Stock price forecasting over adaptive timescale using supervised learning and receptive fields

Mario G.C.A. Cimino[1], Federico Dalla Bona[2], Pierfrancesco Foglia[1],
Manilo Monaco[1], Cosimo A. Prete[1], and Gigliola Vaglini[1]

[1] Department of Information Engineering, University of Pisa, 56122 Pisa, Italy
[2] Trading Methods SL - 35100 San Bartolomé de Tirajana - Las Palmas, Gran Canaria, Spain
{m.cimino, p.foglia, m.monaco, a.prete, g.vaglini}@ing.unipi.it
federicodallabona@tradinglift.com

Abstract. Pattern recognition in financial time series is not a trivial task, due to level of noise, volatile context, lack of formal definitions and high number of pattern variants. A current research trend involves machine learning techniques and online computing. However, medium-term trading is still based on human-centric heuristics, and the integration with machine learning support remains relatively unexplored. The purpose of this study is to investigate potential and perspectives of a novel architectural topology providing modularity, scalability and personalization capabilities. The proposed architecture is based on the concept of Receptive Fields (RF), i.e., sub-modules focusing on specific patterns, that can be connected to further levels of processing to analyze the price dynamics on different granularities and different abstraction levels. Both Multilayer Perceptrons (MLP) and Support Vector Machines (SVM) have been experimented as a RF. Early experiments have been carried out over the FTSE-MIB index.

Keywords: Stock price forecasting, Pattern recognition, Artificial neural network, Support vector machine.

1 Introduction and motivation

Stock price forecasting is a challenge in financial systems. Although there is an inherent level of chaos in markets dynamics, what really matters for decision-making is upward and downwards movements. In this field, even small improvements in performance can be very profitable [1].

Typically, *long-term trading* investors use *fundamental analysis*, which involves the study of many indicators over months, concerning overall economy, industry condition, financial condition and management of companies, earnings, expenses, assets and liabilities and so on. In this field, many studies have proposed various methods for forecasting stock prices. Most of these approaches require to select carefully the input variables, to establish a predictive mathematical model under professional knowledge, to adopt various statistical methods for prior analysis of indicators.

With the rapid growth of internet-based transactions, the frequency for performing operations on the stock market has increased to fractions of seconds. In the last decade, several automatic traders emerged in *short-term trading*, operating in markets without the human intervention [2,3]. Nowadays, automatic trading is efficient for intraday operations, but starts to become inefficient on *medium-term trading*, i.e., ranging from some days to a couple of weeks, over which events may have a nonlinear impact on dynamics.

The focus of this paper is on *medium-term trading*, where financial gains are not trivial to obtain. Typically, medium-term traders use *technical analysis*, adopting stock price and, eventually, volume charts to identify patterns and trends. Current research on medium-term stock price forecasting involves artificial intelligence techniques and real time computing. A fundamental process in technical analysis is *pattern recognition*. The purpose is to identify repeating patterns in prices, and figure out the ongoing pattern in order to predict the pattern completion in the close future [4]. In contrast to a case of function approximation, the investor aims to get a maximal profit rather than a minimal standard deviation. Thus, results depend on the predicted sign of course changes, and are based on the speculation for the rise and fall of prices. Therefore, the patterns of interest represent changes of price instead of course value.

In technical analysis, many patterns have been proposed. A graphical pattern emerges from the operational decisions of buyers and sellers and summarizes the global behavior of a market in a specific time interval. Common chart patterns are *head-and-shoulders*, *double-top*, *triple-top*, *spike-top*, *1-2-3-low*, *1-2-3-high*, and so on [4]. Different people use different methods to consider the favorite patterns. There are many ways to be successful and no one strategy is the best. However, patterns are not significant without *context indicators*. Examples of context indicators are simple moving average, support, resistance, trend lines, the degree of prices volatility and momentum-based indicators. In brief, support and resistance are certain predetermined levels of the price at which the price tends to stop and reverse [5]. Under proper patterns, practitioners look to buy at support and sell at resistance. Trend lines give defined entry and exit points, which can be utilized when moving to a new context, where there is no price history. High volatility in financial time series results in a greater difficulty in forecasting [6]. Momentum-based indicators, such as Bollinger bands, tend to be used in range-bound or trendless markets [1,7]. Thus, it is the current context which assigns a certain reliability to a given pattern. Another fundamental issue is the time frame (or horizon), to which the analysis should be carried out. The horizon is not a constant notion: it is iteratively adjusted to focus the expected patterns at different granularities, potentially ranging from days to weeks. Indeed, many patterns are inherently recursive, manifest a fractal character and can therefore be found on different time scales, and with different duration in time [8].

Given the above requirements, a novel architecture based on machine learning has been designed and realized for supporting strategies of identification of investment opportunities. Overall, the strategies focus on short-term and medium-term patterns, but taking into consideration also short-term and medium-term information furnished by indicators as a reference context. The purpose of this study is to investigate potential and perspectives of the proposed approach in terms of accuracy, but also modularity, scalability and personalization.

More technically, the architecture is organized in terms of Receptive Fields (RFs), i.e., sub-modules focusing on specific groups of patterns, that can be connected to further levels of processing [9]. This topology allows to analyze the price dynamics on different granularities and different abstraction levels [10]. Both Multilayer Perceptrons

(MLP) and Support Vector Machines (SVM) have been experimented as a receptive field for detecting sub-group of patterns. Early experiments have been carried out over the FTSE-MIB index, an Italian stock market index.

The remainder of the paper is structured as follows. Section 2 discusses the architectural design. Experiments are covered in Section 3. Finally, Section 4 summarizes conclusion and future work.

2 Architectural design

This section focuses on the design of an architecture supporting strategies of identification of investment opportunities. Such strategies consider the stock price oscillation in a given time window. As an example, and without losing generality, Fig. 1 shows a basic breakout strategy, using the *1-2-3 low* pattern. The strategy assumes that, under an overall rising trend, and after occurrence of waves 1, 2, and 3, the opportunity is identified when the stock price *breaks* the resistance level. To detect this opportunity, the *1-2-3 low* pattern must be recognized, as well as its context (i.e., rising trend, support and resistance levels). To recognize patterns in financial time series is not a trivial task, due to additional fluctuations (noise), the volatile context, the lack of formal definitions. Technical patterns are not exact, and can be very different over occurrences, in terms of a different amplitude and different duration, and visually the same pattern occurrences can look different despite being equivalent. Moreover, patterns do not cover every time point in the series. In addition, the classical technical patterns are often personalized by technical analysts, producing a high number of pattern variants. As a consequence, to develop and manage algorithms for automated heuristics for financial pattern detection is often neither convenient nor effective.

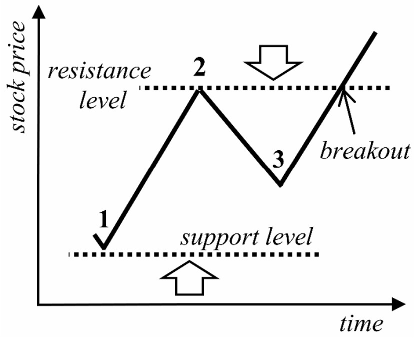


Fig. 1. Breakout trading strategy.

For the above reasons, during the past decades machine learning models have been developed to learn the key characteristics and to recognize technical patterns [4]. The major challenges in the field are: to recognize the input pattern reliably, to provide scalable, modular and adaptable architectures, to allow a dynamic inclusion/exclusion of personalized patterns. In particular, current research partially considers the manageability and the integrability with user's cognitive processes. For this purpose, the architectural design should explicitly address the application of the support system to human strategies. Traditionally, attention is primarily paid to interface design [13] and usability [14] or to performance evaluation. With regard to the latter aspect, the use of machine learning raises the following issues: (i) for an increasing number of patterns, there is an explosion of units needed, leading to increasing difficulties in learning; (ii) when patterns not previously considered are taken into account, the overall global learning process must be repeated. To foster scalability issues, a recent research trend considers that deep learning topology could be applied to time series forecasting, by extracting robust features that capture relevant information for each architectural layer [7]. However, this field still remains relatively unexplored, and then can produce systems that are costly, i.e., complex and engineered, and not very profitable, i.e., usable only under specific circumstances.

In this paper we propose a flexible architectural topology, where each RF can be considered as a modular "plugin" specialized for families of patterns. This approach enables a dynamic customization of patterns. Thus, to add, remove or update a family of patterns is relatively simple, since it involves mainly the training over the data of the new family. Furthermore, this modular organization allows the reusability of RFs. Fig. 2 shows the overall topology, based on a collection of RFs.

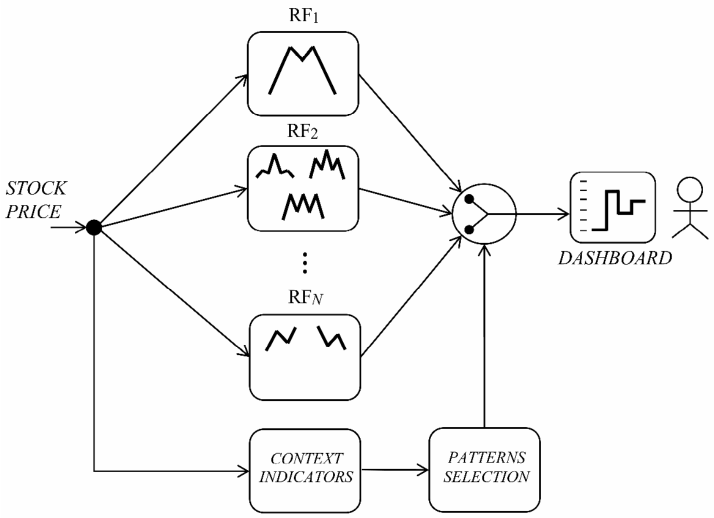


Fig. 2. Architectural topology of a system supporting strategies of identification of investment opportunities.

Each RF is specialized on a family of patterns, it takes as an input the stock price time series, and provides as an output the related pattern label. Some examples of family of patterns are shown in figure: *double-top* (RF1), *spike-top*, *head-and-shoulders*, *triple-top* (RF2), ... , *1-2-3 low*, *1-2-3 high* (RF*N*), and so on. On the bottom, context indicators are calculated, to determine the significance and the reliability of the recognized patterns. The output is controlled through a pattern selector. The outcome is a temporal sequence of patterns. The pattern abstraction level sensibly reduces the complexity and the cognitive effort for the technical analyst, and can be easily represented in a dashboard.

Fig. 3 represents the internal structure of a single RF. As specified, the input is made of samples of the stock price over time, $p(t)$, with sampling rate σ. To allow a scanning over a specified time scale, an adaptable windowing is carried out as a first step, selecting the values that are enclosed in the current sliding window with size τ. Second, the ripple is removed, by using a triangular moving average with size ρ, as a smoothing function [11]. It is a weighted moving average commonly used in financial time series, providing a good balance between signal and micro noise. Third, a resampling is carried out to deliver a prefixed number of data samples T, independently of the time scale. Fourth, min-max normalization is made, obtaining data values between 0 and 1 aside from the price scale. Fifth, the pattern recognition system can produce none, one or more recognized patterns. In case of unknown pattern, the process restarts, adapting the time window size τ, through a zoom in/out function. The process ends when setting the right time scale for that pattern, or when the minimum/maximum scale is reached.

To speed up the execution time of the overall process, in systems where time constraints are a concern, the pipelined structure of the RFs and/or the loops can be parallelized on advanced multicore systems [15].

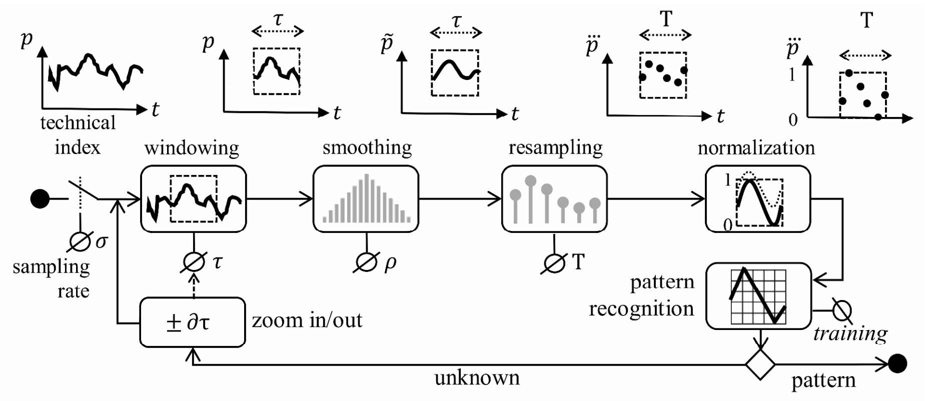


Fig. 3. Structure of a RF.

For the pattern recognition task, two different solutions have been studied and experimented: a Multi-Layer Perceptron (MLP), and a Support Vector Machine (SVM).

2.1 Multilayer perceptron as a receptive field

The MLP is fed by T input samples and consists of one hidden layer with size (i.e., number of neurons) h, and an output layer with P output, where P is the number of patterns that can be recognized in the family. The activation is based on a softmax function, which is commonly used in multiclass classification methods. Given the feed-forward and fully connected layers of a MLP, the overall number of parameters of a RF measures its complexity, according to the following formula:

$$\text{RFcomplexity(MLP)} = T \cdot (h + 1) + h \cdot (P + 1) \tag{1}$$

Given a set of labelled patterns, for a given family of patterns, the 80% of randomly extracted patterns is used as a training set, and the remaining 20% is used as a test set. The training is based on the minimization of the mean squared error with respect to the numerical output expected for each label, made through the backpropagation algorithm. The best value of the hyperparameter h is determined by calculating the 99% confidence interval of the classification rate for each h in a specified interval.

2.2 Support Vector Machine as a receptive field

The SVM architecture adopted is a soft-margin SVM known as C-SVM classifier. Since an SVM is designed for binary classification, the multiclass RF able to recognize a family of P patterns is created decomposing the problem into a combination of the $P(P$-1)/2 outputs of binary classifiers, i.e., according to a *one-vs.-one* strategy. Given the number s_i of supporting vectors of each i-th binary classifier, $S_i = \{v_j : v_j \in \Re^T, j = 1, \ldots, s_i\}$ the overall number of parameters of an RF measures its complexity, according to the following formula:

$$\text{RFcomplexity(SVM)} = T \cdot \sum_1^{P(P-1)/2} s_i \qquad (2)$$

In an SVM architecture, the hyperparameter C is called regularization term, and controls the trade-off between maximizing the margin and minimizing the training error. The kernel function used is the Radial Basis Function (RBF), whose standard deviation is controlled via the hyperparameter γ, which can be considered the inverse of the radius of influence of support vectors. The hyperparameter γ controls the tradeoff between error due to bias and variance in your model: a small gamma can produce a model with low bias and high variance while a large gamma can produce a model with higher bias and low variance. The best values of hyperparameters C and γ have been set using a grid-search with exponentially growing sequences.

3 Experimental studies

The data set used for experiments comes from the stock trading prices of the FTSE-MIB index, sampled from Oct 19, 2017 to May 2, 2018, i.e., in 134 total days. Each day contains 511 observations: an observation is sampled every minute (σ = 1 sample per minute) during the opening time, i.e. from 9:00 to 17:30, i.e., 8.5 hours. Overall, the data set contains 68,474 total samples. The window size τ is set within the interval [30, 120] minutes (or samples). The smoothing size is ρ = 15 minutes. The resampling size is T = 50 samples. Finally, the zoom in/out variation is $\partial\tau$ = 1 minute.

The number of instances generated from the dataset is 142 for each of the patterns *1-2-3 low* and *1-2-3 high*. A perturbation function has been applied to increase the instances up to 426 per pattern. The perturbation function adds or subtracts a small random value to each sample. After perturbation, the quality of the generated samples has been verified using a segmentation function based on the Perceptually Important Points (PIP) algorithm [12] to verify the statistics for each pattern.

Fig. 4a shows an occurrence of *1-2-3 low* pattern, before and after smoothing. Fig. 4b shows the signal after resampling and normalization. It is apparent that differences in micro noise, scale prices and time extension are almost completely removed. Fig. 5 shows the 426 occurrences of *1-2-3 low* pattern. It is clear that the occurrences of the same pattern can look very different.

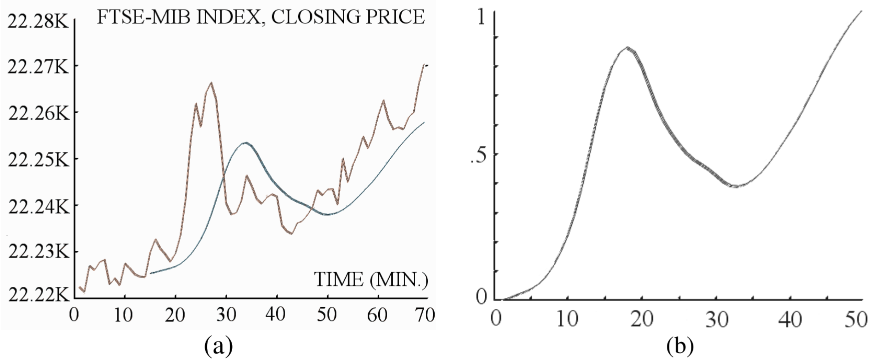


Fig. 4. Real sample of pattern *1-2-3 low*: (a) before and after smoothing; (b) after resampling and normalization

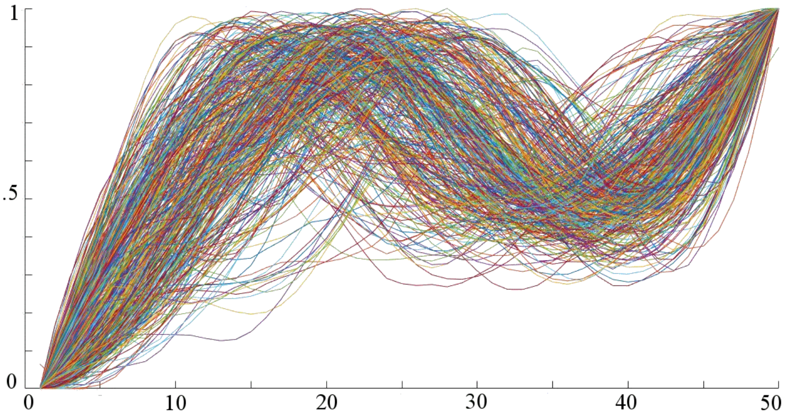
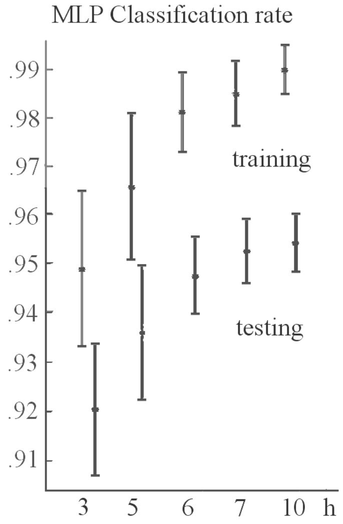


Fig. 5. Occurrences of *1-2-3 low* pattern available in the data set.

To compare the performances of the MLP-based and SVM-based RF, Fig. 6a shows their classification rate for different values of the respective hyperparameters. For the MLP, the best performance on the test set is achieved with 10 neurons in the hidden layer: 95.3%. For the SVM, the best performance is achieved with $C = \{9{,}10\}$ and $\gamma = 0.6$: 100% (training) and 98.61% (testing). It is apparent that the SVM outperforms the MLP. However, according to Formula (1) and (2), and considering the “unknown” pattern as a third class, the number of parameters is $50 \cdot 11 + 10 \cdot 4 = 590$ for the MLP, and $50 \cdot (70 + 50 + 73) = 9{,}650$ for the SVM. Thus, in terms of complexity, SVM is more than one order of magnitude larger than MLP.



(a)

C	γ	**Train Set**	**Test Set**
10	.1	99.61	97.22
1	.1	97.03	94.44
100	.1	100	95.83
9	.1	99.45	97.22
8	.1	99.37	97.22
11	.1	99.61	97.22
10	.01	94.84	93.06
10	1	100	97.22
10	.5	100	95.83
10	**.6**	**100**	**98.61**
9	**.6**	**100**	**98.61**
8	.6	100	97.22

(b)

Fig. 6. Performances of a RF with 3 patterns and for different values of hyperparameters: (a) Classification rate of MLP against h; (b) Classification rate of SVM against C and γ.

Let us consider the effects of an increase of the number of patterns, by introducing in the same RF the following two patterns: *upward head-and-shoulders* and *downwards head-and-shoulders*. Again, a perturbation function has been applied to increase the instances up to 426 per pattern. After training, the RF is able to recognize 5 total patterns. Fig.7 shows the performances provided by the MLP-based and SVM-based RF with the related hyperparameters. Results on the test set show that SVM outperforms the MLP. However, according to Formula (1) and (2), the number of parameters is now $50 \cdot 11 + 10 \cdot 6 = 610$ for the MLP, and $50 \cdot 291 = 14{,}550$ for the SVM. Thus, the complexity of SVM is dramatically increased. However, the complexity of SVM is not an issue, since families of patterns are typically made by less than 5 patterns.

RF: hyperparameters	**Train Set**	**Test Set**
MLP: $h = 6$	$.973 \pm .015$	$.938 \pm .02$
MLP: h =10	$.991 \pm .003$	$.948 \pm .02$
SVM: C=12, γ=0.1	$.998$	$.983$

Fig. 7. Classification rate of a RF with 5 patterns, made with MLP and SVM.

4 Conclusions

In this paper we have designed and experimented a flexible architectural topology based on Receptive Fields, for supporting strategies of identification of financial investment opportunities based on machine learning. The approach enables a dynamic customization of patterns and foster reusability of RF. The overall system provides a pattern-based abstraction level, and sensibly reduces the complexity and the cognitive effort for the technical analyst involved in medium-term technical analysis. By improving the comfort via timely and tailored information, the system can be well integrated with the investor's decision process, thus reducing errors resulting from bias and emotion.

The paper illustrates the design of an RF considering noise, variability of amplitude, duration, shape, of equivalent pattern occurrences. MLP and SVM have been used as a RF. Experiments have shown that SVM outperforms MLP, although the SVM complexity is sensibly higher. However, such complexity is not an issue since each RF is typically devoted to few patterns.

As a future work, different machine learning techniques, which incorporate the concept of time into their operating model, will be adopted for RFs.

Acknowledgements

This work was carried out in the framework of the SCIADRO project, co-funded by the Tuscany Region (Italy) under the Regional Implementation Programme for Underutilized Areas Fund (PAR FAS 2007-2013) and the Research Facilitation Fund (FAR) of the Ministry of Education, University and Research (MIUR).

The authors thank Marco Gasperini for his work on the subject during his thesis.

References

1. Zhou, X., Pan, Z., Hu, G., Tang, S., Zhao, C.: Stock Market Prediction on High-Frequency Data Using Generative Adversarial Nets. Mathematical Problems in Engineering, 2018.
2. Jabbur, E., Silva, E., Castilho, D., Pereira, A., & Brandão, H. (2014, August). Design and evaluation of automatic agents for stock market intraday trading. In Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 03 (pp. 396-403). IEEE Computer Society.
3. Wang, X., Bao, S., & Chen, J.: High-frequency stock linkage and multi-dimensional stationary processes. Physica A: Statistical Mechanics and its Applications, 468, 70-83, 2017.
4. Gong, X., Si, Y. W., Fong, S., & Biuk-Aghai, R. P. (2016). Financial time series pattern matching with extended UCR Suite and Support Vector Machine. Expert Systems with Applications, 55, 284-296.
5. Zapranis, A., & Tsinaslanidis, P. E. (2012). A novel, rule-based technical pattern identification mechanism: identifying and evaluating saucers and resistant levels in the US stock market. *Expert Systems with Applications, 39*(7), 6301–6308.
6. Yi, X., Jin, X., John, L., & Shouyang, W. (2014). A multiscale modeling approach incorporating ARIMA and ANNS for financial market volatility forecasting. Journal of Systems Science and Complexity, 27, 225–236.
7. Bao, W., Yue, J., & Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. PloS one, 12(7), e0180944.
8. Volna, E., Kotyrba, M., & Jarusek, R. (2013). Multi-classifier based on Elliott wave's recognition. Computers & Mathematics with Applications, 66(2), 213-225.

9. Cimino, M. G., Pedrycz, W., Lazzerini, B., & Marcelloni, F. (2009). Using multilayer perceptrons as receptive fields in the design of neural networks. Neurocomputing, 72(10-12), 2536-2548.
10. Alfeo, A. L., Barsocchi, P., Cimino, M. G., La Rosa, D., Palumbo, F., & Vaglini, G. (2018). Sleep behavior assessment via smartwatch and stigmergic receptive fields. Personal and Ubiquitous Computing, 22(2), 227-243.
11. Wang, L., An, H., Xia, X., Liu, X., Sun, X., & Huang, X. (2014). Generating moving average trading rules on the oil futures market with genetic algorithms. Mathematical Problems in Engineering, 2014.
12. Fu, T. C., Chung, F. L., Luk, R., & Ng, C. M. (2007). Stock time series pattern matching: Template-based vs. rule-based approaches. Engineering Applications of Artificial Intelligence, 20(3), 347-364.
13. Dini, G., Foglia, P., Prete, C. A., & Zanda, M. (2014). Social and Q&A interfaces for app download. Information Processing & Management, 50(4), 584-598.
14. Foglia, P., Prete, C. A., & Zanda, M. (2008, May). Relating GSR signals to traditional usability metrics: Case study with an anthropomorphic web assistant. In Instrumentation and Measurement Technology Conference Proceedings, 2008. IMTC 2008. IEEE (pp. 1814-1818). IEEE.
15. Bartolini, S., Foglia, P., & Prete, C. A. (2018). Exploring the relationship between architectures and management policies in the design of NUCA-based chip multicore systems. Future Generation Computer Systems, 78, 481-501.