Use Of Neural Network Ensembles for Portfolio Selection and Risk Management

D.L.Toulson S.P.Toulson†

Intelligent Financial Systems Ltd.,
8 Orsett Terrace, London, W2 6AZ
Email:ifsys@ifsys.demon.co.uk

†London School Of Economics,Houghton Street, London,WC2A 2AE

NeuroCOLT Technical Report Series

NC-TR-96-046

May 1996[1]

Produced as part of the ESPRIT Working Group
in Neural and Computational Learning,
NeuroCOLT 8556

NeuroCOLT Coordinating Partner

Royal Holloway
University of London

Department of Computer Science
Egham, Surrey TW20 0EX, England

For more information contact John Shawe-Taylor at the above address
or email neurocolt@dcs.rhbnc.ac.uk

[1] Received May 2, 1996

Abstract

A well known method of managing the risk whilst maximising the return of a portfolio is through Markowitz Analysis [11] of the efficient set. A key pre-requisite for this technique is the accurate estimation of the future expected returns and risks (variance of returns) of the securities contained in the portfolio along with their expected correlations. The estimates for future returns are typically obtained using weighted averages of historical returns [19] of the securities involved or other (linear) techniques. Estimates for the volatilities of the securities may be made in the same way or through the use of (G)ARCH [5] [3] or stochastic volatility (SV) [7] techniques.

In this paper we propose the use of neural networks to estimate future returns and risks of securities. The networks are arranged into *committees* [6]. Each committee contains a number of independently trained neural networks. The task of each committee is to estimate either the future return or risk of a particular security. The inputs to the networks of the committee make use of a novel discriminant analysis technique we have called *Fuzzy Discriminants Analysis.*

The estimates of future returns and risks provided by the committees are then used to manage a portfolio of 40 UK equities over a five year period (1989-1994). The management of the portfolio is constrained such that at any time it should have the same risk characteristic as the FTSE-100 index. Within this constraint, the portfolio is chosen to provide the maximum possible return. We show that the managed portfolio significantly outperforms the FTSE-100 index in terms of both overall return and volatility.

1 Introduction

In this paper we present a forecasting and trading methodology for financial markets. Figure 1 shows a general overview of the system. The raw data from the markets is pre-processed for outliers and other errors and then stored to form a database of historical time series. Prediction models for the future returns and volatilities of selected securities are formulated using transformed features of the price histories and other economic data from the database. Combinations of these predictors are then used as inputs to trading models. A variety of different prediction and trading models are available to the system. This allows a comparison of the relative performance of different models in terms of both predictive ability and profitability to be made. A more detailed description of each aspect of the system follows.

1.1 Pre-processing

The raw economic data available to the system will be provided from services like Datastream, Reuters or Telerate. The data is filtered and pre-processed using methods appropriate for the types of time series examined. It is then stored in the historical database and made available to subsequent feature extraction and feature transformation processes. The following types of pre-processing are used.

Most time series are subject to outliers. These outliers may be rejected using methods from robust statistics. Irregularly spaced data (i.e. tick by tick

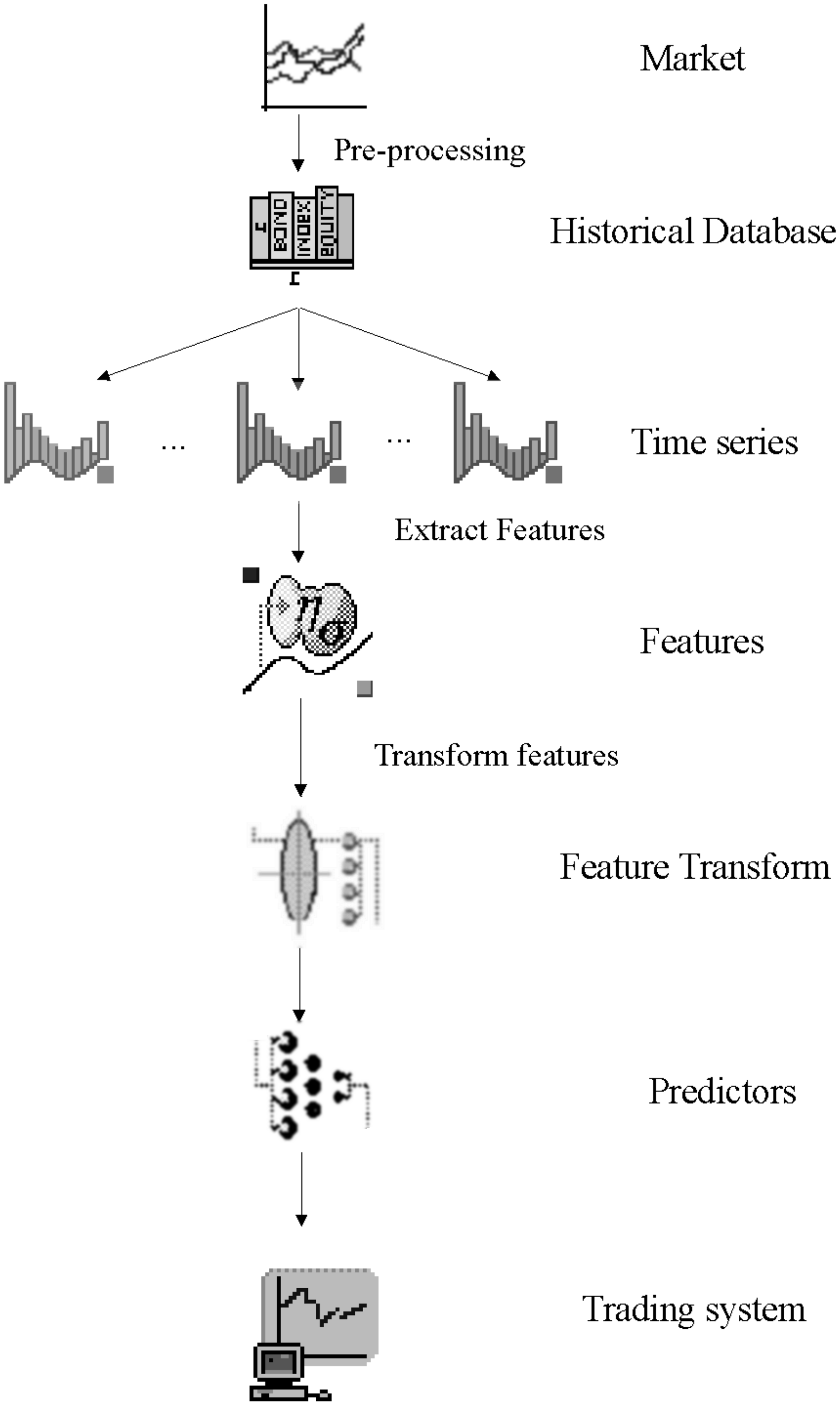

Figure 1 Overview of the management process

FX/FX) may cause problems for some types of prediction models. In such cases the raw data can be transformed into a time series sampled at standardised intervals [13]. Bond and equity data should be adjusted for the effects of dividends and coupons.

1.2 Feature Extraction

The pre-processed data is made available to a number of feature extraction modules within the system. Each feature extractor collects information available up to time t about a particular time series (t is the time from which we wish to make a forward prediction). The feature extractor then produces some representation of this information. As an example, one possible feature could consist of the first n lagged first differences of the time series of interest. Another possibility would be a feature consisting of the moments of these lagged first differences.

1.3 Feature Transforms

The features extracted in the previous section will hopefully include sufficient information for better than random predictions to be made about the time series to be predicted. However, it is likely that a good deal of redundant information will be included in the features. This redundant information makes the process of prediction more difficult. It is useful to be able to reduce the dimension of the input information through a transform that will maintain the significant information whilst eliminating much of the redundant information. We discuss in some detail methods for achieving this type of transformation in Section 3.

1.4 Prediction

The task of the prediction modules is to take in a number of feature vectors and to make a prediction for a particular security. In practice this will be either the expected mean or the expected volatility of the future return. The types of predictors available to the system include neural networks (multi-layer perceptron [20]), k-nearest neighbour models, ARIMA models [14], (G)ARCH and stochastic volatility models.

1.5 Prediction Combination

It is possible to use a single predictor to obtain estimates of the expected return and volatility of each of the securities in the portfolio. However, it has been shown by a number of authors [9] that a linear combination of a number of predictors whose parameters have been estimated in independent manners often give superior results. We use weighted combinations of predictors (committees) whose weightings are found by OLS optimisation on the training data.

1.6 Trading Model

Once predictions about the future returns have been obtained for all of the securities the final task is to convert these predictions into an actual set of trading recommendations. A number of different trading models may be applied.

A *speculative trading model* is a trading system working on a single time series. Its task is to recommend a *long*, *short* or *flat* position at any time t based on information available up to that time.

An *option pricing model* takes the estimates of return and volatility about a security and then prices options on it using a Black-Scholes [17] type model.

A *portfolio management model* manages the allocation of securities in a portfolio. The management process usually involves maximising the return of the portfolio whilst keeping the risk of the portfolio below a certain level.

2 Portfolio Management of 40 UK Equities

We shall now illustrate the steps involved in the implementation of a particular instance of the general system by managing a portfolio of 40 UK equities.

Firstly, we shall examine how a novel feature transformation may be used to reduce the dimension of the feature vectors. We show how the use of this transform leads to a significant improvement in the performance of the prediction models.

We then examine the relative performances of a number of possible prediction models (using the feature transformed data as input). We show that a committee of neural networks is an efficient prediction model able to forecast both future returns and volatilities of securites.

Finally, we examine two possible trading models that may be used to manage a portfolio after the prediction networks have been trained. We show that both trading models are able to significantly outperform the return of the FTSE-100 index over a five year test period. In the case of the second trading model, we are also able to show that the volatility of return (risk) of the managed portfolio can be controlled and was found to be marginally less than the volatility of the FTSE-100 index itself, whilst still significantly outperforming the index in terms of return.

2.1 Data

The financial data used in this paper was composed of 11 years of daily closing prices for 40 UK equities obtained from Datastream (30 December 1983 - 30 December 1994). The equities chosen needed to pass two requirements. Firstly, they needed to be members of the FTSE-100 index and secondly, they needed to be quoted over the full 11 year period. The 40 securities were chosen at random from the qualifying securities and ordered according to their betas. Along with the securities, daily closing prices for the FTSE-100 index were obtained for the same 11 year period.

The data was separated into three periods. The first period was used to train the neural networks and to optimize other predictors (*training set*). The second period was set aside for the validation or out of sample testing of the models (*validation set*). The third period was used to asses the general performance of the system on unseen data (*test set*). The data in the test set was used only AFTER all of the prediction and trading models had been trained and optimized. All results given in this paper will be quoted in terms of this test set which consists of the five year period 30 December 1989 - 30 December 1994.

2.2 Pre-processing and Feature Extraction

The Datasteam data contained only a few outliers which were rejected using a seven standard deviation threshold applied to the returns of the equities. Care was taken not to edit out the crash of 1987. For the purposes of input to the predictors, the outliers were replaced by simple interpolation. However, the days that these prices fell on were marked as untradable for that security. In terms of managing a portfolio this meant that the holding of that particular security could not be changed on that day.

Two features were extracted from each timeseries. The first feature was obtained by concatenating the past 60 normalized daily returns. The second feature contained estimates of the first four moments of the daily returns for each time series. These estimates of the mean, variance, skewness and kurtosis of daily returns were calculated using a range of between the last 10 and 120 lagged observations.

The input to each predictor consisted of these two features calculated for both the security whose return was being estimated and the FTSE-100 index. The untransformed input vectors to the predictors were thus 128 dimensional.

3 Feature Transformation

As mentioned in the previous section, the input vector to each predictor is 128 dimensional. This is high compared to the number of samples (1498) we have available from which we must construct example vectors to train the neural networks and other prediction models.

In this section we discuss methods of reducing the dimension of the input data whilst retaining as much of the original **relevant** information content as possible.

3.1 Principal Components Analysis

A number of authors have suggested Principal Components Analysis (PCA) [15] as a method of reducing the dimension of the input vectors for financial and other pattern recognition applications. PCA works by finding a subspace of the original input space that preserves the maximum information content (variance) in the original data when it is projected from the original space onto the subspace. The projection of the original data onto the subspace is then used as the input to the predictors. It is thus possible to reduce the input vector from N dimensions to M dimensions, $M \leq N$.

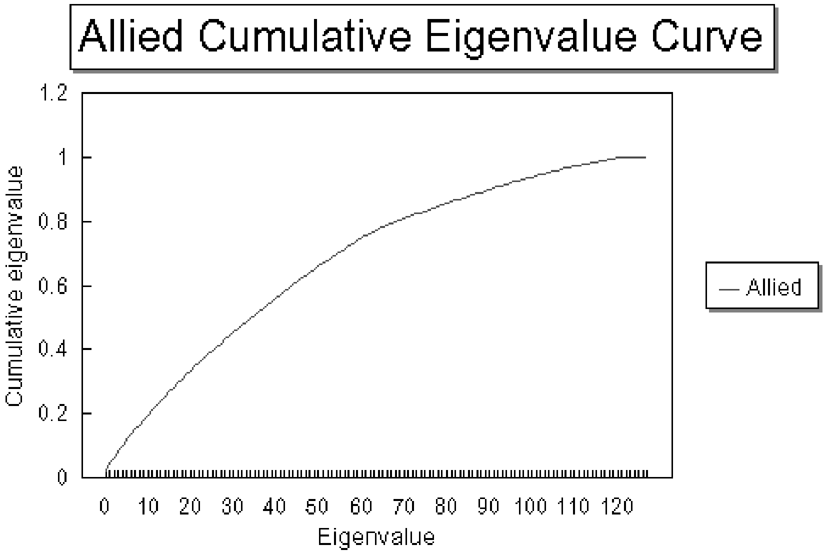

Figure 2 Cumulative eigenvalue curve for the security Allied

The first requirement for the application of the PCA transform is an analysis of the eigenvectors and eigenvalues of the correlation (or covariance) matrix formed from the original input data. Let the eigenvalues of the covariance matrix be denoted $\lambda_i, i = 1, ..., N$ and arranged such that they are monotonically decreasing i.e. $\lambda_i \geq \lambda_{i+1}$. The cumulative eigenvalue curve $c(x)$ is defined to be

$$c(x) = \frac{\sum_{i=1}^{x} \lambda_i}{\sum_{i=1}^{N} \lambda_i}. \tag{1}$$

The interpretation of this curve is that the value $c(x)$ represents the amount of *information* maintained in the input vectors if we project them onto the subspace spanned by the top x eigenvectors. A feature transformation that, for instance, retains 95 percent of the original information (variance) of the input data can be obtained by selecting the appropriate value for x. An example of a real cumulative eigenvalue curve obtained for the original 128 dimensional input vectors for the security *Allied* is shown in Figure 2.

Care must be taken when using Principal Components Analysis as a method of dimension reduction. In particular,

- ◇ The scale and variance of the input components should be similar. This may be achieved either by using the correlation matrix (rather than the covariance matrix) or other techniques, such as Benzecri Normalisation [2].
- ◇ In using PCA to reduce the dimension of the data we are equating *relevant information* in the input vectors with variance. As we shall see in the next section this is not necessarily what we desire.

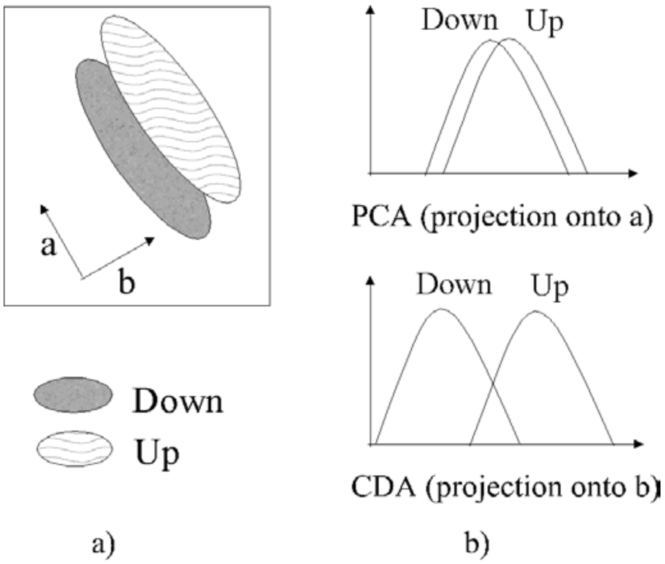

Figure 3 Simple two class problem demonstrating the potential weakness of PCA

3.2 Canonical Discriminants Analysis

Consider the distribution of samples from a simple two-dimensional two-class problem shown in Figure 3 (a). The first principal component of the data is shown as vector $\vec{a}$. If we now consider reducing the dimension of the two dimensional input vectors to one dimension using Principal Components Analysis, the projection of the input samples onto the first principal component is shown in Figure 3 (b). Clearly we have chosen the worst possible subspace onto which to project the original input vectors as the two classes are now completely inseparable! With a little thought, one can see that the actual vector onto which we would wish to project the data to maintain the information necessary to distinguish examples from the two classes is the vector labelled $\vec{b}$. The properties of this vector are that it

1. Minimises the *within class variance* (W) of the projection of the input vectors.

2. Maximises the *between class mean separation* (B) of the projection of the input vectors.

A transform that is formulated to achieve the above goals by maximising the ratio of mean separation to variance is the Canonical Discrimanant Analysis transform [24]. This transform is essentially the generalisation of Fisher's linear discriminant function to multiple dimensions [4]. The canonical discriminants may be found in the following manner.

Let us consider a sample of data $X = \{\vec{x}_i,\ i = 1, ..., T\}$ of size T. Each $N \times 1$ sample vector $\vec{x}$ is either a member of class 1 ($\vec{x} \in \mathcal{H}_1$) or class 2 ($\vec{x} \in \mathcal{H}_2$). For the case of our data we can assume that the two classes refer to samples of either positive or negative future returns for the security being examined. Zero returns can either be excluded or assumed to be positive.

Let us define the *Within Class Scatter Matrix* S_W as

$$S_W = \sum_{c=1}^{2} \left(\sum_{\vec{x} \in \mathcal{H}_c} (\vec{x} - \vec{m}_c)\, (\vec{x} - \vec{m}_c)^T \right) \tag{2}$$

where $\vec{m}_c$ is the mean for class c, defined

$$\vec{m}_c = \frac{1}{n_c} \sum_{\vec{x} \in \mathcal{H}_c} \vec{x}, \tag{3}$$

and n_c is the number of samples belonging to class c.

Let us also define the *Between Class Scatter Matrix* S_B as

$$S_B = \sum_{c=1}^{2} n_c\, (\vec{m}_c - \vec{m})\, (\vec{m}_c - \vec{m})^T, \tag{4}$$

where $\vec{m}$ is the mean of all of the samples (i.e. samples from both classes).

Now consider the projections of the original N dimensional sample vectors $\vec{x}$ onto a smaller M dimensional subspace. We can represent this projection as a linear transformation

$$\vec{y} = A^T \vec{x}, \tag{5}$$

where A is an $N \times M$ dimensional matrix whose rows are the axes of the subspace onto which the original vectors, $\vec{x}$ are being projected.

Let P_W denote the within class scatter matrix and P_B denote the between class scatter matrix of the projected samples. The within class and between class scatter matrices for the projected vectors can be written in terms of the transformation matrix A

$$P_W = A^T S_W A, \tag{6}$$

$$P_B = A^T S_B A. \tag{7}$$

Recalling the discussion earlier in this section, we seek a transformation matrix A that will maximise the ratio of P_B to P_W (i.e. large between class mean separations, small within class variances). The rows of the optimal transformation matrix A can be found by solving the generalised eigenvalue problem

$$S_B \vec{a}_i = \lambda_i S_W \vec{a}_i. \tag{8}$$

for the eigenvectors $\vec{a}_i$. The top M eigenvectors then define the transfromation matrix A.

3.3 Fuzzy Discriminants Analysis

In the previous section we saw how we could treat each of the samples used to train the future return predictors as belonging to one of two distinct classes, namely those with positive or negative returns. We then derived a transform that would reduce the dimension of the input vectors whilst retaining the maximum amount of separation between samples of different classes.

A criticism of this approach is that it treats all positive and all negative returns equally. Thus, a transform that causes a small positive and a small negative return to map close together in the transformed space is penalised in the same way as a transform that maps a large positive return close to a large negative one. For the purposes of maximising the profitability of our trading system we are most interested in correctly distinguishing between *large* positive and *large* negative returns. In this section we derive a slightly modified version of the CDA transform that achieves this objective.

The main difficulty with our use of the two-class CDA formulation lies in the artificial discretisation of what is really a continuous variable (the future returns of the securities). To overcome this difficulty we shall define each sample vector $\vec{x}$ as belonging to both the *up* and *down* class with *fuzzy* weightings. The weightings will be a function of the future returns of the sample vector $r(\vec{x})$. The weightings are constrained such that the sum of up and down weightings for each sample is 1.00. A number of suitable functions might be used, we have chosen the following. Let the *down* weighting of sample vector $\vec{x}$ be denoted by $\omega(1|\vec{x})$ and the *up* weighting be denoted $\omega(2|\vec{x})$. We define the weightings to be

$$w(1|\vec{x}) = \frac{1}{1 + exp\left(D(r(\vec{x}))\right)}, \tag{9}$$

$$w(2|\vec{x}) = \frac{1}{1 + exp\left(-D(r(\vec{x}))\right)}. \tag{10}$$

where D(.) is a function of the mean μ_r and standard deviation σ_r of the future returns sampled over the training set and the actual return itself $r(\vec{x})$

$$D(r(\vec{x})) = \frac{|r(\vec{x}) - \mu_r|}{\sigma_r}. \tag{11}$$

Now that we have obtained expressions for the class weightings for each sample vector we re-write Equations 2 - 3 of the CDA analysis in terms of these weightings.

$$S_W = \sum_{c=1}^{2} \left(\sum_{\vec{x}} \omega(c|\vec{x}) \, (\vec{x} - \vec{m}_c) \; (\vec{x} - \vec{m}_c)^T \right), \tag{12}$$

where $\vec{m}_c$ is the weighted class conditional mean, defined by

$$\vec{m}_c = \frac{1}{W_c} \sum_{\vec{x}} \omega(c|\vec{x})\vec{x}, \tag{13}$$

and W_c is the weighted number of sample vectors belonging to class c

$$W_c = \sum_{\vec{x}} \omega(c|\vec{x}) \tag{14}$$

The derivation of the optimal subspace then proceeds as before. It is hoped that by re-formulating the problem in these terms the transformation obtained

Model	Mean square error	% Accuracy all	% Accuracy large
MLP 128	27.9	52.67	53.67
MLP 40 PCA	19.82	52.24	53.81
MLP 10 CDA	19.34	53.72	57.15
MLP 10 FDA	17.97	53.63	58.87

Figure 4 Table comparing the performance of the different feature transforms

will provide good separation in input space between samples having large positive and large negative future returns. We shall refer to this transform as a *Fuzzy Discrimanants Analysis*, the term *fuzzy* being a reference to the pseudo-probabilistic assignment of samples to the up and down return classes.

3.4 Results for Feature Transformations

A relative performance comparison of the three different feature transforms was carried out using the UK equity data. Figure 4 shows the relative performances of multi-layer perceptron predictors trained using each of these three feature transforms as input. The predictors were trained to forecast 7 day ahead returns. A benchmark result was also obtained using the original 128 dimensional input vectors denoted by *MLP 128*. The subspace dimension chosen for each feature transform was optimized over the range 5 to 40. It was found that a 10 dimensional input space gave the best results for CDA and FDA. Forty dimensions gave the best results for PCA.

The results in Figure 4 quote both mean square prediction error and correct directional prediction percentages (the percentage of times that the predictor correctly forecasts the *sign* of the future return). A result is also given for the percentage accuracy of the predictors in forecasting the sign of *large* absolute returns. A large return is defined to be a return whose absolute magnitude is greater than 2 percent.

The performances shown in Figure 4 are averaged for 20 different securities and are quoted in terms of the test set consisting of the previously unseen data period 30 December 1989 to 30 December 1994.

Clearly all the predictors that used feature transforms achieve lower prediction errors than the untransformed benchmark predictor. In addition, the CDA and FDA schemes significantly outperform the untransformed and PCA transformed data schemes. The CDA technique performs slightly better on percentage accuracy, whilst the the FDA does better on mean square prediction error. It can be seen that the FDA performs best when considering its accuracy of predicting large changes.

In this section, comparisons have been made between the use of different feature transforms using a multi-layer perceptron as a predictor. In the next section we will compare the predictive abilities of the multi-layer perceptron, a committe network and a k-nearest neighbour predictor for forecasting future returns. We shall also compare the performance of a committee network and a stochastic volatility model for forecasting future volatilities.

4 Prediction

In this section we compare the predictive ability of three different predictors. These are the multi-layer perceptron (used to predict future returns and volatilities), a k-nearest neighbour model (used to predict future returns) and a stochastic volatility model (used to predict future volatilities). Before presenting the relative performances of the different models we will briefly review them giving details of their particular implementation in this work.

4.1 Multi-Layer Perceptron

Neural networks have recently received considerable attention in the financial community [23] [18]. The most commonly used neural network architecture is the multi-layer perceptron shown in Figure 5.

The basic building block of the multi-layer perceptron is the artificial neuron shown in Figure 6. A neuron operates by summing the input it receives via weighted links from other neurons and then outputs a value that is a non-linear function of this input activation. Let x_i be the total activation of neuron i, y_i the output of neuron i and ω_{ji} the weighted link between neurons j and i then

$$\begin{aligned} x_i &= \sum_{j=1}^{n} \omega_{ji} y_i, \\ y_i &= \theta(x_i), \end{aligned} \tag{15}$$

where $\theta(.)$ is some non-linear function, usually taken to be sigmoidal,

$$\theta(x_i) = \frac{1}{1 + exp(-x_i)}. \tag{16}$$

The multi-layer perceptron is composed of hierarchical layers of such neurons arranged so that information flows from the input layer to the output layer of the network, i.e. no feedback connections are allowed. The device hence provides a non-linear mapping of input vectors to output responses i.e.

$$\mathcal{F} : \mathcal{R}^{I_n} - > \mathcal{R}^{O_n} \tag{17}$$

where I_n is the number of neurons in the input layer and O_n is the number of neurons in the output layer. The particular mapping performed by the multi-layer perceptron is determined by the adjustable weighted links between nodes. It can be shown that a three layer multi-layer perceptron with an arbitarily large number of nodes in the hidden layer acts as a *universal approximator* and can realize any continuous function [10].

The use of the multi-layer perceptron in this work is shown in Figure 7. Features obtained from the security whose return is being forecast and the

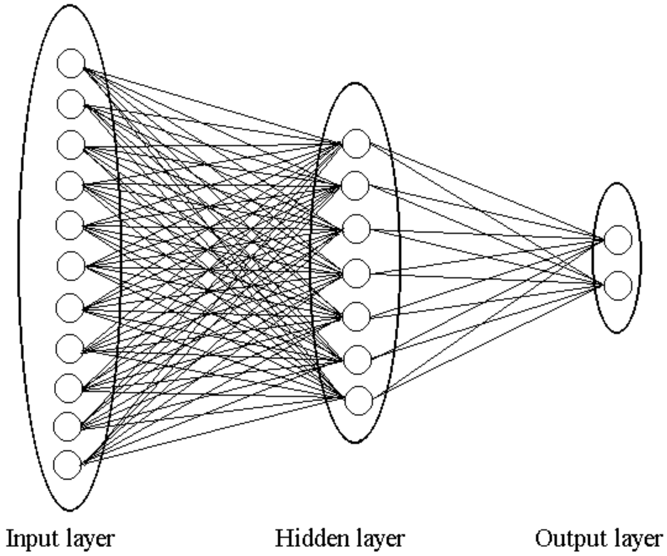

Figure 5 The multi-layer perceptron

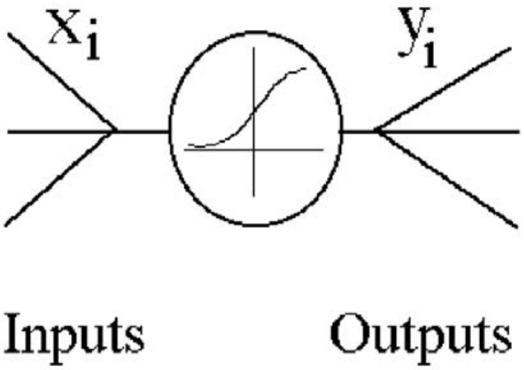

Figure 6 The neuron

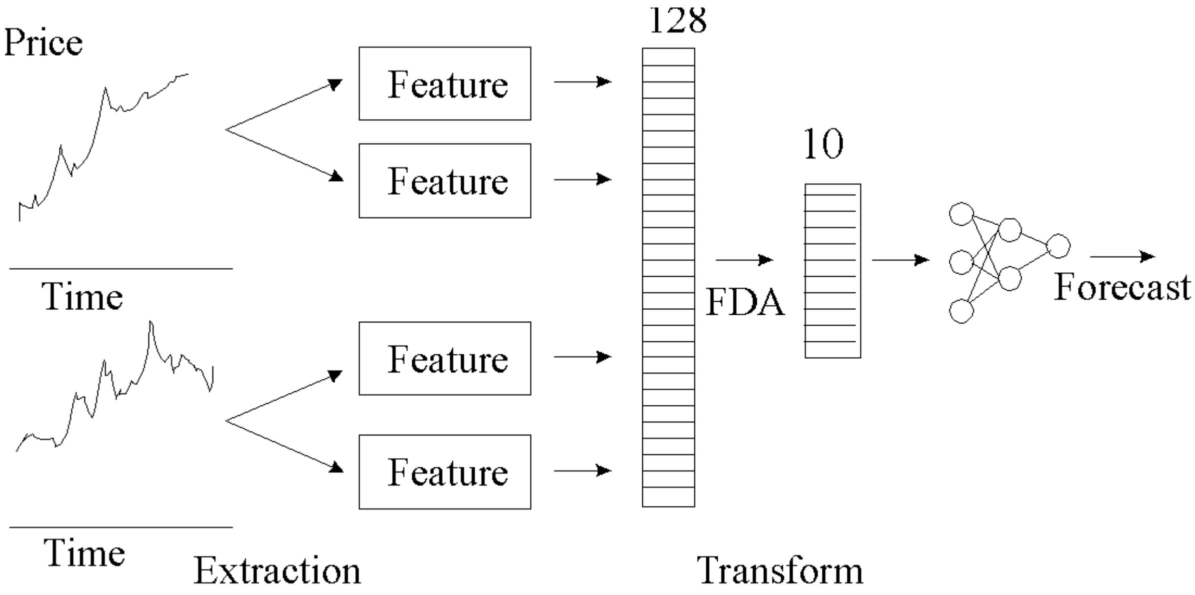

Figure 7 Use of the nulti-layer perceptron to forecast future returns and risks of securities.

FTSE-100 index are transformed using Fuzzy Discriminants Analysis and then fed into the input layer of a single hidden layer multi-layer perceptron. The output layer of the network, comprising a single neuron, is then required to give an output response that is a simple coding of the predicted future return or risk of the security for a 7 day ahead prediction horizon.

The network is trained using a set of desired input/output vector pairs termed the training set. The training process involves the iterative adaptation of the weighted links between neurons to minimize the mean square error between the desired outputs and the actual ones produced by the net. A number of techniques may be used to achieve this. First order techniques such as backprop [22] are often used, though in this work a more optimal method, scaled conjugate gradient [12], was preferred.

To avoid overfitting the data in the learning process a method of concurrent descent is used whereby the training data is split into training and validation sets. Training is halted at the point at which the error of the validation set is minimized. The performance of the network on unseen data is then assessed by applying it to a separate test set. To further improve the generalisation performance of the networks, weight decay was applied to the networks during training.

The optimal number of hidden nodes to use for each architecture was determined by an exhaustive search between 2 and 32 hidden nodes for each architecture. This involved training nets with the number of hidden nodes in this range and then choosing the five architectures providing the lowest validation error. These five architectures were then placed into a *committee network* structure. The committee provides predictions that are weighted linear combinations of the predictions of each of its members. The optimal weightings for each of the committee members can be found through a direct ordinary least squares method applied to the training data [6].

4.2 Stochastic Volatility Model

For most financial time series groups of highly volatile observations are interspersed with tranquil periods. Modelling volatility becomes important if markets are efficient, since the observations p_t of a particular financial time series should just follow a martingale. However, it is then possible to find non-linear models for the change in variance which might provide some predictive ability.

Changes in the variances y_t can be set up as a Gaussian white noise process ε_t which is multiplied by the standard deviation σ_t given by

$$y_t = \sigma_t \varepsilon_t, \; \varepsilon_t \sim IID(0,1). \tag{18}$$

The stochastic volatility model assumes that the variance σ_t is an unobservable process and the volatility at time t given all the information up to time $t-1$ is random. Let the log volatility h_t denote a normally distributed unobservable random variable

$$h_t \sim N(0, \sigma_h^2) \tag{19}$$

then

$$\sigma_t^2 = \exp h_t. \tag{20}$$

Thus σ_t^2 can be generated by a linear stochastic process such as a first-order autoregression AR(1). The stationary AR(1) stochastic volatility model is given by

$$\begin{aligned} y_t &= \varepsilon_t \exp\left(\frac{h_t}{2}\right), \\ h_{t+1} &= \gamma + \phi h_t + \eta_t, \end{aligned} \tag{21}$$

where $0 \leq \phi \leq 1$ and

$$\eta_t \sim NID(0, \sigma_\eta^2) \tag{22}$$

The non-linear equation 21 can be modified to fit in a linear state space model. Assuming normality for ε_t it can be shown that $\log \varepsilon_t^2$ has a mean of -1.2704 and a variance of $(\pi^2/2)$ [1].

Let

$$\varepsilon_t^* = \log \varepsilon_t^2 + 1.27, \tag{23}$$

Squaring the observations and taking logs, the stochastic volatility equation 21 is given is state space form as

$$\begin{aligned} \log y_t^2 &= -1.27 + h_t + \varepsilon_t^*, \\ h_{t+1} &= \gamma + \phi h_t + \eta_t. \end{aligned} \tag{24}$$

Assuming that $\log \varepsilon_t^2$ is normally distributed a quasi-likelihood estimation can be carried out by applying the Kalman filter to the state space form (equation 24). An in depth discussion of the Kalman filter and its application to stochastic volatility models can be found in [7] and [8].

Model	Mean square error	% Accuracy all	% Accuracy large
FDA MLP	17.97	53.63	58.87
Committee MLP	16.81	54.2	58.24
k-NN	19.86	52.14	51.91

Figure 8 Table giving results for predicting future returns

Model	Mean square error
Committee MLP	15.67
Stochasitc volatility	15.94

Figure 9 Table giving results for predicting future variances

4.3 Results

To compare the various prediction models, we again use a set of 20 UK equities. The prediction models were trained and validated on the first six years of the data. The prediction errors and percentage accuracies quoted in this section were then calculated on the unseen test set consisting of the last 5 years of data (30 December 1989 to 30 December 1994).

Return Prediction

The models used for predicting future returns were a single multi-layer peceptron, a committee network predictor containing 5 multi-layer perceptrons and a k-nearest neighbour predictor (the optimal k value was found using the training and validation sets).

All of the models were trained to predict the 7 day ahead return of each of the 20 securities. Figure 8 shows the relative performances of the predictors on the unseen data. The performance of the committe network is shown to be slightly better than the single multi-layer perceptron and considerably better than the k-nearest neighbour model.

Volatility Prediction

Here, we compare the performance of a committee multi-layer perceptron and a stochastic volatility model applied to the prediction of future volatilities. It can be seen from Figure 9 that the committee network and the stochastic volatility model give similar results, with the prediction error of the network being slightly smaller. This result is consistent with previous work carried out examining FX/FX data [21].

Position	Long	Flat	Short
Equity assets	S	0	-S
Monetary assets	0	S	2S

Figure 10 Table showing the possible positions that may be held for each security

5 Trading Models

We have shown how to obtain good estimates for the future returns and volatilities of securities using an intelligent feature transformation (Fuzzy Discriminants Analysis) and committes of neural networks. We now wish to convert these estimates into a trading strategy. We examine two separate (realistic) trading strategies and assess the profitability of trading these systems over the 5 years of test data set aside from our original data set.

5.1 Simple Speculative Model

We assume we start with a total of $40S$ in assets. We divide the assets equally between the 40 securities. At any particular time, we adopt one of the three positions (long, short or flat) for each security detailed in Figure 10.

When taking a long position we hold S of our assets in shares for a particular security and nothing is invested at the risk free rate. For a flat position we hold nothing of our assets in shares in the equity but invest S of the assets at the risk free rate. For a short position, we short sell S of our assets in shares and invest $2S$ at the risk free rate.

Whenever switching between positions in a particular security, a combined transaction cost and slippage allowance of 1% of the price of the equity is subtracted from the total assets.

To trade each security we trained three independent predictors, each predictor being a committee network containing 5 multi-layer perceptrons. The committee networks were trained to predict the future returns of the equity at three different return horizons, namely 7, 14 and 28 days ahead. At time t we form a trading signal $s(t)$ that is a weighted combination of the three predictions i.e.

$$s(t) = \sum_{i=1}^{3} P_i(t) \tag{25}$$

where $P_i(t)$ is the return estimated by predictor i. Although the predictors use different prediction horizons to train the networks, the predictions $P_i(t)$ are always in terms of estimated *one* day ahead forecasts. The conversion is made by simply dividing the prediction reponse by the number of days ahead that the particular predictor is trained on.

We then use the trading signal $s(t)$ at time t and apply the trading rules shown in Figure 11. If the trading rules call for a change of position we apply

Current position	Test	Action: Go...
Flat	if $s(t) > \alpha$	Long
Flat	if $s(t) < -\alpha$	Short
Long	if $s(t) < -\beta$	Flat
Short	if $s(t) > \beta$	Flat

Figure 11 Table showing the trading rules

Model	Profitability
Buy & Hold	27.88%
FTSE-100 Index	26.53%
Speculative Trading System	52.01%

Figure 12 Table giving results for profitability of trading strategies over the test period

transaction costs and re-assess the total assets held in that particular security.

The two trading parameters α and β are used as thresholds to decide when to change position. Suitable values for both parameters are found by optimizing the profitability of the system applied to the training and validation data. The actual profitability of the system is then determined by trading it on the test data.

Results

Figure 12 shows the overall returns of this trading model applied to the 40 equities. Clearly the trading strategy suggested by the system significantly outperforms the simple returns of both the individual equities and the FTSE-100 index itself. What is not clear, however, is the amount of market risk we expose ourselves to by adopting this methodolgy. In the next section we consider a trading model that provides high return and attempts to manage risk.

5.2 Portfolio Management Model

We assume we start with a total of S in assets. At any time t we hold $\omega_i(t)S$ of shares in equity i. These weights are constrained such that

$$\sum_{i=1}^{40} \omega_i(t) = 1.00 \tag{26}$$

The *risk* associated with the portfolio at time t is

$$V(t) = \sum_{i=1}^{40} \sum_{j=1}^{40} \omega_i(t)\omega_j(t)\sigma_{ij}(t) \tag{27}$$

Model	Profitability
Buy & Hold	27.88%
FTSE-100	26.53%
Managed portfolio	48.74%

Figure 13 Table giving results for profitability of trading strategies

where $\sigma_{ij}(t)$ is the covariance of the expected return between securities i and j at time t. These covariances may be calculated using the committee network predictions of the volatility (σ_i^2) coupled with estimates of the inter-security return correlations ρ_{ij}. The inter-security returns may be estimated using simple historical averages [19]. The covariances are then simply

$$\sigma_{ij}(t) = \rho_{ij}\sigma_i(t)\sigma_j(t) \tag{28}$$

The *expected return* of the portfolio $R(t)$ at time t is simply

$$R(t) = \sum_{i=1}^{40} r_i(t)\, \omega_i(t) \tag{29}$$

where $r_i(t)$ is the expected return of security i evaluated by the committee network at time t.

During the five year testing period we adopted the following trading strategy. At weekly intervals, the portfolio weightings were adjusted such that the expected return $R(t)$ was maximised subject to the constraints of Equation 26 and the further constraint that the risk of the portfolio $V(t)$ must be no greater than the risk of the FTSE-100 index at all times. The method used to find the portfolio weights was a direct method based on the use of Lagrangian Multipliers [16]. We then added further constraints relating to transaction charges incurred by changing positions. When changing portfolio weightings a round-trip transaction/slippage allowance of 1 percent was subtracted from the assets.

Results

Figure 13 shows the performance of the managed portfolio over the five test years. In comparison we show the returns of both the FTSE-100 index and the return of a simple *average* portfolio of the 40 equities with constant, equal weightings (i.e. the average return of the securities in the portfolio). Clearly, the return of the managed portfolio is superior to both the FTSE-100 index and the average portfolio. We also found that the volatility of the managed portfolio was marginally lower than that of the FTSE-100 index.

6 Conclusions

In this paper we have given an overview of a general prediction and trading methodology covering all aspects from obtaining raw market information to making actual trading recommendations. We have demonstrated the use of a novel discriminant analysis technique to reduce the input space for predictors such as neural networks or k-nearest neighbour models. It was shown that these input reduction techniques achieved a performance increase in terms of mean square error and prediction accuracy.

Committees of neural networks were used to make predictions about the future returns and volatilities of securities using inputs transformed using Fuzzy Discriminats Analysis. These estimates were then used in two trading models applied to the management of a portfolio of 40 UK equities. The equities were managed over a five year period with realistic, somewhat excessive, transaction and slippage costs introduced into the simulated trading.

In the first trading model we showed that by adopting simple speculative long, short or flat positions based on the predictions of the committee networks, we were able to outperfom the return of the FTSE-100 index by just under 100 percent. In the second model, we showed that by using a Markowitz type methodology we were able to generate a similar amount of excess return *and* at the same time manage the risk of the portfolio to be always less than or equal to that of the FTSE-100 index.

References

[1] M Abramovitz, N C Stegun, **Handbook of Mathematical Functions**, Dover Publications,New York, 1972.

[2] J P Benzecri **L'analyse des correspondances**, Dunod, Paris 1973.

[3] T Bollerslev, **Generalized Autoregressive Conditional Heteroskedasticity**, Journal of Econometrics, 31, 307-327, 1986.

[4] R O Duda P E Hart **Pattern Classification and Scene Analysis** Chapter 4, Wiley-InterScience. 1972.

[5] R F Engle, **Autoregressive Conditional Heteroskedasticity With Estimates of the Variance of United Kingdom Inflation**, Econometrica,50, 987-1008, 1982.

[6] L K Hanson, P Salamon, **Neural Network Ensembles**, IEEE Transactions on Pattern analysis and Machine Intelligence PAMI-12, 10,993-1001,1990.

[7] A Harvey, E Ruiz, N Shephard, **Multivariate Stochastic Variance Models**, Review of Economic Studies, 61, 247-264, 1994.

[8] A Harvey, **Forecasting, structural time series models and the Kalman filter**, Cambrigde University Press, 1991.

[9] - S Hashem, B Schmeiser, **Approximating a function and its derivatives using MSE-optimal linear combinations of trained feedforward neural networks**, Proceedings World Congress on Neural Networks WCNN-93,I-617-620, 1993.

[10] K Hornik **Approximation Capabilities Of Multi-layer Feedforward Networks**, Neural Networks, Vol.4, No.5, pp.251-258.

[11] H M Markowitz,**Portfolio Selection: Efficient Diversification of Investments**, Wiley, New York, 1959.

[12] M F Moller **A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning**, PB-339, Pre-print Computer Science Department, University of Aarhus, Denmark.

[13] U A Muller, M M Dacorogna, R B Olsen, O V Pictet, M Schwarz, C Morgenegg, **Statistical Study Of Foreign Exchange Rates, Empirical Evidence Of A Price Change Scaling Law, And Intraday Analysis**, Journal of Banking and Finance, 14, 1189-1208, 1990.

[14] - G E P Box G M Jenkins **Time Series Analysis: Forecasting and Control** revised edition, San Francisco, Holden Day 1976.

[15] E Oja **Neural Networks, Principal Components and Subspaces** Intl. Journal On Neural Systems, Vol. 1 p. 61-68, 1989.

[16] R A Haugen **Modern Investment Theory** Prentice Hall, 1993.

[17] Black F, Scholes M. **The Pricing of Options and Corporate Liabilities**, Journal of Political Economy, May-June 1973.

[18] A N Refenes, M Azema-Barac, S A Karoussos, **Currency Exchange Rate Prediction And Neural Network Design Strategies**, Neural Computing And Applications, 1 1 46-58, 1993.

[19] $RiskMetrics^{TM}$ **- Technical Document**, Second Edition, JP Morgan, New York, 1994.

[20] D E Rummelhart, G E Hinton, R J Williams. **Learning Internal Representations by Error Propagation in Parallel Distributed Processing**, Vol 1, Ch 8. MIT Press.

[21] S P Toulson. **Forecasting Level and Volatility Of Exchange Rates: A Comparative Study**, Vol.1 212-216, World Congress on Neural Networks (WCNN-95). 1995

[22] P Werbos **Beyond Regression: New Tools For Prediction And Analysis In The Behavioral Sciences**, Ph.D Dissertation, Harvard University, Department of Applied Mathematics.

[23] H White, **Economic Prediction Using Neural Networks: The Case OF IBM Daily Stock Returns**, Proc. IEEE Int. Conference on Neural Networks, San Diego, II-451-459,1988.

[24] J Wiles J Stewart A Bloesch **Patterns of Activations Are Operators in Recurrent Networks**, Technical Report 189, Department Of Computing Science, Queensland, Australia.