Expert Systems With Applications 90 (2017) 290–302

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A stacked generalization system for automated FOREX portfolio trading

Anastasios Petropoulos[a,b], Sotirios P. Chatzis[a,*], Vasilis Siakoulis[b], Nikos Vlachogiannakis[b]

[a] Department of Electrical Engineering, Computer Engineering and Informatics, Cyprus University of Technology, Limassol 3036, Cyprus
[b] Bank of Greece, Athens 10250, Greece

ARTICLE INFO

Article history:
Received 19 May 2017
Revised 16 July 2017
Accepted 4 August 2017
Available online 18 August 2017

Keywords:
Forex forecasting
Algorithmic trading
Portfolio management
Machine learning
Stacked generalization

ABSTRACT

Multiple FOREX time series forecasting is a hot research topic in the literature of portfolio trading. To this end, a large variety of machine learning algorithms have been examined. However, it is now widely understood that, in real-world trading settings, no single machine learning model can consistently outperform the alternatives. In this work, we examine the efficacy and the feasibility of developing a stacked generalization system, intelligently combining the predictions of diverse machine learning models. Our approach establishes a novel inferential framework that comprises the following levels of data processing: (i) We model the dependence patterns between major *currency pairs* via a diverse set of commonly used machine learning algorithms, namely support vector machines (SVMs), random forests (RFs), Bayesian autoregressive trees (BART), dense-layer neural networks (NNs), and naïve Bayes (NB) classifiers. (ii) We generate implied signals of exchange rate fluctuation, based on the output of these models, as well as appropriate side information obtained by analyzing the correlations across currency pairs in our training datasets. (iii) We finally combine these implied signals into an aggregate predictive waveform, by leveraging majority voting, genetic algorithm optimization, and regression weighting techniques. We thoroughly test our framework in real-world trading scenarios; we show that our system leads to significantly better trading performance than the considered benchmarks. Thus, it represents an attractive solution for financial firms and corporations that perform foreign exchange portfolio management and daily trading. Our system can be used as an integrated part in international commercial trade activities or in a quantitative investing framework for algorithmic trading and carry-trade speculation.

1. Introduction

The foreign exchange market (FOREX, FX) is a global over-the-counter (OTC) market for trading one currency in exchange of another. After the collapse of the Bretton Woods system in 1973, the daily average turnover of the FX market has immensely grown to reach $5.1 trillion per day in April 2016. This is almost twice the daily average volume for OTC interest rate products, thus rendering FX the currently most liquid market.[1] The evolution of exchange rates is not only an important determinant of the value of cash flows denominated in foreign currencies for international businesses; most importantly, it is a variable of interest to investors that seek to identify opportunities of speculating on any imminent market movements (Ozturk, Toroslu, & Fidan, 2016). At the same time, FX rates are significantly influenced by monetary policy authorities. Indeed, this is the case since FX rates constitute a means for maintaining price stability, via controlling money supply, inflation, and interest rates. Thus, broad empirical evidence proves that geopolitical, monetary, and economic developments can significantly influence the fluctuation of FX financial time series, with their price essentially functioning as an aggregator of all economic activity (Beine, Laurent, & Palm, 2009).

Today, there are approximately 182 official currencies worldwide; however, only a small fraction of them is traded in the FX market. Indeed, over 95% of all daily FX transactions involves only eight currencies, namely the U.S dollar (USD), the Euro (EUR), the British pound (GBP), the Japanese yen (JPY), the Swiss franc (CHF), the New Zealand dollar (NZD), the Australian dollar (AUD) and the Canadian dollar (CAD). This liquidity profile is reasonable, since these currencies belong to economic areas with stable governments, respected central banks and relatively low inflation. In turn, the most traded currency pairs comprise the four *majors*, i.e.

* Corresponding author.
E-mail addresses: tasospetropoulos@gmail.com (A. Petropoulos), soteri0s@me.com (S.P. Chatzis), VSiakoulis@bankofgreece.gr (V. Siakoulis), NVlachogiannakis@bankofgreece.gr (N. Vlachogiannakis).

[1] Triennial Central Bank Survey of FX and OTC derivatives markets, 2016.

http://dx.doi.org/10.1016/j.eswa.2017.08.011
0957-4174/

EUR/USD, USD/JPY, GBP/USD, and USD/CHF, followed by the three *commodity* pairs, i.e. AUD/USD, USD/CAD, and NZD/USD. Due to these facts concerning market liquidity, FX portfolio trading typically focuses on this set of seven major currency pairs, with the occasional addition of few extra currencies pertaining to peripheral economies of the European Union (Égert & Kocenda, 2014).

In this context, this work attacks the problem of designing effective automated portfolio trading systems for the FX market. This is a problem that has been at the epicenter of financial analysis, econometrics, as well as machine learning research, for quite long a time (Dymova, Sevastjanov, & Kaczmarek, 2016; Ince & Trafalis, 2006; Sermpinis, Dunis, Laws, & Stasinakis, 2012). Indeed, the considered problem has been proven to be quite hard, since it poses significant challenges to time-series modeling algorithms. Specifically, financial time series are noisy, since their fluctuation may be affected by factors that are hard to record and objectively quantify in a way that allows for them to be used as independent variables in the context of a time-series modeling algorithm. They are also known to exhibit a deterministically chaotic behavior; this means that financial variables behave as random variables in the short term, while their long-term trends exhibit a clear deterministic behavior, since they tend to converge towards their equilibrium levels. Further, financial time series are clearly non-stationary signals; in other words, their underlying distribution does not remain the same over time (Ohnishi et al., 2012). Indeed, this non-stationary nature can be attributed to structural breaks that occasionally arise from political events, government policies, as well as changes in the expectations and in the risk preferences of investors. Finally, their underlying patterns entail clear nonlinear patterns of dependencies; hence, intricate nonlinear time-series models must be developed for the purpose of accurately predicting their future price fluctuation.

Motivated from these facts, significant research effort has been devoted to the development of novel time-series modeling algorithms, capable of accounting for the noisy, nonlinear, and volatile nature of FX rates time-series. However, in addition to these fundamental challenges to the employed time-series modeling algorithms, the FX market is also characterized by a number of peculiarities that must be carefully taken care of when designing an automated FX trading system. These include: (i) the very high liquidity of the FX market; indeed, its huge trading volume represents the largest asset class in the world; (ii) its geographical dispersion; (iii) its continuous 24 h per day/ 5 days per week operation (i.e., trading from 22:00 GMT on Sunday (Sydney) until 22:00 GMT Friday (New York)); (iv) the variety and limited predictability of factors that affect exchange rates, e.g. related to geopolitical developments; (v) the low margins of relative profit compared to other markets of fixed income; and (vi) the widespread use of leverage as a means of enhancing profit and loss margins. Despite these challenges, though, these very characteristics also render FX the market closest to the ideal of perfect competition, notwithstanding currency intervention by central banks. Thus, they represent a huge opportunity that is ripe to be exploited by informed portfolio managers and algorithm developers.

In this paper, we address these challenges by introducing an automated FX trading framework that is complicated in nature yet, at the same time, parsimonious in the intuition. To capture nonlinear dependence patterns we consider employing a variety of state-of-the-art machine learning techniques, shown to perform well in real-world modeling scenarios, where the absence of linearities is the norm. To account for the underlying nonstationary nature of the modeled time-series, we ensure adjustment and adaptation of the employed predictive models, by means of continuous retraining.

To provide an effective means of overcoming the limitations posed by the noisy nature of the modeled data, we perform FX time-series modeling under a multidimensional modeling setup. Specifically, instead of merely training distinct time-series models on datasets pertaining to each one of the considered currency pairs, we also infer implied correlation signals across currency pairs. This way, we progress well beyond the univariate modeling setup adopted by existing FX portfolio trading systems, by allowing for the structural relationships between currency pairs to be inferred from the data, and be taken into consideration for the purpose of price fluctuation prediction and trading decision generation. Hence, this methodological aspect of our approach, targeting the utilization of additional information hidden in the correlation pairs, constitutes a major contribution this paper brings to the literature.

Indeed, the importance of modeling and studying currency correlation has been extensively studied in economics literature. For instance, Mizuno, Takayasu, and Takayasu (2006) have thoroughly analyzed the correlation networks among currencies; this revealed important information regarding currency dependencies around the globe, with special focus on the peripheral currencies. In a similar vein, Keskin, Deviren, and Kocakaplan (2011) presented a topology of correlation networks among 34 major currencies; this revealed significant correlation information regarding the dominance of EUR and USD as world currencies, as well as the fact that contagion patterns during the recent financial crisis differed by currency and by region. Inspired from these outcomes, several researchers have recently proposed various alternative approaches for extracting correlation information regarding the movements in the FX market (e.g., Miśkiewicz, 2016; Casarin, Tronzano, & Sartore, 2013).

Despite this extensive research scrutiny, though, to the best of our knowledge, such correlation information has never been leveraged in the context of FX portfolio trading systems that learn to effectively fuse information of diverse nature. On the contrary, the main thesis of this work is that the incorporation of such inferred currency correlation information can offer significant performance gains to FX portfolio trading systems, by allowing for better data modeling capacities. Besides, a system utilizing correlations among currency pairs can also provide diversification and money management benefits, as is the case for all financial assets.

In a nutshell, we devise a novel stacked generalization technique that allows for intelligently combining predictive models driven from individual currency exchange rate signals, as well as inferred cross-currency correlation signals. The main characteristics of our approach, the *unique combination of which* endow it with significant advantages over the state-of-the-art, are the following:

1. Our proposed FX trading system jointly models the performance of 10 currency pairs, by inferring their correlations in the context of a multidimensional modeling setup, and leveraging them for the purpose of prediction generation. This is in contrast to the existing literature, where the used time-series models do not leverage such across-the-board correlation information.
2. We perform forecasting using a moving window retraining approach. In other words, the used predictive models are retrained on a moving window, so as to account for the fact that FX dynamics may change over time, influenced by monetary policy divergences, political instability, and economic growth. This is congruent with recent trends in the literature (e.g. Byun et al., 2015; Costantini, Cuaresma, & Hlouskova, 2016; Shen, Chao, & Zhao, 2015), and is contrast to older approaches that either are incapable of rapidly adapting to structural breaks in the world economy, or employ complex Markov-switching models that are highly prone to overfitting (e.g., Bahrepour et al., 2011; Sermpinis et al., 2012).
3. We establish a multilevel, multicomponent modeling framework for FX price forecasting, driven by machine learning al-

gorithms, and stacked generalization techniques. Stacking takes place at a diverse set of system levels: both at the level of computing implied signals through moving window correlations, as well as at the level of combining distinct predictive components via machine learning techniques. Such a stacked generalization framework, which also takes into account inferred implied correlation signals, apart from raw exchange rate signals, is a novel solution that has never been reported in the past.

We perform an exhaustive experimental evaluation of our system, using market data over a long time-period which spans a total of 15 years. The long time period considered under our experimental setup allows to explore system performance under volatility-stress periods and smooth trending periods alike.

The remainder of this study is organized as follows. In Section 2, we provide a brief overview of related empirical work, and explore the most commonly used modeling techniques for FX rates forecasting. In Section 3, we elaborate on the dataset used for developing and evaluating our system. In Section 4, we provide a thorough description of the motivation and the architecture of our developed FX portfolio trading system. In Section 5, we elaborate on our experimental setup, and present all our empirical results. In addition, we assess our empirical findings by analyzing their statistical significance and resulting real-world trading performance. Finally, in Section 6 we draw our conclusions, while also indicating directions for future research.

2. Related work

There is an abundance of research work in automated trading systems. Most of these are generic asset trading systems, as opposed to systems specifically tailored to the unique characteristics of the FX market. In general, these systems are based on time-series analysis techniques, nonlinear regression models, and, more recently, advanced machine learning techniques (Patel, Shah, Thakkar, & Kotecha, 2015). Similar is the landscape of FX trading systems, with advanced machine learning algorithms lying at their core.

In this context, neural network (NN) models (LeCun, Bengio, & Hinton, 2015) have been examined in multiple papers as an effective means for financial time-series forecasting. Dense-layer architectures have been the most typically used ones, while recurrent neural networks are the second most-used approach. However, the empirical evidence obtained so far regarding their efficacy remains rather inconsistent. Indeed, while some studies have shown promising results, there is also a plethora of recent works, including papers on FX fluctuation forecasting, where NNs failed to outperform the maximum benchmarks in all cases (Alfaro, García, Gamez, & Elizondo, 2008; Dunis & Huang, 2002; Ince & Trafalis, 2006; Lam, 2004). Better and more consistent results have been recently reported by leveraging another type of NNs, namely restricted Boltzmann machines (RBMs) (Carreira-Perpinan & Hinton, 2005). However, the efficacy of this method has not been scrutinized in a variety of experimental scenarios, apart from the ones reported in Takeuchi and Lee (2013); therefore, its strengths and weaknesses are yet to be convincingly examined.

On the other hand, support vector machines (SVMs) have also been extensively used for predicting future prices in FX time-series (Vapnik, 1998). Indeed, classical works, such as Kim (2003), have reported better performance using SVMs compared, e.g., to dense-layer NN-based approaches. Markov-switching models constitute another extensively considered framework (Frömmel, MacDonald, & Menkhoff, 2005; Swishchuk, Tertychnyi, & Elliott, 2014). Their main advantage consists in their inherent capability of detecting regime switches in observed sequential data. However, they also suffer from high computational complexity, and quite often, poor generalization capacity due to overfitting. Markov state space models, such as Kalman filters (Harvey, 1990), have also been used as a continuous latent variable alternative (Dunis & Shannon, 2005). Finally, random forests (RFs) (Breiman, 2001) have also been considered due to their capability of handling noisy data, and their robustness against overfitting (Krauss, Do, & Hu, 2017).

Despite these advances, as we have already discussed, no single predictive model has managed to consistently dominate other state-of-the-art alternatives in all experimental scenarios. Different models work best at different settings, even when considering trading of the same type of assets. Therefore, it is reasonable to posit that developing methods for intelligently combining predictions stemming from alternative trained models might allow for increased predictive performance compared to any individual trained model. Indeed, this is an idea that has been previously considered by many researchers, in the context of several markets. For instance, Swanson and Zeng (2001) used a simple ridge regression model that learns to combine underlying predictive models, while Chan, Stock, and Watson (1999) used Bayesian Information Criteria. However, real-world financial applications of this idea have failed to corroborate its efficacy; instead, the reported outcomes are rather contradictory. Characteristic examples are reported applications in output growth projection (Stock & Watson, 2004), US employment growth forecasting (Rapach & Strauss, 2008), and forecasting price changes for the NYSE Composite Index (Leigh, Purvis, & Ragusa, 2002).

3. Data collection and preprocessing

Our dataset is downloaded from Bloomberg (US close session time). It comprises the daily FX rates of 10 currencies; these include EUR, JPY, GBP, CAD, CHF, AUD, NZD, SEK, NOK and DKK. Thus, our dataset covers more than 95% of FX market's liquidity, including three categories of currencies: major currencies, commodity currencies, and European peripheral currencies. Our dataset covers a 15-year period, which spans from 1/1/2001 up to 31/12/2015; this results in around 4000 observations for each currency pair. All currency pairs are quoted against USD *(base currency)*.

As expected based on the related literature, the so-obtained exchange rate time-series of all the considered 10 currency pairs (denominated against the USD) are non-normal and no-stationary. We have confirmed this fact in a statistical way by using the Jarque–Bera statistic; the outcome suggests non-normality at the 99% confidence interval. To account for the non-stationary nature of our data, we transform all the obtained exchange rate time-series into time-series of *daily rates of returns*. Denoting as $\{p_t\}_{t=1}^T$ the observed exchange rate time-series, the corresponding returns series $\{r_t\}_{t=1}^T$ yields $r_t \triangleq \log p_t - \log p_{t-1}$. Thus, the returns series $\{r_t\}_{t=1}^T$ is eventually the one we use to develop our system.

We split the so-obtained dataset into two parts: (i) The first 500 observations of each returns time-series are used to select/fit each machine learning technique (c.f., Section 4.1). (ii) The rest of the observations is split into three periods, which correspond to different economic cycles of the US economy. The first corresponds to a weak period for the USD, the second corresponds to a volatile period, coinciding with the last Great Recession, and the third corresponds to a strong period for the USD. Thus, this sort of data split allows for us to thoroughly examine the specific performance characteristics of our approach under different economic environments. This split of our available data is summarized in Table 1.

In Fig. 1, we illustrate the currency exchange rate movements for the USD/EUR pair during the three considered periods. To further illustrate the qualitative differences between these three periods, we plot the ten-day exchange rates volatility time-series in Fig. 2. Our observations corroborate our intuition that these periods are characterized by a significantly different economic envi-

Table 1
Summary of analyzed dataset subsets.

Periods	Number of Trading Days	Start Date	End Date
Model Training	510	1/1/2001	13/12/2002
Period #1 (Weak USD Index)	1130	16/12/2002	13/4/2007
Period #2 (Volatile USD Index)	1130	16/4/2007	12/8/2011
Period #3 (Strong USD Index)	1130	15/8/2011	11/12/2015

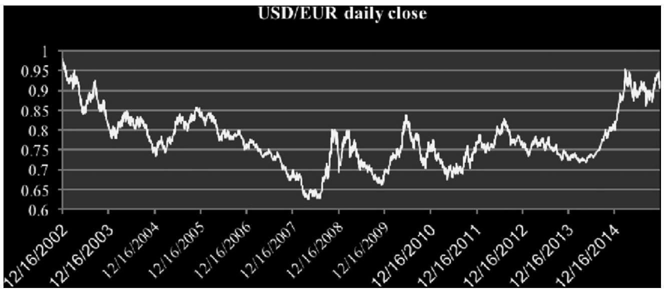

Fig. 1. Time-series of USD/EUR currency exchange rate movements.

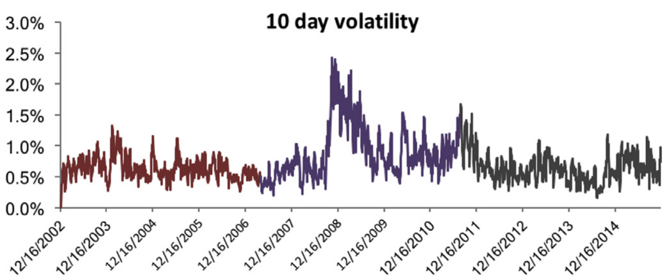

Fig. 2. Ten-day exchange rates volatility for the USD/EUR pair.

ronment: A consistently weak period for the USD, followed by a volatile period for the USD and the world economy alike, which is succeeded by a consistently strong period for the USD.

Finally, in Table 2 we illustrate some key statistical properties of the analyzed return time-series, and how they fluctuate among the considered three periods. Based on the skewness and kurtosis of the return time-series, we conclude that daily returns may be normally distributed. Hence, machine learning methods suitable for modeling normally-distributed data are expected to perform well in our setting.

4. Proposed approach

The proposed automated FX trading system comprises three subsequent levels of functionality, as shown in Fig. 3. At the first level, we postulate and fit a variety of machine learning algorithms for generating *original predictive signals* of exchange rate fluctuation (upward or downward). Specifically, we use a naïve bayes (NB) binary classifier, as well as the following regression models: SVM-regression, NNs, RFs, and a related "sum-of-trees" model where averaging is performed under a Bayesian inference rationale, namely Bayesian additive regression trees (BART) (Chipman, George, & McCulloch, 2010). Note that BART has never been considered before as a method for financial time-series modeling. For each type of machine learning method, we fit a different model to perform prediction for each of the ten considered currency pairs. This makes a total of 50 trained models (5 methods times 10 different currency pairs). For each trained model, we use as independent variables (i.e., model input) the 5 last observations from all the modeled currency pairs (i.e., the log-returns pertaining to the last 5 days). This results in an input vector of size 50 (i.e., the log-returns pertaining to the last 5 days times 10 modeled currency pairs).

At the second level, we compute the correlations across currency pairs to generate additional *implied predictive signals* that arise as derivatives stemming from the respective original predictive signals. For instance, let as suppose that we compute a positive correlation coefficient between the USD/EUR exchange rate signal and the USD/JPY exchange rate signal. Let us also assume that the trained RF model pertaining to the USD/EUR signal generates a positive ("buy") predictive decision. Then, since the two currency pairs are reckoned to be positively correlated, an*implied buy signal* is also inferred for USD/JPY. This inferential process is implemented for all trained models and all currency pairs. Thus, for each currency pair, and on each trading day, we obtain a total of 45 implied signals, by making use of the computed correlations with the other 9 currency pairs, and the generated predictive signals of the 5 machine learning models fitted to each currency pair.

Finally, at the third level of the system, we devise an intelligent and effective method for aggregating the obtained predictive signals into a global predictive signal for each currency pair, that can be reliably used for FX trading. Indeed, we have examined a number of alternative approaches for best addressing this challenging task. This way, we have devised a novel stacked generalization approach for FX trading. We outline it in the following Sections, along with the details of the fitting and development procedures that we adopted for each employed machine learning algorithm.

4.1. First level: configuration of the used machine learning models

As briefly outlined in Section 3, the first 500 observations of the available time series are used to fit each machine learning model. The criterion we use for this purpose is the in-sample predictive accuracy of each fitted model configuration. As we have already explained, each postulated model is fitted to perform prediction for one considered currency exchange rate. To overcome model specification errors and to avoid co-integration issues, the independent (input) variables of each model constitute the vector of log-returns of all the considered exchange rates pertaining to the last 5 trading days (a 50-dimensional vector of real numbers). This choice provides a more robust approach to data modeling, which increases our forecasting ability and avoids contemporary causality issues. The output variable of the fitted NB classifiers takes in the set $\{+1, -1\}$, representing a prediction of a positive or negative move of the modeled exchange rate on the next trading day (hence, corresponding to a "buy" or "sell" predictive suggestion, respectively). Regarding the rest of the fitted models, which constitute regression methods, the outputs represent the predicted log-return of the modeled exchange rate pertaining to the following trading day.

At this point, we underline that our selection of the number of input variables to be limited to the (log-returns of the) last five days is based on the fact that the log-returns of financial time series are stationary and exhibit insignificant serial autocorrelation. On this basis, and in order to determine the number of past observations to use as model input variables, we run a series of autoregressive (AR) models for each currency pair and derived the maximum number of lags for which the relevant coefficient of the AR model was found statistically significant. This analysis showed that the maximum AR model lag found to be statistically significant is the *fifth* for all the 10 currency pairs.

The used SVMs employ a radial basis function (RBF) kernel with the default hyperparameter values, as defined in the *e1071* package of the R programming language, which we used to develop our system. The employed dense-layer NNs[2] comprised two hidden

[2] Our implementation was based on the *neuralnet* package of the R programming language.

Table 2
Key statistical properties of the analyzed returns time-series during the considered three periods.

Period 1	USD/EUR	USD/JPY	USD/GBP	USD/CAD	USD/CHF	USD/AUD	USD/NZD	USD/SEK	USD/NOK	USD/DKK
Mean	−0.02%	0.00%	−0.02%	−0.03%	−0.02%	−0.03%	−0.03%	−0.02%	−0.02%	−0.02%
Standard Error	0.02%	0.02%	0.02%	0.02%	0.02%	0.02%	0.02%	0.02%	0.02%	0.02%
Median	−0.02%	0.00%	−0.02	−0.03%	0.01%	−0.05%	−0.07%	−0.02%	−0.01%	−0.03%
Standard Deviation	0.6%	0.5%	0.5%	0.5%	0.6%	0.6%	0.7%	0.7%	0.7%	0.6%
Sample Variance	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Kurtosis	0.62	0.99	0.36	0.48	0.65	0.94	0.89	0.42	0.49	0.64
Skewness	0.02	(0.17)	0.03	0.03	(0.09)	0.37	0.41	(0.05)	0.15	0.04
Range	3.87%	4.57%	3.76%	3.45%	4.79%	4.55%	5.40%	4.72%	5.15%	3.86%
Minimum	−1.90%	−2.37%	−2.05%	−1.72%	−2.54%	−1.89%	−2.47%	−2.46%	−2.70%	−1.90%
Maximum	1.97%	2.21%	1.71%	1.73%	2.26%	2.66%	2.93%	2.25%	2.45%	1.96%
Period 2	USD/EUR	USD/JPY	USD/GBP	USD/CAD	USD/CHF	USD/AUD	USD/NZD	USD/SEK	USD/NOK	USD/DKK
Mean	0.00%	−0.04%	0.02%	−0.01%	−0.04%	−0.02%	−0.01%	−0.00%	−0.01%	0.00%
Standard Error	0.02%	0.02%	0.02%	0.02%	0.02%	0.03%	0.03%	0.03%	0.03%	0.02%
Median	−0.02%	−0.02%	0.00%	−0.04%	−0.03%	−0.10%	−0.09%	−0.05%	−0.04%	−0.02%
Standard Deviation	0.7%	0.8%	0.7%	0.8%	0.8%	1.2%	1.1%	1.0%	1.0%	0.7%
Sample Variance	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Kurtosis	1.41	4.16	2.28	1.63	4.21	7.87	2.20	1.98	2.38	1.41
Skewness	(0.09)	0.14	0.42	0.18	(0.22)	0.42	0.38	0.03	0.09	(0.09)
Range	5.92%	9.01%	6.39%	7.25%	9.37%	15.58%	10.81%	9.20%	9.84%	5.93%
Minimum	−3.48%	−3.50%	−2.92%	−4.00%	−4.69%	−8.27%	−4.29%	−4.98%	−4.97%	−3.50%
Maximum	2.44%	5.50%	3.47%	3.25%	4.68%	7.31%	6.52%	4.22%	4.87%	2.44%
Period 3	USD/EUR	USD/JPY	USD/GBP	USD/CAD	USD/CHF	USD/AUD	USD/NZD	USD/SEK	USD/NOK	USD/DKK
Mean	0.02%	0.04%	0.01%	0.03%	0.02%	0.03%	0.02%	0.02%	0.04%	0.02%
Standard Error	0.02%	0.02%	0.01%	0.01%	0.03%	0.02%	0.02%	0.02%	0.02%	0.02%
Median	0.00%	0.02%	0.00%	0.02%	0.03%	0.02%	0.03%	0.02%	0.02%	0.01%
Standard Deviation	0.6%	0.6%	0.5%	0.5%	0.9%	0.7%	0.7%	0.7%	0.7%	0.6%
Sample Variance	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Kurtosis	2.13	4.14	0.88	1.11	222.39	1.60	1.16	1.08	1.46	2.15
Skewness	(0.09)	0.22	0.08	0.18	(8.67)	(0.09)	0.11	0.02	0.35	(0.10)
Range	5.15%	6.50%	3.18%	3.74%	28.47%	6.17%	5.81%	5.29%	6.05%	5.26%
Minimum	−3.01%	−3.02%	−1.55%	−1.75%	−19.38%	−3.13%	−2.67%	−2.85%	−2.22%	−3.01%
Maximum	2.15%	3.49%	1.64%	1.99%	9.09%	3.05%	3.14%	2.44%	3.82%	2.25%

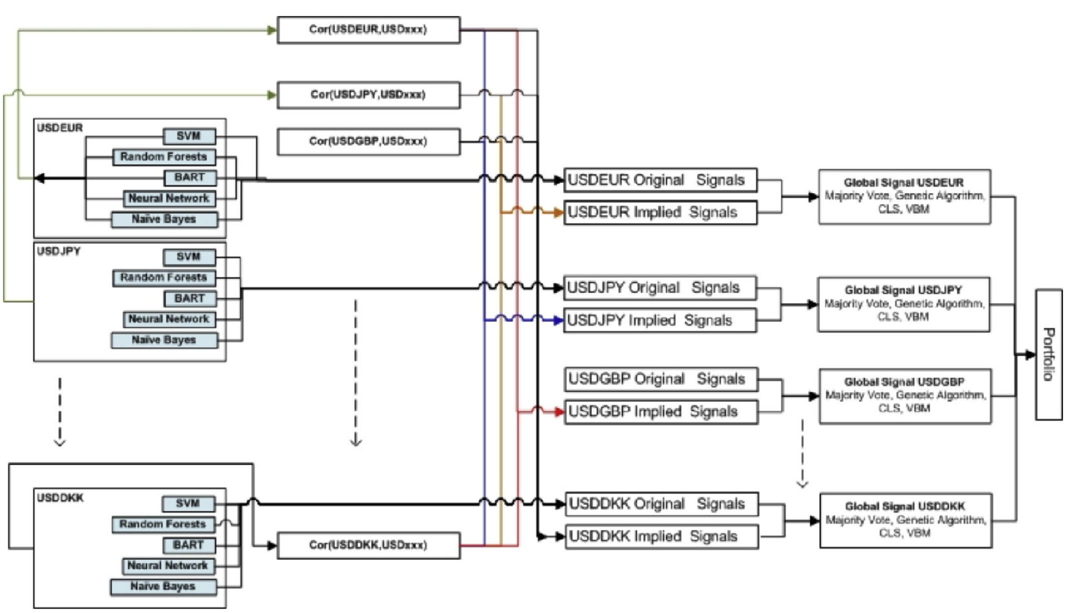

Fig. 3. A flowchart of the proposed system.

layers of 15 and 5 neurons with *tanh* activation functions; this configuration was selected on the grounds of the obtained in-sample predictive accuracy in the training set, as described previously. The employed regression RFs comprised 300 random trees; we adopted the default setup of their training algorithm, as implemented in the *randomForests* package of the R programming language. BART was initialized with 50 additive trees; subsequently, it was allowed to perform its entailed sampling procedures so as to determine a data-driven "best" number of trees. We adopted the default configuration of its Markov chain Monte Carlo (MCMC) training algorithm implemented in the *bartMachine* package of the R programming language.

Finally, to account for changes in the economic conditions, the employed models are not trained once and for all. Instead, we fol-

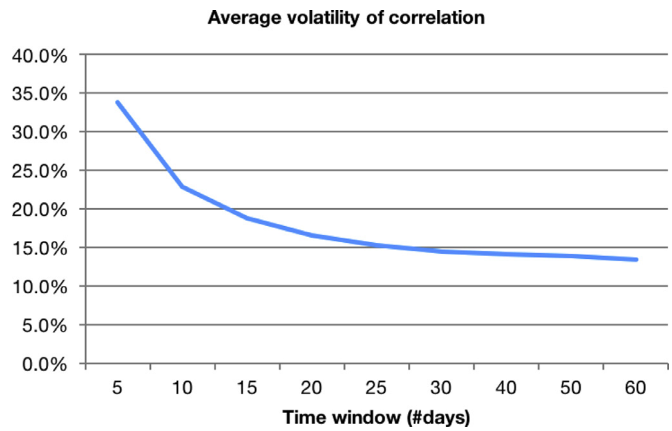

Fig. 4. Average volatility of computed correlation coefficients over time windows of different lengths.

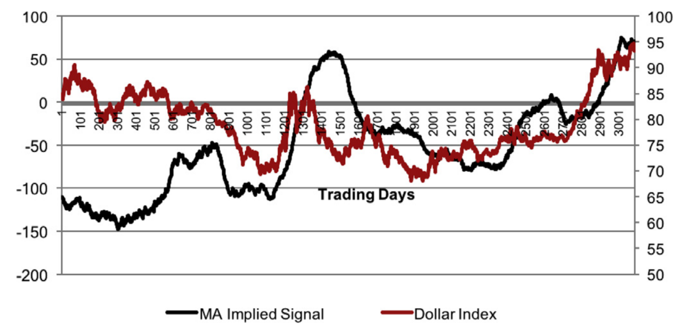

Fig. 5. Predictive value of the constructed predictive implied signals, and their correlation with the USD index movement.

low a moving window-based retraining approach, under which: (i) We initially train all the postulated machine learning models on the first 500 data points; we use the corresponding in-sample performance statistics to also determine any applicable hyperparameter values, e.g. number of neurons in the trained NNs. (ii) After 30 days, we retrain all model parameters on the data pertaining to the last 500 days (but retain their hyperparameter selection). (iii) We repeat this procedure until the end of the period under examination.

4.2. Second level: implied predictive signals derivation

Implied predictive signals derivation is performed by computing all the pertinent correlation coefficients across currency pairs. To properly do so, we need to decide upon the time window over which we perform this kind of computation. This entails keeping a balance between two opposing objectives: (i) ensuring the stability of the estimated correlation coefficients, so that they reflect the realized level of co-movement between different currencies in the short to medium term; and (ii) allowing for quick adaptation to the current market conditions, by promptly capturing the volatile relationships among currency pairs in the FX market.

To this end, we performed an initial investigation of this matter, by leveraging the observations pertaining to the first 500 days in our dataset to calculate the correlations between currency pairs over time-windows of variable lengths, ranging from 5 days to 60 days. Subsequently, we estimated the volatility (standard deviation) of the so-obtained correlation coefficients, in order to determine the window length that exhibits the most plausible behavior, i.e. not too volatile but still able to swiftly capture dependence pattern changes due to specific monetary fundamentals shifts. The outcomes of this analysis are briefly summarized in Fig. 4. As we observed, the use of a 30-day window achieves the best balance between these objectives; correlation windows shorter than 30 days tend to increase the volatility of the estimated correlation coefficients, thus increasing the noise in our projections; longer correlation windows appear to make our inferred implied signals less responsive, thus risking our forecasting process to be contaminated by rather obsolete data.

Eventually, we augment the set of original predictive signals, generated by the employed machine learning models, with additional (implied) predictive signals that may also be of utility to the developed FX trading system. For instance, in case the trained RF pertaining to the USD/EUR currency pair indicates an appreciation of EUR, and we reckon that EUR appears to have a positive correlation with JPY in the last 30 days, then we extract an implied appreciation signal also for USD/JPY. This way, on each day, and for each exchange rate, we generate 50 predictive signals: 5 stemming from the original machine learning models trained at the first level, and 45 implied signals obtained by utilizing the computed correlations.

Having obtained this set of 50 predictive signals for each currency pair, we further proceed to binarize them so as to yield the eventually sought *buy or sell signals* that are useful for driving an automated FX trading system. To this end, we use a simple step function filter, which assigns a -1 value (*sell signal*) when the prediction indicates a downward movement, and 1 (*buy signal*) otherwise. The so-obtained signals carry significant inferential power, as they provide an indicator about the USD strength over all the considered currencies.

To better illustrate this fact, in Fig. 5 we plot the *summation* of the so-obtained 50 signals across all the examined exchange rates (black line). This results in a predictive index with values ranging in the interval $[-500, 500]$; this corresponds to 5 original predictive signals from the postulated machine learning models, in addition to another 45 implied signals, that are computed for a total of 10 currency exchange rates, and take values in the set $\{-1, +1\}$. Values close to 500 are expected to correlate with an across-the-board strong USD, while values much below zero should indicate a period of notable USD weakness. To corroborate this intuitive claim, in the same figure we also plot the USD market index over the same 15-year period (red line). As we observe, the economic meaning of the generated predictive index is profound: In Period #1 (16/12/2002–13/4/2007), during which the USD consistently depreciates across the board, the inferred predictive index is negative. The opposite happens during Period #3 (15/8/2011–11/12/2015), which coincides with a period of USD appreciation, due to the tightening of the FED monetary policy.

4.3. Third level: predictive signal aggregation

The set of 50 binary predictive signals (buy/sell values), obtained at the second level of our system, must be aggregated in an effective way, so as to generate the final predictive information that will be used to drive our automated FX trading system. The simplest way of doing so is to postulate a committee machine, which simply averages the predictions corresponding to each one of the traded currency pairs. Such a procedure can be motivated from a frequentist perspective by considering the tradeoff between bias and variance; this decomposes the error due to a predictive model into the bias component that arises from differences between the model and the true function to be predicted, and the variance component that represents the sensitivity of the model to the individual data points. However, for this procedure to be valid, we need to ensure that the postulated committee comprises uncorrelated predictive models (Bishop, 2006, Chapter 14).

To confirm this, in Table 3 we compute the correlation of the original predictive signals produced by the trained machine learning models, across all the examined currency pairs. It is evident

Table 3
Correlation values between the generated predictions of the employed machine learning algorithms.

Correlation	RF	BART	NN	NB
SVM	40%	52%	25%	13%
RF		52%	19%	30%
BART			25%	26%
NN				13%

that the considered machine learning techniques exhibit weak correlation with each other; hence, our postulated committee machine is a theoretically sound method to use. Nevertheless, coming up with an effective, *stackedgeneralization mechanism,* that allows for assigning different weights to each one of the obtained predictive signals, for the purpose of their aggregation, should be well-expected to yield improved predictive performance.

We follow this route in this paper, considering three alternative approaches. The first one is simple *majority voting.* In this context, we have examined both utilization of all the obtained predictive signals, as well as utilizing only the original predictive signals, while skipping the implied ones. We have found that majority voting using the available predictive signals performs better than each separate machine learning model, as well as compared to majority voting among only the original predictive signals (c.f. Section 5). This provides strong empirical evidence that the obtained implied signals contain valuable information regarding FX movements, which can be leveraged to increase the performance of our trading strategy.

Another alternative we have considered is the use of a weighted averaging scheme, where the weights are selected by application of a genetic algorithm (GA). Specifically, once we fit our machine learning models on the available training dataset, we use a genetic algorithm to select five candidate sets of weights, for aggregating all the obtained predictive signals into a single global one. This selection is performed on the grounds of yielding the best predictive performance during Period #1. Of these 5 alternatives, we eventually retain the one that works best in Period #2.

Further, another alternative we consider is the well-known constrained least squares (CLS) forecast combination scheme. This is based on the modeling scheme

$$\mathrm{obs}_t = c + \sum_{i=1}^{p} w_i \hat{\mathrm{obs}}_{it} + e_t \tag{1}$$

where obs_t is the observed signal at time t, $c \in R$ is a trainable constant, p is the number of distinct predictor signals that we aggregate over, $w_i \in \mathbb{R}$ is the weight assigned to each of them, $\hat{\mathrm{obs}}_{it}$ is the corresponding predictive signal at time t, and e_t is the error at time t. Then, training of the postulated aggregator consists in estimating the $\{w_i\}_{i=1}^{p}$ and c values that minimize the total prediction mean square error (MSE) in the targeted dataset, under the constraint:

$$\sum_{i=1}^{p} w_i = 1 \tag{2}$$

To this end, we here utilize the data pertaining to Periods #1 and #2, as outlined in Table 1.

Finally, we also consider a variance-based method, also based on the weighted average scheme outlined in Eq. (1). The difference between this method and CLS consists in how the weights $\{w_i\}_{i=1}^{p}$ are computed. Indeed, the employed variance-based method adopts the simple scheme:

$$w_i = \frac{\mathrm{MSE}_i^{-1}}{\sum_{j=1}^{p} \mathrm{MSE}_j^{-1}} \tag{3}$$

where MSE_i is the mean square error obtained by only using the ith available signal to perform prediction. Thus, this approach assigns more weight to methods that yield higher accuracy in the used development set, which in our case remains the data pertaining to Periods #1 and #2.

4.4. Trading system

4.4.1. Trading strategy setup

Our automated trading strategy consists in investing one lot of funds on each currency pair every working day. This takes place after the US financial markets close and before the Asian markets open. Our strategy adheres to the buy or sell signals generated by our system; each trade remains open for as many consecutive days as the system-generated signal remains pointing to the same direction. In case the signals changes (from buy to sell or the other way around), we reverse the trade; that is, we close the current position and open a new position to the opposite direction. Our profit and loss (P&L) is calculated based on the agreement between the forecasts our system generates and the actual currency-pair move. Hence, in case the available forecast points to the direction the market eventually moves (i.e., appreciation or depreciation of the USD relative to the other currency in the pair), we consider that our P&L increases by the change in the exchange rate. On the contrary, in case the forecast points to the opposite direction, then the P&L of our strategy decreases by the change in the exchange rate. We assume that each day we invest the same amount of money, except for the case where the trade remains open. In the latter case, the invested amount is compounded by the P&L incurred in the previous day. Finally, note that we also introduce a "no trade" switch in our system. As we shall explain next, this provides a mechanism that allows for one to opt out from trading, under some specific conditions. In such a case, the P&L measurement of our system does not alter. Instead, we assume that we do not perform any transaction and have no open position.

4.4.2. Accounting for transaction costs and slippage

In order to establish a real-world trading strategy, we have to take into account that the actual execution of a trade bears some transaction costs. Necessarily, these must be taken into account by the decision generation algorithm of our trading system, and be reflected in our computation of the obtained P&L performance. Expectably enough, transaction costs are not the same for all retail FX providers. To account for this fact, we have thoroughly examined the costs charged by two leading FX trading platforms, namely FXCM and Saxotrader, and devised the following formula for our automated trading system

$$\text{Total cost per transaction} = (\text{Spread} + \text{Slippage}) \times (\text{PipCost}) \times N + C \tag{4}$$

In this expression, N is the number of lots that we trade, C is the round-turn-commission charged by the provider, the Spread term refers to the difference between the exchange rate applied to a customer willing to purchase a quote currency and the price of quote currency selling, while the PipCost is determined by each retail FX provider.

As we observe in Eq. (4), in our employed transaction costs formula we have also introduced slippage. This refers to the change of the currency price between the time a trader asks for a position to be opened or closed on their behalf, and the actual time this transaction takes place. To calculate a representative (expected) slippage value that can be used by our system, we performed a simulation analysis to estimate the gap in the currency exchange market. To allow for the reader to obtain a feeling of what the annualized trading costs look like for each considered currency (against the USD), in Table 4 we depict the mean transactions cost stemming

Table 4
Annualized trading costs for each considered currency (against USD).

Currency	Cost
SEK	1.8%
NOK	2.0%
JPY	1.5%
EUR	1.5%
DKK	1.8%
NZD	2.1%
GBP	1.7%
CAD	2.0%
AUD	2.1%
CHF	1.6%

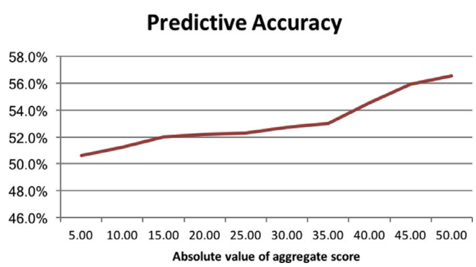

Fig. 6. Predictive value of the aggregate predictive signal that our system generates.

Table 5
Considered leverage strategies: number of lots comprising the position initiated for a specific currency.

Aggregate Predictive Signal (Absolute Value)	≤ 5	$\in (5, 15)$	≥ 15
Strategy #1: Number of lots	1	2	3
Strategy #2: Number of lots	1	3	5

from Eq. (4), computed over the considered test period (Period #3), assuming execution of 100 trades per calendar year (evenly distributed over time).

4.4.3. Leverage strategies

In real-world trading scenarios, it is frequently the case that a trader may be strongly confident in their prediction of the upcoming price fluctuation regarding a traded currency. In such a case, one may opt for simultaneously opening multiple lots of the same (initial) position. Indeed, such a leverage mechanism can have a multiplicative effect on profit, if the available predictions turn out to be successful. To examine whether we can feasibly allow for leverage introduction in our proposed trading system, we calculate a simple sum of all the obtained predictive signals for each traded currency. Since, for each currency, we obtain 5 binary predictive signals from the fitted machine learning models, plus another 45 binary implied (cross-correlation) signals, these computed aggregate signal values (summations) take on the interval $[-50, 50]$. Then, we assess whether higher absolute values of the computed aggregate signals correspond to more confident predictions regarding the actual market movements.

In Fig. 6, we plot the outcomes of this analysis; we observe that the predictive accuracy of the so-obtained aggregate signals increases with their absolute value. Specifically, a positive (buy) aggregate signal, or conversely, a negative (sell) aggregate signal is confirmed by an upward, or conversely, downward, market movement in around 50.5% of the time, on average. This increases to 56.5% in case the obtained absolute signal value takes on [45, 50]. Inspired from these findings, we consider introduction of the leverage strategies outlined in Table 5. These alternative strategies consist in the initiated positions comprising more than one lots, depending on how strong the absolute values of the computed aggregate predictive signals are. We evaluate both of them in the experimental section of this paper.

4.4.4. Introducing a stop-loss switch

Trading strategies usually adopt the philosophy "cut your losses quick, let your winning trades run." To investigate what the impact of such a philosophy would be on the P&L performance of our proposed automated FX trading system, we introduce a simple stop-loss threshold. Its functionality consists in simply closing a position when its aggregate loss exceeds that preset threshold.

4.4.5. The effect of volatility

News announcements and central bank decisions usually cause significant intraday volatility that may lead to sharp movements in the FX market (Bauwens, Omrane, & Giot, 2005). Such intraday volatility effects typically break short-term market movement patterns, thus prodding the investors to rapidly unwind positions or increase hedges. Indeed, as it has been shown in the recent literature (e.g., Sermpinis et al., 2012), such volatility outbreaks give rise to a "buy the rumor sell the news" investor behavior, whereby short-term FX movements defy the patterns anticipated by carefully analyzing both the relevant technical indicators of the economy as well as the available historical data.

To allow for our system to account for these facts, we introduce a no-trade switch that prevents trading on specific days. The employed switch is activated on the following occasions: (i) when a policy decision is published by the FED; (ii) when the US non-farm payrolls are announced (first Friday of each month); (iii) when the 5-day volatility of daily returns surpasses a threshold of 1.5%. In essence, this list of criteria comprises: (i) the most significant USD-related events that affect volatility; this selection is reasonable, since all the considered currency exchange rates are USD-denominated; and (ii) a back-stop criterion, that relies on the 5-day volatility to capture all other significant events affecting volatility in exchange rates movements (e.g., interventions from other central banks, political instability etc.). Thus, it is our firm belief that, although this list of criteria is not exhaustive, it can certainly offer a useful no-trade switch. We evaluate its effect on the P&L of the proposed trading system in the experimental results section that follows.

5. Experimental evaluation

In this Section, we provide a thorough experimental evaluation of our approach, along with comparisons to some baseline models and strategies. Specifically, the considered baselines comprise:

1. Naïve strategy: Under this strategy, we consider that the current-day currency movement will be repeated on the next day.
2. AR: We use a standard linear model, highly popular in the field of econometrics, namely the AR model (Asteriou & Hall, 2011). The model postulated for each currency is composed of five lags from that currency as well as one lag from the rest of the considered currencies. Training is performed similar to the algorithms employed by our system; we use a moving window of 500 days, and update the models every 30 days.
3. Buy and hold: Under this strategy, a long position is initiated for all currencies and held constant for the analyzed period.
4. Sell and hold: Under this strategy, a short position is initiated for all currencies and held constant for the analyzed period.
5. Random Walk: This method is useful for testing the random walk hypothesis against the employed machine learning techniques. It allows for examining how statistically significant is the difference between the obtained predictive signals and simple random selection. It consists in employing Monte Carlo sim-

Table 6
First system level: statistical performance analysis.

Method	SVM	RF	BART	NN	AR
MAE	0.54%	0.53%	0.54%	0.58%	0.64%
MAPE	2352.86%	4945.80%	2097.86%	1858.58%	5816.98%
RMSE	0.74%	0.73%	0.75%	0.81%	0.84%

ulation for all currencies with random scenarios, and aggregating the resulting P&L on portfolio level.

5.1. Performance analysis at the first system level

We commence our investigations considering the individual performance of each employed machine learning algorithm at the first level of the proposed system. We perform evaluations both from a predictive performance perspective as well as from a trading performance perspective. Statistical performance is assessed on the grounds of the obtained root mean square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) between the predicted currency returns and their actual values; these are standard error metrics employed in the related literature (e.g., Plakandaras, Papadimitriou, & Gogas, 2015; Sermpinis et al., 2012; Costantini et al., 2016). The obtained results are provided in Table 6. Note that, therein, we do not provide performance measurements for NB, as this is a binary classification method. That is, its generated predictive signals are not a posteriori binarized into buy and sell values by the system; rather, the method originally generates binary signals. Apparently, the accuracy of predictive signals of such a binary nature cannot be reliably measured by means of the MAE/MAPE/RMSE metrics. We observe that the employed machine learning models perform much better than the considered econometric model, i.e. AR. No machine learning model appears to outperform all the rest.

Turning to the trading performance of the employed methods, to obtain an objective assessment we convert their generated predictive signals into buy/sell values (binarization), and use them to implement our automated FX trading system, outlined previously. On this basis, we proceed to evaluate the trading performance of our system, considering that only one machine learning algorithm is employed, and that we do not introduce a stop-loss switch, a no-trade switch, or leverage strategy. Comparisons are performed considering the baselines described above. Our evaluation metrics comprise three standard metrics, namely the average annualized return, the annualized volatility, and the obtained Sharpe ratio, as well as the maximum drawdown, which is calculated as the maximum losses experienced as percentage of the P&L.

The obtained results are illustrated in Table 7. Based on the these results, we observe that SVM, NB, and RF exhibit the highest annualized return. All machine learning techniques perform better in all metrics against the benchmark models. In addition, all machine learning methods perform higher in Period #1, while performance in Periods #2 and #3 does not change dramatically compared to Period #1, regardless of the different volatility regimes. This is a significant merit of our approach, as it proves to exhibit consistent performance under different economic environments.

Further, in Table 8 we provide a breakdown of the annualized return obtained by the considered machine learning algorithms for each different currency. It is evident that there is no model performing best across all currencies, but there is a slight observed superiority of NB and RF over the rest. It is also interesting to notice that USD/JPY proves to be the most difficult currency exchange rate for financial modeling. Indeed, this is intuitively expected, since Japan authorities have been actively pursuing a policy of managing its currency strength for trade and economic reasons. Such a political intervention makes it difficult for machine learning models to infer relevant economic patterns. In addition, we observe increased performance in peripheral currencies, namely SEK and NOK. This is despite these currencies being less liquid and more volatile. This seemingly counterintuitive result implies that these currencies exhibit patterns that are more easily captured by machine learning models.

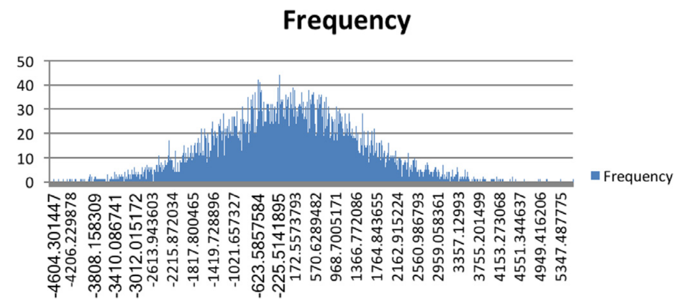

Fig. 7. Distribution of simulated random walk data.

Finally, to test our method against the random walk hypothesis, i.e. the randomness in the performance of each model, we produce a distribution of random walk simulations. To this end, we assume a trading system of one lot per currency exchange rate on a daily basis, and aggregate the results across all three scenarios. The statistical results of the simulation are depicted in Fig. 7. On this basis, we compare the performance of each model against the random walk distribution; the obtained results are illustrated in Table 9. We observe that all machine learning models reject the random walk hypothesis with 95% probability. This is not the case, though, for the considered benchmarks.

5.2. Performance analysis of the overall system

We begin our analysis of the overall system performance by scrutinizing the outcomes of majority voting. Specifically, in Table 10 we show how trading performance changes if we perform majority voting using only the original predictive signals, only the inferred implied signals, as well as all the predictive signals obtained by our system. These results concern evaluation over Period #1, Period #2, and Period #3, alike; we do not employ leverage, no-trade, or stop-loss switches in this case. As we observe, the proposed approach of performing majority voting over all the obtained predictive signals outperforms all the alternatives.

Further, in Table 11 we illustrate how majority voting compares to the considered alternative stacked generalization approaches, namely GA, CLS, and the variance-based approach outlined in Section 4.3. These results concern performance only during Period #3, as the data from Period #1 and Period #2 have been used for method development. As we observe, the proposed predictive signal weighting scheme produced better results than simple majority voting. Out of the alternative methods considered for the purpose of training the employed weights, GA was shown to exhibit the highest drawdown, predictive accuracy, and net annualized return after costs.

As discussed in Section 4.4.3, introduction of leverage is certainly expected to allow for a considerable increase in the net annualized return of our automated trading system. To verify this, in Tables 12(a)-(b) we illustrate how performance changes by applying either of the leverage strategies outlined in Table 5. These results concern performance only during Period #3, as the data from Period #1 and Period #2 have been used for method development; we have not employed the no-trade and stop-loss switches. As we observe, the comparative performance outcomes reported in Table 11 do not change in this case: GA proves to be the best-performing predictive signal weighting scheme out of the consid-

Table 7
First system level: trading performance analysis.

Method	SVM	Proposed		Approach		Naïve strategy	Buy-hold	Sell-hold	AR
		RF	BART	NN	NB				
Period 1 (Weak USD) Annualized Return	4.4%	6.0%	4.4%	2.3%	5.9%	−3.0%	−5.5%	5.5%	2.4%
Period 2 (Volatile USD) Annualized Return	1.4%	2.6%	1.3%	−0.7%	4.2%	0.0%	−3.1%	3.1%	1.9%
Period 3 (Strong USD) Annualized Return	2.8%	0.3%	0.3%	4.0%	1.3%	−5.1%	6.5%	−6.5%	0.7%
Average Annualized Return	2.87%	2.96%	1.99%	1.89%	3.79%	−2.72%	−0.72%	0.72%	1.64%
Annualized Volatility	6.52%	6.29%	6.42%	5.17%	6.09%	6.61%	8.64%	8.64%	6.11%
Sharpe Ratio	0.44	0.47	0.31	0.37	0.62	−0.41	−0.08	0.08	0.27
Maximum drawdown	−11%	−9%	−10%	−11%	−8%	−73%	−41%	−27%	−13%

Table 8
First system level: trading performance breakdown.

Currency	SVM	RF	BART	NN	NB
SEK	1.88%	5.72%	2.36%	6.64%	−0.77%
NOK	5.00%	8.22%	8.16%	3.56%	1.98%
JPY	0.79%	1.48%	−0.83%	−1.60%	0.66%
EUR	1.94%	3.44%	1.73%	1.42%	2.70%
DKK	3.37%	1.16%	2.88%	1.86%	3.72%
NZD	6.54%	−0.09%	3.36%	−1.95%	10.51%
GBP	1.85%	1.88%	0.38%	3.94%	2.67%
CAD	3.89%	1.17%	−0.47%	−1.09%	7.50%
AUD	1.25%	3.08%	−0.27%	1.01%	7.72%
CHF	2.16%	3.51%	2.59%	5.10%	1.18%

Table 9
p-values pertaining to the random walk hypothesis.

Method	SVM	RF	BART	NN	NB	Naïve strategy	AR	Buy-hold	Sell-hold
p-value	0.26%	0.24%	2.67%	3.11%	0.02%	99.68%	5.9%	76.64%	15%

Table 10
Majority voting: performance over all time periods; we do not employ leverage, no-trade, or stop-loss switches.

Method	Implied Signals	Original Signals	All Signals
Annualized return portfolio	4.70%	4.00%	5.00%
Maximum drawdown	−9.30%	−9.20%	−8.10%
Annualized volatility	6.40%	6.40%	6.60%
Information ratio (Sharpe ratio)	0.73	0.63	0.76

ered alternatives. On the other hand, it becomes apparent that the use of leverage leads to a remarkable increase in the annualized return**.** Note though that this appears to come at the cost of increased uncertainty and risk, which manifests itself as higher trading volatility. Hence, it is quite reasonable that Leverage Strategy #1, which is the more conservative of the two proposed alternatives, appears to yield a much better Sharpe ratio.

The effect of introducing the proposed stop-loss switch is summarized in Table 13; therein, we show the performance obtained by introducing that switch. We consider three different values for the stop-loss threshold, taking in {1%, 1.5%, 2%}. Interestingly, all these three threshold values yield exactly the same performance change for all the employed predictive signal weighting techniques. In all cases, the observed impact on the obtained system performance is minuscule; expectably enough, the best performer (GA) experiences the highest (yet minimal) performance reduction among the alternatives. This is due to a combination of two contradictory effects: (i) the stop-loss switch may reduce the losses stemming from very volatile losing trades; (ii) the stop-loss switch may increase the losses incurred in cases when, after the stop-loss threshold is hit, the trade recovers and becomes profitable or at least exhibits smaller losses. Based on these observations, we deem that the introduction of the proposed stop-loss switch is auspicious for our system: it bears a minimal reduction of trading performance in our test set, while acting as a good safeguard against the occurrence of extreme negative shocks in the economy.

Finally, in Table 14 we show how our system performs if we also activate the proposed volatility switch. As expected, we observe that the proposed no-trade switch does offer a notable increase to overall trading performance. This corroborates the findings of other researchers (e.g., Sermpinis et al., 2012) who have also reported that accounting for short-term volatility in a trading strategy can improve its profitability.

5.3. Further insights

To gain further insights into the performance of our overall system, we explore how it varies over the considered currencies. We depict the outcomes of this analysis in Table 15; these results concern the fully-fledged configuration of our model (i.e., including the considered leverage strategy as well as both the employed switches). As we observe, the trading performance of our system is higher when dealing with peripheral European currencies. This behavior is well-expected: major currencies include a lot of noise, as they behave as aggregators of global news, country-specific news, monetary policies, and geopolitical risks. Thus, majors tend to exhibit rapid and sharp intraday movements, which have a negative impact on system performance. The same fact, combined with the exceptionally long history of active market intervention of the Japanese authorities, help us explain what is the main driving force for the notably inferior performance of our system when it comes

Table 11
Overall system performance in test period (Period #3); we do not employ leverage, no-trade, or stop-loss switches.

Method	Majority Voting (all)	GA	CLS	Variance-Based
Annualized return portfolio	4.6%	6.8%	4.8%	4.7%
Maximum drawdown	−10%	−7.6%	−3.7%	−3.5%
Annualized volatility	7.1%	5.9%	4.1%	4.2%
Sharpe ratio	65.6%	116.1%	116.8%	112.9%
Predictive accuracy	51.7%	52.1%	51.1%	50.9%
Annualized number of trades	106	105	102	101
Annualized cost	2.25%	2.23%	2.18%	2.15%
Net annualized return after costs	2.35%	**4.57%**	2.62%	2.55%

Table 12
Overall system performance in test period (Period #3); we do not employ the no-trade and stop-loss switches.

(a) Leverage Strategy #1.

Method	Majority Voting (all)	GA	CLS	Variance-Based
Annualized return portfolio	11.0%	15.9%	11.8%	11.7%
Maximum drawdown	−21.7%	−16.1%	−7.4%	−7.2%
Annualized Volatility	18.1%	16.0%	10.8%	11.1%
Information Ratio (Sharpe Ratio)	61.2%	99.1%	**109.0%**	105.6%
Average Leverage	1.83	1.83	1.83	1.83
Number of trades annualized	106	105	102	101
Cost annualized	4.1%	4.1%	4.0%	3.9%
Net annualized return after costs	6.88%	11.81%	7.81%	7.76%

(b) Leverage Strategy #2.

Method	Majority Voting (all)	GA	CLS	Variance-Based
Annualized return portfolio	17.4%	24.9%	18.8%	18.7%
Maximum drawdown	−28.3%	−21.1%	−10.1%	−13.1%
Annualized Volatility	29.3%	26.3%	17.7%	18.2%
Information Ratio (Sharpe Ratio)	59.5%	94.8%	106.1%	103.1%
Average Leverage	2.83	2.83	2.83	2.83
Number of trades annualized	106	105	102	101
Cost annualized	6.4%	6.3%	6.2%	6.1%
Net annualized return after costs	11.02%	18.58%	12.63%	12.60%

Table 13
Overall system performance in test period (Period #3); we do not employ the no-trade switch: (a) 1% stop-loss threshold; (b) 1.5% stop-loss threshold; (c) 2% stop-loss threshold. We consider application of Leverage Strategy #1, which is the one yielding the highest Sharpe ratio.

(a)

Method	Majority Voting (all)	GA	CLS	Variance-Based
Annualized return portfolio	11.13%	16.05%	11.81%	11.85%
Maximum drawdown	−21.35%	−15.74%	−6.99%	−6.95%
Annualized volatility	17.93%	15.86%	10.69%	11.03%
Sharpe ratio	58.89%	99.68%	105.81%	97.06%
Average leverage	1.83	1.83	1.83	1.83
Annualized number of trades	110	105	104	103
Annualized cost	4.24%	4.11%	4.07%	3.98%
Net annualized return after costs	6.89%	11.94%	7.74%	7.87%

(b)

Method	Majority Voting (all)	GA	CLS	Variance-Based
Annualized return portfolio	11.13%	16.10%	11.95%	11.70%
Maximum drawdown	−21.51%	−15.90%	−7.28%	−7.05%
Annualized volatility	18.04%	15.80%	10.70%	11.03%
Sharpe ratio	59.36%	100.67%	102.39%	103.40%
Average leverage	1.83	1.83	1.83	1.83
Annualized number of trades	108	107	103	103
Annualized cost	4.24%	4.11%	4.07%	3.98%
Net annualized return after costs	6.89%	11.99%	7.88%	7.72%

(c)

Method	Majority Voting (all)	GA	CLS	Variance-Based
Annualized return portfolio	11.11%	16.07%	11.75%	11.80%
Maximum drawdown	−21.56%	−16.04%	−7.32%	−7.13%
Annualized volatility	18.08%	15.90%	10.63%	10.98%
Sharpe ratio	60.89%	97.04%	104.08%	101.40%
Average leverage	1.83	1.83	1.83	1.83
Annualized number of trades	107	106	103	102
Annualized cost	4.19%	4.17%	4.03%	3.99%
Net annualized return after costs	6.92%	11.90%	7.72%	7.81%

to the USD/JPY rate, among all major currencies. Finally, it is notable that the performance of our system achieves its highest value in the case of the USD/CAD exchange rate, while its poorest performance occurs in the case of the USD/NZD. The latter is probably due to a structural break during Period #3, not captured in the training set, related to a breakthrough mixture of monetary policy adopted by the Reserve Bank of New Zealand.

Finally, another aspect of our system that is worth to examine concerns the potential of increasing its performance by adopting a margin-based trading scheme. Indeed, typical retail brokers do not need to invest 1 USD for each USD worth of the positions they open. On the contrary, margin requirement for typical retail brokers is around 10–15%. In the previous experimental results, we assumed that the necessary margin amount for trading during Period #3 equals the maximum drawdown amount. This implies the

Table 14
Overall performance of our (full) system, in test period (Period #3); we use Leverage Strategy #1.

Method	Majority Voting (all)	GA	CLS	Variance-Based
Annualized return portfolio	12.2%	16.4%	13.3%	13.4%
Maximum drawdown	−16.7%	−11.9%	−7.1%	−7.2%
Annualized volatility	16.6%	14.8%	10.0%	10.2%
Sharpe ratio	73.6%	110.8%	133.2%	130.4%
Average leverage	1.79	1.79	1.79	1.79
Annualized number of trades	101.00	99.00	97.00	98.00
Annualized cost	3.9%	4.0%	3.9%	3.9%
Net annualized return after costs	8.29%	12.50%	9.40%	9.48%

Table 15
Breakdown of the overall system performance (total return) in test period (Period #3); we use Leverage Strategy #1.

Currency	Majority Voting (all)	GA	CLS	Variance-Based
SEK	78%	85%	65%	51%
NOK	41%	72%	23%	31%
JPY	65%	67%	20%	17%
EUR	83%	82%	59%	59%
DKK	80%	82%	57%	49%
NZD	29%	60%	−1%	−25%
GBP	37%	66%	77%	68%
CAD	41%	110%	71%	97%
AUD	−9%	41%	96%	107%
CHF	51%	46%	60%	73%

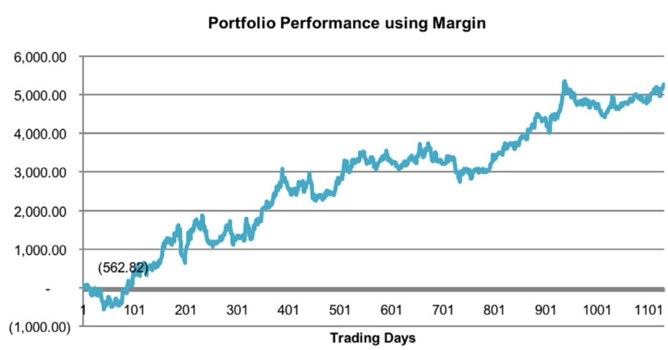

Fig. 8. System performance when trading is performed using margin.

assumption that a margin call will take place at the beginning of the trading period. Yet, this is not the actual case in our experimental results, since the maximum drawdown takes place after a significant positive P&L has been obtained. Besides, we observe that the actual margin requirement in Period #3 amounts to 560, while the overall P&L in Period #3 is almost $5000.

Thus, the total portfolio performance obtained by our system rises to around 900% during that period, when margin is used. These findings are graphically depicted in Fig. 8. This corresponds to an annualized return ranging in the interval 150%–200%, depending on the applicable transaction costs. These are strong empirical findings which corroborate our claims that our trading system can provide a very good return on investment. Indeed, our approach clearly outperforms the performance of the S&P 500 index in the same period, which is widely considered as a standard industrial benchmark for evaluating portfolio management strategies.

6. Conclusions

The thrust of this empirical study was focused on creating an automated trading system tailored to 10 currency pairs traded against the USD. We proposed a novel FX rate portfolio forecasting model, leveraging the attractive properties of popular machine learning algorithms. Our approach exploited well-established knowledge regarding the interconnections among currencies, as reflected in their pairwise correlations.

We performed an extensive experimental evaluation of our approach, using data from a 15-year period. The obtained out-of-sample performance results showed that our FX trader can manage to achieve a yearly return that may exceed 18%, depending on the leverage profile. Our analyses also provided interesting insights regarding the prevalence of structured patterns underlying the behavior of various currency exchange rates, and the potential of successfully inferring them to facilitate and inform the trading process. Thus, our system proves to have the potential of becoming a valuable tool for FX traders, that can be used to provide forecasts for the contemporary evolution of multiple FX rates on a daily basis.

An aspect this work has not considered concerns also examining a setup denominated in alternative major currencies, e.g. the EUR, and combining the predictive outcomes stemming from different denominations at a fourth level of analysis. Exploring the utility of an even more diverse set of machine learning algorithms, at the first system level, is also something that would be worth of investigation. The value of such novel developments remains to be examined in our future research endeavors.

References

Alfaro, E., García, N., Gamez, M., & Elizondo, D. (2008). Bankruptcy forecasting: An empirical comparison of AdaBoost and neural networks. *Decision Support Systems, 45*(1), 110–122.

Asteriou, D., & Hall, S. G. (2011). ARIMA models and the Box–Jenkins methodology. In *Applied econometrics* (pp. 265–286). Palgrave MacMillan.

Bahrepour, M., et al. (2011). An adaptive ordered fuzzy time series with application to FOREX. *Expert Systems with Applications, 38*(1), 475–485.

Bauwens, L., Omrane, W. B., & Giot, P. (2005). News announcements, market activity and volatility in the Euro/Dollar foreign exchange market. *Journal of International Money and Finance, 24*(7), 1108–1125.

Beine, M., Laurent, S., & Palm, F. C. (2009). Central bank FOREX interventions assessed using realized moments. *Journal of International Financial Markets, Institutions, and Money, 19*, 112–127.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. New York: Springer.

Breiman, L. (2001). Random forests. *Machine Learning, 45*, 5–32.

Byun, H. W., et al. (2015). Using a principal component analysis for multi-currencies-trading in the foreign exchange market. *Intelligent Data Analysis, 19*(3), 683–697.

Carreira-Perpinan, M., & Hinton, G. E. (2005). On contrastive divergence learning. In *proceedings of the 10th international conference on artificial intelligence and statistics (AISTATS 2005)* (pp. 59–66).

Casarin, R., Tronzano, M., & Sartore, D. (2013). A Bayesian stochastic correlation model for exchange rates. *Advances in latent variables-methods, models and applications*. Springer.

Chan, Y., Stock, J., & Watson, M. (1999). A dynamic factor model framework for forecast combination. *Spanish Economic Review, 1*(2), 91–121.

Chipman, H., George, E., & McCulloch, R. (2010). BART: Bayesian additive regression trees. *The Annals of Applied Statistics, 4*(1), 266–298.

Costantini, M., Cuaresma, J. C., & Hlouskova, J. (2016). Forecasting errors, directional accuracy and profitability of currency trading: The case of EUR/USD exchange rate. *Journal of Forecasting, 35*(7), 652–668.

Dunis, C., & Huang, X. (2002). Forecasting and trading currency volatility: An application of recurrent neural regression and model combination. *Journal of Forecasting, 21*(5), 317–354.

Dunis, C., & Shannon, G. (2005). Emerging markets of South-East and Central Asia: Do they still offer a diversification benefit? *Journal of Asset Management, 6*(3), 168–190.

Dymova, L., Sevastjanov, P., & Kaczmarek, K. (2016). A Forex trading expert system based on a new approach to the rule-base evidential reasoning. *Expert Systems with Applications, 51*, 1–13.

Égert, B., & Kocenda, E. (2014). The impact of macro news and central bank communication on emerging European Forex markets. *Economic Systems, 38*, 73–88.

Frömmel, M., MacDonald, R., & Menkhoff, L. (2005). Markov switching regimes in a monetary exchange rate model. *Economic Modelling, 22*(3), 485–502.

Harvey, A. (1990). *Forecasting, structural time series models and the Kalman filter*. Cambridge, U.K.: Cambridge University Press.

Ince, H., & Trafalis, T. B. (2006). A hybrid model for exchange rate prediction. *Decision Support Systems, 42*, 1054–1062.

Keskin, M., Deviren, B., & Kocakaplan, Y. (2011). Topology of the correlation networks among major currencies using hierarchical structure methods. *Physica A: Statistical Mechanics and its Applications, 390*(4), 719–730.

Kim, K. (2003). Financial time series forecasting using support vector machines. *Neurocomputing, 55*(1–2), 307–319.

Krauss, C., Do, X. A., & Hu, N. (2017). Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. *European Journal of Operational Research, 259*(2), 689–702.

Lam, M. (2004). Neural network techniques for financial performance prediction: Integrating fundamental and technical analysis. *Decision Support Systems, 37*, 567–581.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 512*, 436–444.

Leigh, W., Purvis, R., & Ragusa, J. (2002). Forecasting the NYSE composite index with technical analysis, pattern recognizer, neural network, and genetic algorithm: A case study in romantic decision support. *Decision Support Systems, 32*(4), 361–377.

Miśkiewicz, J. (2016). Cross-correlations of the Forex market using power law classification scheme picture. *Acta Physica Polonica A, 129*(5), 917–921.

Mizuno, T., Takayasu, H., & Takayasu, M. (2006). Correlation networks among currencies. *Physica A: Statistical Mechanics and its Applications, 364*, 336–342.

Ohnishi, T., Takayasu, H., Ito, T., Hashimoto, Y., Watanabe, T., & Takayasu, M. (2012). On the nonstationarity of the exchange rate process. *International Review of Financial Analysis, 23*, 30–34.

Ozturk, M., Toroslu, I. H., & Fidan, G. (2016). Heuristic based trading system on Forex data using technical indicator rules. *Applied Soft Computing, 43*, 170–186.

Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications, 42*, 259–268.

Plakandaras, V., Papadimitriou, T., & Gogas, P. (2015). Forecasting daily and monthly exchange rates with machine learning techniques. *Journal of Forecasting, 34*(7), 560–573.

Rapach, D., & Strauss, J. (2008). Forecasting US employment growth using forecast combining methods. *Journal of Forecasting, 27*(1), 75–93.

Sermpinis, G., Dunis, C., Laws, J., & Stasinakis, C. (2012). Forecasting and trading the EUR/USD exchange rate with stochastic neural network combination and time-varying leverage. *Decision Support Systems, 54*, 316–329.

Shen, F., Chao, J., & Zhao, J. (2015). Forecasting exchange rate using deep belief networks and conjugate gradient method. *Neurocomputing, 167*, 243–253.

Stock, J., & Watson, M. (2004). Combination forecasts of output growth in a seven–country data set. *Journal of Forecasting, 23*(6), 405–430.

Swanson, N., & Zeng, T. (2001). Choosing among competing econometric forecasts: regression-based forecast combination using model selection. *Journal of Forecasting, 20*(6), 425–440.

Swishchuk, A., Tertychnyi, M., & Elliott, R. (2014). Pricing currency derivatives with Markov-modulated Lévy dynamics. *Insurance: Mathematics and Economics, 57*, 67–76.

Takeuchi, L., & Lee, Y.-Y. A. (2013). Applying deep learning to enhance momentum trading strategies in stocks. *Technical Report*. Stanford University.

Vapnik, V. N. (1998). *Statistical learning theory*. New York: Wiley.