



Missouri University of Science and Technology
Scholars' Mine

Electrical and Computer Engineering Faculty
Research & Creative Works

Electrical and Computer Engineering

01 Jan 1995

Conservative Thirty Calendar Day Stock Prediction Using a Probabilistic Neural Network

H. Tan

Donald C. Wunsch

Missouri University of Science and Technology, dwunsch@mst.edu

Danil V. Prokhorov

Follow this and additional works at: https://scholarsmine.mst.edu/ele_comeng_facwork



Part of the [Electrical and Computer Engineering Commons](#)

Recommended Citation

H. Tan et al., "Conservative Thirty Calendar Day Stock Prediction Using a Probabilistic Neural Network," *Proceedings of the IEEE/IAFE 1995 Computational Intelligence for Financial Engineering, 1995*, Institute of Electrical and Electronics Engineers (IEEE), Jan 1995.

The definitive version is available at <https://doi.org/10.1109/CIFER.1995.495262>

This Article - Conference proceedings is brought to you for free and open access by Scholars' Mine. It has been accepted for inclusion in Electrical and Computer Engineering Faculty Research & Creative Works by an authorized administrator of Scholars' Mine. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact scholarsmine@mst.edu.

Conservative Thirty Calendar Day Stock Prediction Using a Probabilistic Neural Network

Hong Tan, Danil V. Prokhorov and Donald C. Wunsch II
 Applied Computational Intelligence Laboratory
 Department of Electrical Engineering, Texas Tech University
 Lubbock TX 79409-3102

Abstract

We describe a system that predicts significant short-term price movement in a single stock utilizing conservative strategies. We use preprocessing techniques, then train a probabilistic neural network to predict only price gains large enough to create a significant profit opportunity. Our primary objective is to limit false predictions (known in the pattern recognition literature as false alarms). False alarms are more significant than missed opportunities, because false alarms acted upon lead to losses. We can achieve false alarm rates as low as 5.7% with the correct system design and parameterization.

Problem Statement

Our problem is to predict if the closing price of a particular stock will go up enough to create a profit opportunity in the next 30 calendar days (about 22 trading days). This information can then be used to analyze the attractiveness of call options, short-term trading, buying on margin, or other risky vehicles requiring conservatism in stock selection to offset risk. We are investigating a variety of stocks and provide an analysis of Apple Computer here as an example of our technique. The methodology can be applied to any stock. In this paper we define a profit opportunity as a price increase of more than 2% in 30 days to account for trading costs, the cost of the time value of stock options, borrowing costs, and the lost opportunity for alternative investments. This number can be refined by conventional risk analysis techniques. Our predictions are made each day.

Predictability and Preprocessing

Utilizing the technique in [1], we analyzed the predictability of several attractive stocks. (Attractiveness of stocks was based on a preliminary manual determination.) The Apple stock daily closing price used as an example throughout this paper exhibits a semi-chaotic behavior implying its partial predictability.

In this example we preprocess the raw closing price data [2]. For each pattern, 19 inputs, including the features defined below are used. The definition for our features are:

$$\text{Level-0: feature (t) = log } \frac{\text{value of close(t)}}{\text{exponential moving average of close(t)}}$$

$$\text{Level-1: feature (t) = } \frac{\text{close(t) - BA(t-n)}}{\text{close(t) + BA(t-n)}}$$

Where:

$$\text{BA(t-n) = } \frac{1}{m+1} \sum_{k=-m/2}^{m/2} \text{close(t-n+k)}$$

The index n determines how far back in time the center of the block is situated. [2] gives a table for different n and m :

n	1	2	3	4	5	7	9	13	17	25	33	49	65	97	129	193	257	385
m	0	0	0	0	0	2	2	4	4	8	8	16	16	32	32	64	64	128

After several tests, we selected $n = 17$. Thus almost one month (about 22 working days) of historical data are used to predict the trend of the stock closing price in the next month.

Probabilistic Neural Network for Prediction

The Probabilistic Neural Network (PNN) is a computationally efficient algorithm for a Bayesian-based function approximation[3]. In our example, it consists of four layers of dedicated nodes (Fig. 1).

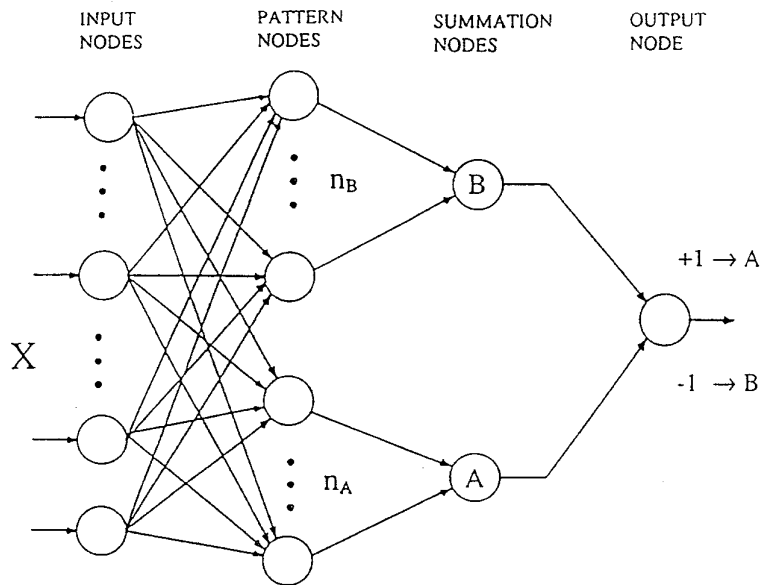


Figure 1. Formulation of Probabilistic Neural Network (PNN) for this problem.

Nineteen input nodes are fully connected with the next layer of pattern nodes. Input nodes simply distribute components of an input X . The i -th pattern node output function is:

$$y_i = \exp(- (W_i - X)^t (W_i - X) / 2 \sigma^2), \quad (1)$$

where W_i is the i -th training pattern, and σ is the smoothing parameter of the Gaussian kernel. Other alternatives to (1) are available [3], including (1) with adaptable σ [4] or full covariance matrices instead [5]. This calls for one pattern node for every pattern in the training data set. In our system and many others there is one training pattern for each day of historical data. This requires much bigger networks than many competing models. However, performance is very competitive with other approaches. The requirement for one pattern node per training data pattern can be alleviated if it becomes unduly burdensome, but we have experienced no problems to date. The third layer is formed by summation nodes which sum the outputs (1) of those pattern units that correspond to a

certain category of predictions. In our case, category A corresponds to our desired prediction of significant profits in the next 30 calendar days, and B is the opposite.

The output node adds outputs of two summation nodes, with only the output of the summation node (B) being weighted by the following parameter C:

$$C = -L (nA / nB), \quad (2)$$

where nA is the number of training patterns from the category (A), nB is the number of training patterns from the category (B), and L is the ratio of losses associated with wrong predictions. We have used $L > 1$ emphasizing the importance of avoiding false alarms. In effect we have a voting mechanism between the training patterns. $L > 1$ multiplicatively biases the output against making predictions by giving greater weight to B node votes than A node votes. At the output, we have a hard-limiting threshold: +1 whenever an input pattern X belongs to category (A) and -1 if it is from category (B).

Training of this predictor is just memorization of all patterns from the training set and assigning a separate pattern node to every training pattern. Such a technique is justified in our case due to availability of data and sufficient computing power. The training is similar to that in [6]. Alternative techniques for similar problems are certainly available and we have seen them perform well in the past [7]. We have used additional proprietary techniques to further enhance the conservatism of our estimates. The results presented below represent these enhanced conservative estimates. Without these enhancements, the approach described above would result in higher false alarm rates than reported here.

Results

Our results are given in the tables and figures below. The network was trained on data from January 1, 1987 through September 30, 1993. These tables show results for out-of-sample data, known in the neural network literature as test data; that is, data that the neural network never was trained on. The test data set goes from October 1, 1993 to September 16, 1994. Table 1 shows predictions for $L = 2$ with some representative values of σ . At $\sigma = 0.0149$ we achieve a false alarm rate of only 10.4%.

Table 1. $L=2$

σ	total predictions	#up > %2	% of success
0.0086	107	87	85.6%
0.0149	77	69	89.6%

Figure 2 shows these predictions (for $L = 2$, $\sigma = 0.0149$) plotted with price data. The straight line shows daily closing price, the filled in triangles denote days when strong price increase within 30 days was predicted, and the open triangles show the false alarms when a prediction was made but the desired price increase within 30 days did not occur.

Table 2 shows predictions for $L = 4$ with some representative values of σ . At $\sigma = 0.0149$ we achieve a false alarm rate of only 5.7%.

Table 2. $L=4$

σ	total prediction s	# up > %2	% of success
0.0086	84	73	87.9%
0.0149	35	33	94.3%

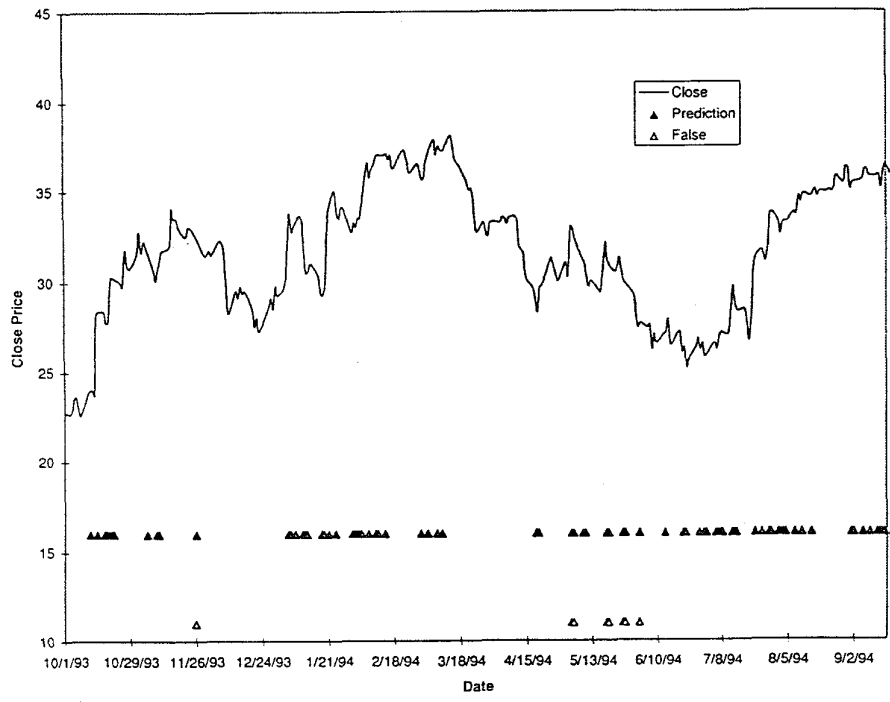


Figure 2. PNN Predictions plotted with price data. $L = 2$, $\sigma = 0.0149$

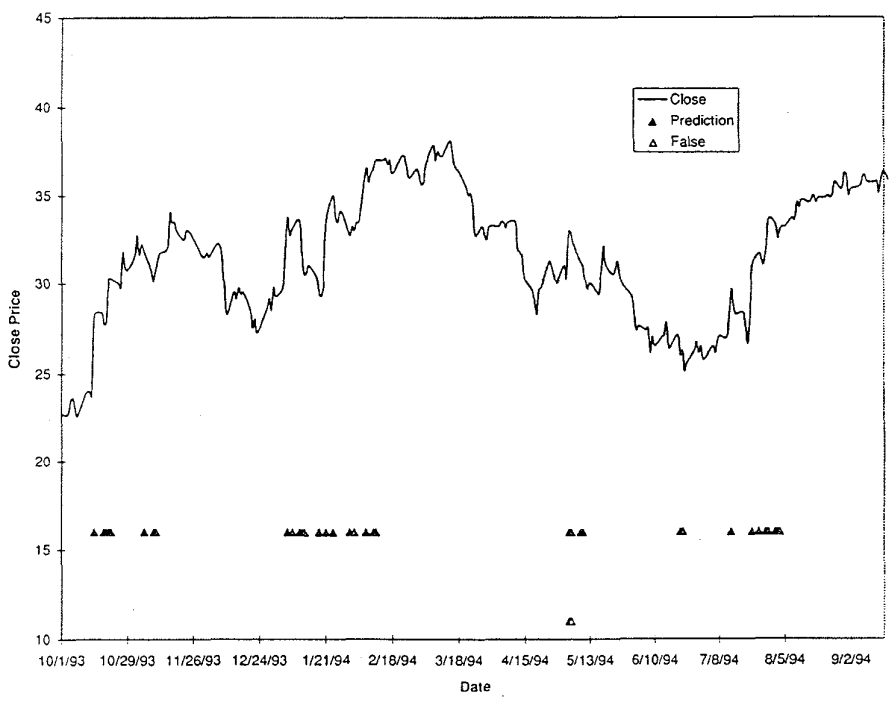


Figure 3. PNN Predictions plotted with price data. $L = 4$, $\sigma = 0.0149$

Figure 3 shows the predictions (for $L = 4$, $\sigma = 0.0149$) plotted with price data. Note the enhanced conservatism results in fewer predictions. We therefore are achieving higher accuracy at the expense of potential lost opportunities. However, our philosophy is that a smaller number of higher quality trading ideas is better than constantly trading. After all, trading invokes transaction costs. Furthermore, by applying this method to a large portfolio of attractive stocks, more than enough good trading opportunities should be available. We therefore consider the choice of making fewer predictions in favor of higher accuracy to be the best policy.

Conclusion

We have demonstrated an approach to short-term stock forecasting. By taking a design approach designed to minimize false alarms, we have achieved high accuracy in return for a lower number of predictions. Use of the PNN has worked well for this approach. Research is currently under way to compare performance of different models (not limited to neural networks) for this problem.

References

1. Klimasauskas, C. "Basics of Building Market Timing Systems: Making Money with Neural Networks," Tutorial at the IEEE World Congress on Computational Intelligence, Orlando, 1994.
2. Jurik, Mark, "The care and feeding of a neural network". *Futures*, Vol 21, pp. 40 - 42, Oct. 1992, Cedar Falls, Iowa.
3. Specht, D. "Probabilistic Neural Networks," *Neural Networks*, Vol. 3, pp. 109-118, 1990.
4. Specht, D., and Romsdahl, H. "Experience with Adaptive Probabilistic Neural Networks and Adaptive General Regression Neural Networks," *Proceedings of the IEEE World Congress on Computational Intelligence*, Vol. 2, pp. 1203-1208, 1994.
5. Musavi, M., Kalantri, K., Ahmed, W., and Chan, K. "A Minimum Error Neural Network," *Neural Networks*, Vol. 6, pp. 397-407, 1993.
6. Prokhorov. D., and Wunsch, D. "A Threshold Polynomial Neural Network," To appear in *Scientific Siberian*, AMSE Press.
7. "A Commodity Trading Model Based on a Neural Network-Expert System Hybrid," Karl Bergerson and Donald C. Wunsch II, *Neural Network in Finance and Investing*, Robert R. Trippi and Efraim Turban, eds., Probus, Chicago, 1993.