An Artificial Intelligence Analysis of Stock Returns

Mohamad Shaaf
Professor of Economics
College of Business Administration
University of Central Oklahoma
Edmond, OK 73034

Maryellen P. Epplin
Professor of Finance
College of Business Administration
University of Central Oklahoma
Edmond, OK 73034

Abstract

Previous studies used regression to examine the relationships between stock returns and macroeconomic variables. This study uses an artificial intelligence method to further investigate this relationship. The results suggest that the relative importance of dividends is 51.5 percent, making them the most important factor influencing the rate of return on the New York Stock Exchange index. The order of relative importance of the remaining variables was as follows: industrial production 29.1 percent, money supply 16.5 percent, yield on AAA corporate bonds 1.4 percent, Treasury Bill rates 0.7 percent, consumer price index 0.6 percent, and stock market volatility 0.3 percent.

Introduction

Theoretically, stock returns are based on fundamental economic information such as macroeconomic variables and firm-specific factors. Empirically, many researchers have investigated the link between those variables and stock returns. No empirical study has provided a satisfactory explanation for the behavior of stock market. For instance, Cutler, Poterba, and Summers (1989) performed regression analysis on the real dividend-inclusive return on the value-weighted New York Stock Exchange (NYSE) composite portfolio over the 1926 – 1985 period using seven variables (1987). The explanatory factors were real dividend payments, industrial production, the real money supply, the nominal long-term interest rate, the nominal short-term interest rate, the monthly Consumer Price Index (CPI) inflation rate, and market volatility, measured as suggested by French, Schwert, and Stambaugh (1987).

More recently, a branch of artificial intelligence modeling, called neural networks, has been used to analyze security returns. Two advantages of neural network models relative to those of econometrics are: 1) they are nonparametric, and 2) they allow for nonlinearity in variables. Since econometric models, which have been used in some earlier studies, might violate one or both of these conditions, the results could be undermined. The theory of neural networks states that they detect patterns in a way similar to the human brain. Artificial neural networks are multiple-layer configurations, consisting of simple processing elements or nodes that interact with each other through weighted connections in the system. In this learning design the network is set for pattern recognition using actual cross-sectional or time-series data as the external teacher.

Neural network applications have been used by mutual fund managers, pension fund managers, and capital management companies to pick stocks for their portfolios (Davidson, 1999). Industry participants have described them as "a valuable tool in the trader's arsenal" for enhancing fund performance (Ruggiero, 1999). Eakins, Stansell, and Buck (1998) investigated the factors used by institutional investors in stock selection. Their results indicate that factors other than systematic (general) risk are important to these investors and that neural networks models produce the best forecasts of institutional ownership.

Several studies have used neural networks to explore the use of various financial and economic variables. Qi and Maddala (1999) used them to forecast stock returns. When portfolios based on linear regression forecasts and on neural network forecasts were constructed, those based on the neural network method earned higher risk-adjusted returns. Racine (2001) was unable to replicate Qi's (1999) results. In this study, neural network-based portfolios yielded lower accumulated wealth and higher risk than linear regression-based portfolios. Chuah (1993) was not able to time the market using neural network results, but the neural network forecasts led to much larger profit than those generated using a linear model or a buy and hold strategy. Gencay (1988) and Gencay and Stengos (1988) used neural networks to predict the Dow Jones Industrial Average using past buy and sell signals based on moving average trading rules. The networks were able to effectively predict the index.

The objective of this study was to use neural network modeling to further investigate the relationships between stock returns and financial and economic variables. Specifically, the variables used were those suggested by Cutler, Poterba, and Summers (1989). A neural network model was used to evaluate whether the real, dividend-inclusive return on the value-weighted NYSE index can be forecasted based on these seven variables. A further aim of the study was to investigate the relative importance of the factors influencing stock returns. The results could prove useful to both institutional and individual investors in portfolio management.

The findings of this study confirm those obtained earlier by other methods, and suggest that dividends, industrial production, and money supply are the most important factors influencing stock returns. The technical details relating to neural network modeling are explained in the Appendix. The following section explains the nature of the data, their form in the model, and the composition of inputs and output. The empirical findings of the study and the simulation of the model are presented next, followed by the summary and conclusions.

Input and Output of the Neural Networks

In this study the following output and inputs are used.

Output:

ROR = real dividend-inclusive monthly return on the value-weighted New York Stock Exchange index.

Inputs:

LDIV = logarithm of real dividend payment on the value-weighted New York Stock Exchange portfolio deflated by monthly Consumer Price Index.

LIP = logarithm of industrial production.

LMS = logarithm of the M1 money supply.

AAAYLD = nominal long-term interest rate, measured as Moody's AAA corporate bond yield.

TBR = nominal short-term interest rate, measured as the yield on three-month Treasury bill rates.

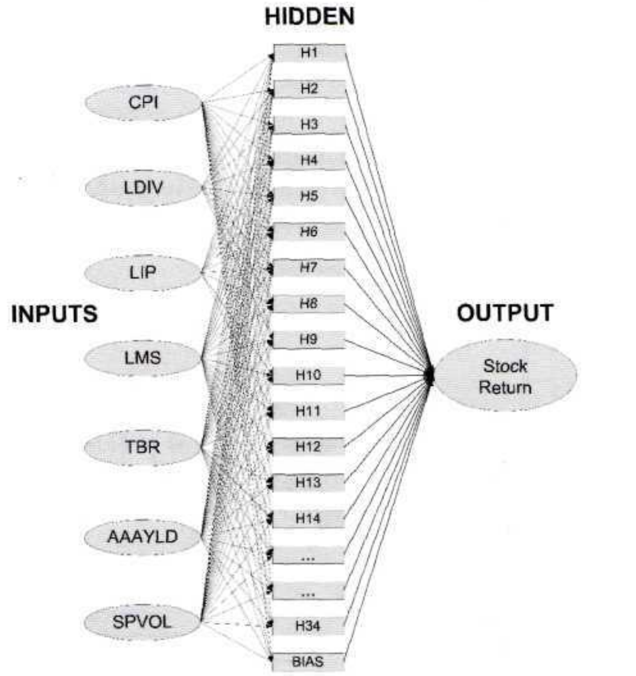
CPI = monthly Consumer Price Index.

SPVOL = logarithm of stock market volatility, defined as the average squared daily return on the Standard & Poor's Composite Index within the month.

The output and inputs of the model are shown in Figure 1, and expressed by the following general function:

ROR = F (LDIV, LIP, LMS, AAAYLD, TBR, CPI, SPVOL)

Figure 1
Artificial Neural Network Analysis of Stock Returns



Empirical Results and Simulation

The neural network used monthly data from 1966 to 1999 for seven inputs (independent variables) as determinants of the rate of return on the New York Stock Exchange Index (dependent variable).[1] The findings of the model are shown in Table 1. This process is referred to as the learning process, where the neural network program is trained relative to the behavior of the variables being studied. Accordingly, the neural networks employed 34 hidden layers, with resulting optimum hidden neurons of 32. The resulting R^2 of this input mix was 0.27, the average error of the estimate of ROR was 0.0295, the correlation coefficient of the actual and estimated output was 0.523, the mean square error was 0.001, and the root mean square error was 0.037.

Table 1
Results of the Artificial Neural Network: The Determinants of Stock Returns

Number of Inputs	7
Number of Hidden Neurons Trained	34
Optimal Number of Hidden Neurons	32
R^2	0.27
Average Error	0.0295
Correlation	0.523
Mean Square Error	0.001
Root Mean Square Error	0.037
Variables:[a]	**Relative Importance (%)**
LDIV	51.5
LIP	29.1
LMS	16.5
AAAYLD	1.4
TBR	0.7
CPI	0.6
SPVOL	0.3
Total	100

[a] Output:

ROR = Real dividend-inclusive monthly return on the value weighted New York Stock Exchange index. Source: Yahoo Finance, http://chart.yahoo.com/m?s=^NYA

Inputs:

LDIV = Logarithm of real dividend payment on the value-weighted New York Stock Exchange portfolio deflated by Consumer Price Index. Source: Bureau of Economic Analysis, http://ww.stls.frb.org/fred/data/gdp/dividend

LIP = Logarithm of industrial production. [3] Source: Federal Reserve Board, http://www.stls.frb.org/fred/data/business/update/bus01

LMS = Logarithm of the M1 money supply. Source: Federal Reserve Board, http://www.stls.frb.org/fred/data/monetary/m1sl

AAAYLD = nominal long-term interest rate, measured as Moody's AAA corporate bond yield. Source: Federal Reserve Board, http://www.bog.frb.fed.us/releases/h15/data/m/aaa.txt

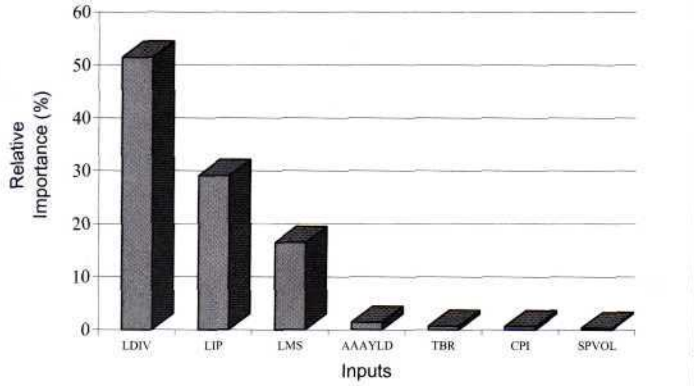
TBR = Nominal short-term interest rate, measured as the yield on three-month Treasury bill rates. Source: Federal Reserve Board, http://www.stls.frb.org/fred/data/irates/tb3ms

CPI = Monthly Consumer Price Index. Source: Bureau of Labor Statistics, http://146.142.4.24/cgi-bin/srgate

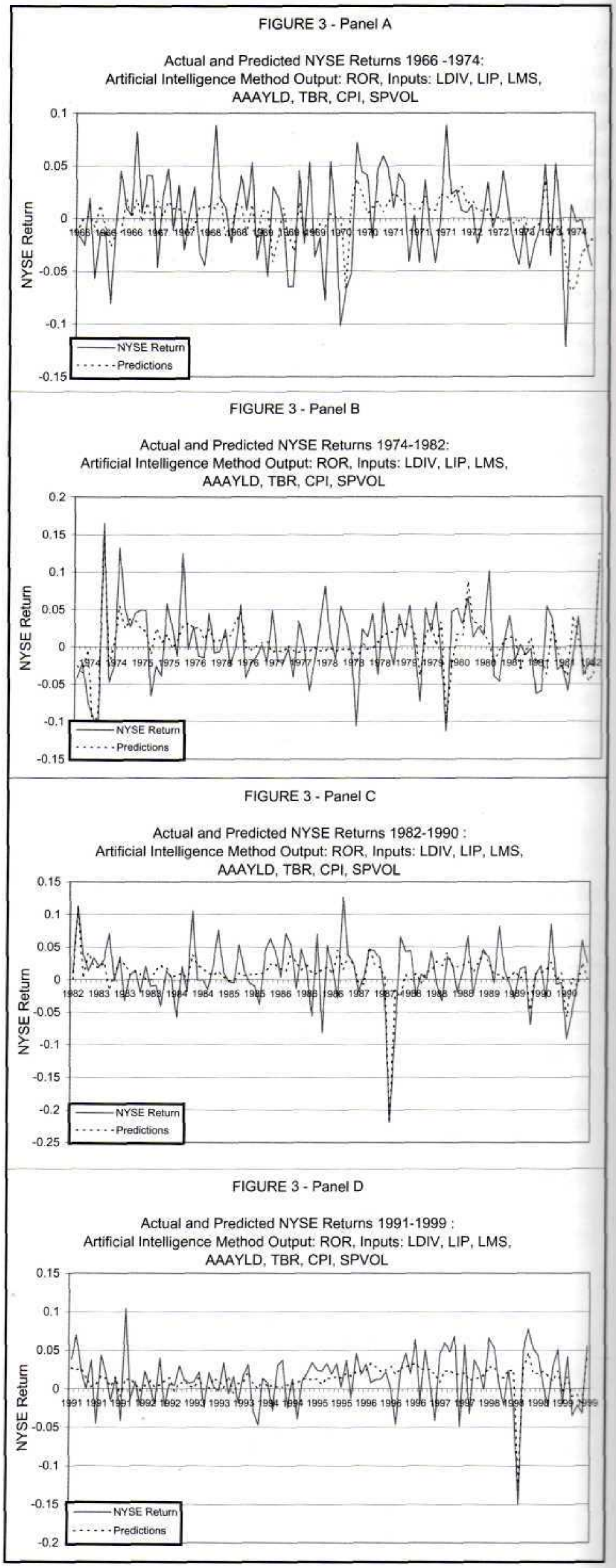
SPVOL = *Logarithm of stock market volatility, defined as the average squared daily return on the Standard & Poor's Composite Index within the month.* Source: Standard & Poor's, http://hart.yahoo.com/d?s=^SPC

The relative importance of these variables was determined by the weight captured in the learning process, as seen in Table 1 and shown in Figure 2. The model found that the relative importance of dividends (LDIV) was 51.5 percent, making them the most important factor influencing the rate of return and the level of the stock index. The relative importance of the six remaining variables was as follows: industrial production (LIP) 29.1 percent, money supply (LMS) 16.5 percent, yield on AAA corporate bonds (AAAYLD) 1.4 percent, Treasury Bill rates (TBR) 0.7 percent, consumer price index (CPI) 0.6 percent, and the volatility of stock market (SPVOL) 0.3 percent.

FIGURE 2
RELATIVE IMPORTANCE OF SEVEN INPUTS ON STOCK RETURNS



To test the accuracy and the validity of the neural network prediction, the model was simulated and the results of the estimated (learned) output (ROR) were compared with the actual values, as shown in Figure 3.



Summary and Conclusion

The empirical results of earlier studies suggest that real dividend payments, industrial production, real money supply, nominal long-term interest rates, nominal short-term interest rates, the monthly Consumer Price Index, and stock market volatility influence stock returns. One of the shortfalls of those studies is the use of econometric models that are parametric, which may not meet the assumptions required for explaining the behavior of stock returns. The purpose of this study was to investigate whether the real dividend-inclusive return on the value-weighted NYSE index can be explained by these seven variables using neural network analysis, which is non-parametric. Furthermore, the purpose of this study was to investigate the relative importance of the variables (inputs) influencing the value-weighted NYSE index.

Using monthly data from 1966 to 1999, the findings confirmed earlier studies that used econometric methods. This study found that, in decreasing importance, dividends, money supply, yield on AAA corporate bonds, Treasury-Bill rates, the consumer price index, industrial production, and stock market volatility influenced the return on the NYSE index. Dividends had the highest relative importance of 51.5 percent, industrial production 29.1 percent, and money supply had 16.5 percent. The remaining three inputs each had less than a 2 percent influence on the stock returns. The neural network model also explained the movement in stock returns somewhat more accurately than studies using regression.

Endnote

[1] Data after 1999, considered to be outliers, were not included due to the beginning of the decline of the stock market during that period. For the sources of data see footnote to Table 1. The Augmented Dickey-Fuller (ADF) method was used to test the time series ROR, LDIV, LIP, LMS, AAAYLD, TRB, CPI, and SPVOL in the model. The results suggested that all were stationary.

References

Chuah, K. L. (1993). *A nonlinear approach to return predictability in the securities markets using feedforward neural network.* Unpublished dissertation, Washington State University, Department of Economics.

Cutler, D. M., Poterba, J. M., & Summers, L. H. (1989). What moves stock prices? *Journal of Portfolio Management, 15* (3), 4-12.

Davidson, C. (1999). Still fuzzy after all these years. *Securities Industry News, 11* (24), 26-28.

Eakins, S. G., Stansell, S. R., & Buck, J. F. (1998). Neural networks vs. Tobit models: analyzing the nature of institutional demand for common stocks. *Quarterly Journal of Business & Economics, 37* (2), 33-48.

French, K. G., Schwert, W., & Stambaugh, R. (1987). Expected stock returns and stock market volatility. *Journal of Financial Economics, 19*, 3-30.

Gencay, R. (1998). The predictability of security returns with simple technical trading rules. *Journal of Empirical Finance, 5*, 347-359.

Gencay, R., & Stengos, T. (1998). Moving average rules, volume and the predictability of security returns with feedforward networks. *Journal of Forecasting, 17*, 401-414.

Qi, M., & Maddala, G. S. (1999). Economic factors and the stock market: a new perspective. *Journal of Forecasting, 18* (3), 151-166.

Racine, J. (2001). On the nonlinear predictability of stock returns using financial and economic variables. *Journal of Business & Economic Statistics, 19* (3), 380-382.

Ruggiero Jr., M. A. (1999). Birth of a neural network. *Futures: News, Analysis & Strategies for Futures, Options & Derivatives Traders*, March, 62-65.

Appendix:

Nature and Mathematics of Artificial Neural Networks

A typical neural network consists of: 1) one input (independent-variable) layer, 2) one or more intermediate or hidden layers, and 3) one output (dependent-variable) layer. Similar to the simultaneous equations in econometric modeling, neural network architecture can have several independent variables and several dependent variables. A network similar to the one used in this study, with seven independent variables (inputs), a bias, and one dependent variable (output) is shown in Figure 1. Each connection from the input layer to the hidden layer or from the hidden layer to the output layer has a weight. These weights represent the coefficients or the parameters of the model. The size of each weight represents the relative strength of the connection. Each node computes a weighted-sum of the incoming values and passes this sum through a nonlinear or linear function as output. The output of one layer serves as the input to the next.

The hidden layer(s) is a processor of information and a bridge between the independent variable(s) and dependent variable(s). Since this layer is connected to independent variables (inputs) on one hand and dependent variables (output) on the other, and is hidden from the outside world, it is called the hidden layer. Besides its link to the hidden layer, the input layer can be directly connected to the output layer. Similar to the econometric method, the back propagation design is an optimization technique for finding optimum values of the weights as parameters. Furthermore, there is a bias processing element that is similar to the intercept notion in equations of econometric models.

The three main phases in the operation of a network are learning, recall, and testing. In the learning phase, the neural network recognizes a pattern between independent variable(s) and dependent variable(s) and estimates the final value at the output layer. Subsequently, this estimated value is compared with the actual output, and errors are calculated. These errors become the factors used to adjust the weights and, subsequently, to reduce the errors and readjust the weights of the connections. This process of weight adjustment continues further until the error declines to an acceptable level, when possible. Through this process, the neural network learns the rules and adapting patterns for processing the information. The resulting pattern of linking the independent variables to the dependent variables in the learning process can be used for testing the accuracy of the networks. In addition, the pattern can be used for prediction of the dependent variable(s), given the independent variable(s).

Suppose there are I number of independent variables, J hidden nodes in the hidden layer, and k output nodes in the output layer. Accordingly, back propagation computes the summation of multiplication of independent variables, X_i, by their corresponding weights (from the input layer to the hidden layer), and adds bias weights (intercept) to it as follows:

$$u_j = w_{0j} + \sum_{i=1}^{I} w_{ij} * X_i \qquad (1)$$

where w_{0j} is the bias weight j, and w_{ij} are the weights from the input i to hidden node j. The weighted summations, u_j, in equation (1) are then transformed by a transfer function in the hidden layer. This output, y_j, from the hidden layer enters the output node as input, and subsequently computes the output of the network with k output nodes, z_k, as:

$$z_k = g(v_k), \qquad k = 1, 2, \ldots, K \qquad (2)$$

and $g(v_k)$ is the sigmoid function of v_k as:

$$v_k = w_{0k} + \sum_{j=1}^{J} w_{jk}\, y_j \qquad (3)$$

where w_{jk} are weights on the links from hidden node j to the output k, and w_{ok} is the bias weight of output node k.

The next step in the neural network is the calculation of the error. Suppose the estimated final dependent variable at the output layer that is produced by the network with N observations is z_{nk}, and the actual dependent variable is A_{nk}, then the mean square error is computed as:

$$E = \sum_{n=1}^{N} \sum_{k=1}^{K} (Z_{kn} - A_{kn})^2 / 2NK \tag{4}$$

Here, E is the global error function, which is differentiable of all the connection weights in the network.

The subsequent stage in the artificial neural network is to adjust the weights (parameters) of the model. The strength of the connections (weights) changes continuously for the minimization of the error as the objective function. There are two major categories of weights in the neural networks. The first category connects the hidden nodes to the output nodes. The second connects the input nodes to the hidden nodes. They are explained next.

Hidden-to-Output-Node Weights

When the error E is determined, the network back propagates it for the increase or decrease of the weights. First, it selects some arbitrary point in weight space and computes the slope of the error surface at that point. Then, it computes the change or derivative of the error, E, with respect to the weight from the hidden node j to the output k, $\delta E/\delta w_{jk}$, through a chain-rule multiplication of three terms as follows:

$$\delta E / \delta w_{jk} = (\delta E/\delta z_k)\ (\delta z_k/\delta v_j)\ (\delta v_j/\delta w_{jk}) \tag{5}$$

The first term on the right side of the equation, $\delta E / \delta z_k$, is the derivative of the error with respect to the dependent variable k. The second term in the chain is the derivative of that node's output with respect to its weighted sum of inputs from the hidden layer to the output layer. The last term, $\delta v_j/\delta w_{jk}$, is the derivative of that sum with respect to the weight connected to the output node k. According to equation (3), the last term in equation (5) depends on the output in the hidden node, y_j, as

$$\delta v_j / \delta w_{jk} = y_j \tag{6}$$

Input-to-Hidden-Node Weights

Similarly, the neural networks compute the relative change of the error, E, with respect to a change in the weights (from input nodes to the hidden nodes), $\delta E / \delta w_{jk}$, as a chain-rule multiplication of five terms as follows:

$$\delta E/\delta w_{ij} = [\ \sum_{k=1}^{K} (\delta E/\delta z_k)(\delta z_k/\delta v_k)(\delta v_k/\delta y_j)](\delta y_j/\delta u_j)(\delta u_j/\delta w_{ij}) \tag{7}$$

The first two terms on the right side of equation (7) are similar to those in equation (5). The third term, $\delta v_k/\delta y_j$, is the derivative of the input from the hidden layer to the output layer with respect to the output j to the hidden node. The fourth term, $\delta y_j/\delta u_j$, is the derivative of the output in the hidden node with respect to the weighted sum of inputs in that node. The fifth term is the derivative of that sum with respect to the weight connecting inputs to the hidden node j.

The next task of the neural network is to find the direction of the weight adjustment in which the error surface declines the most. The network must determine the direction and the magnitude of the change in weight, which results in the greatest reduction of error, as follows:

$$w_p = w_{p-1} - lc \sum_{n=1}^{N} (\delta E / \delta w_p) \tag{8}$$

where w_p and w_{p-1} are the new weights at process numbers p and $p-1$ respectively; lc is the learning coefficient as a parameter, which can be set by the user. It actualizes the proportion and the speed of the weight change. Its best value is determined by trial and error. The last term on the right side of equation (8) is the total derivative of error with respect to each weight in the weight surface. The negative sign in equation (8) changes the direction in which the error declines rather than increases. With the new weight values, the process continues until the magnitude of the error approaches the minimum point in the error surface.

About the Authors

Mohamad Shaaf is a Professor of Economics at the University of Central Oklahoma. He has research interests in diverse areas of economics and finance using artificial intelligence models, econometric methods, and optimization tools.

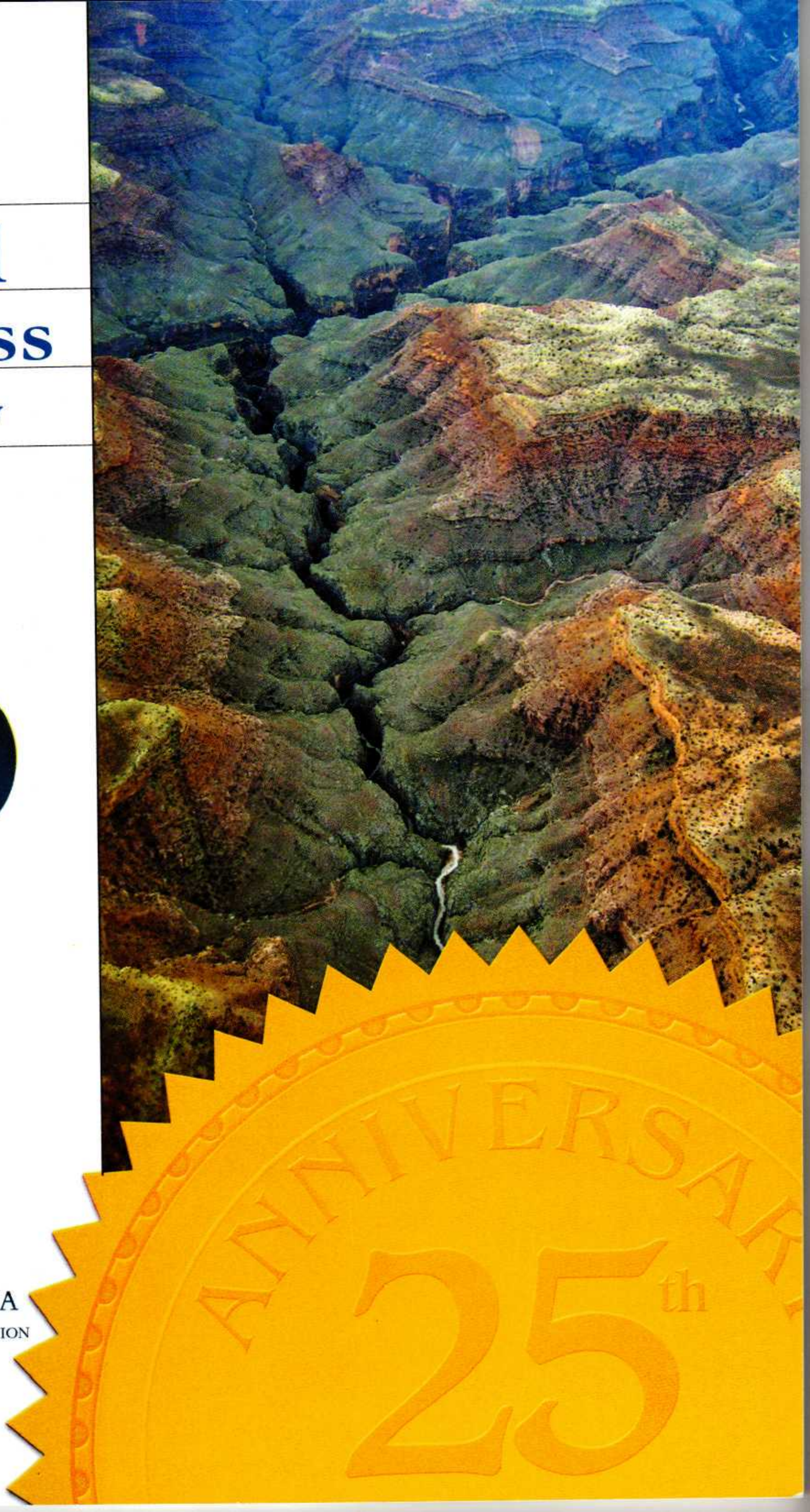
E-mail: mshaaf@ucok.edu

Maryellen P. Epplin is a Professor of Finance at the University of Central Oklahoma. She earned her Ph.D. from Oklahoma State University and is a CFA charterholder. She teaches Business Finance, Intermediate Finance, Investments, and Computer Applications in Finance. Her research interests are in the areas of investments and corporate finance.

E-mail: mepplin@ucok.edu

Central Business Review

WINTER 2006
VOLUME XXV NO. 1



UNIVERSITY OF
CENTRAL OKLAHOMA
COLLEGE OF BUSINESS ADMINISTRATION