time, or when the signal changes sign.

Similarly, the strategy involves trailing stop / loss function for avoiding big losses. We run the random signals generator 10,000 times per dataset, per time frame, per currency pair and we report average net profits and statistics.

3.3 Data setup for testing the strategies

The initial setup consists of intraday data with a time-frame period of 1 min of totally 8 instruments, namely: Australian Dollar/Japanese Yen (AUD/JPY), Euro/Canadian Dollar (EUR/CAD), Euro/Japanese Yen (EUR/JPY), EUR/USD, British Pound/Australian Dollar (GBP/AUD), British Pound/Swiss Franc (GBP/CHF), British Pound/Japanese Yen (GBP/JPY) and British Pound/US Dollar (GBP/USD). This dataset starts from Jan 2009 until end of Dec 2013 and contains approximately 14.8 million entries (for 1 min). The row data format for each entry is as follows:

Symbol, Date Time < *range of "tf" min* >, *open price, high price, low price, close price* where the "*tf*" is the time frame corresponding to the data row.

Time series data come from raw tick format data of records of the form <*time* (in seconds), *ask price*, *bid price*> and all data row values (open, high, low, close) correspond to the *average* of bid and ask price. We use threefold cross-validation test for pattern discovery/validation, while the strategy runs on out-of-sample data following a path similar to the walk forward

```
Define randomly the number of signals to be created (between 70% and 200% of the average number trades
produced by the trading systems 1-3).
Given the number of rows of the time series data create an array of index values:  TS_index = {0, 1, 2,..$n$}
For ($s$ = 1 to the number of signals to be created)
  Select randomly an index number ($i$) from TS_index and remove 5 items data ( from $i$-2 to $i$+2) of the
TS_index   // remove neighbour signals.
  Assign randomly to the index a signal number (-1 short, 0 = close all positions, 1 = long) as follows:
  Generate zero signal with 20 % probability, same signal (long or short) with 30% probability, and opposite
signal with 50% probability.
  Store index and signal number in an object list
End For
```

Fig. 8 Random signals generator algorithm

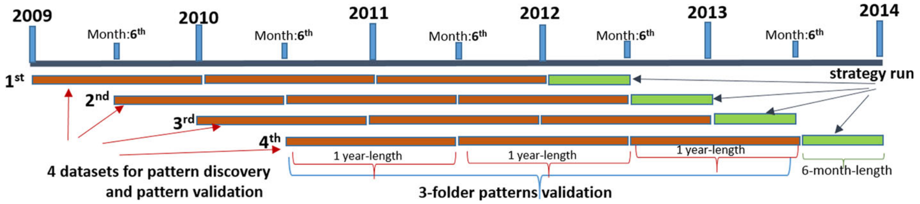


Fig. 9 Dataset splitting for pattern discovery/validation and strategy training

testing. The data were divided into four datasets where each dataset shifts 6-month ahead from the previous as shown in Fig. 9. For the strategy trading, we decided to use the walk forward testing because it results in robust trading systems and is closer to how the real markets work. According to the walk forward testing method, the training (parameters optimization) takes place with data representing a specific period of historical time series, while the next time window is out-of-sample data reserved for testing. Then, this method is repeated by shifting forward the training data by the period covered by the out-of-sample test, while the out-of-sample data similarly shift forward. The datasets used for our experiments are as follows (note 1st semester is from Jan 1 to June 30 and 2nd semester is from July 1 to Dec 31):

- 1st data setup: threefold cross-validation for patterns: Years [2009, 2010 and 2011]. Strategy Training Data: 1st semester of Year [2012]
- 2nd data setup: threefold cross-validation for patterns: Years [2nd semester 2009, 2010, 2011 and 1st semester of 2012]. Strategy Training Data: 2nd semester of Year [2012]
- 3rd data setup: threefold cross-validation for patterns: Years [2010, 2011, and 2012]. Strategy Training Data: 1st semester of Year [2013]
- 4th data setup: threefold cross-validation for patterns: Years [2nd semester 2010, 2011, 2012 and 1st semester of 2013]. Strategy Training Data: 2nd semester of Year [2013]

The cross-validation system uses 2 years for training and 1 year for testing. For example, for the 1st data setup we have the following tests: (1) training for years 2009 and 2010—validate patterns for year 2011, (2) training for years 2010 and 2011—validate patterns for year 2009 and (3) training for years 2009 and 2011—validate patterns for year 2010. The 6-month period of data used by the strategy training system always follows the 3-year pattern discovery/validation process, and it is out-of-sample data.

3.4 Prediction accuracy of the discovered patterns

The threefold cross-validation test was executed separately 3 times per instrument (all 8 currency pairs in total), per dataset (numbered #1, # 2, #3, #4), per time frame, and per type of template grid (totaling 1152 times). Table 3 summarizes the accuracy of the algorithm per dataset, per instrument and per time frame: the column named “Cross Test” depicts the average of prediction accuracy (Definition 9) of the threefold validation tests, and the next column “Training” is the prediction accuracy of total 3-year-data. The results shows that four currency pairs (AUD/JPY, EUR/USD, EUR/CAD, GBP/USD) at specific time frames (left side of the table) have cross-test prediction accuracy less than 60%. On the other hand, on the right side of the table, the other four currency pairs pass all validation tests and more specifically some of them have great prediction accuracy (such as GBP/AUD, GBP/CHF).

Table 3 Prediction accuracy of the discovered prototype patterns

Dataset	Instrument	Timeframe	Cross test	Training	Instrument	Timeframe	Cross test	Training	Dataset	Instrument	Timeframe	Cross test	Training	Instrument	Timeframe	Cross test	Training
1	AUD/JPY	1	61.18	62.48	EUR/USD	1	61.89	63.59	1	EUR/JPY	1	63.06	63.73	GBP/JPY	1	64.55	64.90
2		1	61.19	63.10		1	60.76	62.94	2		1	61.98	63.26		1	63.51	64.58
3		1	61.04	63.17		1	61.12	63.10	3		1	61.57	63.35		1	63.26	64.46
4		1	60.73	62.13		1	60.02	62.68	4		1	61.79	63.41		1	63.25	63.96
1		20	63.03	64.46		20	66.37	65.04	1		20	68.08	66.34		20	**72.78**	69.91
2		20	62.39	62.83		20	65.01	63.95	2		20	64.46	65.09		20	67.72	66.79
3		20	**58.78**	61.18		20	61.06	63.08	3		20	62.10	63.92		20	63.61	65.00
4		20	60.36	60.99		20	**59.91**	62.46	4		20	64.40	63.82		20	65.26	64.83
1		60	66.51	65.76		60	69.96	66.51	1		60	**71.54**	68.10		60	**75.23**	72.42
2		60	64.09	64.17		60	67.64	64.90	2		60	66.99	65.75		60	**71.59**	68.80
3		60	**58.27**	63.77		60	61.73	64.49	3		60	63.97	64.70		60	66.83	65.74
4		60	62.49	63.40		60	60.62	63.71	4		60	66.27	64.85		60	68.35	65.54
1	EUR/CAD	1	62.00	65.03	GBP/USD	1	61.96	63.99	1	GBP/AUD	1	65.23	67.40	GBP/CHF	1	63.74	64.36
2		1	60.99	62.80		1	60.42	63.42	2		1	63.49	64.70		1	63.22	64.06
3		1	60.27	62.01		1	**59.74**	63.00	3		1	62.58	64.18		1	62.97	64.00
4		1	60.44	61.63		1	**58.94**	62.11	4		1	62.96	63.68		1	62.57	63.73
1		20	67.47	65.99		20	67.73	66.46	1		20	**78.09**	75.58		20	**71.50**	69.02
2		20	63.26	63.22		20	64.58	65.29	2		20	**73.82**	71.62		20	67.28	66.01
3		20	**57.32**	61.52		20	61.22	63.90	3		20	68.71	67.12		20	62.53	65.16
4		20	**59.19**	61.36		20	60.63	62.70	4		20	66.15	65.22		20	62.94	64.23
1		60	69.16	67.30		60	**73.35**	70.08	1		60	**79.96**	77.00		60	**74.92**	72.16
2		60	64.67	65.32		60	68.70	66.60	2		60	**76.10**	72.86		60	**70.29**	67.70
3		60	**56.26**	64.23		60	60.98	65.33	3		60	**70.64**	67.98		60	61.66	65.66
4		60	**59.53**	64.05		60	61.12	64.43	4		60	68.33	66.23		60	62.12	65.02

These resulting figures are shown in bold

3.5 Strategy performance

In this subsection, we report our experimental results. To run our experiments, regarding money management, we opted for the simplest scheme possible, called "Fixed Dollar Amount." The concept is simple: for every trade we use the same fixed amount from our account. We use 1 lot size contract (which amounts to USD $100,000). We made the very conservative assumption that the margin level is low for the spot forex, and in particular 0.1 (1:10). The initial capital used for running the strategies was USD $20,000 with leverage 1:10 and exposure to margin account 50%; 1:10 leverage means that for buying 1 lot, it is required to have in the trading account USD $10,000; with exposure 50%, the initial account must have at least USD $20,000. All calculations are done without any profit reinvestments, and each trade is 1 lot.

It is important to mention that all calculations shown in the rest of this section are made with the above settings. Had we used no leverage at all (1:1 leverage, in which case "exposure" becomes meaningless), the initial capital needed would have been USD $100,000 and the net profits reported in Tables 4, 5, 6 and 7 would need to be divided by 5; the Sharpe and Sortino ratios would have increased as well, showing much reduced risk taking in the process.

During the trading simulation, if at any point the losses decrease the account below the $20,000 limit that is required for trading 1 lot, the simulation decreases the amount for trading to maintain the 50% exposure. In any case, *there is never any "new money" added to the account during the simulation*, so that in case the accumulated losses result in a zero account balance, the simulation simply stops and reports that all the initial account investment was lost.

Another issue that requires mention is the spread commissions that must be taken into account in order to accurately estimate trade performance of any system. Each currency pair has different spreads that vary with time. When calculating the net profits of each trade, we subtract the cost of spread which is the difference between the buy and sell rate of the currency pair. Some forex providers offer fixed-spread values, others variable spreads. For all calculations we used reasonable fixed-spread values that are currently in use by many providers. For example, for EUR/USD, in Sect. 3, we run experiments assuming 0.5 pip spread, which means for each trading of 1 lot, we estimate $0.5 \times \$10 = \$5$ per transaction. Since a complete trade has two transactions (open and close position) the spread commissions cost is estimated at $10 per trade. However, we also ran experiments with much higher spread values (for EUR/USD equal to 0.7 and 0.9), and even without using any leverage at all, and report results in Table 12 (Sect. 5).

3.5.1 Technical trading systems report

The performance results of the selected eight (8) instruments (currency pairs) showed that, as expected, in lower time-frames (1, 20 min) technical trading systems using only technical indicators as defined in Sect. 3.1 cannot "beat the market". Actually, for 1 min time frame, none of the three (3) trading systems using technical indicators managed to bring positive net profits (lost all capital in the first 2–3 months of trading). For 20 min time frame, four (4) currency pairs lost all capital (−100%), while for the rest four (4), the net profits were between 4.89 and 22.82% having very small Sharpe ratio (< 0.3). For the higher time frame of 60 min, the results were better, the net profit percent was between −16.06 and 53.80% having however very low Sharpe ratio value (< 1). The very limited positive results brought by the three technical trading systems must be attributed to the use of trailing stops with low value.

Table 4 Performance of random signal-based strategy: worst and best-case report

	% avg profit	AVG t test	% win trades	% win runs		% avg profit	AVG t test	% win trades	% win runs
GBP/AUD					EUR/JPY				
01 min					01 min				
Dataset 1	−63.21	−2.48	41.34	5.40	Dataset 1	11.21	0.19	48.11	57.50
Dataset 2	−69.48	−3.22	39.01	1.20	Dataset 2	−38.47	−0.06	47.72	50.10
Dataset 3	−61.31	−2.34	41.19	5.80	Dataset 3	43.32	0.66	48.12	64.30
Dataset 4	−56.58	−1.69	41.65	9.40	Dataset 4	−42.96	−0.05	47.73	48.40
20 min					20 min				
Dataset 1	−47.24	−1.67	43.87	10.20	Dataset 1	10.61	0.17	48.17	58.90
Dataset 2	−52.74	−2.15	42.60	4.70	Dataset 2	0.71	−0.05	48.11	50.80
Dataset 3	−48.12	−1.62	43.29	10.50	Dataset 3	13.84	0.14	46.56	56.30
Dataset 4	−41.22	−1.20	43.44	16.50	Dataset 4	1.28	0.00	48.11	52.20
60 min					60 min				
Dataset 1	−28.10	−0.95	44.61	24.00	Dataset 1	8.03	0.11	47.58	56.00
Dataset 2	−26.10	−1.01	45.00	20.40	Dataset 2	2.34	−0.01	47.72	51.30
Dataset 3	−23.86	−0.78	44.01	28.00	Dataset 3	−0.41	−0.13	43.98	48.50
Dataset 4	−26.76	−0.80	42.81	26.10	Dataset 4	1.16	0.00	47.64	51.60

Table 5 Overall comparison between strategies of net profits percentage across currency pairs and time frames

	AUD/JPY	EUR/CAD	EUR/JPY	EUR/USD	GBP/AUD	GBP/CHF	GBP/JPY	GBP/USD
01 min								
Grid patterns								
Net profit%	87.95%	41.01%	155.78%	98.30%	9.33%	13.68%	183.35%	61.02%
Sharpe	2.74	1.74	3.54	2.94	0.16	0.38	3.46	2.11
Trading system								
Net profit%	−100.00%	−100.00%	−100.00%	−100.00%	−100.00%	−100.00%	−100.00%	−100.00%
Radom strategy								
Net profit%	−11.63%	−49.74%	−6.73%	1.89%	−26.21%	−57.61%	21.99%	−8.17%
20 min								
Grid patterns								
Net profit%	59.10%	32.48%	93.42%	72.53%	31.79%	37.02%	107.74%	79.58%
Sharpe	1.98	1.40	2.43	2.65	1.20	1.36	2.49	2.83
Trading system								
Net profit%	−100.00%	−100.00%	7.42%	22.82%	−100.00%	−100.00%	7.68%	4.89%
Radom strategy								
Net profit%	−13.51%	−33.73%	6.61%	1.12%	−47.33%	−41.85%	7.92%	−4.29%
60 min								
Grid patterns								
Net profit%	36.89%	46.75%	70.94%	33.72%	20.97%	18.19%	38.74%	47.59%
Sharpe	1.30	1.96	2.19	1.41	0.83	0.64	1.15	2.20
Trading system								
Net profit%	42.20%	−16.06%	53.80%	41.57%	51.78%	22.63%	43.10%	35.60%
Radom strategy								
Net profit%	−7.00%	−42.09%	2.78%	0.68%	−26.21%	−23.09%	1.75%	0.09%

Table 6 Detailed comparison performance report for AUD/JPY, EUR/CAD, EUR/JPY and EUR/USD of 60 min time-frame data

	Data set no. 1	Data set no. 2	Data set no. 3	Data set no. 4	Data set no. 1	Data set no. 2	Data set no. 3	Data set no. 4
Time frame: 60 min	Australian Dollar versus Japanese Yen (AUD/JPY)				EURO versus Canadian Dollar (EUR/CAD)			
Net profit amount($)	14,636.68	5934.25	5314.93	3627.57	10,881.16	11,145.13	7257.11	8114.52
Spread commissions (S)	5376.53	4636.41	788.42	1395.10	4036.09	5829.08	2927.19	1587.07
Net profit percent (%)	73.18%	29.67%	26.57%	18.14%	54.41%	55.73%	36.29%	40.57%
Number of total trades	307	263	54	99	169	241	124	69
Percent (%) winning trades	46.58%	46.01%	33.33%	39.39%	52.07%	47.30%	42.74%	37.68%
Exposure to market	80.80%	89.10%	24.20%	46.50%	73.10%	85.50%	63.50%	63.70%
t test statistics	1.79	0.92	0.87	0.86	2.1	1.85	1.25	1.79
Sharpe ratio	2.33	1.02	1.13	0.7	2.35	2.1	1.57	1.82
Sortino ratio	4.48	1.86	3.4	1.32	4.46	4.19	3.53	5.2
Time frame: 60 min	EURO versus Japanese Yen (EUR/JPY)				EURO versus US Dollar (EUR/USD)			
Net profit amount ($)	12,455.46	14,685.68	16,734.80	12,872.40	13,416.00	6666.00	1925.00	4965.00
Spread commissions (S)	2795.83	3090.36	1664.19	1380.23	1135.00	1080.00	755.00	270.00
Net profit percent (%)	62.28%	73.43%	83.67%	64.36%	67.08%	33.33%	9.62%	24.83%
Number of total trades	319	352	227	195	227	216	151	54
Percent (%) winning trades	47.85%	50.28%	41.41%	50.26%	46.70%	46.30%	45.70%	44.44%
Exposure to market	72.30%	83.30%	53.60%	71.30%	72.60%	68.70%	45.70%	67.40%
t test statistics	1.36	1.89	1.85	1.88	2.13	1.32	1.1	1.03
Sharpe ratio	1.57	2.59	2.37	2.24	2.75	1.55	0.35	0.98
Sortino ratio	2.94	5.098	5.61	4.47	5.33	2.81	0.61	2.65

Table 7 Detailed comparison performance report for GBP/AUD, GBP/CHF, GBP/JPY and GBP/USD of 60 min Time-frame Data

	Data set no. 1	Data set no. 2	Data set no. 3	Data set no. 4	Data set no. 1	Data set no. 2	Data set no. 3	Data set no. 4
Time frame: 60 min	Pound Sterling versus Australian Dollar (GBP/AUD)				Pound Sterling versus Swiss Franc (GBP/CHF)			
Net profit amount ($)	5265.05	3226.05	5014.97	3259.22	7393.26	1164.05	3357.16	2636.44
Spread commissions (S)	2757.90	3945.71	4455.79	2946.35	4036.77	3926.49	3254.85	1282.35
Net profit percent (%)	26.38%	16.13%	25.07%	16.30%	36.97%	5.82%	16.79%	13.18%
Number of total trades	67	95	110	80	150	149	122	47
Percent (%) Winning trades	40.30%	46.32%	40.91%	41.25%	44.00%	44.30%	42.62%	38.30%
Exposure to market	40.20%	45.70%	47.90%	28.50%	67.10%	75.40%	54.50%	32.10%
t test statistics	0.96	0.68	0.76	0.67	1.2	0.24	0.54	0.66
Sharpe ratio	0.88	0.65	1.09	0.7	1.32	0.15	0.5	0.59
Sortino ratio	1.86	1.13	2.45	1.47	3.02	0.26	0.97	1.31
Time frame: 60 min	Pound Sterling versus Japanese Yen (GBP/JPY)				Pound Sterling versus US Dollar (GBP/USD)			
Net profit amount ($)	10,998.30	3548.39	11,254.14	5194.54	14,538.00	7452.00	7768.00	8313.00
Spread commissions (S)	1059.26	1362.68	788.60	520.04	1584.00	336.00	352.00	1224.00
Net profit percent (%)	54.99%	17.74%	56.27%	25.97%	72.69%	37.26%	38.84%	41.57%
Number of total trades	141	181	125	86	198	42	44	153
Percent (%) Winning trades	39.01%	44.20%	47.20%	36.05%	47.98%	52.38%	45.45%	45.75%
Exposure to market	47.20%	71.10%	32.10%	35.30%	71.50%	46.10%	28.80%	62.90%
t test statistics	1.28	0.5	1.5	0.82	2.35	1.84	1.61	1.53
Sharpe ratio	1.51	0.49	1.64	0.97	2.65	2.23	2.14	1.78
Sortino ratio	4.35	0.91	3.3	2.48	5.6	5.95	5.98	3.74

3.5.2 Random signals-based strategy report

We ran the random strategy per currency pair, per time frame, per dataset. Table 4 shows the results of the worst performing instrument, which was the GBP/AUD, and the best performing, which was the EUR/JPY. The column "% Avg Profit" presents the average net profit% of 10,000 runs, while the "AVG t test" the average of all t test statistics. The column "% win trades" presents the average (of 10,000 runs) of percentage of number of winning trades divided by the number of all trades. The next "% win runs" is the percentage of the number of runs that have positive net profit.

In Table 4, surprisingly, for the currency pair (EUR/JPY), in most datasets and time frames the random signal-based strategy did bring on average positive net profits. But the t statistic value for all cases remains very close to zero or negative, indicating that indeed, *these averages happen to be positive only by chance*.

3.5.3 Patterns strategy performance: comparison with random-based strategy and technical trading systems

In Table 5, we have summarized all results across all currency pairs and time frames for the three categories of the trading systems (our proposed "Grid Patterns"-based strategy shown in the correspondingly labeled rows, *the average of the three technical indicators-based trading systems* (in the rows labeled "Trading System"), and the random strategy, in the rows labeled "Random Strategy"). The Net Profit% and the Sharpe ratio displayed in the table (in the rows labeled "Sharpe") is the average value across the four datasets. Our Grid Patterns Strategy is far superior in lower time frames, while the technical trading systems produce a lot of false signals (in general the average of winning trades was approximately 35–37% for those systems).

The details of the performance of our Grid Patterns Strategy for the 60 min time frame are shown in Tables 6 and 7. Statistical significance of the results is provided by means of the t test statistic in the row labeled "t test Statistics." To get an idea of the risk taking involved, we report the Sharpe and Sortino ratios in the rows labeled as such (see [24] for respective definitions). Our strategy seems to perform best with low-risk values for AUD/JPY, EUR/JPY and GBP/JPY instruments.

3.5.4 Patterns strategy performance: comparison with other recent studies

In this section, we compare our system against the heuristic rule-based system results reported in Ozturk [28], and the headlines news–mining-based trading system results reported in Nassirtoussi et al. [26]. To make the comparison fair, we use exactly the same forex training and testing dataset and parameter settings (initial capital, spread, leverage etc.) that the authors of these studies used, except for the news headlines data of the latter study which our method does not utilize at all.

In Table 8 we show the results our system obtains versus the best, minimum and average results reported in Ozturk [28] on EUR/USD and GBP/USD in the time periods and time frames (1 min) the author used. For each dataset, the column "Grid Patterns" refers to the results obtained by our proposed system, and the column "Ozturk 2015 best(min/avg)" shows in the first row of results, the *best* net profit amount produced by Ozturk's systems, and within parentheses, separated by "/", the minimum, and average net profit the author reports. In the rest of the rows, the row labeled "Spread Commissions ($)" indicates the spread commission

Table 8 Comparison of grid patterns strategy versus heuristic rule-based trading system [28]

Time frame: 1 min	EUR/USD [1/5/2013–30/6/2013]		EUR/USD [1/9/2013–31/12/2013]		GBP/USD [1/5/2013–30/6/2013]	
	Grid patterns	Ozturk [28] best (min/avg)	Grid patterns	Ozturk [28] best (min/avg)	Grid patterns	Ozturk [28] best (min/avg)
Net profit amount ($)	11,407.81	13363 (−3729/**7489**)	15,214.83	14584 (−1648/**5380.5**)	5,009.85	6332 (−3037/**2612.4**)
Spread commissions (S)	0	0	0	0	0	0
Net profit percent	114.08%	133.36%	152.15%	145.84%	50.10%	62.32%
Number of total trades	615	702	607	2723	80	775
Percent (%) winning trades	61.83%	61.97%	61.29%	59.53%	56.25%	62.84%
Average profit per trade	18.54	19.04	25.06	5.36	62.62	8.04
Exposure to market	0.95	N/A		N/A		N/A
t test statistics	2.68	N/A		N/A		N/A
Sharpe ratio	4.30	N/A		N/A		N/A
Sortino ratio	11.99	N/A		N/A		N/A

Bold values indicate the average net profit amount of the system by Ozturk used to calculate the results, the row “Net Profit Percent(%)” shows the net profits percentage obtained by our system, and the *best* net profits percentage obtained by Ozturk, the row “Number of Total Trades” indicates the total number of trades required by our system and Ozturk’s *best* system, the row “Percent(%) Winning Trades” refers to the percentage of winning trades of the two systems, the row “Average Profit per Trade” refers to the average profit per trade obtained by our system and by Ozturk’s *best* system, “Exposure to Market” refers to our system’s calculated exposure to market (not available for Ozturk’s system, as the author does not report this metric and is not possible to derive from his published data), and finally, “*t* test Statistics,” “Sharpe Ratio” and “Sortino Ratio” refer to our system’s *t* test, Sharpe ratio and Sortino ratio computed values (again, not available from Ozturk’s publication). In his study, Ozturk used an initial capital of USD $10,000, zero transaction costs and a 1:100 leverage, and he experimented with a large number of algorithms and rule variations to see if a hybrid rule-based local search or GA-based search could come up with a profitable strategy for trading on the forex market. On the largest dataset Ozturk tried (1-year EUR/USD data), our results outperform his *best* obtained result by more than 4.3%, and on the same dataset, our results are more than 182.8% better than the average of his experiments (see mid-columns of Table 8). For the other two smaller datasets, even though the best single experiment the author reports is better than the result we obtain, on average, our net profits are more than 52% better for the EUR/USD 6-month dataset (left columns of Table 8) and 91% better for the GBP/USD 6-month dataset (right columns of Table 8). More importantly, our results show that we always need much fewer trades than Ozturk’s systems; for EUR/USD, our system issues between 25 and 87% the number of trades of Ozturk’s system, and for GBP/USD, our system issues only about 10% the number of trades of the other system. In the face of any nonzero spread commissions, this fact alone would magnify very significantly the differences between the two systems, in our system’s favor.

Finally, in Table 9, we present the results of testing the accuracy of the validated patterns our system produces when trained on the exact same dataset used in the study of Nassirtoussi et al. [26], and tested on the last 16 days of 2011. Nassirtoussi et al. report a rapidly deteriorating system accuracy when their testing dataset exceeds a few points (representing the last few hours of 2011). In particular, when their test set size exceeds 30, the accuracy of their system is between 52 and 60%. On the contrary, our system, as can be seen in the 2nd and 4th column and the last row of Table 9, produces an accuracy around 69% even when tested on a much larger super-set of the test set of Nassirtoussi et al. Apparently, even without utilizing any external information (headline news) at all, our system is capable of producing patterns whose accuracy on the test set compares very favorably to the predictions of the study of Nassirtoussi and his co-authors.

Table 9 Grid patterns accuracy tested on Dec. 15, 2011–Dec. 31, 2011

Grid	Down accuracy	Down # predictions	Up accuracy	Up # predictions	Total predictions
10 × 10	59.09%	44	71.41%	29	73
15 × 10	60.00%	30	83.33%	18	48
20 × 15	85.71%	7	71.40%	7	14
25 × 15	83.33%	12	71.42%	7	19
Total predictions	106/154				
Accuracy	68.83%				

Dataset used in headline news-based mining for forex [26]

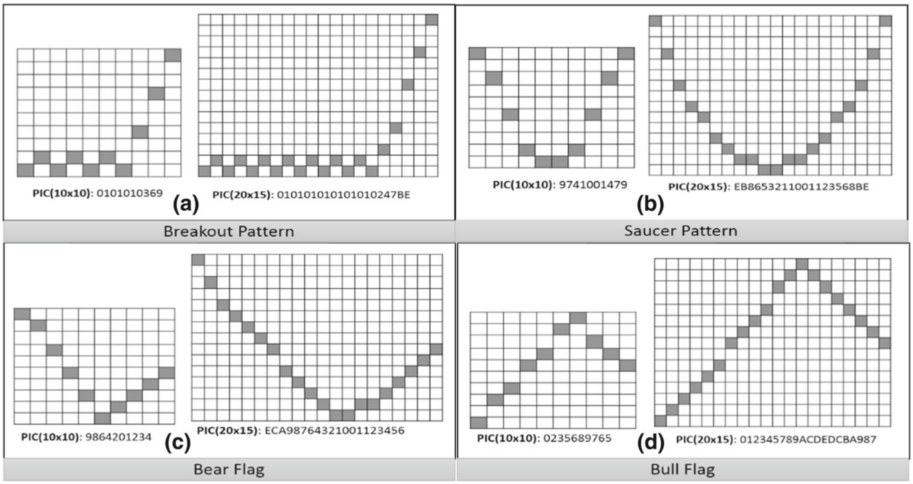


Fig. 10 Four (4) technical analysis chart patterns in the form of template grids of 10 × 10 and 20×15 dimensions

4 Analysis of existing and new chart formations

4.1 Analyzing existing technical analysis chart formations

From the patterns we have discovered, we wanted to find out which, if any, of the existing popular technical analysis chart patterns based on our array of template grids method have forecasting power. These patterns should exist in the database if they have forecasting power more than 60%. The results were very interesting and surprisingly different from what we expected from the viewpoint of technical analysis. Below, we present the results of four (4) different chart formations (breakout, saucer, bear flag and bull flag) which are very popular in the technical analysis community (shown in Fig. 10).

Our findings refer to four instruments (EUR/JPY, EUR/USD, GBP/JPY and GBP/USD) for a period of 3 years (2011–2013) intraday data.

(a) Breakout Pattern. Parracho and Neves [29] used a template breakout pattern with the combination of the uptrend pattern to produce buy signals for 100 stocks of S&P500 index using daily data. Here, we tested this formation using two different in size template grids (10 × 10, 20 × 15) across three time frames (1, 20, 60 min). The results are shown in Table 10. The column "Occurrences and Type of formation" shows how many times the patterns appeared during that period (3 years) and if pattern is Bear (price will increase within "Periods Ahead" interval above its highest level within the pattern time frame) or Bull (price will decrease), the column "Periods Ahead" shows the periods of prediction, and the column "Prediction Accuracy" is the average of pattern prediction. We can see, surprisingly, that the breakout pattern is not actually Bull (buy signal) as it would be expected from technical analysis. Actually, it acts as bear formation and it has the same behavior across all intraday time frames, different in size template grids and currency pairs! This, however, does not contradict with the findings of Parracho and Ferreira because the forex market is a completely different market.

Table 10 The prediction accuracy of the breakout pattern

Break out pattern ('0101010369'), ('010101010101010247BE')	Timeframe 1 min			Timeframe 20 min			Timeframe 60 min		
	Occurrences and type of formation	Periods ahead	Prediction accuracy (%)	Occurrences and type of formation	Periods ahead	Prediction accuracy (%)	Occurrences and type of formation	Periods ahead	Prediction accuracy (%)
EUR/JPY									
10 × 10	1356/Bear	5	69.91	872/Bear	5	68.81	349/Bear	5	64.18
20 × 15	1188/Bear	5	66.08	314/Bear	5	70.38	97/Bear	5	67.01
EUR/USD									
10 × 10	1250/Bear	5	67.68	797/Bear	5	68.38	339/Bear	5	58.11
20 × 15	1167/Bear	5	61.7	333/Bear	5	72.37	113/Bear	5	67.26
GBP/JPY									
10 × 10	1657/Bear	5	73.33	899/Bear	5	76.08	341/Bear	5	71.55
20 × 15	1334/Bear	5	69.19	284/Bear	5	76.76	120/Bear	5	67.5
GBP/USD									
10 × 10	1124/Bear	5	64.95	746/Bear	5	72.92	329/Bear	5	62.61
20 × 15	1073/Bear	5	60.39	306/Bear	5	68.63	131/Bear	5	71.76

(b) Saucer Pattern (Bottom). Wang and Chan [36] examined the saucer pattern in trading stocks using a template grid method as Bull pattern. We tested the behavior of saucer (bottom) formation in forex intraday market. The results are shown in Table 11. Note that the "Undefined" word means low forecasting accuracy (less than 55%). Reading carefully the results, we can conclude that the template grid size and time frame define the behavior of the pattern to be Bear or Bull!

(c) Bull and Bear Flag. Bull and Bear Flags were examined by Leigh et al. [23] by using template matching techniques on over 35 years of daily closing price data of stocks of New York Exchange Composite Index. Our findings are consistent with what researchers found for stocks and how they are used by technical analysis practitioners. Tables 12 and 13, summarize results for Bull and Bear Flag patterns. We can see that Bull Flag is truly Bull formation and Bear Flag is truly Bear formation for intraday forex data. But, this is not an important pattern (that can be exploited for trading) for all currency pairs. Actually, for GBP/USD the Bear Flag pattern has low forecasting accuracy (<55%). In case of GBP/JPY for 60 min, the 20×15 is both Bull and Bear which usually implies a very volatile market.

4.2 Analyzing frequent new patterns

Furthermore, from the new discovered price patterns we tried to focus on only those having high frequency of appearance and having consistent behavior across all currency pairs. The number of discovered price patterns which appeared very frequently is more than 200. Some of them appear to have complex behaviors which varies per currency pair and grid size. Our purpose was to show that there are a lot more chart formations than those known in technical analysis which may be more profitable and worth further researching to discover their complex behaviors.

5 Discussion

5.1 Further analysis of pattern prediction results

The main goal of this research was to find whether there exist unknown price patterns that have predictive power; during this research we discovered a lot of different price patterns (totally 3518) with significant forecasting power that are not described in technical analysis. Analyzing further the data, the question arises if patterns have similar predictive power uniformly across all types of instruments. To answer this question, we present an example. In the EUR/AUD of time frame 60 min we have 327 different prototype patterns (of all dimensions) that have forecasting power. But 211 of them (of 327) have no forecasting power to the EUR/CAD of the same time frame, and this happens across time frames for the same instrument. Therefore, there is a diversity in the behavior in intraday price movements, which is very important to know and exploit for trading in the forex market. Another important point, worth mentioning, is that the initial template grid method, as described in previous studies, has one disadvantage: it does not take into account the degree of volatility of prices, and this sometimes results in low prediction accuracy of the method (50–60%). As we briefly mentioned in Sect. 2.2.2, in order to overcome this problem we introduced and used the "pips range" factor and related filter. This range parameter represents the difference between the highest and lowest value of the grid template expressed in pips. By involving a threshold for

Table 11 The prediction accuracy of the saucer pattern

Saucer pattern ('9741001479'), ('EB8653211001123568BE')	Timeframe 1 min			Timeframe 20 min			Timeframe 60 min		
	Occurrences and type of formation	Periods ahead	Prediction accuracy (%)	Occurrences and type of formation	Periods ahead	Prediction accuracy (%)	Occurrences and type of formation	Periods ahead	Prediction accuracy (%)
EUR/JPY									
10 × 10	154/Bull	5	61.04	230/Bear	5	62.17	98/Bull	5	58.7
20 × 15	Undefined			Undefined			46/Bear	5	67.39
EUR/USD									
10 × 10	1250/Bull	5	69.61	211/Bear	5	60	80/Bull	20	61.25
20 × 15	Undefined			Undefined			64/Bear	5	62.5
GBP/JPY									
10 × 10	193/Bear	5	63.21	266/Bear	5	63.16	59/Bull	20	62.71
20 × 15	506/Bear	5	60.67	162/Bear	5	61.73	42/Bear	20	71.43
GBP/USD									
10 × 10	74/Bear	5	60	Undefined			60/Bull	5	61.67
20 × 15	Undefined			Undefined			53/Bear	5	64.15

Table 12 The prediction accuracy of the bull flag pattern

Bull flag ('0235689765'), ('012345789ACDED-CBA987')	Timeframe 1 min			Timeframe 20 min			Timeframe 60 min		
	Occurrences and type of formation	Periods ahead	Prediction accuracy (%)	Occurrences and type of formation	Periods ahead	Prediction accuracy (%)	Occurrences and type of formation	Periods ahead	Prediction accuracy (%)
EUR/JPY									
10×10	786/Bull	5	61.07	728/Bull	5	60.44	241/Bull	5	56.02
20×15	500/Bull	5	60	428/Bull	5	57.71	121/Bull	5	61.16
EUR/USD									
10×10	669/Bull	5	60	Undefined			Undefined		
20×15	Undefined			Undefined			106/Bull	20	55.66
GBP/JPY									
10×10	987/Bull	5	64.34	691/Bull	5	61.51	235/Bull	5	60.43
20×15	634/Bull	5	62.93	425/Bull	5	56.71	110/Both	20	60
GBP/USD									
10×10	Undefined			Undefined			Undefined		
20×15	Undefined			Undefined			78/Bear	20	55.13

Table 13 The prediction accuracy of the bear flag pattern

Bear Flag ('9864201234'), ('ECA98764321001123456')	Timeframe 1 min			Timeframe 20 min			Timeframe 60 min		
	Occurrences and type of formation	Periods ahead	Prediction accuracy (%)	Occurrences and type of formation	Periods ahead	Prediction accuracy	Occurrences and type of formation	Periods ahead	Prediction accuracy
EUR/JPY									
10 × 10	874/Bear	5	60.53	646/Bear	5	61.61	173/Bear	20	64.16
20 × 15	Undefined			286/Bear	1	55.85	141/Bear	20	60.28
EUR/USD									
10 × 10	719/Bear	5	61.89	589/Bear	5	59.42	130/Bear	20	61.54
20 × 15	740/Bear	5	55.54	265/Bear	5	55.85	Undefined		
GBP/JPY									
10 × 10	1132/Bear	5	63.43	660/Bear	5	61.21	176/Bear	20	61.93
20 × 15	1062/Bear	5	60.36	278/Bear	5	63.67	143/Bear	20	62.24
GBP/USD									
10 × 10	Undefined			Undefined			Undefined		
20 × 15	Undefined			Undefined			166/Bear	20	55.42

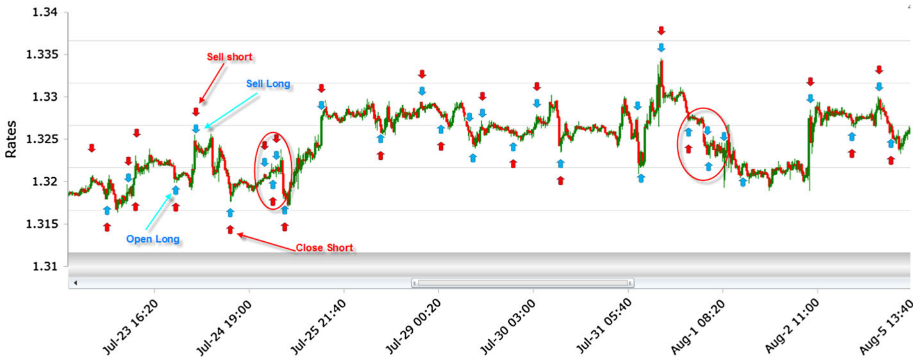


Fig. 11 Trading signals by patterns strategy for EUR/USD, 20 min time frame, from Jul-22 to Aug-5, 2013

pips range (for 1 min time frame: pips range from 20 to 130 pips, for 60 min time-frame pips range, is between 30 and 300) in the pips-range filter, our algorithms are able to reject patterns with low pips range and thus improve accuracy significantly.

5.2 Graphical analysis of the patterns strategy performance

In Fig. 11, we present the trading signals created in a particular time window by our proposed Patterns Strategy for EUR/USD for the 20 min time frame (each candlestick represents period of 20 min) for 2013. The red arrows correspond to short positions only, while the blue arrows correspond to long positions only. In the beginning of the time window (left side of figure), the system opens a long trade (see label "Open Long") and in the next top the long position is closed (see label "Sell Long"), while at the same time it opens short position (see label "Sell Short"). Then, the short position is closed in the second bottom. We investigated visually a lot of parts of signals created by the system. In cases we have profitable trades almost all signals are created in tops or bottoms or a little earlier (1–2 candlesticks), as this chart (which is an actual sample) shows. It is easy to see that, as desired, the system gives signals *before* the new trend/reversal occurs! Signals created by technical indicators, on the other hand, are always created *after* the trend has started.

5.3 Limitations: spread effects

Comparing pattern strategy performance results across time frames, it seems that using 1 min time frame brings in most cases better results. The t statistic is very high (well above 1.6) in specific currency pairs and time frames, providing strong evidence that our trading system produces very profitable trades that cannot be attributed to chance. On the other hand, based on the t test statistics, to be confident at the 95% level, the Walk Forward Test passes 5 out of the 8 currency pairs (AUD/JPY, EUR/JPY, EUR/USD, GBP/JPY and GBP/USD) for 1 min time frame, while for 20 min it passes 4 of 8 currency pairs (EUR/JPY, EUR/USD, GBP/JPY and GBP/USD). For 60 min time frame no currency pair passes the t test!

Generally, if we take into account only the net profits, the patterns strategy brings robust high profits in most cases we tested. More specifically, the system brought low or negative profits only for GBP/AUD and GBP/CHF. But, the spread used for trading in GBP/AUD was 4 pips which implies cost on average $70 fixed cost per trade, while, for example, the cost of EUR/USD is only $10. Similarly, the high cost of trading for the GBP/CHF is at least

Table 14 Performance of patterns strategy under different spread values and leverage for 8 instruments

		AUD/JPY	EUR/CAD	EUR/JPY	EUR/USD	GBP/AUD	GBPCHF	GBP/JPY	GBP/USD
Leverage 1:10									
Low spread (in pips)		1.4	2.3	0.7	0.7	4	2.5	1.7	0.8
High spread (in pips)		2.1	2.6	1.2	0.9	4.4	2.6	2	1
Net profit %—low spread (leverage 1:10)	1 min	87.95%	41.01%	155.78%	98.03%	9.33%	13.68%	183.35%	61.02%
Net profit %—high spread (leverage 1:10)		68.15%	37.17%	122.55%	89.31%	7.27%	12.89%	150.10%	57.25%
Net profit %—low spread (leverage 1:10)	20 min	59.42%	32.48%	93.42%	72.53%	31.79%	37.02%	107.74%	79.58%
Net profit %—high spread (leverage 1:10)		44.74%	30.80%	81.28%	68.41%	28.45%	35.86%	94.70%	75.74%
Net profit %—low spread (leverage 1:10)	60 min	36.89%	46.75%	70.94%	33.72%	20.97%	18.19%	38.74%	47.59%
Net profit %—high spread (leverage 1:10)		31.77%	44.91%	64.57%	32.10%	19.61%	17.56%	36.88%	46.50%
Leverage 1:1									
Net profit %—low spread (no leverage)	1 min	17.59%	8.20%	31.16%	19.61%	1.87%	2.74%	36.67%	12.20%
Net profit %—high spread (no leverage)		13.63%	7.43%	24.51%	17.86%	1.45%	2.58%	30.02%	11.45%
Net profit %—low spread (no leverage)	20 min	11.88%	6.50%	18.68%	14.51%	6.36%	7.40%	21.55%	15.92%
Net profit %—high spread (no leverage)		8.95%	6.16%	16.26%	13.68%	5.69%	7.17%	18.94%	15.15%
Net profit %—low spread (no leverage)	60 min	7.38%	9.35%	14.19%	6.74%	4.19%	3.64%	7.75%	9.52%
Net profit %—high spread (no leverage)		6.35%	8.98%	12.91%	6.42%	3.92%	3.51%	7.38%	9.30%

partially the reason for the low profits our system brings in this instrument too. In Table 14, therefore, we show how net profits vary depending on the spread used for trading across various instruments, and also depending on the leverage used (1:10 or no leverage at all, i.e., 1:1). As can be seen from this table, *even without using any leverage, and under very high spreads, the system always maintains positive (and mostly very good) net profits*, some times exceeding 30% (in the GBP/JPY pair).

Two other significant factors that affects the trading performance are the *latency* and the *slippage*. We did not take into account these two factors in our calculations. Latency from forex perspective is how much it takes to execute an order (from time the computer sends the order to time receives the confirmation). So, selecting a broker which uses low latency technologies is clearly an important factor. The slippage is another factor that could decrease the profitability. It is the difference between expected price of a trade and the price at which the order is filled. In our strategy, we buy at close price by using market orders. Here, we expect the slippage to be very small because the Forex market is very "liquid." But when there is high volatility and/or big orders, the slippage becomes important. In our strategy we traded 1 lot (USD $100,000) and brought very high net profits percent even with very low leverage (1:10). If we increase significantly the size of the order, say up to 1000 lots, the order will be likely filled partially at different prices because there is imbalance of buyers and sellers, and this could possibly lead to further losses.

6 Conclusions and future work

Before the 1980s but even to this day, many academic researchers have been skeptical about the ability to predict prices, especially via technical analysis, and the general consensus was that it is not possible to produce as good results as the buy–hold strategy ([1,9,15]) . Later studies showed the opposite [2,4,23] . *Today*, there is a growing number of academics and practitioners alike willing to acknowledge at least that the efficient market hypothesis or the closely related random-walk hypothesis does not always hold strictly [14].

In this paper, we have presented a framework and complete expert system for discovering frequent profitable intraday patterns in the Forex market. Our *prototype pattern discovery algorithm* is efficient and is capable of processing tens of millions of time-series patterns in a few hours. Via analysis of the discovered patterns, we have shown that there are new, previously unknown chart formations which have similar or better forecasting power than the well-known patterns from technical analysis. The discovered patterns are not described in technical analysis and yet have very high support levels *depending on currency pair and time frame*. In this study we tried to quantify the pattern behavior by employing many predicate variables to map the future direction of the prices. Then, we showed how to exploit these patterns in a straightforward Patterns Strategy that can be characterized as "leading strategy" because in most cases creates signals *before* the reversal or change of the trend. The simulated trading performance results, although not yet validated in a real-time live demonstration system receiving feeds in real time from a broker organization, were nevertheless *very encouraging even for very low or no leverage levels and high spreads commissions*. On the other hand, it seems that our framework would be very well suited to institutional investors (banks, hedge funds and others managing large portfolios): aside from slippage issues mentioned above, using higher leverage levels, such as 1:100 which are quite common in the real world, and being able to demand lower spread commissions, the net profits that can be realized by our system seem to reach very high levels; such performance would rank our proposed expert system among the top traders in the forex market (there are a few signal providers in the web-site www.zulutrader.com that achieve similar or better profits). As should be expected, however, when trading with such leverage, the Sharpe and Sortino ratios decrease, indicating the higher risk that the system would expose the user to. A future research question to be answered then is how to compute the optimal leverage and order sizes to trade with so as to optimize the trading performance of the system subject to the risk as measured by the Sortino ratio being kept above certain minimum thresholds.

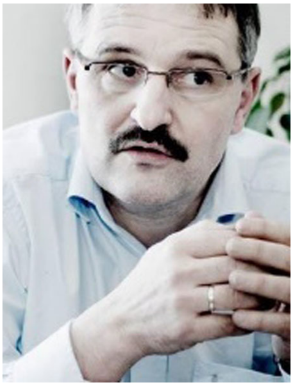
References

1. Alexander SS (1961) Price movements in speculative markets: trends or random walks. Ind Manag Rev 2:7–26
2. Bessembinder H, Chan K (1995) The profitability of technical trading rules in the Asian stock markets. Pac Basin Finance J 3(2–3):257–284
3. Bo L, Linyan S, Mweene R (2005) Empirical study of trading rule discovery in China stock market. Expert Syst Appl 28:531–535
4. Brock W, Lakonishok J, Lebaron B (1992) Simple technical trading rules and the stochastic properties of stock returns. J Finance 47:1731–1764
5. Bulkowski TN (2008) Encyclopedia of candlestick charts, 2nd edn. Wiley, Hoboken
6. Caginalp G, Laurent H (1998) The predictive power of price patterns. Appl Math Finance 5:181–205
7. Cheng C-H, Chen T-L, Wei L-Y (2010) A hybrid model based on rough sets theory and genetic algorithms for stock price forecasting. Inf Sci 180:1610–1629
8. Duda R, Hart P (1973) Pattern classification and scene analysis. Wiley, New York
9. Fama EF (1970) Efficient capital markets: a review of theory and empirical work. J Finance 25:383–417
10. Goumatianos N, Christou IT, Lindgren P (2013a) Useful pattern mining on time series: applications in the stock market. In: Proceedings of 2nd international conference on pattern recognition applications and methods (ICPRAM 2013), Barcelona, Spain, pp 608–612
11. Goumatianos N, Christou IT, Lindgren P (2013) Stock selection system: building long/short portfolios using intraday patterns. Procedia Econ Finance 2:296–307 (Proc. Intl. Conf. Appl. Econ. 2013)
12. Haeri A, Hatefi S-M, Rezaie K (2015) Forecasting about EUR/JPY exchange rate using hidden Markova model and CART classification algorithm. J Adv Comput Sci Technol 4(1):84–89
13. Hung K-K, Cheung Y-M, Xu L (2003) An extended ASLD trading system to enhance portfolio management. IEEE Trans Neural Netw 14(2):413–425
14. Ilmanen A (2011) Expected returns: an investor's guide to harvesting market rewards. Wiley, Chichester
15. Jensen MC, Bennington GA (1970) Random walks and technical theories: some additional evidence. J Finance 25(2):469–482
16. Kao L, He T (2009) Developing actionable trading agents. Knowl Inf Syst 18(2):183–198
17. Keogh E, Pazzani M (2000) A simple dimensionality reduction technique for fast similarity search in large time series databases. In: Proceedings of fourth Pacific-Asia conference on knowledge discovery and data mining, pp 122–133
18. Kong X, Qiang W, Guoqing C (2010) An approach to discovering multi-temporal patterns and its application to financial databases. Inf Sci 180:873–885
19. Lee KH, Jo GS (1999) Expert system for predicting stock market timing using a candlestick chart. Expert Syst Appl 16:357–364
20. Leigh W, Modani N, Purvis R, Roberts T (2002a) Stock market trading rule discovery using technical charting heuristics. Expert Syst Appl 23(2):155–159
21. Leigh W, Purvis R, Ragusa JM (2002b) Forecasting the NYSE composite index with technical analysis, pattern recognizer, neural network, and genetic algorithm: a case study in romantic decision support. Decis Support Syst 32:161–174
22. Leigh W, Modani N, Hightower R (2004) A computational implementation of stock charting: abrupt volume increase as signal for movement in New York stock exchange composite index. Decis Support Syst 37:515–530
23. Leigh W, Flobish C, Hornik S, Purvis R, Roberts T (2008) Trading with a stock chart heuristic. IEEE Trans Syst Man Cybern Part A 38(1):93–104
24. Maginn J, Tuttle D, McLeavey D, Pinto J (eds) (2007) Managing investment portfolios: a dynamic process, 3rd edn. Wiley, Hoboken , NJ, USA

25. Marshall BR, Young MR, Rose LC (2006) Candlestick technical trading strategies: can they create value for investors? J Bank Finance 30:2303–2323
26. Nassirtoussi AK, Aghabozorgi S, Wah T-Y, Ngo DCL (2015) Text-mining of news-headlines for FOREX market prediction: a multi-layer dimension reduction algorithm with semantics and sentiment. Expert Syst Appl 42:306–324
27. Ney H, Steinbiss V, Haeb-Umbach R, Tran B-H, Essen U (1994) An overview of the Phillips research system for large vocabulary continuous speech recognition. Int J Pattern Recognit Artif Intell 8(1):33. doi:10.1142/S0218001494000036
28. Ozturk M (2015) Heuristic-based trading system on Forex data using technical indicator rules. M.Sc. thesis, Comp. Eng. Dept. Middle East Technical University
29. Parracho P, Neves RF (2011) Trading with optimized uptrend and downtrend pattern templates using a genetic algorithm kernel. In: Conference: proceedings of IEEE congress on evolutionary computation, New Orleans, LA, USA, 5–8 June 2011
30. Petrov V-Yu, Tribelsky MI (2015) FOREX trades: can the Takens algorithm help to obtain steady profit at investment reallocations? Pis'ma v ZhETF 102(12):958–961
31. Poh KL (2000) An intelligent decision support system for investment analysis. Knowl Inf Syst 2(3):340–358
32. Raudys S (2013) Portfolio of automated trading systems: complexity and learning set size issues. IEEE Trans Neural Netw Learn Syst 24(3):448–459
33. Theofilatos K, Likothanasis S, Karathanasopoulos A (2012) Modeling and trading the EUR/USD exchange rate using machine learning techniques. Eng Technol Appl Sci Res 2(5):269–272
34. Toshniwal D, Joshi RC (2005) Similarity search in time series data using time weighted slopes. Informatica 29(1):79–88
35. Wang J-L, Chan S-H (2007) Stock market trading rule discovery using pattern recognition and technical analysis. Expert Syst Appl 33:304–315
36. Wang J-L, Chan S-H (2009) Trading rule discovery in the US stock market: an empirical study. Expert Syst Appl 36:5450–5455
37. Walid B, Van Oppens H (2006) The performance analysis of chart patterns: Monte-Carlo simulation and evidence from the euro/dollar foreign exchange market. Empir Econ 30:947–971
38. Wu J-L, Yu L-C, Chang P-C (2014) An intelligent stock trading system using comprehensive features. Appl Soft Comput 23:39–50
39. Xu L, Cheung Y-M (1997) Adaptive supervised learning decision networks for trading and portfolio management. J Comput Finance 13(2):806–816
40. Zhang D, Zhou L (2004) Discovering golden nuggets: data mining in financial application. IEEE Trans Syst Man Cybern Part C 34(4):513–522
41. Zhang Z, Jiang J, Liu X, Lau R, Wang H, Zhang R (2010) A real-time hybrid pattern matching scheme for stock time series. In: Proceedings of 21st Australasian conference on database technologies, vol 104, pp 161–170

Nikitas Goumatianos holds a B.Sc. from the Hellenic Air Force Academy (1989), Athens, Greece, an MBA (2008) from the Monterey Naval Postgraduate School, Monterey, CA, USA, and a Ph.D. (2016) in Computer Science from the Center for Tele-Infrastructure (CTiF), Aalborg University, Aalborg, Denmark. He is a retired Brigadier General of the Hellenic Air Force. His research interests include Artificial Intelligence and Machine Learning and its applications in Algorithmic Trading, and Financial Modeling.

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, Athens, Greece (2006). He has been with Delta Technology Inc. as Senior Developer, with Intracom S.A. Development Programmes Dept. as Area Leader in Data & Knowledge Engineering, and with Lucent Technologies Bell Labs as Member of Technical Staff. He has also been an adjunct assistant professor with the University of Patras, Greece, and an adjunct professor at Carnegie-Mellon University, Pittsburgh, PA, USA. He has also served as CTO and co-founder of IntelPrize, a Greek Big-Data start-up company. He is currently Professor at Athens Information Technology, Athens, Greece, and has published more than 70 articles in top-tier scientific journals and peer-reviewed conferences. His research interests include Machine Learning, Data Mining and Analytics, and Parallel/Distributed Computing & Optimization.



Peter Lindgren holds a B.A. (1983) in Business Administration from Copenhagen Business School, Copenhagen, Denmark, an M.A. (1985) in Business Administration from Aarhus School of Business, Aarhus, Denmark, and a Ph.D. (2003) from Aalborg University, Copenhagen, Denmark. He has been Associate Professor at the Dept. of Mechanical Engineering, Aalborg University, Aalborg, Denmark from 1999 to 2014. He is a Professor with the Dept. of Business Development and Technology at Aarhus University, Aarhus, Denmark. His research interests include business models and high-speed innovation.

Ramjee Prasad holds a B.Sc. in Electronics Engineering (1968) from Bihar Institute of Technology, Sindri, India, and an M.Sc. in Electronics Theory and Microwave Engineering (1970) and Ph.D. in Plasma Physics (1979), both from Birla Institute of Technology, Mesra, Ranchi, India. He is the recipient of many international academic, industrial and governmental awards and distinctions of which the most recently is the cross of the order of chivalry from the Danish Queen due internationalization of top-class telecommunication research and education. He holds the Professorial Chair of Wireless Information and Multimedia Communications at Aalborg University, Denmark (AAU) since June 1999. Since 2004 he is the Founding Director of the Center for TeleInfrastruktur (CTIF), established as a large multi-area research center at the premises of Aalborg University. He has published many hundreds of journal and conference articles in top-tier journals and conferences. Prasad is a Fellow of IEEE, the IET and IETE.