Neural Networks, Fuzzy Inference Systems and Adaptive-Neuro Fuzzy Inference Systems for Financial Decision Making

Pretesh B. Patel and Tshilidzi Marwala

School of Electrical and Information Engineering, University of the Witwatersrand, Private Bag 3, 2050, Johannesburg, South Africa
p.patel@ee.wits.ac.za, t.marwala@ee.wits.ac.za
http://dept.ee.wits.ac.za/~marwala/

Abstract. This paper employs pattern classification methods for assisting investors in making financial decisions. Specifically, the problem entails the categorization of investment recommendations. Based on the forecasted performance of certain indices, the Stock Quantity Selection Component is to recommend to the investor to purchase stocks, hold the current investment position or sell stocks in possession. Three designs of the component were implemented and compared in terms of their complexity as well as scalability. Designs that utilized 1, 4 and 16 classifiers, respectively, were developed. These designs were implemented using Artificial Neural Networks, Fuzzy Inference Systems as well as Adaptive Neuro-Fuzzy Inference Systems. The design that employed 4 classifiers achieved low complexity and high scalability. As a result, this design is most appropriate for the application of concern.

1 Introduction

Pattern recognition could be defined as the study of the ability of machines to observe the environment, learn to differentiate between patterns of interest from their backgrounds and formulate reliable as well as sensible decisions about the categories of the patterns [1]. This is a complex task that is an innate ability for humans. However, to develop a system to solve such problems poses formidable research challenges.

This research focuses on a pattern classification problem utilized within an application that could assist individual as well as institutional investors in making financial decisions. It is anticipated that this application would be used in conjunction with other financial analysis methodologies. As a result, such an application should be employed to confirm an investment decision.

Pattern classification is the process of assigning an input pattern to one of a predefined set of classes. It consists of developing a functional relationship between the input features and the target classes. Accurately estimating such a relationship is vital to the success of a classifier. Specifically, the quantity of stocks or shares to be purchased based on the forecasted performance of certain indices is the pattern classification problem. The Dow Jones Industrial Average,

I. King et al. (Eds.): ICONIP 2006, Part III, LNCS 4234, pp. 430–439, 2006.

Johannesburg Stock Exchange or the JSE Securities Exchange (JSE) All Share, Nasdaq 100 and Nikkei 225 Stock Average indices are considered. However, the computational intelligent techniques as well as their implementation methodology utilized in this research could be adapted for decision making systems in other industry sectors.

The classification of data into various classes has been an important research area for many years. Artificial neural networks (ANNs) have been applied to pattern classification [2]. Research has also been conducted on fuzzy classification. This resulted in many algorithms, such as fuzzy K-nearest neighbour [3] and fuzzy c-means [4], being applied to decision making systems. Fuzzy systems constructed using genetic algorithms have been utilized [5][6]. Fuzzy neural networks have also been employed in pattern classification applications [7][8]. Support Vector Machines have been applied to multi-category classification problems [9]. These classification tasks have also been implemented by combining multiple simpler specialized classifiers [10][11].

In this research, artificial neural network (ANN) architectures, Fuzzy Inference Systems (FISs) as well as Adaptive Neuro-Fuzzy Inference Systems (ANFISs) have been considered. Specifically, the Multi-Layer Perceptron (MLP) and the Radial Basis Function (RBF) neural network architectures have been considered. FISs developed employed subtractive clustering to generate the required membership functions and set of fuzzy inference rules. Information on these computational intelligent techniques can be found in [12], [13] and [14], respectively.

The next section briefly examines the application of concern. Thereafter, the implementation methodology is described. The paper concludes with the comparison of the various models developed and the selection of the superior classifiers.

2 The Developed System

The developed system is to be used in assisting an investor in making financial decisions. As a result, the system should be based on a profitable trading strategy. There are numerous trading strategies available. This research focuses on the "Buy low, sell high" trading strategy. The strategy has been implemented as well as compared to the "Buy and hold" trading strategy in terms of profits generated.

The "Buy low, sell high" trading strategy entails purchasing certain stocks at a low price and selling these stocks when the price is high. The "Buy and hold" trading strategy, as the name suggests, involves an investor purchasing certain stocks and retaining them for a particular duration. The method used to implement the "Buy low, sell high" trading strategy involved classifying the change in index or delta into certain categorizes. Delta is defined as the difference between the closing index value for the next day and the closing index value for the previous day. This functionality has been implemented within the Forecasting Component (FC). Depending on the classification of this component, the strategy would recommend the investor to purchase stocks, hold the current investment position or sell stocks. This responsibility can be found in the Stock

Quantity Selection Component (SQSC). Table 1 illustrates the forecasted classes as well as the corresponding investment recommendation.

It has been determined that the "Buy low, sell high" trading strategy, with the percentage threshold combination of 0.8% and -0.20% of the closing value for the previous day, is most profitable. As a result, the system has been based on this trading strategy. Further information on the comparison of the 2 trading strategies considered can be found in [15].

Pattern classification problems can be grouped as either dichotomous or polychotomous problems. Dichotomous classification can be interpreted as 2-class classification problems, whereas polychotomous classification involves problems with more than 2 classes to be categorized. The SQSC module is the center of this research. Based on the forecasted performance of the closing price of the index, the component is to recommend the investor to purchase stocks, hold the current investment position or sell stocks. It is evident that this is a polychotomous classification problem as there are more than 2 classes. Further information on the FC module can be found in [15].

Various classifier designs of the SQSC module were considered. Each of these designs were developed using both ANNs as well as fuzzy logic techniques. The first design employed 1 classifier. This classifier consisted of 16 inputs and 16 outputs. The inputs to the model are the forecasted performance of the closing price of the indices considered. The outputs of the classifier are the investment recommendations for the indices. The second design involved 4 classifiers. Each classifier has 4 inputs and 4 outputs. Each classifier is used to generate an investment recommendation for an index considered. The input to a classifier is the forecasted performance of the closing price of an index. The output of a classifier is the investment recommendation for the index. The third and final design considered utilized 16 classifiers. Each classifier has 4 inputs and 1 output. Each classifier is employed to categorize whether or not to execute an investment recommendation. The input to a classifier is the same as design 2 above. The outputs of the classifiers are fed into an interpretation function that generates the final investment recommendations for the indices. This design has been implemented to investigate the method of utilizing simpler classifiers to generate a multi-category classifier.

3 Implementation Methodology

The data used to develop the SQSC module has been generated based on the 4 forecasted closing price performance classes illustrated in Table 1. The development process was divided into various stages.

The following procedure has been pursued in the creation of the various classifiers employed:

1. Selection and processing of data to be used by the classifiers during training, validation and testing.
2. Optimization of the classification threshold of the various classes to be categorized.

Table 1. The "Buy low, sell high" trading strategy categorizes

Class	Requirement	Investment recommendation
Large Rise (LR)	Delta > Positive threshold percentage of previous day closing price.	If LR is forecasted for the next day, sell stocks in possession.
Slight Rise (SR)	0 < Delta <= Positive threshold percentage of previous day closing price.	If SR is forecasted for the next day, hold current investment position.
Slight Drop (SD)	Negative threshold percentage of previous day closing price <= Delta <= 0.	If SD is forecasted for the next day, buy stocks to the value of 15 % of available trading capital.
Large Drop (LD)	Delta < Negative threshold percentage of previous day closing price.	If LD is forecasted for the next day, buy stocks to the value of 25 % of available trading capital.

3. Optimization of the classifier architectures.
4. Comparison of the various classifiers developed and the selection of the superior model.

The remainder of this section will elaborate on the various stages of implementation mentioned above.

3.1 Selection and Processing of Data

The data utilized in developing and testing the various classifiers has been created by analyzing all the possible combinations of the 4 forecasted closing price performance classes. As a result, the entire data set consisted of 256 unique data records.

In order to present the forecasted closing price performance classes to the classifiers, a binary notation is employed. These inputs are presented to the classifier using 4 inputs. This input representation format is used for all indices considered. A similar binary notation scheme is also utilized to present the investment recommendation outputs. Table 2 illustrates the manner in which the inputs and outputs of the component are to be interpreted. As previously mentioned, design 1 has 16 inputs and 16 outputs. The input representation format is the same as above. However, the first group of 4 inputs corresponds to the forecasted performance of the Dow Jones Industrial Average index. Similarly, the second, third and fourth group of 4 inputs characterizes the forecasted performance of the JSE All Share, Nasdaq 100 and Nikkei 225 Stock Average indices, respectively. The outputs are to be interpreted in a similar manner.

The data is divided into a training, validation and test set. During the implementation of all 3 designs considered, the training data set consisted of all data records where the inputs were classified into 2 of the 4 closing price performance classes. However, the models developed were validated and tested with the remaining possible closing price performance class combinations. The

Table 2. Classifier input and output representation

Classifier inputs					Classifier outputs				
Input	1	2	3	4	Output	1	2	3	4
LR	1	0	0	0	Sell stocks in possession	0	1	0	0
SR	0	1	0	0	Hold current position	1	0	0	0
SD	0	0	1	0	Buy stocks to the value of 15 % of available trading capital	0	0	1	0
LD	0	0	0	1	Buy stocks to the value of 25 % of available trading capital	0	0	0	1

training data set is used to train the ANN to find the general pattern between its inputs and outputs. The validation data set is used to assess the network and the test data is employed to confirm the classification quality of the developed model.

The training data set is used to create the cluster centers within the FISs. However, the validation and test data sets are utilized to assess the classification ability of the inference systems.

3.2 Optimization of the Classification Threshold

MLP and RBF neural network architectures were utilized in the classification of investment recommendations. The MLP and RBF neural network architectures are possibly the most extensively employed ANNs in pattern classification [2]. Due to the non-linear capabilities of these networks, they are said to be excellent universal approximators that provide highly accurate solutions. As a result, these networks produce very practical tools for classification and inversion problems [12].

It has been stated that a network with 1 hidden layer, provided with sufficient data, can be used to model any function [12]. As a result, the ANN architectures employed consisted of only 1 hidden layer. The MLP network hidden layer consists of non-linear activation functions. The choice of the activation function is largely dependent on the application of the model [12]. However, it has been found that the hyperbolic tangent activation function offers a practical advantage of faster convergence during training [2]. As a result, this function has been employed within the MLP network.

The MLP network output layer also contains activation functions. There are 3 major forms of the function that should be considered. These are the linear, logistic sigmoidal and softmax activation functions [2]. It has been stated that the appropriate selection of the output layer activation function for a classification problem is the logistic sigmoidal function [2]. As a result, this function has been employed within the output layer of the MLP network. The RBF networks that have been developed contained a Gaussian activation function within its hidden layer and a linear activation function within its output layer.

As previously mentioned, the FISs developed utilized subtractive clustering to create the required membership functions and set of fuzzy inference rules. During this stage of implementation, the number of hidden nodes within the ANNs and the cluster radius utilized by the cluster centers within the FISs were assumed to be arbitrary. This will be optimized at a later stage of development. During this stage of development, the number of hidden nodes within the ANNs as well as the cluster radius utilized by the FISs was 10 and 0.5, respectively. This stage of implementation involved the optimization of the interpretation of the classifiers. As a result, this involved the selection of an appropriate classification threshold value that would yield the most accurate results.

The classification threshold has been optimized by minimizing an error function that mapped the classification thresholds to the accuracy of the developed classifiers. The process has been performed on the validation data set.

Since this is a classification implementation, the accuracy of the models can no longer be calculated using the sum of square error of the difference between the target and investment recommendation classifier output. Instead a confusion matrix is utilized to identify the number of true and false classifications that are generated by the models developed. This is then used to calculate the true accuracy of the classifiers, using the following equation:

$$Accuracy = \sqrt{\frac{TP * TN}{(TP + FN) * (FP + TN)}} \tag{1}$$

where
TP is the true positive (1 classified as a 1),
TN is the true negative (0 classified as a 0),
FN is the false negative (1 classified as a 0),
FP is the false positive (0 classified as a 1).

The classification threshold was optimized by initially creating classifiers utilizing a threshold value of 0.5. This implies that if the classifier outputs a value less than 0.5, the output will be regarded as a 0. Similarly, if the output value is larger than or equal to 0.5, the output will be interpreted as a 1. This threshold value of 0.5 proved to be adequate for the MLP networks as well as the FISs implementations. The threshold value resulted in 100% accurate classifications. This has been demonstrated on the training as well as validation data sets. However, the RBF classifier employed in design 1 did not perform well utilizing this threshold value. As a result, the classification threshold of this model had been varied from 0.1 to 0.5 in iterations of 0.01. Table 3 illustrates the threshold values that resulted in the largest accuracy value for the validation data set. The threshold value of 0.5 proved to be satisfactory for design 2 and design 3 RBF classifiers.

3.3 Optimization of the Classifier Architectures

This stage of implementation involved the optimization of the ANN and Fuzzy Inference System (FIS) architectures. As a result, this step of development involved the selection of the correct number of hidden neurons that would yield the most accurate results. It also entailed selecting the correct cluster radius that would concede the largest investment recommendation classification accuracy.

Table 3. Results of varied classification threshold for design 1 RBF classifier. DJ, JSE, Nas and Nik corresponds to Dow Jones Industrial Average index, JSE All Share index, Nasdaq 100 index and Nikkei 225 Stock Average index, respectively.

	Classification thresholds			
Class.	DJ	JSE	Nas	Nik
Sell stocks in possession	0.19	0.23	0.20	0.11
Hold current position	0.17	0.10	0.12	0.17
Buy stocks to the value of 15 % of available trading capital	0.19	0.16	0.13	0.17
Buy stocks to the value of 25 % of available trading capital	0.24	0.17	0.17	0.19

The number of hidden neurons or nodes has been optimized by minimizing an error function that mapped the number of hidden nodes to the accuracy of the developed network. The process was performed on the validation and test data sets.

The hidden nodes were optimized by creating various MLP and RBF ANNs with hidden nodes of 1 to 75. As a result, 150 ANNs were developed. These developed networks employed the classification thresholds stated in the previous section. The networks also utilized the same activation functions mentioned in the previous section. Utilizing the training data set, these networks are trained. The validation and test data are then presented to the ANN. Thereafter, the accuracies for the training, validation and test data sets are calculated. When presented with the validation and test data, ANNs that resulted in the largest accuracy were analyzed.

Fig. 1 illustrates the sell stocks in possession at the next day closing price investment recommendation results of design 1. Similar results were achieved for the other design implementations as well as investment recommendations. Similar results were also obtained for the other indices considered.

The investigation revealed that a design 1 MLP and RBF network with number of hidden nodes larger than 12 and 52, respectively, yield 100% accurate models for categorizing the investment recommendations appropriately. The investigation also determined that design 2 MLP and RBF ANNs with number of hidden nodes greater than 2 and 5, respectively, achieved the same results. Similar results were obtained with design 3 MLP and RBF networks that contained more than 1 hidden neuron.

The cluster radius indicates the range of influence of a cluster. A small cluster radius results in small clusters in the data and, therefore, many fuzzy rules. Large cluster radii yield few large clusters in the data and, hence, fewer fuzzy rules [13]. The cluster radius has been optimized by minimizing an error function that mapped the radius to the accuracy of the developed inference systems. This process was performed on the validation and test data sets.

During this step of implementation, the optimization process entailed the construction of various inference systems with the cluster radius ranging from 0.01 to 1. The investigation determined that design 2 FISs with a cluster radius equal to or greater than 0.01 achieve 100% accuracy in categorizing the investment recommendations appropriately. However, the design 1 FIS did not achieve 100% investment recommendation classification accuracies. It has been determined that a cluster radius of 0.11 achieved the most accurate results. The lowest accuracy value attained was 83%. The largest accuracy value was 100%.

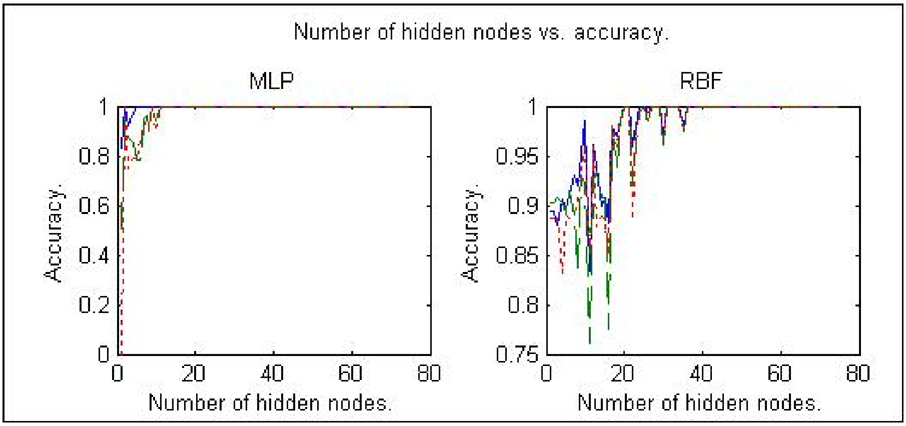

Fig. 1. This figure illustrates the results of sell stocks in possession at the next day closing price investment recommendation for design 1. The number of hidden nodes was varied and the corresponding accuracy values achieved were noted. The solid, dashed and dotted line represent the training, validation and test data sets, respectively.

3.4 Comparison of the Various Designs Implemented and the Selection of the Superior Model

This stage of implementation entailed the comparison of the various designs that were developed. It also involves the selection of the best design to classify the investment recommendations. Table 4 illustrates the various models that have been created. The above designs have been compared in terms of their complexity as well as scalability. Complexity, in this context, is defined as the number of classifiers employed by the design. Scalability is defined as the ability of the design to accommodate the classification of additional investment recommendations.

It is evident that design 1 has low complexity and low scalability. When additional investment recommendations are to be added to the component, the classifier employed is to be re-trained. However, design 2 has low complexity as there are only 4 classifiers utilized. The design also has high scalability. It is not required to re-create the existing classifiers, when additional recommendations are added. Table 4 indicates that design 3 has high complexity. The design contains 16 classifiers. In order to add investment recommendations to the component, the existing classifiers do not have to be re-created. As a result, the design has high scalability.

Table 4. This table illustrates the various models that were created. Accuarcies are presented as percentages.

Design	Classifier topology	Hidden nodes	Fuzzy rules	Membership functions	Accuracy (Training)	Accuracy (Validation)	Accuracy (Test)
1	MLP	12	-	-	100	100	100
1	RBF	52	-	-	100	100	100
1	FIS	-	85	1360	100	83	87
2	MLP	2	-	-	100	100	100
2	RBF	5	-	-	100	100	100
2	FIS	-	4	16	100	100	100
3	MLP	1	-	-	100	100	100
3	RBF	1	-	-	100	100	100
3	ANFIS	-	4	16	100	100	100

Due to the above analysis, design 2 is most appropriate for this application. It does not employ many classifiers and the design does not require re-work when additions are to be made. It is evident from Table 4 that both the ANN and FIS implementations of design 2 perform satisfactorily. As a result, either of the classifier architectures could be used.

4 Conclusion

This research involved the development of a component that could categorize investment recommendations, based on the forecasted performance of indices, appropriately. The Dow Jones Industrial Average, JSE All Share, Nasdaq 100 and Nikkei 225 Stock Average indices were considered.

Various designs of the component were considered. Designs that utilized 1, 4 and 16 classifiers were implemented. The development methodology employed in the creation of these designs, initially, involved the selection of appropriate classification thresholds. Thereafter, the number of hidden nodes within the ANNs as well as the cluster radius of the cluster centers within the FISs was varied. This resulted in creating acceptable classifier architectures. Acceptable investment recommendation classification accuracies were achieved.

The designs were compared in terms of complexity as well as scalability. Complexity is concerned with the number of classifiers that are used within the design. Scalability is the ability of the design to accommodate the classification of additional investment recommendations. Design 2 has low complexity and high scalability. This design consisted of 4 classifiers. Each classifier has 4 inputs and 4 outputs. This design is most appropriate for the application of concern.

References

1. Jain, A. K., Duin, R. P. W., Mao, J.: Statistical pattern recognition: A review. IEEE Transactions on Pattern Analysis and Machine Intelligence. **22** (2000) 4–37
2. Nabney, I. T.: Netlab: Algorithms for Pattern Recognition. Springer (2002)

3. Keller, J. M., Gray, M., Givens, J.: A fuzzy k-nearest neighbor algorithm. IEEE Transaction on System, Man and Cybernetics. **15** (1985) 580–585
4. Bezdek, J. C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum (1981)
5. Ishibuchi, H., Nozaki, K., Yamamoto, N., Tanaka, H.: Construction of fuzzy classification systems with rectangular fuzzy rules using genetic algorithms. Fuzzy Sets and Systems. **65** (1994) 237–253
6. Russo, M.: FuGeNeSys A fuzzy genetic neural system for fuzzy modeling. IEEE Transaction on Fuzzy Systems. **6** (1998) 373–388
7. Nauck, D., Kruse, R.: A neuro-fuzzy method to learn fuzzy classification rules from data. Fuzzy Sets and Systems. **89** (1997) 277–288
8. Sun, C-T., Jang, J-S. R.: A neuro-fuzzy classifier and its applications. Proceedings of the IEEE International Conference on Fuzzy System. **1** (1993) 94–98
9. Hsu, C-W., Lin, C-J.: A comparison of methods for multi-class support vector machines. IEEE Transactions on Neural Neworks. **13** (2002) 415–425
10. Friedman, J. H.: Another approach to polychotomous classification. Department of Statistics, Stanford University. Technical Report. (1996)
11. Schurmann, J.: Pattern Classification. A Unified View of Statistical and Neural Principles. John Wiley & Sons Inc. (1996)
12. Bishop, C. M.: Neural Networks for Pattern Recognition. Oxford University Press. (1995)
13. Chiu, S.: Fuzzy Model Identification Based on Cluster Estimation. Journal of Intelligent and Fuzzy Systems. **2** (1994) 267–278
14. Jang, J-S. R.: ANFIS: Adaptive-Network-Based Fuzzy Inference System. IEEE Transaction on System, Man and Cybernetics. **23** (1993) 665-685
15. Patel, P. B., Marwala, T.: Forecasting closing price indices using neural networks. IEEE System, Man and Cybernetics Conference. (to appear)