Expert Systems With Applications 67 (2017) 126–139

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Forecasting daily stock market return using dimensionality reduction

Xiao Zhong[a], David Enke[b,*]

[a] Department of Engineering Management and Systems Engineering, Missouri University of Science and Technology, 204 Engineering Management, 600 W. 14th Street, Rolla, MO 65409-0370, USA

[b] Laboratory for Investment and Financial Engineering, Department of Engineering Management and Systems Engineering, Missouri University of Science and Technology, 221 Engineering Management, 600 W. 14th Street, Rolla, MO 65409-0370, USA

ARTICLE INFO

Article history:
Received 24 July 2016
Revised 6 September 2016
Accepted 16 September 2016
Available online 21 September 2016

Keywords:
Daily stock return forecasting
Principal component analysis (*PCA*)
Fuzzy robust principal component analysis (*FRPCA*)
Kernel-based principal component analysis (*KPCA*)
Artificial neural networks (*ANNs*)
Trading strategies

ABSTRACT

In financial markets, it is both important and challenging to forecast the daily direction of the stock market return. Among the few studies that focus on predicting daily stock market returns, the data mining procedures utilized are either incomplete or inefficient, especially when a large amount of features are involved. This paper presents a complete and efficient data mining process to forecast the daily direction of the S&P 500 Index ETF (SPY) return based on 60 financial and economic features. Three mature dimensionality reduction techniques, including principal component analysis (*PCA*), fuzzy robust principal component analysis (*FRPCA*), and kernel-based principal component analysis (*KPCA*) are applied to the whole data set to simplify and rearrange the original data structure. Corresponding to different levels of the dimensionality reduction, twelve new data sets are generated from the entire cleaned data using each of the three different dimensionality reduction methods. Artificial neural networks (*ANNs*) are then used with the thirty-six transformed data sets for classification to forecast the daily direction of future market returns. Moreover, the three different dimensionality reduction methods are compared with respect to the natural data set. A group of hypothesis tests are then performed over the classification and simulation results to show that combining the *ANNs* with the *PCA* gives slightly higher classification accuracy than the other two combinations, and that the trading strategies guided by the comprehensive classification mining procedures based on *PCA* and *ANNs* gain significantly higher risk-adjusted profits than the comparison benchmarks, while also being slightly higher than those strategies guided by the forecasts based on the *FRPCA* and *KPCA* models.

1. Introduction and methodology

Analyzing stock market movements is extremely challenging for both investors and researchers. This is mainly due to the stock market essentially being a dynamic, nonlinear, nonstationary, nonparametric, noisy, and chaotic system (Deboeck, 1994; Yaser & Atiya, 1996). In fact, stock markets are affected by many highly interrelated factors. These factors include: 1) economic variables, such as interest rates, exchange rates, monetary growth rates, commodity prices, and general economic conditions; 2) industry specific variables, such as growth rates of industrial production and consumer prices; 3) company specific variables, such as changes in company policies, income statements, and dividend yields; 4) psychological variables of investors, such as investors' expectations and institutional investors' choices; and 5) political variables, such as the occurrence and the release of important political events (Enke & Thawornwong, 2005; Wang, Wang, Zhang, & Guo, 2011). Each of these factors interacts in a very complex manner Yao, Tan, & Poh, 1999. Above all, the efficient market hypothesis states that current stock values reflect all available information in the market at that moment, and that the public cannot make successful trades based on that information, further adding to the difficulty of understanding and predicting stock market movements.

However, it is believed by some researchers that the markets are inefficient, in part due to psychological factors of the various market participants, along with the inability of the markets to immediately respond to newly released information (Jensen, 1978). Financial variables, such as stock prices, stock market index values, and the prices of financial derivatives are therefore thought to be predictable. This allows one to gain a return above the market average by examining information released to the general public, with results that are better than random (Lo & MacKinlay, 1988). For decades, investors and researchers have been attracted to try and make significant profit due to potential market inefficiencies by improving trading strategies based on increasingly accurate forecast of financial variables.

* Corresponding author.
E-mail addresses: xz4y4@mst.edu (X. Zhong), enke@mst.edu (D. Enke).

http://dx.doi.org/10.1016/j.eswa.2016.09.027
0957-4174/

There exist different categorizations among previous stock market forecasting technologies. For instance, given the number of input variables, financial time series forecasting can be classified as either univariate or multivariate analysis. In univariate analysis, only the financial time series itself is considered as the input, while in multivariate analysis the input variables can be a lagged time series, or another type of data, such as a technical, fundamental, or inter-market indicator. With regard to the techniques used to analyze the stock markets, both statistical and artificial intelligence methods have been explored. One group of statistical approaches are based on the autoregressive moving average (*ARMA*), the autoregressive integrated moving average (*ARIMA*), the generalized autoregressive conditional heteroskedastic (*GARCH*) volatility (Franses & Ghijsels, 1999), and the smooth transition autoregressive model (*STAR*) (Sarantis, 2001). These statistical techniques also fall into the category of univariate analysis since they use the financial time series itself, as well as a lagged time series as input variables. Other types of statistical approaches often employed include linear discriminant analysis (*LDA*), quadratic discriminant analysis (*QDA*), linear regression (*LR*), and support vector machines (*SVM*), each of which usually includes multiple input variables. With the assumptions of linearity, stationarity, and normality, most of the statistical analysis methods listed above have been restricted within the area of financial forecasting. On the contrary, artificial intelligence models, such as artificial neural networks (*ANN*s), fuzzy systems, and genetic algorithms are driven by multivariate data with no required assumptions. Many of these methodologies have been applied to forecast financial variables. For instance, see Armano, Marchesi, and Murru (2005), Cao and Tay (2001), Chen, Leung, and Daouk (2003), Chun and Kim (2004), Thawornwong and Enke (2004), Enke and Thawornwong (2005), Hansen and Nelson (2002), Kim and Han (2000), Shen and Loh (2004), Ture and Kurt (2006), Vellido, Lisboa, and Meehan (1999), Wang (2002), and Zhang (2003). A comprehensive review of these studies can be found in Atsalakis and Valavanis (2009) and Vanstone and Finnie (2009). Often, the developed price forecasting and stock market timing systems are used in conjunction with trading rules to develop an intelligent, autonomous, and/or adaptive decision support system. For instance, see Barak, Dahooie, and Tichý (2015), Cervelló-Royo, Guijarro, and Michniuk (2015), Chen and Chen (2016), Chiang, Enke, Wu, and Wang (2016), Chourmouziadis and Chatzoglou (2016), Enke and Mehdiyev (2013), Jaisinghani (2016), Kim and Enke (2016), Monfared and Enke (2014), and Thawornwong, Enke, and Dagli (2001).

With nonlinear, data-driven, and easy-to-generalize characteristics, multivariate analysis through the use of *ANN*s has become a dominant and popular analysis tool in finance and economics. Refenes, Burgess, and Bentz (1997) and Zhang, Patuwo, and Hu (1998) provide a review of using *ANN*s as a forecasting method in different areas of finance and investing, including financial engineering. Although *ANN*s seem to be suited for financial time series forecasting, they have some limitations. Saad, Prokhorov, and Wunsch (1998) question the robustness of *ANN* results. Hussain, Knowles, Lisboa, and El-Deredy (2007) and Lam (2004) also state that it is crucial for the *ANN*s to achieve accurate results with a deliberate selection of the input variables and an optimal combination of the network parameters, including the learning rate, momentum, number of hidden layers, and number of nodes in each layer. Atsalakis and Valavanis (2009), Cao, Leggio, and Schniederjans (2005), and Thawornwong and Enke (2004) demonstrate that designing an *ANN* with the least complexity and the most relevant and influential input variables can improve the efficiency and accuracy of financial time series forecast. As mentioned earlier, stock markets are affected by various factors, many of which are utilized as possible input variables during the development of a stock market forecasting system. Thus, it is necessary to choose the most influential and representative inputs if an *ANN* is expected to produce an efficient and accurate prediction. This type of selection is the main task of dimensionality reduction technology.

Strictly speaking, the dimensionality reduction can be performed in two different ways: either by selecting the most relevant variables from the original data set (usually called as feature selection) or by generating a smaller group of new variables, each being a certain combination of the older input variables. Researchers in Statistics, Computer Science and Applied Mathematics have worked in this field for many years and developed a variety of linear and nonlinear dimensionality reduction techniques. Van der Maaten, Postma, and Van den Herik (2009) present a review and systematic comparison of these techniques. Sorzano, Vargas, and Pascual-Montano (2014) also categorize the plethora of dimension reduction techniques with the mathematical insight behind them.

Principal component analysis (*PCA*) is the most classical and well-known statistical method for extracting important features from high-dimensional data space. This methodology dates back to Pearson (1901), and is based on the idea of defining a new coordinate system or space where the raw data can be expressed in terms of many less variables without a significant loss of information. Nonetheless, there are some concerns that this linear technique cannot adequately handle complex nonlinear data. Therefore, a number of nonlinear techniques, including kernel-based principal component analysis (*KPCA*), have been proposed. *KPCA* is a kernel-based dimensionality reduction method that has a broad application in pattern recognition and machine learning. The *KPCA* method gained more interest after *SVM* was introduced by Vapnik (1998). Van der Maaten, Postma, and Van den Herik (2009) compare *PCA* with twelve front-ranked nonlinear dimensionality reduction techniques, such as Multidimensional Scaling, Isomap, Maximum Variance Unfolding, *KPCA*, Diffusion Maps, Multilayer Autoencoders, Locally Linear Embedding, Laplacian Eigenmaps, Hessian *LLE*, Local Tangent Space Analysis, Locally Linear Coordination, and Manifold Charting by performing each on artificial and natural tasks. The results show that although nonlinear techniques do well on selected artificial data, none of them outperforms the traditional *PCA* using real-world data. However, they also point out that the selection of a proper kernel function is important for the performance of *KPCA*. In general, the model selection in kernel methods, including the specification of relevant parameters, can lead to high computational costs. Consistently, Sorzano et al. (2014) state that among the available dimensionality reduction techniques, *PCA* and its different versions, such as standard *PCA*, robust *PCA*, sparse *PCA*, and *KPCA* are still the preferred techniques given their simplicity and intuitiveness. Moreover, Van der Maaten, Postma, and Van den Herik (2009) demonstrate the four main weaknesses of the popular dimensionality reduction techniques, including: (1) the susceptibility to the curse of dimensionality, (2) the problems in finding the smallest eigenvalues in an eigenproblem, (3) overfitting, and (4) the presence of outliers.

It is known that many well-accepted techniques are sensitive to noisy data, especially outliers in the data. The quality and performance of such techniques can be significantly affected by missing values and outliers, not to mention incorrect data and mismatches that possibly exist in the data collected from different sources. Properly handling outliers can improve the robustness and accuracy of the dimensionality reduction results and help keep any subsequent classifier from spending too much time trying to find an effective solution. Moreover, if the number of outliers is large, the data cannot be normal or symmetric based on the empirical principle of normality, further reducing classification accuracy for some techniques. Thus, in order to perform an efficient and reliable analysis with reasonably accurate results, it is necessary to conduct a careful data preprocessing at the beginning of any

data mining procedure. Yet, data preprocessing can be very time consuming and somewhat tedious depending on the specific cases. It is not unusual to spend 60–90% of the modeling and testing on cleaning and preprocessing the raw data. Atsalakis and Valavanis (2009) summarize that among the studies of stock market forecasting, some researchers simply preprocess the data by using a logarithmic data transformation or standardization of the raw data, while others do not preprocess the data or give any further details about cleaning the data. There are some techniques from other fields aimed to alleviate and solve this issue. For example, robustness theory is developed for solving problems subject to model perturbation or added noise or outliers; and the theory of fuzzy set proposed by Zadeh (1965) can reduce the effect of outliers or noises when applied to data sets with unmodeled characteristics by assigning a fuzzy membership to each input data point such that different input points can make different contributions to the analyzing process. For almost four decades, statisticians have investigated the robust algorithm of principal component analysis. One outstanding idea is proposed by Xu and Yuille (1995). They adapt the statistical physics approach to define an objective function with the consideration of outliers, and then generalize several commonly used *PCA* self-organizing rules into robust versions. They demonstrate that their method can resist outliers very well. However, it is difficult to choose a hard threshold in their approach. Yang and Wang (1999) extend Xu and Yuille's method by defining a fuzzy objective function and using gradient descent optimization. Their robust principal component analysis algorithm, *FRPCA*, only needs to preset one parameter, the fuzziness variable, which determines the influences of outliers on the results. They developed their algorithm in three different ways based on updating the weights of the data points differently and called them *FRPCA1, FRPCA2*, and *FRPCA3*. Luukka (2011) develops a nonlinear version of the *FRPCA3* algorithm and claims that with outlier removal his algorithm brings promising results in the study of medical data sets.

Data mining, or big data analytics, is focused on analyzing large amounts of data efficiently and extracting important, useful, and hidden information from the data by combining various techniques in different areas, such as pattern recognition, decision making, expert systems, knowledge database discovery, artificial intelligence, and statistics. The main types of data mining include classification mining, cluster mining, association rule mining, text mining, and image mining. Zhong (2000, 2004) and Zhong, Ma, Yu, and Zhang (2001) demonstrate classification and cluster mining in more detail. In general, stock market or financial time series forecasting is focused on developing approaches to successfully forecast or predict index values or stock prices so that the investors can gain high profits using well-defined trading strategies according to the forecasting results. Atsalakis and Valavanis (2009) state that the key to successful stock market forecasting is achieving the best results with both the minimum required input data and the least complex stock market model. Therefore, it is natural to connect data mining with stock market forecasting in order to mine historical data from stock markets to help define better trading strategies. Given the technical challenges and significant potential profits, many researchers find it worthwhile to seek a comprehensive data mining procedure that can produce accurate, consistent, and reliable forecasting results with potential profits. Since a stock market index contains numerous individual stocks and reflects the broader market movement rather than movement of any individual stock, forecasting stock market indices has attracted the attention of many researchers. Some of the studies target monthly data. For example, Thawornwong and Enke (2004), Enke and Thawornwong (2005) and Leung, Daouk, and Chen (2000) forecast the S&P 500 index using monthly data, whereas Wang et al. (2011) analyze historical monthly data to predict the Shanghai Composite index. Other researchers have studied daily data. For example, Guresen, Kayakutlu, and Daim (2011) explore daily data of NASDAQ Stock Exchange index, Kara, Boyacioglu, and Baykan (2011) attempt to predict the direction of movement in the daily Istanbul Stock Exchange (*ISE*) National 100 Index, O'Connor and Madden (2006) predict the daily movements in the Dow Jones Industrial Average index, while Zhu, Wang, Xu, and Li (2008) use daily data to forecast *NASDAQ*, DJIA, and STI indices. A few research groups, such as Armano, Marchesi, and Murru (2005), Cao and Tay (2001), and Niaki and Hoseinzade (2013) work on predicting daily movements of the S&P 500 index. Both Thawornwong and Enke (2004) and Leung et al. (2000) conclude that trading strategies guided by classification models generate higher risk-adjusted profits compared to the benchmark buy-and-hold strategy and those strategies directed by level-estimation based forecasts.

In this paper, the daily direction of SPDR S&P 500 ETF (ticker symbol: SPY) is forecasted using a deliberately designed classification mining procedure. This will begin by preprocessing the raw data to deal with missing values, outliers, and mismatched samples. Three versions of *PCA* are applied next to the cleaned and complete data in order to select the most influential and uncorrelated variables for classification. *ANNs* acting as classifiers are then used with the transformed data sets to forecast the direction of future market returns.

The remainder of the paper is organized as follows. The data description and preprocessing will be discussed next in Section 2, while three different dimensionality reduction techniques will be introduced in Section 3. The proposed classifiers will be briefly reviewed in Section 4, and the data analysis and model development will be illustrated in Section 5. The modeling results will be summarized in Section 6, with the simulation process described in Section 7. Concluding remarks are presented in Section 8. The data sources and descriptions are included in the Appendix.

2. Data description and preprocessing

2.1. Data description

The data set utilized for this study involves the daily direction (UP or DOWN) of the closing price of the SPDR S&P 500 ETF (ticker: SPY) as the output, along with 60 financial and economic factors as the potential features. These daily data are collected from 2518 trading days between June 1, 2003 and May 31, 2013. The 60 potential features can be divided into 10 groups, including the SPY return for the current day and three previous days, the relative difference in percentage of the SPY return, exponential moving averages of the SPY return, Treasury bill (T-bill) rates, certificate of deposit rates, financial and economic indicators, the term and default spreads, exchange rates between the USD and four other currencies, the return of seven world major indices (other than the S&P 500), SPY trading volume, and the return of eight large capitalization companies within the S&P 500 (which is a market cap weighted index and driven by larger capitalization companies). Some of these features are being considered for the first time, while others are a mixture of the features conducted by various research groups (Armano et al., 2005; Cao & Tay, 2001; Thawornwong and Enke (2004), Enke and Thawornwong (2005) and Niaki & Hoseinzade, 2013), as long as their values were released without a gap of more than five continuous trading days during the study period. The details of these 60 financial and economic factors, including their descriptions, sources, and calculation formulas are given in Table A1 of the Appendix. After further analysis, only the most important and influential principal components among all the linear combinations of the 60 factors determined using *PCA, FRPCA*, and *KPCA* will be input into the classifiers to predict the direction of the SPY for the next day.

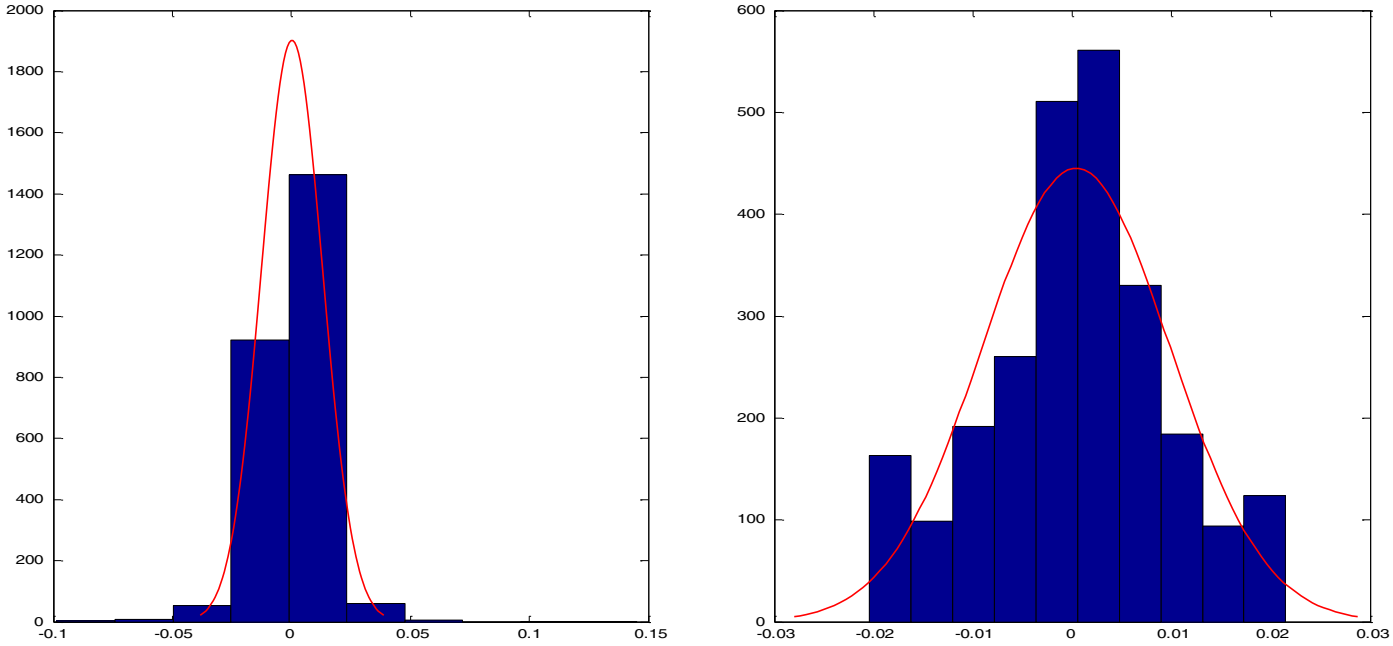

Fig. 1. Histogram of SPY current return (*on the left*); Histogram of adjusted SPY current return (*on the right*).

2.2. Data preprocessing

The data used for this study covers 60 factors over 2518 trading days. As to be expected, for such a large collection of data, there are missing values, mismatching samples, and outliers existing in the raw data. Using the 2518 trading days during the 10-year period as criteria, the collected samples from other days should be deleted. As for the missing values, if there are n values for any variable or column that are missing continuously, the average of the n existing values on both sides of the missing values are used to fill in the n missing values. A simple statistical principle is employed to detect the possible outliers (Navidi, 2011). The possible outliers are then adjusted using a similar method to the one employed by Cao and Tay (2001). Specifically, for each of the 60 factors or columns in the data, any value beyond the interval $(Q_1 - 1.5 * IQR, Q_3 + 1.5 * IQR)$ is regarded as a possible outlier, with the factor value replaced by the boundary of the interval closer to it. Here, Q_1 and Q_3 are the first and third quartile of all the values in that column, and $IQR = Q_3 - Q_1$ is the interquartile of those values. The symmetry of all adjusted and cleaned columns can be checked using histograms or statistical tests. For example, Fig. 1 includes the histograms of factor SPY_t (i.e., the SPY current daily return), before and after data preprocessing. It can be observed that the outliers are removed and the symmetry is achieved after the adjustments.

In this study, the *ANNs* are used as classifiers. At the start of the classification mining procedure, the cleaned data are sequentially partitioned into three parts: training data (the first 70% data), validation data (the last 15% of the first 85% data), and testing data (the last 15% data). The reason for having validation data is to decrease the possibility of overfitting the data, which often happens in *ANN* analysis. Additional details about how the data was used for classification are provided in Section 5.2.

3. Dimensionality reduction using *PCA*, *FRPCA*, and *KPCA*

3.1. PCA

A number of linear or nonlinear techniques have been developed to embed high-dimensional data into a lower dimensional space without much loss of the information. Among them, *PCA* is the most popular unsupervised linear technique for dimensionality reduction. Jolliffe (1986) gives an authoritative and accessible account of this methodology. As one of the earliest multivariate techniques, *PCA* is aimed to construct a low-dimensional representation of the data while keeping the maximal variance and covariance structure of the data. In order to achieve this goal, a linear mapping $\boldsymbol{W}$ that can maximize $\boldsymbol{W}^T var(\boldsymbol{X})\boldsymbol{W}$, where $var(\boldsymbol{X})$ is the variance-covariance matrix of the data $\boldsymbol{X}$, is needed. It is shown that $\boldsymbol{W}$ is formed by the principal eigenvectors of $var(\boldsymbol{X})$. Thus, *PCA* turns out to be an eigenproblem $var(\boldsymbol{X})\boldsymbol{W} = \boldsymbol{\lambda} W$, where $\boldsymbol{\lambda}$ represents the eigenvalues of $var(\boldsymbol{X})$. In addition, it is known that working on the raw data $\boldsymbol{X}$ instead of standardized data with *PCA* tends to give more emphasis to those variables that have higher variances compared to those variables that have very low variances, especially if the units at which the variables are measured are not consistent. In this study, not all variables are measured at the same units. Thus, *PCA* is applied to the standardized version of the cleaned data $\boldsymbol{X}$. In other words, the linear mapping $\boldsymbol{W}*$ is searched such that

$$corr(\boldsymbol{X})\boldsymbol{W}^* = \boldsymbol{\lambda}^*\boldsymbol{W}^*, \tag{1}$$

where $corr(\boldsymbol{X})$ is the correlation matrix of the data $\boldsymbol{X}$.

That is, suppose the data $\boldsymbol{X}$ has the format $\boldsymbol{X} = (\boldsymbol{X_1}\ \boldsymbol{X_2}\ \cdots \boldsymbol{X_M})$, then $corr(\boldsymbol{X}) = \boldsymbol{\rho}$ is a $M \times M$ matrix where M is the dimensionality of the data, and the ij^{th} element of the correlation matrix is

$$corr\left(\boldsymbol{X_i},\ \boldsymbol{X_j}\right) = \rho_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j},$$

where

$$\sigma_{ij} = cov\left(\boldsymbol{X_i},\ \boldsymbol{X_j}\right),\ \sigma_i = \sqrt{var(\boldsymbol{X_i})},\ \sigma_j = \sqrt{var\left(\boldsymbol{X_j}\right)},\ \text{and } i,\ j = 1,\ 2,\ \ldots,\ M. \tag{2}$$

Essentially, the principal components are the linear combinations of all the factors with the coefficients equaling the elements of the eigenvectors, correspondingly. Different amounts of principal components can explain different proportions of the variance-covariance structure of the data. The eigenvalues can be used to rank the eigenvectors based on how much of the

variation of the data is captured by each principal component. In more detail, let $\boldsymbol{\lambda^*} = \{\lambda_i^*\}_{i=1}^M$ denote the eigenvalues of the correlation matrix $corr(\boldsymbol{X})$ such that $\lambda_1^* \geq \lambda_2^* \geq \cdots \geq \lambda_M^*$. Also, let the vectors $\boldsymbol{e_i^T} = (e_{i1}\ e_{i2}\ \cdots\ e_{iM})$ denote the eigenvectors of $corr(\boldsymbol{X})$ corresponding to the eigenvalues λ_i^*, $i = 1,\ 2,\ \ldots,\ M$. It turns out that the elements of these eigenvectors are the coefficients of the principal components. That is, the principal components of the standardized data

$$\boldsymbol{Z} = (\boldsymbol{Z}_1\ \boldsymbol{Z}_2\ \cdots\ \boldsymbol{Z_M}),$$

where

$$\boldsymbol{Z_w^T} = (Z_{1w}\ Z_{2w}\ \cdots\ Z_{Nw}),\ Z_{vw} = \frac{X_{vw} - \mu_w}{\sigma_w},\ v = 1,\ 2,\ \ldots,\ N,\ \text{ and } w = 1,\ 2,\ \ldots,\ M, \quad (3)$$

can be written as

$$\boldsymbol{Y_i} = \sum_{j=1}^{M} e_{ij}\boldsymbol{Z_j},\ i = 1,\ 2,\ \ldots,\ M. \quad (4)$$

Moreover, it is proven that

$$var(\boldsymbol{Y_i}) = \sum_{k=1}^{M}\sum_{l=1}^{M} e_{ik}corr(\boldsymbol{X_k}, \boldsymbol{X_l})e_{il} = \boldsymbol{e_i^T}\boldsymbol{\rho}\boldsymbol{e_i} = \lambda_i^* \quad (5)$$

and

$$cov\left(\boldsymbol{Y_i}, \boldsymbol{Y_j}\right) = \sum_{k=1}^{M}\sum_{l=1}^{M} e_{ik}corr(\boldsymbol{X_k}, \boldsymbol{X_l})e_{jl} = \boldsymbol{e_i^T}\boldsymbol{\rho}\boldsymbol{e_j} = 0 \quad (6)$$

using the Spectral Decomposition Theorem

$$\boldsymbol{\rho} = \sum_{i=1}^{M} \lambda_i^* \boldsymbol{e_i}\boldsymbol{e_i^T} \quad (7)$$

and the fact that both $\boldsymbol{e_i^T}\boldsymbol{e_i} = \sum_{j=1}^{M} e_{ij}^2 = 1$ and the different eigenvectors are perpendicular to each other such that $\boldsymbol{e_i^T}\boldsymbol{e_j} = 0$. Thus, the variance of the ith (largest) principal component is equal to the ith largest eigenvalue, and the principal components are uncorrelated with one another.

Since the total variation of $\boldsymbol{Z}$ is defined as the trace of the correlation matrix $\boldsymbol{\rho}$, that is, $trace(\boldsymbol{\rho}) = \sum_{i=1}^{M} \lambda_i^*$, the proportion of variation explained by the ith principal component is defined to be $\lambda_i^*/trace(\boldsymbol{\rho})$, where $i = 1,\ 2,\ \ldots,\ M$. The proportion of variation explained by the first k principal components is defined to be the sum of the first k eigenvalues divided by $trace(\boldsymbol{\rho})$, that is, $\sum_{i=1}^{k} \lambda_i^* \ / \sum_{i=1}^{M} \lambda_i^*$. Theoretically, if the proportion of variation explained by the first k principal components is large, not much information is lost by reducing the dimensionality of the data space from M to k.

Please note that in general the population variance-covariance matrix $var(\boldsymbol{X})$ is unknown and we may estimate it by the sample variance-covariance matrix $\boldsymbol{S}$ such as

$$\boldsymbol{S} = \frac{1}{N-1}\sum_{v=1}^{N}\left(\boldsymbol{X}_v - \bar{\boldsymbol{X}}\right)\left(\boldsymbol{X}_v - \bar{\boldsymbol{X}}\right)^T. \quad (8)$$

Then estimate the correlation matrix $\boldsymbol{\rho}$ by estimating σ_{ij} as $S(\boldsymbol{X_i}, \boldsymbol{X_j})$, σ_i as $\sqrt{S(\boldsymbol{X_i})}$, and σ_j as $\sqrt{S(\boldsymbol{X_j})}$, where $i,\ j = 1,\ 2,\ \ldots,\ M$. The remaining procedure and the interpretations are the same as described before.

To determine how many and which principal components should be used as inputs to the classifier, it is necessary to find a balance among the expected or required forecasting accuracy, the cost (time and others), and the complexity of the system. That is, the principle components that are chosen must explain the data the best while simplifying the data structure as much as possible. In practice, it is reasonable to consult experts to help determine the proper balance.

3.2. FRPCAs

Given the data $\boldsymbol{X} = \{x_1, x_2, \ldots, x_n\}$, Yang and Wang (1999) propose an optimization function in terms of the data cluster and a noise cluster:

$$RE = \sum_{i=1}^{n}(u_i)^m e(x_i) + \eta\sum_{i=1}^{n}(1-u_i)^m\ , \quad (9)$$

where $u_i \in [0,\ 1]$ and $m \in [1,\ \infty)$. u_i is the membership of x_i belonging to the data cluster and $(1-u_i)$ represents the membership of x_i belonging to the noise cluster. m is the fuzziness variable and the weighting exponent, which determines the influence of small u_i compared to large u_i. $e(x_i)$ is used to measure the error or distance between x_i and the cluster center, and it can be one of the following functions:

$$e_1(x_i) = ||x_i - w^T x_i w||^2, \quad (10)$$

$$e_2(x_i) = ||x_i||^2 - \frac{||w^T x_i||^2}{||w||^2}. \quad (11)$$

This optimization function actually follows the fuzzy clustering approach and essentially calculates the weighted sum of distances between the data and the cluster center, which is equal to 0 in the data set. If $m = 1$ and $u_i \in \{0,\ 1\}$, expression (9) becomes the optimization function proposed by Xu and Yuille (1995).

Since $u_i \in [0,\ 1]$ in expression (9) is continuous, the optimization difficulty caused by the mixture of discrete and continuous variables in Xu and Yuille (1995) is avoided. Using the gradient descent approach, Yang and Wang (1999) derive their robust algorithms of principal component analysis.

FRPCA1 algorithm

Step 1. Initially set the iteration count $t = 1$, iteration bound T, learning coefficient $\alpha_0 \epsilon (0, 1]$, soft threshold η to a small positive value and randomly initialize the weight w. There is no general rule for the setting of m, most papers set $m = 2$ for the reason of simplicity.
Step 2. While t is less than T, do Step 3–9.
Step 3. Compute $\alpha_t = \alpha_0(1 - t/T)$, set $i = 1$ and $\sigma = 0$.
Step 4. While $i < n$, do Step 5–8.
Step 5. Compute $y = w^T x_i\ ,\ u = yw,\ v = w^T u$.
Step 6. Update the weight:

$$w^{new} = w^{old} + \alpha_T \beta(x_i)[y(x_i - u) + (y - v)x_i],$$

where

$$\beta(x_i) = \left(\frac{1}{1 + (e_1(x_i)/\eta)^{1/(m-1)}}\right)^m. \quad (12)$$

Step 7. Update the temporary count: $\sigma = \sigma + e_1(x_i)$.
Step 8. $i = i + 1$.
Step 9. Compute $\eta = \sigma/n$ and $t = t + 1$.

FRPCA2 algorithm

The same as *FRPCA1* except Step 6–7.

Step 6. Update the weight:

$$w^{new} = w^{old} + \alpha_T \beta(x_i)\left(x_i y - \frac{w}{w^T w}y^2\right),$$

where

$$\beta(x_i) = \left(\frac{1}{1+(e_2(x_i)/\eta)^{1/(m-1)}}\right)^m. \quad (13)$$

Step 7. Update the temporary count: $\sigma = \sigma + e_2(x_i)$.

FRPCA3 algorithm

The same as *FRPCA1* except Step 6–7 and $e(x_i)$ below could be set as $e_1(x_i)$ or $e_2(x_i)$.

Step 6. Update the weight:

$$w^{new} = w^{old} + \alpha_T \beta(x_i)\left(x_i y - w y^2\right),$$

where

$$\beta(x_i) = \left(\frac{1}{1+(e(x_i)/\eta)^{1/(m-1)}}\right)^m. \quad (14)$$

Step 7. Update the temporary count: $\sigma = \sigma + e(x_i)$.

The weight updating rule in *FRPCA3* is called the one-unit Oja's algorithm (Oja, 1985). Based on a weight updating rule for classical nonlinear *PCA*, as proposed by Luukka (2011), Oja (1995) developed a nonlinear version of *FRPCA3*:

New Nonlinear FRPCA3 algorithm

The same as *FRPCA3* except Steps 6–7.

Step 6. Calculate $g(y)$, $F = \frac{d}{dy}(g(y))$, $e_3(x_i) = x_i - w^{old} g(y)$, and update the weight:

$$w^{new} = w^{old} + \alpha_T \beta(x_i)\left(x_i e_3(x_i)^T w^{old} F + e_3(x_i) g(y)\right),$$

where

$$\beta(x_i) = \left(\frac{1}{1+(e_3(x_i)/\eta)^{1/(m-1)}}\right)^m$$

and $g(y)$is chosen to be a quite sharp sigmoidal like function

$$g(y) = \tanh(10y). \quad (15)$$

Step 7. Update the temporary count: $\sigma = \sigma + e_3(x_i)$.

3.3. KPCA

KPCA is based on the kernel methods through which the data can almost always be linearly separated and gain independence in a high enough dimensional space if they cannot be linearly separated in lower dimensional space. The transformation from the low dimensional space to the high dimensional space is done by an appropriate kernel function. The kernel function can be linear or nonlinear. If it is linear, then there is no difference between standard *PCA* and *KPCA*. In general, *KPCA* is a reformulation of linear *PCA* in a high-dimensional space constructed using a nonlinear kernel function. This nonlinear extension of *PCA* can improve the quality of dimensionality reduction of the data that have certain types of nonlinearity involved.

Assuming the number of the observations or the instances of the data $\boldsymbol{X}$ is N, instead of directly calculating the eigenvectors of the variance-covariance matrix of the data $\boldsymbol{X}$, *KPCA* first transforms the data $\boldsymbol{X}$ into another high-dimensional space generated by the kernel function κ by computing the kernel matrix $\boldsymbol{K}$ of the data points or vectors $\boldsymbol{x_i}$ such as $k_{ij} = \kappa(\boldsymbol{x_i}, \boldsymbol{x_j})$, where κ is a kernel function; then centers $\boldsymbol{K}$ by the following formula

$$k_{ij} = k_{ij} - \frac{1}{N}\sum_{l=1}^{N} k_{il} - \frac{1}{N}\sum_{l=1}^{N} k_{jl} + \frac{1}{N^2}\sum_{l=1}^{N}\sum_{m=1}^{N} k_{lm}. \quad (16)$$

Now, the N eigenvectors v_i of the centered $N \times N$kernel matrix can be calculated. It is proved that there is a correspondence between $\boldsymbol{v_i}$and the eigenvectors of the covariance matrix of the data transformed into the kernel space $\boldsymbol{\alpha_i}$ via

$$\boldsymbol{\alpha_i} = \frac{1}{\sqrt{\lambda_i}}\boldsymbol{X}\boldsymbol{v_i}. \quad (17)$$

Finally, the lower dimensional representation $\boldsymbol{Y}$ can be obtained by projecting the data $\boldsymbol{X}$ onto the kernel space spanned by $\boldsymbol{\alpha_{i^*}}$, such that $i^* = 1, 2, \ldots, d$, as follows

$$\boldsymbol{y_i} = \left\{\sum_{j=1}^{N}\alpha_1^j k_{ji}, \ \ldots, \sum_{j=1}^{N}\alpha_d^j k_{ji}\right\}$$

where $\alpha_{i^*}^j$ represents the j^{th} value of the eigenvector $\boldsymbol{\alpha_{i^*}}$. Note that $\boldsymbol{Y}$ is a $N \times d$ matrix and $d < N$ is the number of the dimensions to output.

It is obvious that the size of the kernel matrix is proportional to N^2, the square of the number of observations or instances of the data set. This is considered as an important weakness of *KPCA*. In addition, there are many kernel functions that have been developed in the literature, such as simple or linear, polynomial, Gaussian, Cauchy, *ANOVA*, Bayesian, wave, wavelet, Laplacian, and so on. Most of them require a selection of parameters. The influence of different kernels chosen for *KPCA* and various values selected for the relevant parameters on the final results can be significant. However, there is no direct way to make an appropriate selection of the parameters; a large number of experiments are usually performed and tested to identify the best choice. The computational cost can be increased to certain level accordingly. Therefore, choosing an appropriate kernel function with the best parameter setting is critical but difficult for the efficiency of a *KPCA* procedure. The three most commonly used kernels can be expressed as

Linear Kernel:

$$\kappa\left(\boldsymbol{x_i}, \boldsymbol{x_j}\right) = \boldsymbol{x'_i x_j} + c, \quad (18)$$

where c is a constant.

The linear kernel is the simplest kernel function, which is the same as standard PCA theoretically.

Polynomial Kernel:

$$\kappa\left(\boldsymbol{x_i}, \boldsymbol{x_j}\right) = \left(\alpha \boldsymbol{x'_i x_j} + c\right)^d, \quad (19)$$

where α is the slope, c is a constant, and d is the polynomial degree.

The polynomial kernel is a nonstationary kernel. Such kernels are well suited for problems where all the training data are normalized.

Gaussian Kernel:

$$\kappa\left(\boldsymbol{x_i}, \boldsymbol{x_j}\right) = exp\left(-\frac{||\boldsymbol{x_i} - \boldsymbol{x_j}||^2}{2\sigma^2}\right), \quad (20)$$

where σ is the bandwidth.

The Gaussian kernel is an example of radial basis function kernel. The adjustable parameter σ plays a major role in the performance of the Gaussian kernel. The larger the bandwidth, the more linear the function. That is, if σ is overestimated, the exponential will behave almost linearly and the higher-dimensional projection will start to lose its nonlinear power; if σ is underestimated, the Gaussian kernel will lack regularization and the decision boundary will be highly sensitive to noise afterwards. Alternatively, it could also be implemented using

$$\kappa\left(\boldsymbol{x_i}, \boldsymbol{x_j}\right) = exp\left(-\gamma ||\boldsymbol{x_i} - \boldsymbol{x_j}||^2\right), \quad (21)$$

where γ is the parameter to select.

More details about *KPCA* can be found in Schölkopf, Smola, and Müller (1998), Shawe-Taylor and Christianini (2004), and Turk and Pentland (1991).

4. The *ANN* classifiers

Artificial Neural Networks (*ANN*s) were invented to mimic the human brain by carefully defining and designing the network architecture, including the number of network layers, the types

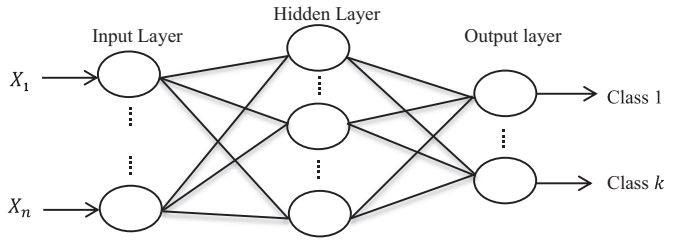

Fig. 2. A three-layer feed-forward neural network used for classification.

of connections among the network layers, the numbers of the neurons in each layer, the learning algorithm, the learning rate, weights between neurons, and the various neuron activation functions. *ANN*s function like a black box that can output prediction or classification results based on the input information.

An efficient *ANN* system usually includes three phases. First, the weights of the connections among the layers and the neurons are adjusted over the training data to achieve a reasonably accurate and reliable prediction or classification result. Second, to avoid overfitting the data and improve the generalization of the *ANN*, validation data are used to determine when the training phase should stop based on an early-stopping rule. Third, the testing data can be input to the trained *ANN* to provide an independent measure of network performance.

There are numerous types of *ANN* that have been explored. They can be categorized according to different aspects, such as the network architecture, the learning algorithm, and the application (Amornwattana, Enke, & Dagli, 2007; Bao & Yang, 2008; Bogullu, Enke, & Dagli, 2002; Chavarnakul & Enke, 2008; Enke, Ratanapan, & Dagli, 2000; Patel, Shah, Thakkar, & Kotecha, 2015; Rather, Agarwal, & Sastry, 2015). Among them, the multi-layer feed-forward *ANN* model with a backpropagation learning algorithm is recognized as one of the most popular financial forecasting tools for its simplicity and efficiency (Vellido, et al., 1999). The configuration of a three-layer feed-forward neural network that is used for classification in this research is given in Fig. 2. In the model, various selected variables, X_1 to X_n, are provided as input to the network. These input variables are selected as discussed in Sections 3 and 4. The network outputs represent the chosen result, which in this study will be the classification of the market direction as either rising or falling over the next period. Neurons in the network will contain a specific activation function, and each neuron will be connected to other neurons in subsequent layers using a weight. The backpropagation learning process will be used to update the individual connection weights to achieve the desired classification accuracy. Further detail of the training, validation, and testing of the *ANN* that is used for classification in this study is provided in Section 5.2.

As mentioned in Section 1, trading strategies guided by classification models usually generate higher risk-adjusted profits (Enke & Thawornwong, 2005; Leung et al., 2000). Therefore, this study will also compare classification-based forecasts through *ANN*s against two defined benchmarks. The specific results can be found in Section 7.

5. Data analysis and model development

5.1. Use PCA, FRPCA, and KPCA to reduce the dimensionality

Background modeling details for the PCA, FRPCA, and KPCA dimensionality reduction techniques are provided in Sections 3.1, 3.2, and 3.3, respectively. The following sections apply each previously described technique to the datasets being tested.

5.1.1. Apply PCA to the data

Using *PCA*, the 60 principal components of the entire data can be generated. The resulting number of principal components that can explain almost 100% ($>99.9999\%$) variation of the data set is 37. More details about the cumulative proportions of variation of the entire data set that can be explained by a different number of principal components corresponding to the data set can be found in Table 1. These principal components are ordered based on their importance or weights. Each principal component corresponds to an eigenvalue of the correlation matrix of the data set. The first principal component listed on the table represents the most influential principal component that is determined by the eigenvector corresponding to the largest eigenvalue of the correlation matrix. This is the same for the subsequent components.

Each principal component is a linear combination of all 60 features. The coefficients or the weights of the 60 features for each principal component imply the level of their importance or influence in the data set. The absolute value of the weight that a feature has will characterize the importance or relevance of that feature to the principal component. Thus, by checking the coefficients of the principal components we can tell which features explain the data better. From Table 1, we observe that the number of principal components that can explain almost 99% variation of the data set is 11. Thus, by observing the coefficients of the first 11 principal components for the data set, a number of conclusions can be drawn. Four groups, including the relative changes in the exchange rates between USD and four other currencies, the SPY return for the current and three previous days, the return of the other seven world major indices, and the return of the eight large market capitalization companies in *S&P* 500 are important. The group of financial and economic indicators are less important, whereas the other groups, such as the relative difference in percentage of the SPY return, exponential moving averages of the SPY return, T-bill rates, certificate of deposit rates, the term and default spreads, and the SPY trading volume have the least impact, and almost zero effect on the first 11 principal components. More specifically, the USD_CNY, i.e., the relative change in the exchange rate between US dollar and Chinese Yuan (Renminbi), is the one feature that plays a much more significant role than the other features in the first principal component of each case. The first principal component can actually explain the majority of the variation for each data set. Table 2 illustrates the coefficients of the first 11 principal components generated from the correlation matrices of the entire data with the size 2518, where the level of importance of each group of features is indicated by different numbers on the leftmost column: 1 = least important; 2 = less important; 3 = important. The *PCA* results are obtained with the MATLAB function *pcacov*, using the methodology described in Section 3.1.

5.1.2. Apply FRPCA to the data

In Section 3.2, four algorithms regarding *FRPCA* are introduced with details. In this paper, the New Nonlinear *FRPCA3* algorithm is applied to the clean and preprocessed data. As with using *PCA* for dimensionality reduction, certain numbers of principal components are chosen and input to the *ANN* classifier for forecasting in Section 5.2.

5.1.3. Apply KPCA to the data

As described in Section 3.3, the polynomial kernel is a nonstationary kernel and is well suited for normalized data. In this paper, the polynomial kernel is used in *KPCA* procedure since the preprocessed data with outlier removal is roughly normal as shown in Fig. 1. Nonetheless, there is no direct way to choose the relevant parameter of the polynomial kernel. To save the computational cost and for the simplicity, we specify $\alpha = 1$, $c = 1$, and $d = 0.5$ in expression (19). The same numbers of principal components, as

Table 1
The results of *PCA* over the entire data.

PCs	Cumulative proportion	PCs	Cumulative proportion	PCs	Cumulative proportion	PCs	Cumulative proportion
1	0.930842	11	0.990272	21	0.998352	31	0.999981
2	0.947948	12	0.991921	22	0.998711	32	0.999987
3	0.961163	13	0.993275	23	0.999035	33	0.99999
4	0.9696	14	0.994235	24	0.999233	34	0.999993
5	0.974644	15	0.995146	25	0.999423	35	0.999996
6	0.978549	16	0.995968	26	0.999604	36	0.999999
7	0.981723	17	0.996543	27	0.999742	37	1
8	0.984181	18	0.997069	28	0.999841	38	1
9	0.986476	19	0.997575	29	0.999933	39	1
10	0.988453	20	0.99797	30	0.999962	40	1

in the cases of *PCA* and *FRPCA*, are selected and used for the forecasting of daily return with the *ANN* classifier in Section 5.2.

5.2. Use ANN to classify the data

The Neural Network toolbox available in MATLAB is used to develop the artificial neural network that is applied to perform the classification in this study. A three-layered feedforward *ANN* structure was used. The network was trained using a scaled conjugate gradient backpropagation algorithm. The number of neurons in the hidden layer was set to 10 based on trial-and-error experience, and for the purpose of comparison. A tangent sigmoid transfer function was selected for the hidden layer. There were two nodes in the output layer representing two classes (UP or DOWN). The output values are actually the probabilities of each input value belonging to the two classes. The larger probability is chosen as the winner. A logistic sigmoid transfer function was used in the output layer. Different numbers of principal components were used as inputs for each cluster and the entire data set.

The Mean Squared Error (MSE) and the confusion matrix were used to evaluate the performance of the *ANN* classifier. MSE is the average squared difference between outputs and targets. Lower values are better. Zero means no error. The confusion matrix consists of four correctness percentages for training, validation, testing, and the total data set that were provided as inputs to the *ANN* classifier. The percent of correctness indicates the fraction of samples that are correctly classified. A value of 0 means no correct classification, whereas 100 indicates maximum correct classifications. In particular, the Neural Network toolbox in MATLAB functions in the following way. The training data are input to train the *ANN* model, and the validation data are input to control the *ANN*'s overfitting problem almost simultaneously. That is, as the *ANN* is trained using the training data, the MSE obtained from classifying the validation data with the trained *ANN* model gets decreased at first and continues to fall for certain amount of time; the MSE of the validation will start to increase when the *ANN* model is having an overfitting problem, resulting in the need for the training phase to be terminated. Thus, the *ANN* model can be trained best in the sense that the validation phase achieves its lowest MSE with the trained model. After the *ANN* is trained and selected, all training data, validation data, and testing data (untouched) are input to and classified by the trained model separately. The percentage of correctly predicted or classified daily directions corresponding to each category can be obtained and recorded.

This study focuses on predicting the daily direction of SPY for next day. The direction can be either UP or DOWN. That is, the output or the response (random) variable has a Bernoulli distribution. In addition, for each selected dimensionality reduction technique, twelve new data sets can be generated by transforming the original cleaned and preprocessed data based on the different number of principal components chosen. In other words, the twelve data sets are a reflection of the original 60-dimensional data in twelve data spaces with various dimensions lower than 60. To show the influence of dimensionality reduction with *PCA, FRPCA*, and *KPCA* on the daily direction classification, *ANN*s are applied to each of the thirty-six transformed data sets. The results are listed in Table 3.

6. Results

The performance of the *ANN* classifier is measured with the rate or percentage of times correctly predicting the direction of the SPY for the next day. Table 3 includes four sections. The leftmost section lists twelve values; each of these values represents the number of principal components based on which one of the twelve new data sets with respect to each of the three dimensionality reduction methods is generated. Moreover, each of the twelve numbers is selected from Table 1 according to the cumulative proportion of variation of the entire data that can be explained by this specific number of principal components. Each row of the other three sections of Table 3 contains classification rates measured for each training, validation, testing, and total data set considered in this study based on *PCA, FRPCA*, and *KPCA*. Each combination of the four rates is chosen from the training results. The criteria of the selection include: all four classification rates are among the highest rates in each of the four categories; all four rates are close to each other as much as possible with the paired difference less than or around 5%.

The rate or percentage of correctness for the testing phase is considered the most important measure to determine the prediction accuracy of the *ANN*s. In order to make a comparison regarding the prediction accuracy among the combining procedures of the *ANN*s and each of the three different dimensionality reduction techniques, a group of paired *t*-tests are performed over the population means of the correctness rates or percentages for all classification models considered in this study. The *P*-values are used as the criteria to draw a conclusion. The hypothesis testing results are given in Table 4.

Assuming the significance level is 0.05 for any hypothesis test, we reject the null hypothesis if the *P*-value is less than 0.05 and favor the alternative hypothesis if the *P*-value is greater than 0.05; the smaller the *P*-value, the more favorable the alternative hypothesis. Therefore, from Table 4, we can conclude that the three *PCA*s do not give significantly different results in average. However, based on the *P*-values, it is fair to say that the standard *PCA* performs slightly better than *FRPCA*, and *FRPCA* performs slightly better than *KPCA* in average. This is consistent with the results demonstrated by Van der Maaten, Postma, and Van den Herik (2009).

In addition, for each version of *PCA* involved, the number of the principal components used as the inputs does not have much impact on the prediction accuracy for the *ANN*s. For example, when standard *PCA* is considered, even if using only the first principal component as the input, the (testing) prediction accuracy for *ANN*s is 56.8% compared to the highest percentage 58.1%, which

Table 2
The allocation of the coefficients of the first 11 *PCA*s from the entire data with size 2518.

Level	Group	Factors	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11
3	SPY return in current and three previous days	SPYt	−0.0019	−0.0402	−0.2349	0.2763	0.0646	0.1976	0.0032	−0.2156	0.2969	−0.4925	0.6027
		SPYt1	−0.0047	0.4022	−0.0010	0.0546	−0.1247	−0.0095	0.0075	−0.0013	−0.2143	−0.0876	0.0500
		SPYt2	−0.0014	0.0096	0.0048	−0.0391	0.6157	−0.0160	−0.0468	0.5000	−0.2365	−0.0956	0.0701
		SPYt3	0.0007	−0.0138	0.0052	0.0203	−0.0551	−0.0087	0.9841	0.0792	0.0739	0.0275	−0.0118
1	Relative difference in percentage of SPY return	RDP5	0.0000	0.0008	−0.0005	0.0007	0.0010	0.0003	0.0018	0.0008	−0.0001	−0.0014	0.0014
		RDP10	0.0000	0.0004	−0.0003	0.0003	0.0006	0.0001	0.0009	0.0004	0.0000	−0.0008	0.0006
		RDP15	0.0000	0.0003	−0.0002	0.0003	0.0004	0.0001	0.0006	0.0003	0.0000	−0.0005	0.0005
		RDP20	0.0000	0.0002	−0.0002	0.0002	0.0003	0.0001	0.0005	0.0003	0.0000	−0.0004	0.0003
1	Exponential moving averages of SPY return	EMA10	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
		EMA20	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
		EMA50	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
		EMA200	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
1	T-bill rates	T1	0.0000	0.0000	0.0000	0.0000	0.0001	0.0000	0.0000	0.0000	−0.0001	0.0001	0.0001
		T3	0.0000	0.0000	0.0000	0.0000	0.0001	0.0000	0.0000	0.0000	−0.0001	0.0001	0.0001
		T6	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	−0.0001	0.0001	0.0001
		T60	0.0000	0.0000	−0.0001	0.0000	0.0001	0.0000	0.0000	0.0000	−0.0002	0.0001	0.0001
		T120	0.0000	0.0000	−0.0001	0.0000	0.0001	0.0000	0.0000	−0.0001	−0.0002	0.0001	0.0001
1	Certificate of deposit rates	CD1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	−0.0001	−0.0001	−0.0001	0.0002	0.0000
		CD3	0.0000	−0.0001	0.0000	0.0000	0.0000	0.0000	−0.0001	−0.0001	−0.0001	0.0002	0.0000
		CD6	0.0000	−0.0001	0.0000	0.0000	0.0000	0.0000	−−0.0001	−0.0001	−0.0001	0.0002	0.0000
2	Financial and economical indicators	oil	−0.0014	0.0055	−0.0865	0.0417	−0.0172	0.0396	−0.0086	0.0117	−0.0429	0.0615	0.0812
		gold	−0.0019	0.0151	−0.2005	−0.1119	−−0.0065	0.0667	0.0069	−0.0256	−0.1725	0.7209	0.6107
		CTB3M	0.0014	0.0008	−0.0022	0.0555	0.0136	0.0008	0.0074	−0.0060	−0.0126	−0.0831	0.1065
		CTB6M	0.0018	−0.0055	0.0007	0.0854	−0.0040	−0.0109	0.0080	−0.0103	−0.0150	−0.0919	0.1217
		CTB1Y	0.0010	−0.0099	0.0042	0.0974	−0.0028	−0.0086	0.0064	−0.0015	−0.0054	−0.0795	0.1075
		CTB5Y	−0.0002	−0.0054	−0.0077	0.0499	0.0009	0.0030	−0.0060	−0.0031	0.0006	−0.0318	0.0272
		CTB10Y	−0.0003	−0.0044	−0.0104	0.0473	0.0012	0.0035	−0.0062	−0.0037	0.0009	−0.0306	0.0248
		AAA	−0.0002	−0.0052	−0.0120	0.0418	−0.0009	0.0017	−0.0059	−0.0049	−0.0043	−0.0252	0.0159
		BAA	−0.0003	−0.0068	−0.0109	0.0447	−0.0036	0.0054	−0.0136	−0.0056	−0.0031	−0.0263	0.0186
1	The term and default spreads	TE1	0.0000	0.0000	0.0000	0.0000	−0.0001	0.0001	0.0000	−0.0001	0.0002	−0.0001	−0.0001
		TE2	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0002	−0.0001	−0.0001
		TE3	0.0000	0.0001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0002	−0.0002	−0.0001
		TE5	0.0003	−0.0001	0.0012	0.0013	0.0001	0.0022	−0.0011	−0.0030	−0.0011	−0.0007	−0.0003
		TE6	0.0001	−0.0004	0.0007	0.0005	−0.0005	0.0012	−0.0015	−0.0016	0.0000	0.0007	−0.0008
		DE1	0.0000	−0.0001	0.0002	−0.0003	−0.0003	0.0002	−0.0002	−0.0001	0.0005	0.0001	−0.0006
		DE2	0.0000	−0.0001	0.0002	−0.0003	−0.0003	0.0002	−0.0002	−0.0001	0.0005	0.0001	−0.0006
		DE4	0.0000	0.0000	0.0000	−0.0001	0.0000	0.0000	0.0000	0.0000	0.0002	−0.0001	−0.0001
		DE5	0.0000	0.0000	0.0000	−0.0001	−0.0001	0.0000	0.0000	0.0000	0.0002	−−0.0001	−0.0001
		DE6	0.0000	0.0000	0.0000	−0.0001	−0.0001	0.0001	0.0000	0.0000	0.0002	−0.0001	−0.0001
		DE7	−0.0002	−0.0007	0.0003	−0.0010	−0.0008	0.0005	−0.0016	−0.0007	0.0002	0.0023	−0.0021
3	Exchange rate between USD and four other currencies	USD_Y	0.0013	−0.1147	0.0669	0.8829	0.0444	−0.3448	−0.0196	0.0323	−0.0938	0.2267	−0.0731
		USD_GBP	0.0027	−0.0441	0.6888	0.1819	0.0071	0.6861	0.0037	0.0213	−0.0762	0.0855	0.0669
		USD_CAD	−0.0002	−0.0016	0.6439	−0.2181	−0.0208	−0.6016	−0.0048	−0.1016	0.0885	−0.0858	0.3690
		USD_CNY	***0.9999***	0.0117	−0.0024	−0.0003	0.0016	−0.0010	−0.0005	0.0005	0.0005	0.0001	0.0022
3	The return of the other seven world major indices	HSI	−0.0021	0.0827	−0.0070	−0.0131	0.3530	−0.0308	0.0866	0.2539	−0.0986	−0.1207	0.0871
		SSE Composite	−0.0008	0.0282	−0.0051	−0.0186	0.1616	−0.0191	0.0723	0.1428	−0.0973	−0.1113	0.1487
		FCHI	−0.0029	0.2522	0.0253	0.0114	0.2961	0.0033	0.0140	−0.2532	0.1875	0.1137	−0.0767
		FTSE	−0.0036	0.3070	0.0334	0.0187	0.3919	0.0153	0.0162	−0.3262	0.2417	0.1489	−0.1197
		GDAXI	−0.0033	0.2562	0.0199	0.0214	0.2888	0.0054	0.0422	−0.2443	0.1863	0.1238	-0.0762
		DJI	−0.0043	0.4239	0.0064	0.0624	−0.1451	−0.0045	0.0211	0.0406	−0.1777	−0.0700	0.0352
		IXIC	−0.0032	0.3070	−0.0031	0.0483	−0.0858	−0.0170	0.0086	−0.0122	−0.2347	−0.0671	0.0417
1	SPY trading volume	V	0.0000	−0.0002	0.0015	−0.0024	0.0011	−0.0016	0.0001	0.0015	−0.0007	0.0016	−0.0057
3	The return of the eight big companies in S&P 500	AAPL	−0.0010	0.1002	−0.0082	0.0134	−0.0383	−0.0125	0.0085	0.0155	−0.1364	−0.0233	0.0090
		MSFT	−0.0023	0.2092	0.0023	0.0310	−0.0886	−0.0114	−0.0027	−0.0081	−0.1133	−0.0314	0.0306
		XOM	−0.0044	0.2196	0.0053	0.0192	−0.0972	−0.0027	0.0083	0.0069	−0.1035	−0.0368	−0.0091
		GE	−0.0027	0.2210	−0.0002	0.0295	−0.0714	0.0175	0.0055	0.0178	−0.0931	−0.0637	0.0015
		JNJ	−0.0039	0.3129	0.0261	0.0684	−0.2218	0.0095	−0.1121	0.6026	0.6577	0.1486	0.0502
		WFC	−0.0011	0.1722	0.0079	0.0224	-0.1032	0.0101	0.0024	−0.0504	−0.1042	−0.0649	0.0050
		AMZN	−0.0012	0.1034	0.0016	0.0130	−0.0330	0.0006	−0.0023	0.0106	−0.0955	−0.0218	0.0396
		JPM	−0.0013	0.1624	0.0009	0.0228	−0.0811	−0.0052	0.0025	−0.0329	−0.1119	−0.0703	0.0065

Table 3
The *ANN* classification results of the 36 transformed data sets based on three *PCAs*.

PCs	PCA				FPCA				KPCA			
	Training	Validation	Testing	Total	Training	Validation	Testing	Total	Training	Validation	Testing	Total
1	54.8	53.6	56.8	54.9	54.8	53.3	57	54.9	55.3	53.3	57	55.2
3	55.2	53.3	57.3	55.2	55.2	53.8	56.8	55.2	55.8	53.6	57	55.6
6	54.9	53.6	57.3	55	57.1	53.6	57	56.6	55.6	53.3	57	55.5
10	56.4	54.6	57.3	56.3	57.1	56.5	56.8	57	56.7	54.6	58.1	56.6
15	56.3	53.3	57.6	56	55.3	55.4	57.8	55.7	56	54.9	57.6	56
22	55.2	54.6	58.1	55.5	56.2	54.9	57.8	56.2	56.6	56	57.8	56.7
26	55.1	53.1	58.1	55.2	56.8	56.5	58.6	57	55.4	54.1	57.8	55.6
31	57.5	57.3	58.1	57.5	56.2	54.4	59.2	56.4	55.7	54.1	57.3	55.7
34	56.2	56	57.3	56.4	56	53.8	58.1	56	55.5	54.4	56.8	55.5
37	55	54.4	57	55.2	56.3	54.1	57.8	56.2	55.7	53.1	57.3	57.6
40	56.2	56.2	56.2	56.2	56	54.1	57.8	56	55.8	59.2	57.6	56.6
60	57.5	54.1	58.1	57.1	56.5	54.4	57.3	56.3	57.4	54.9	58.4	57.1

Table 4
The paired *t*-test results used for the comparison of different classification models with respect to the *PCAs*.

Null hypothesis	Alternative hypothesis	*P*-value
$\mu_{PCA} = \mu_{FRPCA}$	$\mu_{PCA} \neq \mu_{FRPCA}$	0.2989
$\mu_{PCA} = \mu_{KPCA}$	$\mu_{PCA} \neq \mu_{KPCA}$	0.8163
$\mu_{FRPCA} = \mu_{KPCA}$	$\mu_{FRPCA} \neq \mu_{KPCA}$	0.4727
$\mu_{PCA} = \mu_{FRPCA}$	$\mu_{PCA} > \mu_{FRPCA}$	0.8505
$\mu_{PCA} = \mu_{KPCA}$	$\mu_{PCA} > \mu_{KPCA}$	0.5918
$\mu_{FRPCA} = \mu_{KPCA}$	$\mu_{FRPCA} > \mu_{KPCA}$	0.2363

can be achieved in much higher dimensional data spaces. This phenomenon may be interpreted by considering Table 1. From Column 2 of Table 1 we see that the first (and the largest) principal component of the entire cleaned data set can explain 93.08% variation of the data, such that there is not much space left for improvement for the remaining 59 smaller principal components.

7. Trading simulation

After using the *ANNs* to predict the daily SPY direction, it is natural to carry out a trading simulation to see if the higher predictability implies higher profitability. Given that this research study is based on predicting the direction of S&P 500 ETF (SPY) daily returns, we modified the trading strategy for classification models defined by Enke and Thawornwong (2005) as follows:

If $UP_{t+1} = 1$, fully invest in stocks or maintain, and receive the actual stock return for the day $t+1$ (i.e., SPY_{t+1}); if $UP_{t+1} = 0$, fully invest in one-month T-bills or maintain, and receive the actual one-month T-bill return for the day $t+1$ (i.e., $T1H_{t+1}$).

Here UP is the direction of the SPY daily return as predicted by the models described in this paper. The actual one-month T-bill return for the day $t+1$ is:

$$T1H_{t+1} = \frac{discount\ rate}{100} * \frac{term}{360\ days} = \frac{T1_{t+1}}{100} * \frac{28\ days}{360\ days} = \frac{T1_{t+1}}{100} * \frac{7}{90}, \quad (22)$$

where $T1_{t+1}$ is the one-month T-bill discount rate (or risk-free rate) in percentage on the secondary market for business day $t+1$.

Specifically, at the beginning of each trading day, the investor decides to buy the SPY portfolio or the one-month T-bill according to the forecasted direction of the SPY daily return. For simplicity, it is assumed in this paper that the money invested in either a stock portfolio or T-bills is illiquid and detained in each asset during the entire trading day. Dividends and transaction costs are also not considered. Moreover, both leveraging and short selling when investing are forbidden. The two benchmarks used to measure how well the models can perform include investing in a stock portfolio (i.e., buy-and-hold) and purchasing a one-month T-bill at the start of the testing period, and closing the trading at the end of the testing period. The trading simulation is done for all the classification models over each testing period, including 376 samples (excluding the first day of the 377-day testing period because of the lack of direction prediction for that day) of the thirty-six transformed data sets corresponding to the number of principal components involved. The resulting mean, standard deviation or volatility, and Sharpe ratio of the daily returns on investment generated from each forecasting model over each testing data are then calculated. The results are presented in Table 5. In addition to the trading simulation results of the three models for each of the twelve principals components, the 376-day return for both the buy-and-hold and T-bill benchmarks are provided for comparison.

As shown in Table 5, the return from the buy-and-hold benchmark is much higher than one-month T-bill benchmark. By multiplying the mean of the daily return column by 376 and then comparing with the two benchmarks, this comparison indicates that: for all thirty-six transformed data sets, the trading strategies based on the classification models generate higher returns than the one-month T-bill benchmark; the trading strategies based on the *ANNs* combining *PCA* generate higher returns than the buy-and-hold benchmark except for three data sets (PCs = 3, 22, and 31) where the returns are slightly less than the buy-and-hold benchmark; the returns from the trading strategies based on the *ANNs* combining *FRPCA* generate higher returns than the buy-and-hold benchmark except for four data sets (PCs = 1, 3, 10, and 60); and the returns from the trading strategies based on the *ANNs* combining *KPCA* generate higher returns than the buy-and-hold benchmark except for six data sets (PCs = 1, 3, 6, 26, 31, and 34). Six paired *t*-tests are carried out to make a comparison of the mean of daily return from three different model combinations. The results are given in Table 6.

Since all the *P*-values are greater than 0.05, there is no significant difference among the mean of daily returns generated by the models involving *PCA, FRPCA*, and *KPCA* given the thirty-six transformed natural data sets. However, with more careful observation of the *P*-values listed in Table 6, it seems that on average *PCA* performs slightly better than *FRPCA* and *KPCA*, while *FRPCA* performs slightly better than *KPCA*.

The Sharpe ratio is calculated by dividing the mean daily return by the standard deviation of the daily returns. The higher the Sharpe ratio, the higher the return and the lower the standard deviation or volatility, the better the trading strategy. Therefore, another six paired *t*-tests over the Sharpe ratio are performed to

Table 5
Trading simulation results.

Benchmarks	376-Day return								
Buy-and-hold	3.08E-01								
T-bill	3.89E-04								
PCs	Models	Mean of daily return	Std. of daily return	Sharpe ratio	PCs	Models	Mean of daily return	Std. of daily return	Sharpe ratio
1	PCA	8.40E-04	0.0079	0.1011	26	PCA	8.24E-04	0.0077	0.1069
	FRPCA	7.93E-04	0.0079	0.1006		FRPCA	8.81E-04	0.0077	0.1149
	KPCA	7.93E-04	0.0079	0.1006		KPCA	7.52E-04	0.0075	0.1008
3	PCA	7.97E-04	0.0079	0.1012	31	PCA	8.02E-04	0.0077	0.1036
	FRPCA	7.88E-04	0.0079	0.1		FRPCA	8.56E-04	0.0078	0.1097
	KPCA	7.93E-04	0.0079	0.1006		KPCA	7.95E-04	0.0078	0.1019
6	PCA	8.47E-04	0.0078	0.1086	34	PCA	8.61E-04	0.007	0.1235
	FRPCA	9.75E-04	0.0069	0.141		FRPCA	9.00E-04	0.0077	0.1173
	KPCA	7.93E-04	0.0079	0.1006		KPCA	7.83E-04	0.0079	0.0994
10	PCA	8.37E-04	0.0077	0.1084	37	PCA	8.41E-04	0.0074	0.1134
	FRPCA	8.00E-04	0.0073	0.1099		FRPCA	8.89E-04	0.0077	0.1152
	KPCA	9.04E-04	0.0077	0.118		KPCA	8.27E-04	0.0078	0.1055
15	PCA	8.21E-04	0.0079	0.1045	40	PCA	9.61E-04	0.0071	0.1357
	FRPCA	8.53E-04	0.0077	0.1111		FRPCA	8.97E-04	0.0078	0.1157
	KPCA	8.78E-04	0.0073	0.1196		KPCA	0.001	0.007	0.1478
22	PCA	8.07E-04	0.0076	0.1067	60	PCA	9.21E-04	0.0073	0.1264
	FRPCA	9.59E-04	0.0076	0.1269		FRPCA	8.00E-04	0.0079	0.1016
	KPCA	8.63E-04	0.0077	0.1122		KPCA	8.88E-04	0.0077	0.1177

Table 6
The paired *t*-test results used for the comparison of different models with respect to mean of daily return.

Null hypothesis	Alternative hypothesis	*P*-value
$\mu_{PCA} = \mu_{FRPCA}$	$\mu_{PCA} \neq \mu_{FRPCA}$	0.4139
$\mu_{PCA} = \mu_{KPCA}$	$\mu_{PCA} \neq \mu_{KPCA}$	0.6256
$\mu_{FRPCA} = \mu_{KPCA}$	$\mu_{FRPCA} \neq \mu_{KPCA}$	0.3538
$\mu_{PCA} = \mu_{FRPCA}$	$\mu_{PCA} > \mu_{FRPCA}$	0.7931
$\mu_{PCA} = \mu_{KPCA}$	$\mu_{PCA} > \mu_{KPCA}$	0.3128
$\mu_{FRPCA} = \mu_{KPCA}$	$\mu_{FRPCA} > \mu_{KPCA}$	0.1769

Table 7
The paired *t*-test results used for the comparison of different classification models with respect to Sharpe ratio.

Null hypothesis	Alternative hypothesis	*P*-value
$\mu_{PCA} = \mu_{FRPCA}$	$\mu_{PCA} \neq \mu_{FRPCA}$	0.6633
$\mu_{PCA} = \mu_{KPCA}$	$\mu_{PCA} \neq \mu_{KPCA}$	0.6924
$\mu_{FRPCA} = \mu_{KPCA}$	$\mu_{FRPCA} \neq \mu_{KPCA}$	0.5561
$\mu_{PCA} = \mu_{FRPCA}$	$\mu_{PCA} > \mu_{FRPCA}$	0.6684
$\mu_{PCA} = \mu_{KPCA}$	$\mu_{PCA} > \mu_{KPCA}$	0.3462
$\mu_{FRPCA} = \mu_{KPCA}$	$\mu_{FRPCA} > \mu_{KPCA}$	0.2781

compare the three dimensionality reduction technique-relevant forecasting models. The results are listed in Table 7.

The same pattern of *P*-values can be detected from Table 7 as Table 6. Thus, there is no significant difference among the trading strategies generated from three different model combinations, although it seems that *PCA*-relevant strategies perform insignificantly better than the other two, and FRPCA-relevant strategies perform a bit better than the *KPCA* case. This is consistent with the conclusion drawn from Table 6. That is, among the three dimensionality reduction methods, the classification model that is based on *PCA* gives slightly better trading strategy performance with respect to mean of daily return and Sharpe ratio over the thirty-six transformed data sets.

In order to make a statistically meaningful comparison between the returns from the *ANN-PCA*-based trading strategies and the return from the buy-and-hold benchmark, a *t*-test is conducted at the significance level of 0.05. Dividing the 376-day return from buy-and-hold benchmark by 376 gives 0.000 819. Therefore, we define the test as

$$\begin{cases} H_0: \ \mu_{ANN} = 0.000\ 819; \\ H_a: \ \mu_{ANN} > 0.000\ 819. \end{cases}$$

Under H_0, the value of the *t*-test statistic T is $\frac{\bar{x}_{ANN} - 0.000\ 819}{s_{ANN}/\sqrt{12}}$, where $\bar{x}_{ANN}$ is the sample average of the *ANN* mean daily returns from the twelve testing data sets or testing periods and s_{ANN} is the sample standard deviation of the twelve mean daily returns, which equals 1.9586. Then, the *P*-value $= P(T > 1.9586)$, where $T \sim t_{11}$ given H_0 is true. Using the *t*-table or online distribution calculator, the *P*-value = 0.038. Since the *P*-value is smaller than 0.05, we reject the null hypothesis and conclude that the *ANN-PCA*-based trading strategies generate significantly higher (mean) daily return than the benchmark buy-and-hold passive trading strategy.

8. Conclusion

For this research a comprehensive and efficient daily direction of the stock market return forecasting process is presented. The process starts with data cleaning and data preprocessing, and concludes with an analysis of forecasting and simulation results. Often, researchers look to apply the simplest set of algorithms to the least amount of data with both the most accurate forecasting results and the highest risk-adjusted profits. To achieve this goal, three dimensionality reduction techniques, including *PCA*, *FRPCA*, and *KPCA* are introduced and applied to the natural data set involving 60 financial and economic features before the *ANN* classification procedure.

In summary, the mining process using the *ANN-PCA* models gives slightly higher prediction accuracy for the daily direction of SPY for next day compared to the mining process involving *FRPCA* and *KPCA*. Moreover, the trading strategies based on the *ANN-PCA* models gain significantly higher risk-adjusted profits than the comparison benchmarks, and slightly higher than those strategies guided by the forecasts based on *FRPCA* and *KPCA*-relevant models. All classification models-based trading strategies generate higher returns than the benchmark one-month T-bill strategy. As developed, tested, and discussed, analysis has shown that data collection and preprocessing is critical and can help improve the performance of many techniques, such as *PCA* and *ANN*, while decreasing the complexity of the mining procedure and achieving reasonable accuracy and high risk-adjusted profits.

In this study, a natural data set is collected and analyzed. The *ANN* classifiers combing *PCA* are recognized as the simplest, but relatively more accurate procedure. The trading strategies based on this procedure generate slightly higher risk-adjusted profits than the ones based on combing the *ANN*s with either *FRPCA* or *KPCA*. Nonetheless, the selection of a proper kernel function is important for the performance of *KPCA*. In the future, a more delicate selection of the kernel functions and the relevant kernel parameters are suggested.

Appendix

Table A1
The 60 financial and economic features of the raw data.

Group	Name	Description	Source/Calculation
	Date_SPY	trading dates considered	finance.yahoo.com
	Close_SPY	closing prices of SPY on the trading days	finance.yahoo.com
SPY return in current and three previous days	SPYt	The return of the SPDR S&P 500 ETF (SPY) in day t.	finance.yahoo.com / (p(t) - p(t-1))/p(t-1)
	SPYt1	The return of the SPY in day t-1.	finance.yahoo.com / (p(t-1) - p(t-2))/p(t-2)
	SPYt2	The return of the SPY in day t-2.	finance.yahoo.com / (p(t-2) - p(t-3))/p(t-3)
	SPYt3	The return of the SPY in day t-3.	finance.yahoo.com / (p(t-3) - p(t-4))/p(t-4)
Relative difference in percentage of the SPY return	RDP5	The 5-day relative difference in percentage of the SPY.	(p(t) - p(t-5))/p(t-5) * 100
	RDP10	The 10-day relative difference in percentage of the SPY.	(p(t) - p(t-10))/p(t-10) * 100
	RDP15	The 15-day relative difference in percentage of the SPY.	(p(t) - p(t-15))/p(t-15) * 100
	RDP20	The 20-day relative difference in percentage of the SPY.	(p(t) - p(t-20))/p(t-20) * 100
Exponential moving averages of the SPY return	EMA10	The 10-day exponential moving average of the SPY.	p(t)*(2/(10+1))+EMA10 (t-1) *(1-2/(10+1))
	EMA20	The 20-day exponential moving average of the SPY.	p(t)*(2/(20+1))+EMA20 (t-1) *(1-2/(20+1))
	EMA50	The 50-day exponential moving average of the SPY.	p(t)*(2/(50+1))+EMA50 (t-1) *(1-2/(50+1))
	EMA200	The 200-day exponential moving average of the SPY.	p(t)*(2/(200+1))+EMA200 (t-1) *(1-2/(200+1))
T-bill rates (day t)	T1	1-month T-bill rate (in percentage), secondary market, business days, discount basis.	H. 15 Release - Federal Reserve Board of Governors (https://research.stlouisfed.org/fred2/series/DGS5/downloaddata)
	T3	3-month T-bill rate, secondary market, business days, discount basis.	H. 15 Release - Federal Reserve Board of Governors (https://research.stlouisfed.org/fred2/series/DGS5/downloaddata)
	T6	6-month T-bill rate, secondary market, business days, discount basis.	H. 15 Release - Federal Reserve Board of Governors (https://research.stlouisfed.org/fred2/series/DGS5/downloaddata)
	T60	5-year T-bill constant maturity rate, secondary market, business days.	H. 15 Release - Federal Reserve Board of Governors (https://research.stlouisfed.org/fred2/series/DGS5/downloaddata)
	T120	10-year T-bill constant maturity rate, secondary market, business days.	H. 15 Release - Federal Reserve Board of Governors (https://research.stlouisfed.org/fred2/series/DGS10?catbc=1&utm_expid=19978471-Srcl7QpGidAURO4vg_Q.1&utm_referrer=https%3A%2F%2Fresearch.stlouisfed.org%2Ffred2%2Frelease%3Frid%3D18)
Certificate of deposit rates (day t)	CD1	Average rate on 1-month neogtiable certificates of deposit (secondary market), quoted on an investment basis.	H. 15 Release - Federal Reserve Board of Governors
	CD3	Average rate on 3-month neogtiable certificates of deposit (secondary market), quoted on an investment basis.	H. 15 Release - Federal Reserve Board of Governors
	CD6	Average rate on 6-month neogtiable certificates of deposit (secondary market), quoted on an investment basis.	H. 15 Release - Federal Reserve Board of Governors
Financial and economical indicators (day t)	Oil	Relative change in the price of the crude oil (Cushing, OK WTI Spot Price FOB (dollars per barrel)).	Energy Inormation Administration, http://tonto.eia.doe.gov/dnav/pet/hist/rwtcd.htm (work on cleaning the price column first using the SPY dates as control, then calculate the relative change)
	Gold	Relative change in the gold price	usagold.com (use Firefox to Select All, then copy and paste to an Excel file) (the dates used by USAGOLD are not matching with the SPY prices from yahoo.finance. For example, after 06/09/2004. We still clean/estimate/delete the gold prices based on the dates of SPY prices from finance.yahoo.com. Use the same procedure in the whole data set: Take the average of the two closest data with the missing one in the middle. Then delete the mismatching one, and calculate the relatvie difference as before. Another example, the data in 2011, all Friday's prices were recorded as Sunday's prices, so we estimated Friday's prices with the average of Thursday and Sunday's prices. Then deleted Sunday's prices. If there are *n* continuous values missing, then take the average of the *n* available values on each side of these *n* missing values, use the average for all *n* missing values)
	CTB3M	Change in the market yield on US Treasury securities at 3-month constant maturity, quoted on investment basis.	H. 15 Release - Federal Reserve Board of Governors

(*continued on next page*)

Table A1 (*continued*)

Group	Name	Description	Source/Calculation
	CTB6M	Change in the market yield on US Treasury securities at 6-month constant maturity, quoted on investment basis.	H. 15 Release - Federal Reserve Board of Governors
	CTB1Y	Change in the market yield on US Treasury securities at 1-year constant maturity, quoted on investment basis.	H. 15 Release - Federal Reserve Board of Governors
	CTB5Y	Change in the market yield on US Treasury securities at 5-year constant maturity, quoted on investment basis.	H. 15 Release - Federal Reserve Board of Governors
	CTB10Y	Change in the market yield on US Treasury securities at 10-year constant maturity, quoted on investment basis.	H. 15 Release - Federal Reserve Board of Governors
	AAA	Change in the Moody's yield on seasoned corporate bonds - all industries, Aaa.	H. 15 Release - Federal Reserve Board of Governors
	BAA	Change in the Moody's yield on seasoned corporate bonds - all industries, Baa.	H. 15 Release - Federal Reserve Board of Governors
The term and default spreads	TE1	Term spread between T120 and T1.	TE1 = T120 - T1
	TE2	Term spread between T120 and T3.	TE2 = T120 - T3
	TE3	Term spread between T120 and T6.	TE3 = T120 - T6
	TE5	Term spread between T3 and T1.	TE5 = T3 - T1
	TE6	Term spread between T6 and T1.	TE6 = T6 - T1
	DE1	Default spread between BAA and AAA.	DE1 = BAA - AAA
	DE2	Default spread between BAA and T120.	DE2 = BAA - T120
	DE4	Default spread between BAA and T6.	DE4 = BAA - T6
	DE5	Default spread between BAA and T3.	DE5 = BAA - T3
	DE6	Default spread between BAA and T1.	DE6 = BAA - T1
	DE7	Default spread between CD6 and T6.	DE7 = CD6 - T6
Exchange rate between USD and four other currencies (day t)	USD_Y	Relative change in the exchange rate between US dollar and Japanese yen.	http://www.investing.com/currencies/usd-jpy-historical-data
	USD_GBP	Relative change in the exchange rate between US dollar and British pound.	http://www.investing.com/currencies/gbp-usd-historical-data (then, take the opposites to the changes)
	USD_CAD	Relative change in the exchange rate between US dollar and Canadian dollar.	http://www.investing.com/currencies/usd-cad-historical-data
	USD_CNY	Relative change in the exchange rate between US dollar and Chinese Yuan (Renminbi).	http://www.investing.com/currencies/usd-cny-historical-data
The return of the other seven world major indices (day t)	HSI	Hang Seng index return.	finance.yahoo.com
	SSE Composite	Shang Hai Stock Exchange Composite index return.	finance.yahoo.com
	FCHI	CAC 40 index return.	finance.yahoo.com
	FTSE	FTSE 100 index return.	finance.yahoo.com
	GDAXI	DAX index return.	finance.yahoo.com
	DJI	Dow Jones Industrial Average index return.	finance.yahoo.com (no download function for this one); measuringworth.com/datasets/DJA/result.php
	IXIC	NASDAQ Composite index return.	finance.yahoo.com
SPY trading volume (day t)	V	Relative change in the trading volume of S&P 500 index (SPY)	finance.yahoo.com
The return of the eight big companies in S&P 500 (day t)	AAPL	Apple Inc stock return.	finance.yahoo.com
	MSFT	Microsoft stock return.	finance.yahoo.com
	XOM	Exxon Mobil stock return.	finance.yahoo.com
	GE	General Electric stock return.	finance.yahoo.com
	JNJ	Johnson and Johnson stock return.	finance.yahoo.com
	WFC	Wells Fargo stock return.	finance.yahoo.com
	AMZN	Amazon.com Inc stock return.	finance.yahoo.com
	JPM	JPMorgan Chase & Co stock return.	finance.yahoo.com

References

Amornwattana, S., Enke, D., & Dagli, C. (2007). A hybrid options pricing model using a neural network for estimating volatility. *International Journal of General Systems, 36*, 558–573.

Armano, G., Marchesi, M., & Murru, A. (2005). A hybrid genetic-neural architecture for stock indexes forecasting. *Information Sciences, 170*(1), 3–33.

Atsalakis, G. S., & Valavanis, K. P. (2009). Surveying stock market forecasting techniques – Part II: Soft computing methods. *Expert Systems with Applications, 36*(3), 5941–5950.

Bao, D., & Yang, Z. (2008). Intelligent stock trading system by turning point confirming and probabilistic reasoning. *Expert Systems with Applications, 34*, 620–627.

Barak, S., Dahooie, J. H., & Tichý, T. (2015). Wrapper ANFIS-ICA method to do stock market timing and feature selection on the basis of Japanese Candlestick. *Expert Systems with Applications, 42*(23), 9221–9235.

Bogullu, V. K., Enke, D., & Dagli, C. (2002). Using neural networks and technical Indicators for generating stock trading signals. *Intelligent Engineering Systems through Artificial Neural Networks, 12*, 721–726.

Cao, Q., Leggio, K. B., & Schniederjans, M. J. (2005). A comparison between Fama and French's model and artificial neural networks in predicting the Chinese stock market. *Computers & Operations Research, 32*(10), 2499–2512.

Cao, L., & Tay, F. (2001). Financial forecasting using vector machines. *Neural Computing and Applications, 10*, 184–192.

Cervelló-Royo, R., Guijarro, F., & Michniuk, K. (2015). Stock market trading rule based on pattern recognition and technical analysis: Forecasting the DJIA index with intraday data. *Expert Systems with Applications, 42*(14), 5963–5975.

Chavarnakul, T., & Enke, D. (2008). Intelligent technical analysis based equivolume charting for stock trading using neural networks. *Expert Systems with Applications, 34*, 1004–1017.

Chen, T. L., & Chen, F. Y. (2016). An intelligent pattern recognition model for supporting investment decisions in stock market. *Information Sciences, 346*, 261–274.

Chen, A. S., Leung, M. T., & Daouk, H. (2003). Application of neural networks to an emerging financial market: Forecasting and trading the Taiwan stock index. *Computers and Operations Research, 30*(6), 901–923.

Chiang, W. C., Enke, D., Wu, T., & Wang, R. (2016). An adaptive stock index trading decision support system. *Expert Systems with Applications, 59*, 195–207.

Chourmouziadis, K., & Chatzoglou, P. D. (2016). An intelligent short term stock trading fuzzy system for assisting investors in portfolio management. *Expert Systems with Applications, 43*, 298–311.
Chun, S. H., & Kim, S. H. (2004). Data mining for financial prediction and trading: Application to single and multiple markets. *Expert Systems with Applications, 26*(2), 131–139.
Deboeck, G. J. (1994). *Trading on the edge: Neural, genetic, and fuzzy systems for chaotic financial markets*. New York: Wiley.
Enke, D., Ratanapan, K., & Dagli, C. (2000). Large machine-part family formation utilizing a parallell ART1 neural network. *Journal of Intelligent Manufacturing, 11*(6), 591–604.
Enke, D., & Mehdiyev, N. (2013). Stock market prediction using a combination of stepwise regression analysis, differential evolution-based fuzzy clustering, and a fuzzy inference neural network. *Intelligent Automation & Soft Computing, 29*(4), 636–649.
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.
Franses, P. H., & Ghijsels, H. (1999). Additive outliers, GARCH and forecasting volatility. *International Journal of Forecasting, 15*(1), 1–9.
Guresen, E., Kayakutlu, G., & Daim, T. U. (2011). Using artificial neural network models in stock market index prediction. *Expert Systems with Applications, 38*(8), 10389–10397.
Hansen, J. V., & Nelson, R. D. (2002). Data mining of time series using stacked generalizers. *Neurocomputing, 43*(1-4), 173–184.
Hussain, A. J., Knowles, A., Lisboa, P. J. G., & El-Deredy, W. (2007). Financial time series prediction using polynomial pipelined neural networks. *Expert Systems with Applications, 35*(3), 1186–1199.
Jaisinghani, D. (2016). An empirical test of calendar anomalies for the Indian securities markets. *South Asian Journal of Global Business Research, 5*(1), 53–84.
Jensen, M. (1978). Some anomalous evidence regarding market efficiency. *Journal of Financial Economics, 6*(2/3), 95–101.
Jolliffe, T. (1986). *Principal component analysis*. New York: Springer-Verlag.
Kara, Y., Boyacioglu, M. A., & Baykan, O. K. (2011). Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul stock exchange. *Expert Systems with Applications, 38*(5), 5311–5319.
Kim, Y., & Enke, D. (2016). Developing a rule change trading system for the futures market using rough set analysis. *Expert Systems with Applications, 59*, 165–173.
Kim, K. J., & Han, I. (2000). Genetic algorithms approach to feature discretization in artificial neural networks for the predication of stock price index. *Expert Systems with Applications, 19*(2), 125–132.
Lam, M. (2004). Neural network techniques for financial performance prediction: Integrating fundamental and technical analysis. *Decision Support Systems, 37*(4), 567–581.
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.
Lo, A. W., & MacKinlay, A. C. (1988). Stock market prices do not follow random walks: Evidence from a simple specification test. *Review of Financial Studies, 1*(1), 41–66.
Luukka, P. (2011). A new nonlinear fuzzy robust PCA algorithm and similarity classifier in classification of medical data sets. *International Journal of Fuzzy Systems, 13*(3), 153–162.
Monfared, S. A., & Enke, D. (2014). Volatility forecasting using a hybrid GJR-GARCH neural network model. *Procedia Computer Science, 36*, 246–253.
Navidi, W. (2011). *Statistics for engineers and scientists* (3rd ed.). New York: McGraw-Hill.
Niaki, S. T. A., & Hoseinzade, S. (2013). Forecasting S&P 500 index using artificial neural networks and design of experiments. *Journal of Industrial Engineering International, 9*(1), 1–9.
O'Connor, N., & Madden, M. G. (2006). A neural network approach to predicting stock exchange movements using external factors. *Knowledge-Based Systems, 19*(5), 371–378.
Oja, E. (1995). *The nonlinear PCA learning rule and signal separation – mathematical analysis*. Helsinki University of Technology *Technical Report*.
Oja, E., & Karhunen, J. (1985). On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix. *Journal of Mathematical Analysis and Applications, 106*, 69–84.
Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock market index using fusion of machine learning techniques. *Expert Systems with Applications, 42*, 2162–2172.
Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine, 2*(6), 559–572.
Rather, A. M., Agarwal, A., & Sastry, V. N. (2015). Recurrent neural network and a hybrid model for prediction of stock returns. *Expert Systems with Applications, 42*, 3234–3241.
Refenes, A. P. N., Burgess, A. N., & Bentz, Y. (1997). Neural networks in financial engineering: A study in methodology. *IEEE Transactions on Neural Networks, 8*(6), 1222–1267.
Saad, E. W., Prokhorov, D. V., & Wunsch, D. C. (1998). Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks. *IEEE Transactions on Neural Networks and Learning Systems, 9*(6), 1456–1470.
Sarantis, N. (2001). Nonlinearities, cyclical behavior and predictability in stock markets: International evidence. *International Journal of Forecasting, 17*(3), 459–482.
Schölkopf, B., Smola, A., & Müller, K. R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computing, 10*(5), 1299–1319.
Shawe-Taylor, J., & Christianini, N. (2004). *Kernel methods for pattern analysis*. New York: Cambridge University Press.
Shen, L., & Loh, H. T. (2004). Applying rough sets to market timing decisions. *Decision Support Systems, 37*(4), 583–597.
Sorzano, C. O. S., Vargas, J., & Pascual-Montano, A. (2014). A survey of dimensionality reduction techniques. *Cornell University Library Abstracts* (pp. 1–35).
Thawornwong, S., & Enke, D. (2004). The adaptive selection of financial and economic variables for use with artificial neural networks. *Neurocomputing, 56*, 205–232.
Thawornwong, S., Enke, D., & Dagli, C. (2001). Using neural networks and technical analysis indicators for predicting stock trends. *Intelligent Engineering Systems through Artificial Neural Networks, 11*, 05–232.
Ture, M., & Kurt, I. (2006). Comparison of four different time series methods to forecast hepatitis A virus infection. *Expert Systems with Applications, 31*(1), 41–46.
Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience, 3*(1), 71–86.
Van Der Maaten, L., Postma, E., & Van den Herik, J. (2009). Dimensionality reduction: A comparative. *Journal of Machine Learning Research, 10*(1-41), 66–71.
Vanstone, B., & Finnie, G. (2009). An empirical methodology for developing stock market trading systems using artificial neural networks. *Expert Systems with Applications, 36*(3), 6668–6680.
Vapnik, V. (1998). *Statistical learning theory*. New York: Wiley.
Vellido, A., Lisboa, P. J. G., & Meehan, K. (1999). Segmentation of the on-line shopping market using neural networks. *Expert Systems with Applications, 17*(4), 303–314.
Wang, J. Z., Wang, J. J., Zhang, Z. G., & Guo, S. P. (2011). Forecasting stock indices with back propagation neural network. *Expert Systems with Applications, 38*(11), 14346–14355.
Wang, Y. F. (2002). Predicting stock price using fuzzy grey prediction system. *Expert Systems with Applications, 22*(1), 33–39.
Xu, L., & Yuille, A. L. (1995). Robust principal component analysis by self-organizing rules based on statistical physics approach. *IEEE Transactions on Neural Networks, 6*(1), 131–143.
Yang, T. N., & Wang, S. D. (1999). Robust algorithms for principal component analysis. *Pattern Recognition Letters, 20*, 927–933.
Yao, J., Tan, L. C., & Poh, H. (1999). Neural networks for technical analysis: A study on KLCI. *International Journal of Theoretical and Applied Finance, 2*(2), 221–241.
Yaser, S., & Atiya, A. F. (1996). Introduction to financial forecasting. *Applied Intelligence, 6*(3), 205–213.
Zadeh, L. A. (1965). Fuzzy sets. *Information and Control, 8*, 338–353.
Zhang, G. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing, 50*, 159–175.
Zhang, G., Patuwo, B. E., & Hu, M. Y. (1998). Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting, 14*(1), 35–62.
Zhong, X. (2000). *Classification and cluster mining*. Zhejiang University.
Zhong, X. (2004). *A study of several statistical methods for classification with application to microbial source tracking Master thesis*. Worcester Polytechnic Institute.
Zhong, X., Ma, S. P., Yu, R. Z., & Zhang, B. (2001). Data mining: A survey. *Pattern Recognition and Artificial Intelligence, 14*(1), 48–55.
Zhu, X. T., Wang, H., Xu, L., & Li, H. Z. (2008). Predicting stock index increments by neural networks: The role of trading volume under different horizons. *Expert Systems with Applications, 34*(4), 3043–3054.