European Journal of Operational Research 278 (2019) 330–342

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Innovative Applications of O.R.

Large data sets and machine learning: Applications to statistical arbitrage

Nicolas Huck[1]

ICN Business School - CEREFIGE, 86 rue du Sergent Blandan, CS 70148, Nancy Cedex 54003, France

ARTICLE INFO

Article history:
Received 13 December 2017
Accepted 11 April 2019
Available online 16 April 2019

Keywords:
Finance
Big data
Machine learning
Statistical arbitrage

ABSTRACT

Machine learning algorithms and big data are transforming all industries including the finance and portfolio management sectors. While these techniques, such as Deep Belief Networks or Random Forests, are becoming more and more popular on the market, the academic literature is relatively sparse. Through a series of applications involving hundreds of variables/predictors and stocks, this article presents some of the state-of-the-art techniques and how they can be implemented to manage a long-short portfolio. Numerous practical and empirical issues are developed. One of the main questions beyond big data use is the value of information. Does an increase in the number of predictors improve the portfolio performance? Which features are the most important? A large number of predictors means, potentially, a high level of noise. How do the algorithms manage this? This article develops an application using a 22-year trading period, up to 300 U.S. large caps and around 600 predictors. The empirical results underline the ability of these techniques to generate useful trading signals for portfolios with important turnovers and short holding periods (one or five days). Positive excess returns are reported between 1993 and 2008. They are strongly reduced after accounting for transaction costs and traditional risk factors. When these machine learning tools were readily available in the market, excess returns turned into the negative in most recent times. Results also show that adding features is far from being a guarantee to boost the alpha of the portfolio.

1. Introduction

On the financial markets, "saying big data has arrived on Wall Street is an understatement at best" (Blake, 2014, p. 57). The data revolution is a new one after the emergence of mathematicians and physicists in the finance industry in the 80's. These large datasets bring new challenges (Baesens, Bapna, Marsden, Vanthienen, & Zhao, 2016; Chen, Chiang, & Storey, 2012; Fan, Han, & Liu, 2014; George, Haas, & Pentland, 2014; George, Osinga, Lavie, & Scott, 2016; Manyika et al., 2011) not only for practitioners and data scientists but also for academics in a wide variety of fields including applied and computational mathematics, medicine (genomics, neuroscience, etc), engineering, operations management, economics, marketing, and finance. Data are the new raw material of business and economics. George et al. (2014, p. 321) summarize this phenomenon as follows: "Whether it is machine learning and web analytics to predict individual actions, consumer choice, search behavior, traffic patterns, or disease outbreaks, big data is fast becoming a tool that not only analyses patterns, but can also provide the predictive likelihood of an event."

Big data is characterized by massive samples with high dimensionality: these aspects are essential and indicate that the analysis of these datasets require important computational and storage capabilities. As underlined in George et al. (2014), the "bigness" of the data is not all. The development of new statistical methods is among the main questions raised by big data. Fan et al. (2014) present some of the statistical issues data scientists face (heterogeneity, noise accumulation, spurious correlations, incidental endogeneity) and argue that these unique features make the use of some "traditional" statistical techniques inappropriate. The tools from the machine learning field are getting more and more essential and powerful. From a theoretical point of view, some algorithms are universal approximators (Cybenko, 1989; Hornik, Stinchcombe, & White, 1989). The development of the last decade regarding the training of multilayer neural networks and the large increase of the processing power (GPU programming) explain why this field is becoming so popular. The power of machine learning may be illustrated by the victory of AlphaGo, a Go-playing computer, against Lee Sedol, one of the best human players of all times (Silver et al., 2016; The Economist, 2016). The game "Go" has

E-mail addresses: nicolas.huck@icn-artem.com, nicolas.huck@icn-groupe.fr

[1] I wish to thank Matthew Hawkins, Hareesh Mavoori, Olivier Mesly and especially Christopher Krauss for their helpful comments and suggestions.

https://doi.org/10.1016/j.ejor.2019.04.013
0377-2217/

long been considered an Everest for artificial intelligence research. The performance of AlphaGo Zero, a new version of this program, solely based on reinforcement learning is even more impressive (Silver et al., 2017).

Building (profitable) automatic trading algorithms with big data is especially challenging (Dhar, 2015a; 2015b). Blake (2014), Kwan (2014) present, from a practitioner point of view, the impact of large datasets[2] on trading and technology. An important source of (big) data in finance comes from high frequency trading (Seddon & Currie, 2017). This is "now the norm, and it is not going away" (O'Hara, 2014, p. 18). Although this article uses daily data, the datasets are already quite impressive with several hundreds of thousands of observations and several hundreds of predictors. This frequency is in line with the empirical questionings/financial anomalies discussed in this article.

If the introduction of machine learning in finance/portfolio management is natural and not surprising as underlined in Scott (2014), there is still a gap between academic finance on the one hand and the financial industry on the other. In the business community, big data is a reality whereas in the finance academic literature "there is very little published management scholarship that tackles the challenges of using such tools - or, better yet, that explores the promise and opportunities for new theories and practices that big data might bring about" (George et al., 2014, p. 321). Hsu, Lessmann, Sung, and Ma (2016, p. 216) recall that machine learning methods "are rarely considered by financial economists who prefer econometric, often linear methods." As an example with the leading academic finance journal "The Journal of Finance", a search for "machine learning" via the Ebsco database, until 2017, produces no reference.[3] As regards the "Journal of Financial Economics", only one article has been found. 100 capital anomalies are reported in Jacobs (2015). Not a single one employs methods from statistical learning. The recent special issue of the Journal of Business and Economic Statistics (Bai, Fan, & Tsay, 2016) on big data also illustrates this shift: among the fifteen articles, six are on big data finance. Only one article (Taddy, Gardner, Chen, & Draper, 2016) considers machine learning algorithms but it does not deal with (market) finance.

The interest of the academic finance community for machine learning techniques is growing. This movement has been initiated by econometricians and statisticians. The recommendations of Heaton, Polson, and Witte (2016, p. 1) are the following: "Applying deep learning methods to these problems can produce more useful results than standard methods in finance. In particular, deep learning can detect and exploit interactions in the data that are, at least currently, invisible to any existing financial economic theory". Maasoumi and Medeiros (2010), Varian (2014), Mullainathan and Spiess (2017) share this point and expect numerous opportunities for productive collaborations between computer scientists and econometricians in the field of machine learning. Hsu et al. (2016) motivate their research by the need to develop a better understanding of the reasons for the disagreement between the Efficient Market Hypothesis (EMH) and the evidence reported in the machine learning literature and provide a survey of this stream of research. The Adaptive Markets Hypothesis of Lo (2004) and its evolutionary perspective of the financial markets therefore have strong connections with machine learning algorithms. Following this stream, our empirical results confirm a severe drop in performance of trading systems based on machine learning in the most recent years: when these tools became more accessible.

Statistical arbitrage is a natural application field for big data and machine learning. Lo (2010) recalls it involves a large number of securities and substantial computational, trading and information technology infrastructure. Combining these three dimensions (large datasets, machine learning and trading/statistical arbitrage/financial anomalies) is clearly the objective of this article. Although these points are largely discussed in the literature on an individual basis, there is a lack of empirical works dealing with datasets of several hundred variables, state-of-the-art machine learning methods and a discussion of the EMH: the results will present the main trends in terms of portfolio returns. Powerful and well-known algorithms such as Deep Belief Networks, Elastic Net regressions or Random Forests will be considered. Variants (optimization, combination, etc) around these techniques are natural extensions of this work and are only briefly introduced as the idea here is to focus on the financial and empirical contributions. Using a large dataset with various features (including the possibility to test several market anomalies/risk factors, for example: day or month of the week effect, Fama and French factors, etc), this article will explore and assess several machine learning methods and different forecasting horizons from a trading perspective.

The remainder of this article is organized as follows. Section 2 briefly reviews some articles of the relevant literature and specifies the positioning of the research. The different financial data used are presented in Section 3. Section 4 covers the design of the applications: a description of the trading system and the presentation of the statistical/machine learning algorithms considered in this article. Empirical results are presented in Section 5. Section 6 concludes and provides directions for future research.

2. Literature review

The literature dealing with machine learning/neural networks in finance is, in particular, presented and analysed in Wong and Selvi (1998), Atsalakis and Valavanis (2009), Bahrammirzaee (2010), Li, Jiang, Yang, and Wu (2018). Methods and application fields are numerous. As underlined previously, most articles are published in journals from the machine learning/operational research/information system community. The top three finance academic journals (Journal of Finance, Journal of Financial Economics, Review of Financial Studies) are clearly under-represented with only a single citation. This section introduces to a limited number of recent articles which are highly connected to this article.

Huck (2009) presents a prequel to later deep learning applications in finance. In his early algorithm, he deploys Elman recurrent neural networks to perform weekly predictions for all constituents of the S&P 100 universe. A long-short portfolio consisting of the stocks with the highest confidence yields excess returns of 0.8% per week at a 54% directional accuracy. Huck (2010) flexibilizes this approach and presents multi-step ahead forecasts. Takeuchi and Lee (2013) develop a deep learning trading system for all CRSP stocks from 1965 to 2009. They mainly feed cumulative returns as features into a Deep Belief Network, and train it on all stocks from 1965 to 1989. From 1990 to 2009, they use this system to compute the probability that a stock outperforms the cross-sectional median during the next month after the predict date. Going long the top decile and short the flop decile of predictions yields annualized returns of 45.93%. Moritz and Zimmermann (2014) apply Random Forests to the U.S. stock universe from 1968 to 2012. They use return-based features as well as 86 other firm-based characteristics, and achieve risk-adjusted excess returns of 2.28% per month. Heaton et al. (2016) do not develop a specific trading application, but discuss the potential benefits of deep learning in finance in general. Krauss, Do, and Huck (2017) deployed Deep Belief Networks, gradient-boosting, and Random Forests to a survivor-bias free sample of S&P 500 constituents, ranging from 1992 to

[2] There is no official definition of Big Data so that the definition of what is (really) big is relative and subjective.

[3] A research with "neural networks" gives less than twenty articles but an even more limited number of papers really uses these tools in the empirical applications despite the fact they are implemented in statistical packages for many years.

2015. Using return-based features, they find that an ensemble of the above-mentioned methods yields returns of 0.45% per day (before transaction costs). Highest explanatory power stems from the most recent returns prior to portfolio formation. Fischer and Krauss (2018) expand on this work and tailor a long short-term memory network to the same prediction task. Using input sequences of raw returns ranging over 240 days, they find mean returns of 0.46% per day (before transaction costs). Thus, a deep learning model tailored to sequence learning tasks has the potential to outperform a state-of-the-art ensemble model. The authors find that the most important predictors are the returns corresponding to the past five days prior to portfolio formation - confirming the results in Krauss et al. (2017). LTSM models are also considered in Bao, Yue, and Rao (2017), Troiano, Villa, and Loia (2018).

Besides these studies, there are further applications on smaller security universes and other asset classes that employ machine learning methods for capital market predictions. Spreckelsen, Mettenheim, and Breitner (2014) perform real-time pricing and hedging of currency future options, leveraging artificial neural networks. Hsu et al. (2016) show the advantages of machine learning compared to standard econometric methods on 34 financial indices. Chong, Han, and Park (2017) provide a literature review and an application considering 38 stocks from the KOSPI market in an intraday framework. Zhao, Li, and Yu (2017) deploy a deep learning ensemble in order to predict the price of crude oil. Deng, Bao, Kong, Ren, and Dai (2017) introduce a recurrent deep neural network (NN) for real-time financial signal representation and trading.

3. Data, training and trading sets

This section presents the data used in this article and introduces how they are aggregated to form the inputs of the different models. As mentioned previously, one of the goals of this research is to assess the impact of a large number of variables in a statistical arbitrage/long-short portfolio context. In this article, as a particular case, dollar-neutral portfolios are built: dollar amounts of both long and short positions are equal. Furthermore, it is also true at the stock level: each position, long or short, may be normalized to one dollar. The portfolio construction is thus rather simple in this article and more advanced optimization techniques could be used at this stage. DeMiguel, Garlappi, and Uppal (2009) recall the naive $1/N$ portfolio strategy, out-of-sample, is not inefficient. Long-short investments seek to take advantage of inefficient pricing and to minimize market exposure, while profiting from profits in the long positions (finding winners), along with price declines in the short positions (finding losers).

This study covers the American stock market and with a focus on the large capitalization segment. In a first time, via Datastream, the list of all stocks which have been, at least once, part of the S&P 900 Index from 1990 until 2015 are collected: there are about 1900. Then, using Bloomberg and Datastream, the daily stock prices, market capitalizations, trading volumes and the ICB sector classification at the industry level are obtained. The trading system entails two steps: "estimation"/learning of the relationships between the assets and the predictors and then trading based on these forecasts. The management of the portfolios includes the period between January 1993 and June 2015. The pool of data considered for the training sets, updated twice a year (45 batches were built), is formed by stocks free of missing values (adjusted price, market capitalization, volume and sector) during the learning/selection period. Among those stocks, the top 100 or 300 in terms of market capitalization are kept. This choice is motivated by computational feasibility, market efficiency and liquidity. These two ad-hoc values (100 and 300) determine two sizes of information sets/trading environment which approximately represent two popular indexes: the S&P 100 and the MSCI US Large Cap 300. In the US equity market, the first 300 stocks in terms of market capitalization represent approximately 70% of the free float-adjusted market capitalization.

The inputs are divided into four groups/types: Lagged Returns for each stock (i, LR), dummies to IDentify precisely each stock (ii, ID), Time, Industry and Risk (iii, TIR), EXternalities (iv, Ex).

(i) Lagged stock returns (LR):
Lagged returns are the main data in financial time series analysis. Following, for example, Takeuchi and Lee (2013) or Krauss et al. (2017), who develop applications of machine learning in finance, the resolution is higher for the most recent returns. The first five daily returns are computed, then the resolution is reduced. Starting for example at time $t = 0$, the returns are obtained using the following 11 lags/dates: 1, 2, 3, 4, 5, 10, 21, 42, 63, 126 and 252. 21 and 252 (21×12) approximate a trading month and a year, respectively. As an example, the return defined with lags 126 and 252 is the second last semester return.[4] That way, the model knows with a high degree of precision the behavior of the last days and the main trends during farther periods. This sequence of lags is kept for all the additional financial time series used in this article.

(ii) Dummy variables for stock identification (ID):
The economic motivation is quite simple: can the learning algorithm capture stock specific behaviors? Does-it help the trading system in selecting the "best" assets leading to an improvement of the return of the portfolio? Machine learning and dimensionality reduction techniques are interesting to (try to) incorporate, in a single and large model, stock-specific behaviors. In a traditional time series framework, introducing a dummy for each stock would appear confusing. From a technical and statistical point of view, it opens several issues (noise, multicollinearity, dummy variable trap, very sparse predictors, estimation, increase of computation time, etc). Current research problems in medicine/biology (DNA sequences, etc) sometimes have a number of predictors (p) that greatly exceeds the size of the sample (n). The question is thus to find or combine a good subset of variables. At this stage of the presentation of the data, the number of predictors is already quite large and could thus be equal to 110 (11 + (100-1)) or 310 (11 + (300-1)) if each stock is precisely identified via a binary variable.[5] The applications developed in this article are far from being a $p > > n$ study: the ratio n by p will always be greater than 100.

(iii) Time, Industry and Risk (TIR):
Several additional variables are introduced. Dummy variables are first presented. Day of the week (Gibbons & Hess, 1981) and month of the year effect/January effects (Ariel, 1990) are considered via 4 (5-1) and 11 (12-1) dummies, respectively. The stock sector specific behavior (Hong, Torous, & Valkanov, 2007) is taken into account via the ICB classification at the industry level (10 sectors, 9 binary variables). The VIX index (Whaley, 2000) is a well-known investor fear gauge. The allocation of the assets may be VIX dependent and influence the performance (Copeland & Copeland, 1999; Fernandes, Medeiros, & Scharth, 2014; Fleming, Kirby, & Ostdiek, 2001). Five classes (four binary variables) of implied volatility, going from a quiet to a high turmoil market, are built: below 15, between 15 and 20, between 20 and 25,

[4] If P_t^s is the price of a given stock s at time t, this second last semester return is computed as: $R^s_{t,126,252} = \frac{P^s_{t-126}}{P^s_{t-252}} - 1$.

[5] A binary variable is introduced for all but one of the stocks to avoid perfect multicollinearity. From a general point of view, the algorithms used in this article deal with this issue very efficiently.

between 25 and 30 and greater than 30. These splits are linked to the regime switches studied in Baba and Sakurai (2011). The dynamic of the VIX (fear) is also part of the dataset via lagged returns in the fourth group of predictors. That way, two information are given: the level (classes/dummies) and the direction (return). The idea is to be able to capture the notion of extreme correlation. It is known correlation between assets increases in bear markets (crisis periods), but not in bull markets (Longin & Solnik, 2001).

An issue regarding the presentation of the qualitative/categorical variables in the applications is now addressed. All qualitative/categorical variables are transformed into binary variables. This binarization leads to a tremendous number of additional predictors when the number of stocks is high, but this specification of the data fits with all the methods that will be introduced in the next section.

(iv) Externalities (EX):

The predictor list also contains two important commodities: gold[6] and oil (WTI).[7] The impact of these commodities on stock returns is, among others, discussed in Jones and Kaul (1996), Kilian and Park (2009), Black, Klinkowska, McMillan, and McMillan (2014), Caliskan and Najand (2016). For these time series, returns are computed following exactly the rules mentioned above. Some other traditional financial time series (returns) are also included in the data set as potential predictors. The Fama and French (1993) factors (SMB, HML, market premium, 1-month T-bill) are generally used as explanatory factors of a trading strategy. Panopoulou and Plastira (2014) evaluate whether the Fama-French factors influence the predictability of US stocks. This approach is followed and extended to the momentum and short and long-term reversal factors (Carhart, 1997; Jegadeesh, 1990). Sector returns are highly scrutinized by investors. Hence, stocks are divided into 12 industry portfolios.[8] Interest rates are reflected in the one-month T-bill rate (short-term) and the 10-year treasury constant maturity rates (long-term).[9] Discussions on the links between stock returns and interest rates include Laopodis (2013), Huang, Mollick, and Nguyen (2016) With these time series of return, cumulative indexes are first built then lagged returns are computed as defined above.

If all predictors are included, with 300 stocks, the training sets can thus contain up to 592 predictors.[10] With several hundreds of variables/predictors, noise, overfitting and data snooping become important issues. All machine learning algorithms used in this article (see next section) have the ability to avoid/narrow these problems.

The structures (dimensions) of the training and trading sets are now detailed. Different groupings of the variables are proposed. They offer a possibility of evaluating predictor importance:

- The smallest set only incorporates the stock lagged returns: 11 predictors (only group (i)).
- The second set combines stock lagged returns and stock ID (groups (i) and (ii)). The number of predictors is thus 110 (100 stocks: $11+99$) or 310 (300 stocks: $11+299$).
- The third set considers stock lagged returns and the additional variables (groups (i), (iii) and (iv)). The number of predictors is thus 293 ($11+282$) whatever the number of stocks (100 or 300).
- The fourth and last set aggregates all variables (groups (i),(ii), (iii) and (iv)). The number of predictors is thus 392 (100 stocks: $11+99+282$ predictors) or 592 (300 stocks: $11+299+282$ predictors).

The training period lasts two years (504 trading days: $21\times12\times2$). It means the number of samples in the training set equals 50400 with 100 stocks or 151200 with 300 stocks. A trading period lasts 6 months ($6\times21=126$ days). This split over the entire data is repeated 45 times at a semester frequency without the trading periods being overlapping. The stocks in each of these batches are time-varying, depending on data availability and market capitalization. An observation in the input refers to a stock a particular day. The output, constructed for each stock s and each day is a binary variable $Y^s_{t+h}\in\{0,1\}$. h is the estimation/forecast/trading horizon. Two values of h will be assessed: 1 (one day) and 5 (one week). The response Y^s_{t+h} equals one if the h period/day return of stock s is larger than the mean return (equal weighting) computed over all stocks and zero otherwise. This article follows a classification instead of a regression problem, which is linked to the directional forecasting literature (Enke & Thawornwong, 2005; Leung, Daouk, & Chen, 2000). Considering the intriguing decomposition of Christoffersen and Diebold (2006), our focus is on the ability to forecast directions. In a certain way, it is less ambitious than forecasting a return which is the product of a direction and its amplitude. Selecting stocks in a regression framework would lead to choose stocks with a high level of volatility (the stocks with the highest predictions in absolute terms). The classification approach favors stocks which are most likely to be above or below the market. These stocks are less volatile than the ones we could select with the regression approach.

As a consequence, a forecast will be the anticipated probability of a stock to perform better than the market. The trading rule is derived as follows. The stocks selected in the long-short portfolio are those with the highest and lowest probabilities to beat the market, i.e., the top and flop 10 stocks. These selections/rankings are repeated every trading day.

4. Machine learning algorithms

4.1. Basics

Among the various existing statistical methods available for predictive modeling to manage big data,[11] three methods, which are very different in terms of "philosophy", are considered. Deep Belief Networks (DBN) represent a modern form of neural networks able to manage large datasets and the optimization/learning process through several layers (theoretically and in practice). Random Forests come from the field of decision tree learning. With this method, a large number of independent decision trees is considered and the procedure then aggregates the information generated by each tree. The Elastic Net regression is an "advanced" regression specifically designed for problems with a large number of predictors. They are all able to perform classification tasks as demanded by the trading system. Multicollinearity, noise accumulation and overfitting are important issues in machine learning. Random subsampling (Random Forest) and regularization (DBN and Elastic Net regression) provide solutions to these problems.

This article operates an historical performance study. Every 6 months, 45 times in total, each of these algorithms is launched.

[6] Bloomberg Ticker XAUUSD (gold spot).

[7] Data downloaded from the US Energy Information Administration website.

[8] Data from the first two points of this enumeration are extracted from the website of Kenneth French http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html.

[9] Data extracted from the website of Federal Reserve: https://www.federalreserve.gov/releases/h15/.

[10] 592 = 11 (stock lagged returns) + 299 (number of stocks) + 4 (days) + 11 (months) + 4 (VIX) + 9 (sectors) + 11*23(11 lags/returns for 23 time series).

[11] The programming language/software used all over this article is Matlab.

One run is performed for each parameter setting, i.e., the strategies/algorithms are trained and tested for one or all groupings of variables mentioned in Section 2, and for two forecast horizons (one or five days). Results provided in this article have required several hundreds of hours of computation with a recent four-core computer.[12]

The key aim of this large scale backtesting study is not to find an optimal parameter constellation for each of the methods employed or to use the most impressive set of techniques. Ad-hoc choices are done and rules of thumbs and heuristics are used to calibrate/design each of the machine learning algorithms, so that we can evaluate the general impact (main empirical trends), in terms of return/performance, in the trading system of different features and forecast horizons. This strategy is in line with the (empirical) motivations of this article. It is also thrifty from a computational point of view and it is furthermore a way to deal with an issue like data snooping.

Numerous alternatives/variants are of course feasible regarding the methods, the optimization, the hyper-optimization of the parameters. Ensemble learning, aggregation of probability forecasts (Baron, Mellers, Tetlock, Stone, & Ungar, 2014; Dietterich, 2000; Lichtendahl, Yael, Victor Richmond R., & Robert L., 2018; Satopaa, Pemantle, & Ungar, 2016) are also interesting and natural extensions of this article.

The three techniques presented below are now popular in the academic literature and are briefly introduced.

4.2. Deep beliefs networks (DBN): autoencoders, supervised and unsupervised learning

The term Neural Networks is now used for several decades. It refers to computing systems inspired by biological neural networks. In the last years, huge progress have been made to improve the training/optimization of multilayer neural networks. They may fall under the umbrella of what is called Deep Learning. Deep Beliefs Networks are a neural network architecture introduced by Hinton and Salakhutdinov (2006) and Hinton, Osindero, and Teh (2006). Bengio (2009); Hinton (2009) provide a clear description of this framework. This approach combines different elements and steps. The main idea is the stacking of layers firstly trained in an unsupervised fashion followed by a supervised learning phase/fine tuning with backpropagation of the entire architecture. Searching the parameter space of deep architectures is a difficult task. Deep Belief Networks tackle some of these problems with a layer-by-layer procedure and have been used successfully, in various areas, including in image recognition (Hinton et al., 2006).

DBN are based on a composition of simple learning modules/neural networks. Each of these modules can be a restricted Boltzmann machine or an autoencoder as first introduced in Rumelhart, Hinton, and Williams (1986). The aim of an autoencoder is to transform inputs into outputs with the least possible amount of distortion, to learn a representation for a set of data, typically for the purpose of dimensionality reduction (Baldi, 2012). DBN address the problem of "backpropagation without a teacher" setting the target values to be equal to the inputs. They have taken center stage in "deep architecture".

The main specifications of the Deep Belief Networks considered in our computations and applications are now presented. The network starts with two layers of autoencoders and ends with a softmax function which is a common choice for a neural network-based classifier. We opt for the following network architectures, representing a trade-off between the number of observations and features, and the maximum run-time. When the predictors are only the stock lagged returns (11 variables), the first layer has 11 hidden units and the second one 6 (50% of the number of variables rounded upward). In that case, the network is not very large and far from being deep. When the dataset is larger via the inclusion of the dummies identifying stocks and/or the additional variables, the number of hidden units in the first layer equals 25% of the number of features. The second layer contains 50% of the number of units of the first hidden layer. With 300 stocks and all features (592), it means the sizes of the hidden layers are 148 and 74, respectively. For the training of the different layers, the maximum number of iterations is set to 200.

Overfitting is a major problem with Neural Networks. It can be combated with different techniques. For example, regularization modifies the objective function to be minimized by adding terms that penalize large weights. Two common types of regularization are the L1 and L2 regularizations. The first one penalizes the objective function via the absolute magnitude of all parameters and leads the weight vectors to become sparse. The second one operates via the squared magnitude of all parameters. It encourages the networks to use all of these inputs. The combination of the two is called Elastic Net regularization. No optimization of hyperparameters has been performed.[13]

4.3. Random Forest

Random Forests (RF) were introduced by Breiman (2001). They are very powerful tools for regression and classification problems. This algorithm has become highly popular due to its simplicity and has been used in many fields (Grushka-Cockayne, Jose, & Lichtendahl, 2017; Verikas, Gelzinis, & Bacauskiene, 2011; Ziegler & Knig, 2014). In the context of big data, Random Forests are discussed in Genuer, Poggi, Tuleau-Malot, and Villa-Vialaneix (2017). The consistency of Random Forests has been proved by Scornet, Biau, and Vert (2015).

The algorithm to grow a Random Forest is relatively simple. For each tree[14] in the Random Forest, a random subset of the original training data is drawn. Regarding the feature subsampling, the number of variables to select at random for each decision split is set to the square root of the total number of variables. This rule follows the recommendation of James, Witten, Hastie, and Tibshirani (2014).

In order to control the depth of the trees and the computational time, the minimum number of observations per tree leaf must be greater than 0.05% of the total number of observations. The number of trees in the forest is a tradeoff between computational costs and the marginal improvement in learning after each additional tree. The final output is an ensemble of 500 random forest trees, so that classification can be performed via majority vote. This algorithm, based on bootstrapping and aggregation, hence combines many decision trees, so that overfitting is reduced and generalization improved.

As underlined in Section 3, all nominal variables have been binarized (e.g.: with 100 stocks, 99 binary variables are created). With Random Forests and decision trees, another solution exists with a different data type: a categorical predictor. A unique predictor with 100 categories could be created. The question of partitioning nominal attributes in decision trees is addressed in Coppersmith, Hong, and Hosking (1999). This approach is not

[12] However, considering only the forecasting part of the system allows for the potential deployment in a real-time setting in professional finance.

[13] The computations are performed with the Matlab function trainAutoencoder. The hyperparameters are called SparsityProportion, SparsityRegularization and L2WeightRegularization. Values are set to 0.1, 4 and 0.0001. These ad hoc choices are in line with the default parameters or the values reported in the tutorials of the software.

[14] The standard CART (Classification And Regression Tree) algorithm is considered.

followed in this article because DBN and Elastic Net Regression do not support this type of use of the variables.

4.4. Dimensionality reduction: a two-stage methodology with Random Forest Forest and DBN

Random forests are also tools to extract the most important features in a dataset. One of the possible methodologies to perform this task is based on the (increase/modification in) prediction error.[15] For any variable, the values of that variable are permuted across the out-of-bag observations. This is done for every tree, then the prediction error is averaged and normalized. This approach, which implies a huge number of permutations, significantly increases the computation time. This task will be performed with only 200 trees. The variable importance rankings will be used in two different ways. First of all, they are relevant by themselves and will be discussed in Tables 3 and 4.

Second, these ranking will be the basis of a two-step methodology combining random forests and DBN consisting in running the DBN with a smaller dataset than before. At the starting point, the full datasets with respectively 392 (100 stocks) and 592 (300 stocks) features are considered. Only the variables belonging to the first quartile (98 and 148 variables for the 100 and 300 stocks cases) in terms of importance will be part of the inputs of the forecasting model (DBN). One of the questions underlying this strategy deals with the ability of DBN to perform well with very large datasets in the presence of noise or irrelevant features. The design of the DBN, i.e., the number of hidden units in the hidden layers as a function of the number of features, follows the rules mentioned in 4.2 with one exception: the number of hidden units in the first layer equals 50% (instead of 25%) of the number of features.

4.5. Elastic Net

The Elastic Net regression of Zou and Hastie (2005) is a hybrid of two types of regression: the Ridge regression of Hoerl and Kennard (1970) and the Least Absolute Shrinkage and Selection Operator (LASSO) of Tibshirani (1996). It overcomes some of the limitations of these two former methods. These three techniques fall into the literature dealing with a large number of predictors and dimensionality reduction algorithms. Applications in finance using this kind of regressions include Jiaqi and Tindall (2014), Panopoulou and Vrontos (2015).

The LASSO and the Ridge regressions consider L1 and L2 penalty terms when minimizing the sum of squared residuals. On the one hand, the Ridge regression promotes diversity via the L2 penalty as coefficients cannot be zero. The solution is stable but it is not very robust and no variable selection is performed. On the other hand, the LASSO allows continuous shrinkage to zero. The solution is much more parsimonious but it is unstable. From a group of correlated predictors, the method only "arbitrarily" selects one of the predictors.

These regressions are especially relevant when the number of features is important. In the Elastic Net regression,[16] shrinkage is controlled by a parameter λ. An increase of this positive parameter causes the estimated parameters to shrink to zero,while a λ close to zero causes the coefficients to converge to their Ordinary Least Squares counterparts. In the optimization process, the robustness is improved by a cross-validation strategy with 10 folds.[17]

Selecting a subset of variables in a large dataset and then performing forecasts are among the goals of this method. In this article, Elastic Net regressions are only applied when all features are used (392 and 592 variables with 100 and 300 stocks respectively)

5. Results

5.1. First results

The key results are presented in Tables 1 and 2 with Table 1 providing basic statistics and Table 2 providing in-depth analyses about exposure to common sources of systematic risk and performance in different sub-periods. The daily frequency is mainly considered in these tables. Due to the large number of models/specifications (40), the number of performance indicators is limited to the most relevant ones in these overview tables, in order to keep the focus on the key aspects. Starting with Table 1, the most salient findings are as follows:

- All models generate a positive (excess) return before transaction costs between 1993 and 2015. Annualized returns are between 7% and 92% with an average of about 35%, which is well above the performance of the market during that period (about 10%). When including transaction costs, the average annualized return of the 40 models is by 11% (only 6 have a negative return). These strategies, due to a frequent re-balancing (one- or five-day holding period), suffer from very important transaction costs.[18] If these figures are high, they remain credible. The large scale study (about 13000 funds) of Sadka (2010) indicates a 75th percentile of the investment style "equity long-short" at 42% per year[19] (after transaction costs and management and performance fees). In Table 1, the 75th percentile of the 40 models equals 49% and 18% without and with transaction costs.
- The best performance (before transaction costs) is achieved by a random forest only learning with the lagged returns (LR) on a set of 300 stocks. One of the closest works in the literature is that of Krauss et al. (2017). This article, also using Random Forests and focusing on a one-day holding strategy, reports performances greater than our best case. Numerous factors may explain these differences: the number of stocks (500 vs. 300), the duration/length of the learning period, the depth of the trees, the structure of the lagged returns, the decision tree algorithm, etc. Even though returns are different and sensitive to some parameters, they both share the same trends and show these methods were able to provide useful forecasts during the period under study.
- It is also important to recall that most techniques used in our research were not existing and/or implementable during a large part of the trading period. Furthermore, there is no (hyper-)optimization strategy. This, partly, protects again overfitting and data-snooping issues.
- Random Forests appear to be the most effective forecasting technique (which may be different from being the technique leading to the highest global returns including transaction costs) used in our study. Excluding transaction costs, all returns are significantly positive at the 1% level (t-stat with Newey-West one lag correction). These long-short portfolios present daily standard deviations that are about 70% greater than that of the market. On average, portfolios build with the pool of (only) 100 stocks (top-100 stocks in terms of market capitalization) have lower standard-deviations.

[15] From a computational/practical point of view, the Matlab option of the TreeBagger function called OOBPermutedVarDeltaError has been used.

[16] These regressions have been computed with the Matlab function lassoglm. The parameter α dealing with the weight of LASSO versus Ridge is set 0.5. $\alpha = 1$ represents a LASSO regression and an α close to 0 approaches Ridge regression.

[17] The selected model then follows the one standard error rule mentioned in Hastie, Tibshirani, and Friedman (2001, p. 61).

[18] Details about transaction costs are given in the following of this section.

[19] The last part of the trading period, say after 2010, is clearly the worst (see Table 2) one in terms of performance. The comparison with the returns/percentiles reported in Sadka (2010) is against the 40 models developed in these article.

Table 1
Risk-returns characteristics of the portfolios over the period between January 1993 and June 2015 (Part 1) : summary statistics. Each model is a combination between one or two forecasting methods (Deep Belief Network, Elastic Net regression, Random Forest, Elastic Net regression + DBN, a specific type of information set, a forecasting/holding period and a number of stocks in the allowed universe. The best/top three returns/performances are highlighted using bold.

Method	Data and Horizon ID, TIR + EX, H	Nb of Stocks	Before Transaction Costs										After T.C.			Q1	Q2	Q3
			Yearly Ret	Mean	Max	Min	Std	t-stat	Skew.	Kurt.	VAR (1%)	Sharpe Ratio	Max DD	Yearly Ret	Daily Ret	Sharpe Ratio			
DBN	No, No, H=1	100	55,86	0,18	15,08	−12,68	1,75	8,25	0,79	14,01	−4,80	9,69	39	27,60	0,10	**5,12**	−0,56	0,14	0,88
DBN	No, No, H=5	100	31,31	0,11	12,07	−11,88	1,57	5,77	0,39	13,84	−4,53	6,34	40	18,81	0,07	3,80	−0,49	0,09	0,70
DBN	Yes, No, H=1	100	35,13	0,12	21,69	−12,17	1,73	5,81	1,35	21,72	−4,77	6,34	41	10,63	0,04	1,71	−0,56	0,10	0,79
DBN	Yes, No, H=5	100	14,83	0,05	19,75	−10,39	1,62	3,14	1,22	21,08	−4,56	2,44	64	3,90	0,01	−0,02	−0,59	0,05	0,68
DBN	No, Yes, H=1	100	24,92	0,09	12,73	−17,29	2,00	4,06	−0,39	12,96	−5,85	3,98	57	2,27	0,01	−0,02	−0,67	0,09	0,85
DBN	No, Yes, H=5	100	14,85	0,05	12,91	−15,76	1,84	2,91	−0,38	13,24	−5,55	2,15	70	3,92	0,01	−0,02	−0,64	0,07	0,76
DBN	Yes, Yes, H=1	100	22,07	0,08	16,93	−17,76	1,82	3,95	−0,32	15,81	−5,98	3,82	61	−0,07	0,00	−0,57	−0,53	0,11	0,76
DBN	Yes, Yes, H=5	100	8,90	0,03	18,99	−16,02	1,76	2,07	−0,23	18,43	−5,07	1,11	67	−1,47	−0,01	−1,16	−0,61	0,05	0,73
DBN	No, No, H=1	300	**65,17**	**0,20**	19,41	−17,70	2,33	7,28	0,17	12,52	−6,32	8,14	69	**35,22**	**0,12**	4,70	−0,75	0,20	1,17
DBN	No, No, H=5	300	38,84	0,13	44,15	−14,47	1,95	5,72	2,31	59,54	−5,00	6,13	55	25,63	0,09	4,08	−0,69	0,13	0,92
DBN	Yes, No, H=1	300	49,22	0,16	25,20	−11,90	2,15	6,34	1,13	15,38	−5,88	6,96	48	22,16	0,08	3,24	−0,77	0,12	1,05
DBN	Yes, No, H=5	300	19,21	0,07	50,37	−15,13	2,10	3,20	3,41	79,21	−5,37	2,84	52	7,86	0,03	0,93	−0,77	0,06	0,87
DBN	No, Yes, H=1	300	34,11	0,12	15,37	−22,60	2,52	4,37	−0,37	12,00	−7,62	4,35	73	9,79	0,04	1,17	−0,82	0,16	1,12
DBN	No, Yes, H=5	300	17,15	0,06	34,19	−16,55	2,26	2,89	0,33	20,96	−6,79	2,19	72	6,00	0,02	0,42	−0,80	0,11	1,00
DBN	Yes, Yes, H=1	300	26,76	0,09	15,84	−22,21	2,28	3,96	−0,70	16,05	−6,85	3,49	61	3,77	0,01	−0,02	−0,74	0,13	0,99
DBN	Yes, Yes, H=5	300	7,43	0,03	32,13	−16,05	2,08	1,77	0,37	22,06	−6,42	0,94	80	−2,80	−0,01	−0,98	−0,75	0,07	0,91
El.Net	Yes, Yes, H=1	100	44,80	0,15	14,46	−14,86	1,84	6,69	0,25	13,31	−5,38	7,59	58	7,25	0,03	1,07	−0,57	0,10	0,85
El.Net	Yes, Yes, H=5	100	21,61	0,08	17,57	−12,37	1,66	4,13	0,30	16,63	−4,96	4,19	57	10,04	0,04	1,78	−0,55	0,08	0,72
El.Net	Yes, Yes, H=1	300	63,44	0,20	21,37	−20,64	2,33	7,17	0,26	15,03	−6,58	8,14	60	21,06	0,08	2,99	−0,76	0,19	1,15
El.Net	Yes, Yes, H=5	300	21,09	0,08	38,65	−15,18	2,05	3,51	1,37	35,36	−5,56	3,40	70	9,56	0,04	1,44	−0,73	0,07	0,93
RF	No, No, H=1	100	52,51	0,17	14,42	−11,20	1,28	10,32	0,82	16,98	−3,43	**12,47**	33	2,20	0,01	−0,03	−0,38	0,12	0,66
RF	No, No, H=5	100	29,06	0,10	13,73	−7,63	1,12	7,20	0,75	21,69	−3,48	8,00	40	16,78	0,06	4,43	−0,29	0,08	0,47
RF	Yes, No, H=1	100	53,09	0,17	15,66	−14,92	1,46	9,27	0,20	16,53	−4,43	10,93	33	2,59	0,01	−0,03	−0,40	0,13	0,75
RF	Yes, No, H=5	100	30,94	0,11	15,42	−10,92	1,39	6,29	0,29	18,57	−4,51	7,17	46	18,48	0,07	4,29	−0,37	0,10	0,60
RF	No, Yes, H=1	100	41,19	0,14	11,84	−16,17	1,81	6,37	−0,76	16,72	−5,58	7,16	50	−5,39	−0,02	−1,68	−0,50	0,11	0,82
RF	No, Yes, H=5	100	19,52	0,07	12,62	−17,42	1,78	3,63	−0,67	17,83	−5,79	3,35	67	8,14	0,03	1,10	−0,47	0,08	0,67
RF	Yes, Yes, H=1	100	33,27	0,11	16,47	−17,28	1,83	5,36	−0,72	17,88	−5,68	5,44	63	−10,70	−0,05	−3,30	−0,50	0,11	0,80
RF	Yes, Yes, H=5	100	21,02	0,08	16,25	−20,94	2,14	3,43	−0,76	15,21	−6,52	3,25	62	9,50	0,04	1,38	−0,65	0,09	0,87
RF	No, No, H=1	300	**92,51**	**0,26**	21,88	−15,52	1,65	12,51	0,76	16,56	−4,24	**15,13**	32	**29,00**	**0,10**	**5,43**	−0,44	0,19	0,94
RF	No, No, H=5	300	43,49	0,14	15,74	−12,04	1,36	8,41	0,55	18,57	−4,16	9,53	48	**29,83**	**0,10**	**6,59**	−0,37	0,12	0,65
RF	Yes, No, H=1	300	**85,79**	**0,25**	19,00	−15,01	1,91	10,42	0,47	12,98	−5,33	**12,54**	51	24,50	0,09	4,17	−0,58	0,20	1,08
RF	Yes, No, H=5	300	30,92	0,11	15,75	−11,85	1,74	5,26	0,33	13,89	−5,11	5,72	57	18,46	0,07	3,43	−0,58	0,10	0,82
RF	Yes, No, H=5	300	30,92	0,11	15,75	−11,85	1,74	5,26	0,33	13,89	−5,11	5,72	57	18,46	0,07	3,43	−0,58	0,10	0,82
RF	No, Yes, H=1	300	61,12	0,19	17,82	−19,36	2,14	7,47	−0,12	13,58	−6,23	8,39	47	7,97	0,03	0,92	−0,64	0,19	1,08
RF	No, Yes, H=5	300	29,36	0,10	16,10	−21,12	2,09	4,42	−0,75	16,21	−6,15	4,29	58	17,05	0,06	2,37	−0,61	0,12	0,89
RF	Yes, Yes, H=1	300	60,21	0,19	17,90	−25,15	2,18	7,27	−0,45	16,45	−6,13	8,24	55	7,36	0,03	0,90	−0,65	0,19	1,08
RF	Yes, Yes, H=5	300	22,90	0,08	15,66	−19,82	2,13	3,65	−0,63	13,77	−6,67	3,27	62	11,20	0,04	1,39	−0,65	0,13	0,89
RF + DBN	Yes, Yes, H=1	100	22,16	0,08	13,39	−18,32	1,98	3,73	−0,16	12,22	−5,89	3,52	75	0,01	0,00	−0,53	−0,66	0,08	0,87
RF + DBN	Yes, Yes, H=5	100	24,66	0,09	16,52	−14,19	1,76	4,36	−0,10	13,08	−5,52	4,52	51	12,79	0,05	2,25	−0,56	0,08	0,75
RF + DBN	Yes, Yes, H=1	300	21,20	0,08	16,87	−21,96	2,48	3,23	−0,63	13,23	−7,36	2,81	86	−0,78	0,00	−0,42	−0,89	0,12	1,14
RF + DBN	Yes, Yes, H=5	300	15,17	0,06	15,71	−20,20	2,38	2,62	−0,52	11,71	−7,22	2,08	79	4,21	0,02	0,40	−0,79	0,11	1,00
Average across models			35,17	0,12	−0,61	0,11	0,87	5,46	0,23	19,07	−5,58	5,80	57	10,60	0,04	1,57	−0,61	0,11	0,87
Market			9,47	0,04	11,35	−8,95	1,14	2,82	−0,12	11,08	−3,23	2,83	55			2,83	−0,46	0,08	0,50

Table 2
Risk-returns characteristics of the portfolios over the period between January 1993 and June 2015 (Part 2): Augmented Fama and French regression (Momentum + Reversal), sub-periods returns and decomposition of the portfolio between the two legs. Each model is a combination between one or two forecasting methods (Deep Belief Network, Elastic Net regression, Random Forest, Elastic Net regression + DBN, a specific type of information set, a forecasting/holding period and a number of stocks in the allowed universe. The best/top three returns/performances are highlighted using bold.

Method	Data and Horizon ID, TIR + EX, H	Nb of Stocks	Fama-French 3 + 2 reg.				Mean daily returns (Sub-periods)				Long and Short Returns vs Market	
			Int.	Std	t-stat	Int. After T.C.	01/93 12/00	01/01 08/08	09/08 12/09	01/10 06/15		
			A			B	C	D	E	F	G	H
DBN	No, No, H=1	100	0,12	0,02	5,84	0,04	0,28	0,14	**0,33**	0,03	0,10	−0,09
DBN	No, No, H=5	100	0,04	0,02	2,60	0,00	0,20	0,07	0,07	**0,04**	0,06	−0,06
DBN	Yes, No, H=1	100	0,08	0,02	4,28	0,00	0,20	0,09	0,21	0,01	0,07	−0,06
DBN	Yes, No, H=5	100	0,03	0,02	1,62	−0,01	0,09	0,02	0,17	0,02	0,04	−0,02
DBN	No, Yes, H=1	100	0,07	0,02	2,95	−0,01	0,19	0,06	−0,04	0,01	0,06	−0,05
DBN	No, Yes, H=5	100	0,03	0,02	1,46	−0,01	0,17	0,02	−0,19	0,00	0,05	−0,02
DBN	Yes, Yes, H=1	100	0,06	0,02	2,89	−0,02	0,15	0,05	−0,06	**0,05**	0,06	−0,04
DBN	Yes, Yes, H=5	100	−0,01	0,02	−0,53	−0,05	0,11	0,01	−0,15	0,00	0,03	−0,02
DBN	No, No, H=1	300	0,14	0,03	5,30	**0,06**	0,38	**0,17**	−0,04	0,03	**0,11**	**−0,11**
DBN	No, No, H=5	300	0,06	0,02	2,88	0,02	0,23	0,12	−0,05	**0,04**	0,08	−0,07
DBN	Yes, No, H=1	300	0,10	0,02	4,28	0,02	0,27	0,12	**0,30**	0,00	0,09	−0,08
DBN	Yes, No, H=5	300	0,03	0,02	1,17	−0,01	0,13	0,03	**0,34**	−0,03	0,06	−0,03
DBN	No, Yes, H=1	300	0,10	0,03	3,77	0,02	0,22	0,13	−0,20	0,02	0,08	−0,07
DBN	No, Yes, H=5	300	0,03	0,02	1,30	−0,01	0,14	0,06	−0,19	0,01	0,05	−0,03
DBN	Yes, Yes, H=1	300	0,07	0,02	3,11	−0,01	0,18	0,07	−0,07	0,03	0,07	−0,05
DBN	Yes, Yes, H=5	300	−0,01	0,02	−0,67	−0,05	0,11	0,02	−0,31	0,01	0,03	−0,02
El.Net	Yes, Yes, H=1	100	0,13	0,02	6,03	0,01	0,27	0,13	0,12	0,00	0,08	−0,08
El.Net	Yes, Yes, H=5	100	0,04	0,02	2,20	0,00	0,16	0,06	−0,08	0,01	0,05	−0,03
El.Net	Yes, Yes, H=1	300	0,14	0,03	5,27	0,02	**0,41**	0,16	0,00	−0,03	**0,11**	**−0,11**
El.Net	Yes, Yes, H=5	300	0,03	0,02	1,35	−0,01	0,17	0,08	−0,11	−0,03	0,06	−0,03
RF	No, No, H=1	100	0,14	0,02	9,18	−0,02	0,27	0,15	0,15	**0,04**	0,09	−0,09
RF	No, No, H=5	100	0,07	0,01	5,57	0,03	0,19	0,07	0,03	0,03	0,06	−0,05
RF	Yes, No, H=1	100	0,14	0,02	8,40	−0,02	0,31	0,14	0,06	0,03	0,10	−0,08
RF	Yes, No, H=5	100	0,07	0,01	5,18	0,03	0,20	0,08	0,05	0,02	0,06	−0,05
RF	No, Yes, H=1	100	0,11	0,02	5,44	−0,05	0,28	0,10	0,03	0,01	0,08	−0,08
RF	No, Yes, H=5	100	0,03	0,02	1,82	−0,01	0,18	0,02	−0,16	0,03	0,04	−0,04
RF	Yes, Yes, H=1	100	0,08	0,02	4,24	−0,08	0,27	0,07	−0,07	−0,01	0,07	−0,06
RF	Yes, Yes, H=5	100	0,04	0,02	1,56	0,00	0,21	0,01	0,04	−0,02	0,05	−0,04
RF	No, No, H=1	300	**0,22**	0,02	10,96	**0,06**	**0,48**	**0,18**	0,15	**0,07**	**0,16**	**−0,11**
RF	No, No, H=5	300	0,10	0,01	6,61	**0,06**	0,28	0,13	0,01	0,00	0,08	−0,07
RF	Yes, No, H=1	300	**0,20**	0,02	8,90	0,04	**0,43**	**0,21**	0,21	0,03	**0,15**	**−0,11**
RF	Yes, No, H=5	300	0,06	0,02	3,60	0,02	0,22	0,09	0,01	−0,03	0,07	−0,05
RF	No, Yes, H=1	300	0,14	0,02	5,85	−0,02	0,39	0,14	0,05	0,01	0,10	**−0,11**
RF	No, Yes, H=5	300	0,06	0,02	2,86	0,02	0,23	0,05	0,09	−0,01	0,07	−0,06
RF	Yes, Yes, H=1	300	**0,15**	0,02	6,12	−0,01	**0,41**	0,11	0,03	0,00	**0,11**	−0,10
RF	Yes, Yes, H=5	300	0,04	0,02	1,76	0,00	0,21	0,01	0,13	−0,02	0,06	−0,05
RF + DBN	Yes, Yes, H=1	100	0,05	0,02	2,47	−0,03	0,18	0,07	−0,05	−0,03	0,05	−0,05
RF + DBN	Yes, Yes, H=5	100	0,05	0,02	2,74	0,01	0,18	0,05	0,01	0,02	0,05	−0,05
RF + DBN	Yes, Yes, H=1	300	0,06	0,03	2,28	−0,02	0,21	0,05	−0,25	−0,02	0,07	−0,04
RF + DBN	Yes, Yes, H=5	300	0,03	0,02	1,20	−0,01	0,12	0,08	−0,28	0,00	0,04	−0,05
Average return or alpha: all methods			0,08			0,00	0,23	0,09	0,01	0,01	0,07	−0,06
	Market						0,06	0,01	−0,03	0,05		

- On average, the models with one-day forecasting/holding period perform better (excluding transaction costs) than those with a five-day horizon.
- The intersection of big data (using a large number of predictors in market finance) and machine learning techniques is one of the most important drivers of this article. Empirically, excluding transaction costs, with the models studied, on average, a substantial deterioration in performance is observed when all predictors are included (LR + ID + TIR + EX) compared to the case with only stock lagged returns (LR). This decline is more pronounced for DBN compared to Random Forests. In other words, the Random Forests seem less affected by variables which are probably of little importance, add noise and/or could monopolize learning resources. When all features are considered, the annualized returns nevertheless decrease by about one third.
- The Elastic Net regression is a method specifically designed to manage a large number of features and to perform dimensionality reduction. Empirically, returns are slightly lower than those of the two other methods (excluding transaction costs) when only lagged returns are considered. After transaction costs, returns are still positive. In the universe of models including all predictors, the Elastic Net regressions generate the best returns.

Regarding the selection of the predictors inside the Elastic Net regression, the main and average information are the following. Recall this type of computation has been done 45 times (biannual re-estimation of the trading system). In fact, as a consequence of the shrinkage effect, from an original universe of 392 or 592 predictors, on average, only 30 are kept. The selection is in line with the information provided in Table 4 which focus on variable importance using a Random Forest approach. The 11 predictors from the lagged returns group (LR) are very often in the pool. For example, the three most recent lags are selected in about 75% of the batches. Over-represented factors are then the industry (3 to 4 sectors are kept) and the most recent daily returns of other time series. For the remaining factors, no rule emerges. As expected, stock ID, because they are especially sparse, are nearly never selected.

- A two-step methodology combining a random forest (feature selection) and a DBN could potentially be a solution to overcome the difficulty of DBN to manage a large number of predictors. However, when keeping only the first quartile of most important predictors, only a slight increase in the returns, when the trading period lasts five days, is observed.

Besides the discussion on the return, the standard deviation of the portfolios is of major importance and explanations linked to the portfolio construction must be given. As mentioned previously, these equi-weighted dollar-neutral portfolios comprise the daily flop and top ten stocks (according to the forecasts/rankings). When the trading/holding period is five days, each leg of the portfolio is formed, every given day, by 50 positions (5 days $\times$ 10 positions per day): they should be much more diversified than the one-day portfolios (10 stocks per leg) and thus have a lower standard deviation. Comparing pairs of portfolios which only differ by their horizon, the decrease of the standard deviation is rather small and an "incoherence" can be observed (Random Forest with stock ID, additional features, 100 stocks. Std: 1.84 vs 2.14). This issue is inherent to the methodology and the use of lagged returns with a low frequency (e.g. between times $t-126$ and $t-252$/ 6 months and one year). Low frequency returns involve features that are strongly auto-correlated. In the above example, if we shift by one period, 125 of the 126 (daily) elements building this 6-month return would be the same. As a consequence, because some of the predictors are pretty similar (low frequency returns), it becomes likely that a stock is chosen for two (or more) consecutive days in the top/flop portfolios. When the trading/holding horizon is five days, a leg of the portfolio is not formed by 50 different stocks with an equal weighting but by a lower number with different weights. These repetitions also occur when the horizon is just one day: each new one-day portfolio contains some stocks already selected the day before.

Additional risk and performance metrics like the Sharpe Ratio or the maximum DrawDown show the top three strategies also perform well using those criterion.[20] The maximum DrawDown of top strategies is lower than the one of the market (55%). The average maximum DrawDown between all 40 models is nevertheless slightly higher (57%).

The forecasting horizon/the duration of the holding period has a direct consequence on the transaction costs the portfolio manager would face. Estimating precisely, through time, the trading costs of the portfolio managed in this article is quite difficult. Avellaneda and Lee (2010) consider a slippage/transaction costs of 0.05% per trade, which is a reasonable estimate for a universe of highly liquid stocks (always in the top 300 in terms of market capitalization, in the first decile of listed companies[21] in the U.S.). 0.20% per day (leverage 2 portfolio) could be considered a conservative average for the global transaction costs (bid-ask spread + brokerage fees), during the whole trading period, when the horizon is one day (when the "benefits" of repetitions are not integrated). When the horizon is five days, 0.04% is the equivalent (each day, 20% of the capital undergo 0.20% of transaction costs). The effective spreads (and the brokerage fees) have declined substantially during the trading period covered by this study. As reported in Chordia, Roll, and Subrahmanyam (2011), transaction costs were especially high in the nineties (up to 0.30%) compared to the values observed in the most recent years. These average relative spreads are computed for all NYSE-listed stocks. The ones of the top-100 or -300 stocks in terms of market capitalization, not mentioned in Chordia et al. (2011), were probably lower. At this stage, in other words, it is likely that daily transaction costs of 0.20% and 0.04% for one- or five-day holding period portfolios are an underestimation during the beginning of the trading period and an overestimation during the last years.[22]

These "raw" transaction costs must now integrate the impact of repetitions. It is especially important when the trading horizon is one day: the effective holding period of a stock is in fact greater than one day on average. The probability of a given stock to be chosen two consecutive days, the "inertia" of the algorithms in terms of selection, is very different between the three algorithms, Random Forest, DBN and Elastic Net regressions: about 20%, 60% and 40% respectively. In other words, the real average holding period of a stock selected via DBN is 2.5 days and 1,25 days with Random Forest.[23] As a consequence, for these three methods, with one-day holding, transaction costs of 0.16%, 0.08% and 0.12% could be considered. These differences in terms of "inertia" (and the consequences on transaction costs) lead to the following question: why is the probability of a stock to be chosen two consecutive days greater with DBN than with Random Forest. The answer given here is a suggestion: DBN give a (relatively) higher weight to inert inputs. These inputs may be seen as low frequency returns/predictors (e.g. second last semester returns). Each tree in a random forest and Elastic Net Regressions only select a subset of predictors. These two methods foster (the selection of) short-term predictors.

The impact of "inertia" on transaction costs is marginal when the holding period is five days. If DBN do not provide the best forecasts (statistically), lower transaction costs improve the relative performance (in terms of global return) of this method. The same phenomenon appears between one- and five-day forecasting/trading models: the introduction of transaction costs strongly shortens the difference in performance between one-day and five-day portfolios. Mean returns are thus quite similar.

Table 2 provides some additional information regarding the performance of the different portfolios. The exposure of returns to common sources of systematic risk is evaluated via the Fama-French three-factor model (Fama & French, 1993) augmented by momentum and short-term reversal factors (Carhart, 1997; Jegadeesh, 1990). Only the value of the intercepts of the regressions are reported in the Table 2. After accounting for these five traditional factors, most of them keep a positive intercept (38 out of 40, column A). After transaction costs, only 50% of the models still have a positive alpha.[24] In line with Krauss et al. (2017), most portfolios positively (and significantly) load on the market, the momentum and the reversal factors. For the latter two factors, it seems that machine learning algorithms are able to extract such types of patterns from the data. No clear rule emerges with the SMB factor which seems coherent with a universe of large-cap stocks. Most regressions exhibit a negative loading with the HML factors suggesting the selection procedure leads to portfolios with predominantly low book-to-market stocks (high beta/growth stocks). These stocks are the most volatile. A selection based on prediction (top and flop stocks) explains this particular bias in the portfolios. As an example, technology stocks are over-weighted (compared to the relevant stock index) as noted by Krauss et al. (2017).

[20] Various performance measures have been proposed and criticized in the literature. Articles like Sharpe (1994), Scholz (2007), Krimm, Scholz, and Wilkens (2012), with a focus on the Sharpe Ratio, or Ferson and Schadt (1996), Chen and Knez (1996) and the references therein may be useful for the reader.

[21] http://www.nyxdata.com/nysedata/asp/factbook/viewer_edition.asp?mode=table&key=76&category=4.

[22] As an example, in 2013, quoted bid-ask spreads for most of the largest stocks are at the minimum level of one cent: https://www.sec.gov/data/market-structure/spreads-and-depth-by-individual-security.html.

[23] Example with Random Forests: $1.25=\frac{1}{1-0.2}$.

[24] An alpha, α, is a measure of performance (see for example Jensen (1968)) (column B). When it is (significantly) positive, it indicates the strategy has overperformed with respect to an adequate benchmark and appropriate risk factors.

Table 3
Feature importance (per sub-group of variables) according to Random Forests and permutations of the observations. A value at 0 means that randomly reordering the observations of a variable does not change the prediction error (this variable is not important for the accuracy of the model) whereas a positive value indicates that the permutations of a variable negatively affect the prediction error (this variable is important for the accuracy of the model): the higher this indicator, the higher the predictive power of this variable.

	Number of stocks	100	100	300	300
	Horizon (days)	1	5	1	5
Summary statistics (all variables)					
	Mean	0,261	0,297	0,201	0,241
	Max	1,113	1,419	1,246	1,645
	Min	−0,031	0,000	−0,005	0,000
	Std	0,207	0,199	0,239	0,273
	Skewness	1,088	2,445	1,291	1,702
	Kurtosis	6,664	13,118	5,914	8,151
	Median	0,326	0,308	0,006	0,038
	Quartile 3	0,355	0,373	0,399	0,438
Mean importance (per group of variables)					
group *i*: LR	Stock lagged returns	1,039	1,151	1,181	1,433
group *ii*: ID	Stock ID	−0,001	0,100	0,000	0,020
group *iii*: TIR	Day	0,142	0,077	0,183	0,149
	Month	0,068	0,100	0,096	0,139
	Sector (dummies)	0,461	0,624	0,494	0,805
	VIX (dummies)	0,052	0,060	0,062	0,083
group *iv*: EX	Sector returns	0,342	0,339	0,398	0,439
	Momentum and Reversal	0,345	0,358	0,403	0,450
	VIX, Gold, WTI	0,355	0,341	0,415	0,444
	Market (excess return)	0,326	0,331	0,379	0,423
	SMB	0,344	0,348	0,397	0,445
	HML	0,344	0,351	0,402	0,446
	3 month interest rate	0,275	0,334	0,317	0,389
	10 year interest rate	0,287	0,367	0,333	0,420

The global trading period has been divided into 4 sub-periods (columns C to F). The conclusion is highly visible: the trend of the returns is clearly bearish and also confirms the findings in Krauss et al. (2017). During the first period (1993 to 2000, column C), the average daily return of the 40 models is very impressive with 0.23% per day. At that time, tools like random forests and the required computation power were not available. Between 2001 and 2008 (column D), the daily performance is much more modest, 0.09%, but all models generate a positive return. The financial crisis period (Sept 2008/December 2009, only 16 months, column E) shows a large diversity of returns which is hard to explain and the mean is economically non-significant (0.01%). Even inside this highly turbulent sub-period on the market (very high level of the VIX index, a strong drop then a recovery), the performance of each portfolio may alternate rapidly between significant rises and falls. This questions the adaptation of the models to these particular situations. The most recent period (2010-2015, column F) reveals that the different techniques used were not able to generate positive and economically significant trading signals. In some way, this confirms the efficiency (the evolutionary perspective) of the market given that profit opportunities are arbitraged away with increasing popularity of these methods. The last information of Table 2 deals with the sources of the returns. They are investigated via the behavior of the short (column G) and long (column H) legs of the portfolios (difference with the market). Both legs contribute more or less equally to the performance. Inside the sub-periods, this results still holds.

5.2. On feature importance

The first comments provided after Table 1 invite us to have a closer look at variable importance. The influence of each group of variables is provided in Table 3. Table 4 specifically focuses on stock lagged returns (LR), given that these variables have led to the best performance when taken exclusively. The importance values were extracted via the Random Forest algorithm and normalized. The predictor importance does not define the type of patterns/interactions the learning algorithm has found. Four cases are considered as a combination of the number of stocks in the universe (100 or 300) and the holding/forecasting period (one day or one week). As shown by Table 3, if variables are grouped (average for all lags, for all dummies dealing with the day of the week, etc), one type of data clearly dominates the others in terms of importance: the lagged returns of the stocks. Detailed results show these 11 variables/lags/stock returns are always, whatever the batch, in the top 20 of a pool of 392 or 592 variables (100 or 300 stocks). All other continuous variables are about at least three times less important. Binary data, such as the day of the week or the month or the VIX (as four dummy variables) have a small impact. The (average) influence of the stock ID (99 or 299 dummy variables, at most one being equal to one) is very weak but these variables may monopolize part of the computation power and introduce noise as discussed previously.

One group of variables which is sparse, the sector ID (10 sectors, 9 binary variables), presents an importance which is "significantly" greater than those of continuous variables. An alternative (and reduced) set of predictors only formed of stock lagged returns and sector ID, the two most important groups of variables, will thus be tested in the following. It may be seen as a good trade−off between information, computation power and the learning capacities of the algorithms.

Table 4, for the two forecasting/trading horizons and the two different numbers of stocks in the pool, details the importance of each stock lagged return (LR). These indicators are relatively homogeneous with a spread of less than 40% between the most and the least important lags. Even with the one-day horizon, the short-term lags (say $Ret_{0,1}$ and $Ret_{1,2}$) are not necessarily on the top positions in terms of importance. When the forecasting horizon is one week, the importance of low frequency returns (from $Ret_{5,10}$ to $Ret_{126,252}$) is consolidated. The influence of these returns is clearly connected with the repetition/inertia in the stock selection discussed with Table 1 and the significance of momentum and reversal factors in augmented Fama-French regressions in Table 2. With respect to these elements, the next sub-section will present

Table 4
Importance of stock returns (detail for each lag) according to Random Forests and permutations of the observations.

Number of Stocks =100 Horizon = 1 day		Number of Stocks =100 Horizon = 5 days		Number of Stocks =300 Horizon = 1 day		Number of Stocks =300 Horizon = 5 days	
Importance	Lag	Importance	Lag	Importance	Lag	Importance	lag
0,945	$Ret_{4,5}$	0,873	$Ret_{4,5}$	1,108	$Ret_{3,4}$	1,270	$Ret_{2,3}$
0,975	$Ret_{2,3}$	0,913	$Ret_{3,4}$	1,132	$Ret_{2,3}$	1,271	$Ret_{1,2}$
1,008	$Ret_{3,4}$	0,932	$Ret_{2,3}$	1,149	$Ret_{5,10}$	1,299	$Ret_{3,4}$
1,008	$Ret_{10,21}$	0,993	$Ret_{1,2}$	1,154	$Ret_{4,5}$	1,328	$Ret_{4,5}$
1,013	$Ret_{5,10}$	1,087	$Ret_{0,1}$	1,185	$Ret_{10,21}$	1,355	$Ret_{0,1}$
1,043	$Ret_{42,63}$	1,201	$Ret_{5,10}$	1,189	$Ret_{126,252}$	1,397	$Ret_{5,10}$
1,050	$Ret_{21,42}$	1,238	$Ret_{10,21}$	1,193	$Ret_{0,1}$	1,473	$Ret_{10,21}$
1,077	$Ret_{126,252}$	1,292	$Ret_{42,63}$	1,196	$Ret_{1,2}$	1,494	$Ret_{21,42}$
1,089	$Ret_{63,126}$	1,309	$Ret_{21,42}$	1,218	$Ret_{42,63}$	1,592	$Ret_{126,252}$
1,106	$Ret_{0,1}$	1,400	$Ret_{63,126}$	1,222	$Ret_{21,42}$	1,634	$Ret_{63,126}$
1,112	$Ret_{1,2}$	1,419	$Ret_{126,252}$	1,246	$Ret_{63,126}$	1,645	$Ret_{42,63}$

Table 5
Daily portfolio (excess) returns with alternative information sets and an adapted specification of the forecasting method over the trading period between January 1993 and June 2015. The best three returns are highlighted using bold.

Horizon (Days)	Nb Stocks	Features			DBN			Random Forest	
		Stock returns (LR)	Data	Nb of features	Layers size	Before T.C.	After T.C.	Before T.C.	After T.C.
1	100	Reduced	Sector	7 + 9	16/8	**0,10**	0,02	0,18	0,02
5	100	Reduced	Sector	7 + 9	16/8	0,05	0,01	0,09	0,05
1	300	Reduced	Sector	7 + 9	16/8	0,05	−0,03	**0,26**	**0,10**
5	300	Reduced	Sector	7 + 9	16/8	0,05	0,01	0,11	0,07
1	100	All	Sector	11 + 9	16/8	**0,12**	**0,04**	0,18	0,02
5	100	All	Sector	11 + 9	16/8	0,07	0,03	0,10	0,06
1	300	All	Sector	11 + 9	16/8	0,17	**0,09**	**0,28**	**0,12**
5	300	All	Sector	11 + 9	16/8	0,08	**0,04**	0,12	**0,08**
1	100	Reduced		7	7/4	0,01	−0,07	0,15	−0,01
5	100	Reduced		7	7/4	0,01	−0,03	0,08	0,04
1	100	Reduced	ID	7 + 99	27/14	**0,10**	0,02	0,18	0,02
5	100	Reduced	ID	7 + 99	27/14	0,05	0,01	0,09	0,05
1	100	Reduced	TIR + EX	7 + 190	50/25	0,06	−0,02	0,11	−0,05
5	100	Reduced	TIR + EX	7 + 190	50/25	0,05	0,01	0,08	0,04
1	100	Reduced	ID + TIR + EX	7 + 99 + 190	75/38	0,04	−0,04	0,14	−0,02
5	100	Reduced	ID + TIR + EX	7 + 99 + 190	75/38	0,03	−0,01	0,08	0,04
1	300	Reduced		7	7/4	−0,01	−0,09	**0,24**	**0,08**
5	300	Reduced		7	7/4	−0,03	−0,07	0,11	0,07
1	300	Reduced	ID	7 + 299	77/39	**0,12**	**0,04**	**0,24**	**0,08**
5	300	Reduced	ID	7 + 299	77/39	0,07	0,03	0,09	0,05
1	300	Reduced	TIR + EX	7 + 190	50/25	0,07	−0,01	0,18	0,02
5	300	Reduced	TIR + EX	7 + 190	50/25	0,06	0,02	0,09	0,05
1	300	Reduced	ID + TIR + EX	7 + 299 + 190	125/63	0,02	−0,06	0,17	0,01
5	300	Reduced	ID + TIR + EX	7 + 299 + 190	125/63	0,03	−0,01	0,09	0,05

some alternative models and check the robustness of the first conclusions.

5.3. Some alternative specifications

The results of the last two sub-sections have provided directions to propose alternative models in terms of predictor. This is a strategy to control the strength of our first conclusions. They are tested only on the Random Forest and DBN algorithms.

The first alternative specification emphasises on the sector identifier. A set of features combining the stock lagged returns and the sector ID is created. According to Table 3, these features are clearly the most important. The idea is to help the learning algorithms focusing on the essential and to save computational capacities. The second change deals with the number of lagged returns. An alternative (reduced) number of lags is used. In the core results of Table 1, 11 different lags/returns have been defined as inputs/features via the following number of days/lags: 1, 2, 3, 4, 5, 10, 21, 42, 63, 126 and 252. Limiting the number of days/lags to 21 (one month) leads to only 7 different returns. The momentum and reversal factors are significant in the Fama-French regressions of Table 2. Reducing the number of returns/lags is a way to assess the importance of these two risk factors and to evaluate the general sensitivity of the performances/returns. Numerous specifications for the number of lags are possible but the idea is to stay simple. Looking (for a method to determine) for “optimal lags” seems difficult in the context of this article.

Table 5 reports the mean daily return for these different alternative groups of predictors. For each learning algorithm (DBN or RF), 24 models are defined: these are combinations between the different types of variables (stock lagged returns (reduced or not)LR, sector ID, stock ID, TIR and EX) mentioned in this article. The rules given in the Section Methodology are kept. For the DBN, the design/number of autoencoders per layer has been slightly adapted.

The results show that the DBN returns are strongly negatively affected by the absence of information dating from more than one month. In other words, a large of part of the returns reported in Table 1 may be attributed to the ability to exploit momentum and reversal behaviors. Even the combination of all stock lagged returns and the sector ID, which appears in our hypothesis, as the

most favorable case, leads to a decline in return. Table 5 confirms the superiority of the random forests compared to the DBN excluding transaction costs. On average, with Random Forests, only a slight decline of the results is reported when low frequency ("long-term") returns are not available. The combination of stock lagged returns and sector ID leads to a weak increase of the returns. The best return (excluding transaction costs) of this article, 0.28% per day, is thus obtained when the pool contains 300 stocks.

6. Concluding remarks

Using a database with several hundreds of predictors, this article has developed several independent statistical arbitrage strategies based on three methods (Random Forest, Deep Belief Networks and Elastic Net Regression). These algorithms are relevant when the number of predictors is high (several hundreds). The contribution of this work is mainly empirical (What are the most salient trends? Which information is the most valuable?). The main objective of this article is to contribute to a literature combining big data, machine learning and finance (statistical arbitrage, market efficiency/anomalies). While these elements are currently becoming the norm in the market, the academic literature is relatively sparse. The applications and the forecasting models, focusing on the U.S. market, are based on various types of information and report results on a 22-year trading period. The first pillar of the data lies on the stock lagged returns which is common practice. Other predictors may include the precise identification of each stock as well as the sector a stock belongs to. The datasets are enhanced by sector or commodity (oil, gold) returns but also by the VIX Index or the Fama and French factors. In total, a feature space with up to 600 predictors is considered.

Building long-short portfolios based on different algorithms, information sets and forecasting/holding horizons, several major points appear. All raw returns are positive. After transaction costs, most of them remain positive. Denoising returns from traditional risk factors, only 50% of the models have a positive alpha. Adding predictors is not a guarantee to increase the performance of the portfolios. In the applications, the portfolios with the highest returns are mainly those only considering the stock lagged returns as inputs. More predictors means potentially more noise. The underlying question is about the ability of the algorithms to manage this challenge. Among the three tools considered, Random Forests generate the portfolios with the best performances excluding transaction costs. Two trading/forecasting/holding horizons (one day or one week) have been tested. With a one-day horizon, the raw returns (excluding transaction costs) are clearly higher. Including transaction costs, the spread is sharply reduced. A crucial element is the bearish trend of the returns. If annual returns greater than 100% are reported in the first years (in the nineties), in the last years (say after 2010), the results are economically non-significant in line with Krauss et al. (2017). An increase in efficiency of the markets and the diffusion of these tools in the finance industry are potential explanations for this trend.

This article is a step to bridge the gap between academic finance and "professional" finance in questioning big datasets with machine learning methods. This field of research requires extensive additional empirical works. It includes analysis of the forecasting/trading rules/horizons (arbitrage between effective holding periods and transaction costs, the rules to open and close positions). Identifying each stock, trying to capture individual/specific effects, rises the number of predictors and the level of noise and reduces the performance of the trading system. Deeper empirical studies involving the hyper-optimization of all parameters are necessary. This suggests a more general discussion of the advantages and disadvantages between a single and large model (e.g. DBN in this article) versus a combination/aggregation of forecasts/models/smaller information sets (Dietterich, 2000; Huck, 2009) to build a trading system. Further studies could focus on the importance of alternative datasets, such as news, tweets, earnings estimates and other information potentially relevant to the development of the price of stocks.

References

Ariel, R. A. (1990). High stock returns before holidays: Existence and evidence on possible causes. *Journal of Finance, 45*(5), 1611–1626.

Atsalakis, G. S., & Valavanis, K. P. (2009). Surveying stock market forecasting techniques–Part II: Soft computing methods. *Expert Systems with Applications, 36*(3), 5932–5941.

Avellaneda, M., & Lee, J.-H. (2010). Statistical arbitrage in the US equities market. *Quantitative Finance, 10*(7), 761–782.

Baba, N., & Sakurai, Y. (2011). Predicting regime switches in the VIX index with macroeconomic variables. *Applied Economic Letters, 18*(13-15), 1415–1419. doi:10.1080/13504851.2010.539532.

Baesens, B., Bapna, R., Marsden, J. R., Vanthienen, J., & Zhao, J. L. (2016). Transformational issues of big data and analytics in networked business. *MIS Quarterly, 40*(4), 807–818.

Bahrammirzaee, A. (2010). A comparative survey of artificial intelligence applications in finance: Artificial neural networks, expert system and hybrid intelligent systems. *Neural Computing and Applications, 19*(8), 1165–1195. doi:10.1007/s00521-010-0362-z.

Bai, J., Fan, J., & Tsay, R. (2016). Special issue on big data. *Journal of Business & Economic Statistics, 34*(4), 487–488.

Baldi, P. (2012). Autoencoders, unsupervised learning, and deep architectures. In I. Guyon, G. Dror, V. Lemaire, G. Taylor, & D. Silver (Eds.), *Proceedings of icml workshop on unsupervised and transfer learning*. In *Proceedings of Machine Learning Research: 27* (pp. 37–49). Bellevue, Washington, USA: PMLR.

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). doi:10.1371/journal.pone.0180944.

Baron, J., Mellers, B. A., Tetlock, P. E., Stone, E., & Ungar, L. H. (2014). Two reasons to make aggregated probability forecasts more extreme. *Decision Analysis, 11*(2), 133–145.

Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and Trends in Machine Learning, 2*(1), 1–127. doi:10.1561/2200000006.

Black, A. J., Klinkowska, O., McMillan, D. G., & McMillan, F. J. (2014). Forecasting stock returns: Do commodity prices help? *Journal of Forecasting, 33*(8), 627–639.

Blake, M. (2014). Smart technology for big data. *Journal of Trading, 9*(1), 57–66.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.

Caliskan, D., & Najand, M. (2016). Stock market returns and the price of gold. *Journal of Asset Management, 17*(1), 10–21.

Carhart, M. M. (1997). On persistence in mutual fund performance. *The Journal of Finance, 52*(1), 57. doi:10.2307/2329556.

Chen, H., Chiang, R. H. L., & Storey, V. C. (2012). Business intelligence and analytics: From big data to big impact. *MIS Quarterly, 36*(4), 1165–1188.

Chen, Z., & Knez, P. (1996). Portfolio performance measurement: theory and applications. *Review of Financial Studies, 9*(2), 511.

Chong, E., Han, C., & Park, F. C. (2017). Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. *Expert Systems with Applications, 83*, 187–205. doi:10.1016/j.eswa.2017.04.030.

Chordia, T., Roll, R., & Subrahmanyam, A. (2011). Recent trends in trading activity and market quality. *Journal of Financial Economics, 101*(2), 243–263.

Christoffersen, P. F., & Diebold, F. X. (2006). Financial asset returns, direction-of-change forecasting, and volatility dynamics. *Management Science, 52*(8), 1273–1287.

Copeland, M. M., & Copeland, T. E. (1999). Market timing: Style and size rotation using the VIX. *Financial Analysts Journal, 55*(2), 73–81.

Coppersmith, D., Hong, S. J., & Hosking, J. R. (1999). Partitioning nominal attributes in decision trees. *Data Mining and Knowledge Discovery, 3*(2), 197–217. doi:10.1023/A:1009869804967.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS), 2*(4), 303–314. doi:10.1007/BF02551274.

DeMiguel, V., Garlappi, L., & Uppal, R. (2009). Optimal versus naive diversification: How inefficient is the 1/n portfolio strategy? *Review of Financial Studies, 22*(5), 1915–1953.

Deng, Y., Bao, F., Kong, Y., Ren, Z., & Dai, Q. (2017). Deep direct reinforcement learning for financial signal representation and trading. *IEEE Transactions on Neural Networks & Learning Systems, 28*(3), 653–664.

Dhar, V. (2015a). The scope and challenges for deep learning. *Big Data, 3*(3), 127–129.

Dhar, V. (2015b). Should you trust your money to a robot. *Big Data, 3*(2), 55–58.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *Multiple classifier systems* (pp. 1–15). Springer.

Enke, D., & Thawornwong, S. (2005). The use of data mining and neural networks for forecasting stock market returns. *Expert Systems with Applications, 29*(4), 927–940. doi:10.1016/j.eswa.2005.06.024.

Fama, E. F., & French, K. R. (1993). Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics, 33*(1), 3–56. doi:10.1016/0304-405X(93)90023-5.

Fan, J., Han, F., & Liu, H. (2014). Challenges of big data analysis. *National Science Review, 1*(2), 293–314. doi:10.1093/nsr/nwt032.

Fernandes, M., Medeiros, M. C., & Scharth, M. (2014). Modeling and predicting the CBOE market volatility index. *Journal of Banking & Finance, 40*, 1–10.

Ferson, W. E., & Schadt, R. W. (1996). Measuring fund strategy and performance in changing economic conditions. *Journal of Finance, 51*(2), 425–461.

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research, 270*(2), 654–669.

Fleming, J., Kirby, C., & Ostdiek, B. (2001). The economic value of volatility timing. *Journal of Finance, 56*(1), 329–352.

Genuer, R., Poggi, J.-M., Tuleau-Malot, C., & Villa-Vialaneix, N. (2017). Random forests for big data. Big Data Research, *9*, 28–46. https://hal.archives-ouvertes.fr/hal-01233923.

George, G., Haas, M. R., & Pentland, A. (2014). Big data and management. *Academy of Management Journal, 57*(2), 321–326.

George, G., Osinga, E. C., Lavie, D., & Scott, B. A. (2016). Big data and data science methods for management research. *Academy of Management Journal, 59*(5), 1493–1507.

Gibbons, M. R., & Hess, P. (1981). Day of the week effects and asset returns. *Journal of Business, 54*(4), 579–596.

Grushka-Cockayne, Y., Jose, V. R. R., & Lichtendahl, K. C. (2017). Ensembles of overfit and overconfident forecasts. *Management Science, 63*(4), 1110–1130.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). The Elements of Statistical Learning. *Springer Series in Statistics*. New York, NY, USA: Springer New York Inc.

Heaton, J. B., Polson, N. G., & Witte, J. H. (2016). Deep learning for finance: Deep portfolios. *SSRN*. Available at SSRN: https://ssrn.com/abstract=2838013.

Hinton, G. E. (2009). Deep belief networks. *Scholarpedia, 4*(5), 5947. doi:10.4249/scholarpedia.5947. Revision #91189.

Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation, 18*(7), 1527–1554. doi:10.1162/neco.2006.18.7.1527.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science, 313*, 504–507.

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation from nonorthogonal problems. *Technometrics, 12*(1), 55–67.

Hong, H., Torous, W., & Valkanov, R. (2007). Do industries lead stock markets? *Journal of Financial Economics, 83*(2), 367–396.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer network are universal approximators. *Neural Networks, 2*(5), 359–366. doi:10.1016/0893-6080(89)90020-8.

Hsu, M.-W., Lessmann, S., Sung, M.-C., & Ma, T. (2016). Bridging the divide in financial market forecasting: machine learners vs. financial economists. *Expert Systems With Applications, 61*, 215–234.

Huang, W., Mollick, A. V., & Nguyen, K. H. (2016). U.s. stock markets and the role of real interest rates. *Quarterly Review of Economics & Finance, 59*, 231–242.

Huck, N. (2009). Pairs selection and outranking: An application to the S&P 100 index. *European Journal of Operational Research, 196*(2), 819–825.

Huck, N. (2010). Pairs trading and outranking: The multi-step-ahead forecasting case. *European Journal of Operational Research, 207*(3), 1702–1716.

Jacobs, H. (2015). What explains the dynamics of 100 anomalies? *Journal of Banking & Finance, 57*, 65–85. doi:10.1016/j.jbankfin.2015.03.006.

James, G. M., Witten, D., Hastie, T., & Tibshirani, R. (2014). An introduction to statistical learning: With applications in R. *Springer texts in statistics* (corrected). New York: Springer.

Jegadeesh, N. (1990). Evidence of predictable behavior of security returns. *The Journal of Finance, 45*(3), 881. doi:10.2307/2328797.

Jensen, M. C. (1968). The performance of mutual funds in the period 1945-1964. *Journal of Finance, 23*, 389–416.

Jiaqi, C., & Tindall, M. L. (2014). Hedge fund replication using shrinkage methodologies. *Journal of Alternative Investments, 17*(2), 26–49.

Jones, C. M., & Kaul, G. (1996). Oil and the stock markets. *Journal of Finance, 51*(2), 463–491.

Kilian, L., & Park, C. (2009). The impact of oil price shocks on the u.s. stock market. *International Economic Review, 50*(4), 1267–1287.

Krauss, C., Do, X. A., & Huck, 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.

Krimm, S., Scholz, H., & Wilkens, M. (2012). The sharpe ratio's market climate bias: Theoretical and empirical evidence from us equity mutual funds. *Journal of Asset Management, 13*(4), 227–242.

Kwan, M. (2014). Big data's impact on trading and technology. *Journal of Trading, 9*(1), 54–56.

Laopodis, N. T. (2013). Monetary policy and stock market dynamics across monetary regimes. *Journal of International Money & Finance, 33*, 381–406.

Leung, M. T., Daouk, H., & Chen, A.-S. (2000). Forecasting stock indices: A comparison of classification and level estimation models. *International Journal of Forecasting, 16*(2), 173–190.

Li, Y., Jiang, W., Yang, L., & Wu, T. (2018). On neural networks and learning systems for business computing. *Neurocomputing, 275*, 1150–1159. doi:10.1016/j.neucom.2017.09.054.

Lichtendahl, K., Yael, G.-C., Victor Richmond R., J., & Robert L., W. (2018). Bayesian ensembles of binary-event forecasts: When is it appropriate to extremize or anti-extremize? Working Paper 19035. arXiv:1705.02391v2

Lo, A. W. (2004). The adaptive markets hypothesis. *Journal of Portfolio Management, 30*(5), 15–29.

Lo, A. W. (2010). Hedge Funds: An analytic perspective. *Advances in financial engineering* (rev. and expanded). Princeton: Princeton University Press.

Longin, F., & Solnik, B. (2001). Extreme correlation of international equity markets. *Journal of Finance, 56*(2), 649–676.

Maasoumi, E., & Medeiros, M. C. (2010). The link between statistical learning theory and econometrics: Applications in economics, finance, and marketing. *Econometric Reviews, 29*(5/6), 470–475.

Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., & Hung Byers, A. (2011). *Big data: The next frontier for innovation, competition, and productivity*. McKinsey Global Institute.

Moritz, B., & Zimmermann, T. (2014). Deep conditional portfolio sorts: The relation between past and future stock returns. Working paper.

Mullainathan, S., & Spiess, J. (2017). Machine learning: An applied econometric approach. *Journal of Economic Perspectives, 31*(2), 87–106. doi:10.1257/jep.31.2.87.

O'Hara, M. (2014). High-frequency trading and its impact on markets. *Financial Analysts Journal, 70*(3), 18–27.

Panopoulou, E., & Plastira, S. (2014). Fama french factors and us stock return predictability. *Journal of Asset Management, 15*(2), 110–128.

Panopoulou, E., & Vrontos, S. (2015). Hedge fund return predictability; to combine forecasts or combine information? *Journal of Banking & Finance, 56*, 103–122.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Parallel distributed processing: Explorations in the microstructure of cognition, *Learning internal representations by error propagation* (1, pp. 318–362). Cambridge, MA, USA: MIT Press. http://dl.acm.org/citation.cfm?id=104279.104293.

Sadka, R. (2010). Liquidity risk and the cross-section of hedge-fund returns. *Journal of Financial Economics, 98*(1), 54–71. doi:10.1016/j.jfineco.2010.05.001.

Satopaa, V. A., Pemantle, R., & Ungar, L. H. (2016). Modeling probability forecasts via information diversity. *Journal of the American Statistical Association, 111*(516), 1623–1633.

Scholz, H. (2007). Refinements to the sharpe ratio: Comparing alternatives for bear markets. *Journal of Asset Management, 7*(5), 347–357.

Scornet, E., Biau, G., & Vert, J.-P. (2015). Consistency of random forests. *Annals of Statistics, 43*(4), 1716–1741. doi:10.1214/15-AOS1321.

Scott, C. (2014). A conversation with robert litterman. *Journal of Portfolio Management*, 57–63.

Seddon, J. J., & Currie, W. L. (2017). A model for unpacking big data analytics in high-frequency trading. *Journal of Business Research, 70*, 300–307.

Sharpe, W. F. (1994). The sharpe ratio. *Journal of Portfolio Management, 21*(1), 49.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., et al. (2016). Mastering the game of go with deep neural networks and tree search. *Nature, 529*, 484–489.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., et al. (2017). Mastering the game of go without human knowledge. *Nature, 550*, 354–359.

Spreckelsen, C., Mettenheim, H.-J., & Breitner, M. H. (2014). Real-time pricing and hedging of options on currency futures with artificial neural networks. *Journal of Forecasting, 33*(6), 419–432.

Taddy, M., Gardner, M., Chen, L., & Draper, D. (2016). A nonparametric Bayesian analysis of heterogenous treatment effects in digital experimentation. *Journal of Business & Economic Statistics, 34*(4), 661–672.

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

The Economist (2016). A game-changing result. http://www.economist.com/news/science-and-technology/21694883-alphagos-masters-taught-it-game-electrifying-match-shows-what.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 58*(1), 267–288.

Troiano, L., Villa, E. M., & Loia, V. (2018). Replicating a trading strategy by means of LSTM for financial industry applications. *IEEE Transactions on Industrial Informatics, 14*(7), 3226–3234. doi:10.1109/TII.2018.2811377.

Varian, H. R. (2014). Big data: New tricks for econometrics. *Journal of Economic Perspectives, 28*(2), 3–28.

Verikas, A., Gelzinis, A., & Bacauskiene, M. (2011). Mining data with random forests: A survey and results of new tests. *Pattern Recognition, 44*(2), 330–349. doi:10.1016/j.patcog.2010.08.011.

Whaley, R. E. (2000). The investor fear gauge. *Journal of Portfolio Management, 26*(3), 12–17.

Wong, B., & Selvi, Y. (1998). Neural network applications in finance: A review and analysis of literature (1990-1996). *Information & Management, 34*(3), 129–139. doi:10.1016/S0378-7206(98)00050-0.

Zhao, Y., Li, J., & Yu, L. (2017). A deep learning ensemble approach for crude oil price forecasting. *Energy Economics*, 9–16.

Ziegler, A., & Knig, I. R. (2014). Mining data with random forests: current options for real-world applications. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 4*(1), 55–63. doi:10.1002/widm.1114.

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67*(2), 301–320.