

# A Comparative Study of Time Series Prediction Based on Neural Network and the Ornstein-Uhlenbeck Process with Jumps

Yaohui Bai<sup>1,\*</sup>, Yan Tu<sup>1</sup> and Huayang Li<sup>1</sup>

<sup>1</sup>Cloud Computation and Big Data Research Center, Jiangxi University of Finance and Economics, Nanchang, 330013, China

<sup>2</sup>School of Finance, Southwestern University of Finance and Economics, Chengdu, 610074, China

\*Corresponding author

**Abstract**—The research of stock price prediction is very important. Traditionally, the stock price is usually processed as a time series. However, the modelling of such time series is extremely important and vital, and has been attracting the attention of both practitioners and researchers. In this paper, the dynamic neural network model and the OU process with jumps are used to analyze the stock prices respectively, and the two models are compared by fitting and prediction performance. The experimental results show that the OU process with jumps is superior to the dynamic neural network for stochastic time series prediction.

**Keywords**—stochastic time series; dynamic neural network; the ornstein-uhlenbeck process; stock prices

## I. INTRODUCTION

The research of stock prices volatility not only has important academic significance, but also has important practical significance. Traditionally, the stock price is usually processed as a time series. The modelling of such time series is extremely important and vital, and has been attracting the attention of both practitioners and researchers. However, it is also considered a rather difficult problem, due to the many complex features frequently present in stock price series, such as irregularities, volatility, trends and noise, and so on. A number of techniques have been developed in an attempt to model stock prices series based on their present and past behavior.

Traditional time series modelling technologies, such as autoregressive integrated moving average (ARIMA) [1], exponential smoothing [2], decomposition [3], etc., have been widely and successfully used. More recently a number of machine learning techniques, such as neural networks [4], fuzzy systems [5], genetic algorithm [6], and SVM [7] are becoming promising directions in this fields. Some showed improvement compared to traditional models.

Among modern intelligent methods, neural networks method is an important technology. Neural network is usually divided into two categories: dynamic network and static network. The static network is a simple forward network without feedback and delay. The output is calculated directly from a forward mapping relation established by the neural network. Compared with the static neural network, the dynamic

neural network is widely used in time series analysis because of its feedback loop characteristics.

Mean reversion is a mathematical concept, which is often used in stock prices analysis [8]. In general, the essence of this definition is to assume that both the high and low prices of stock are temporary, and that the stock prices always tend to its average. In stock price analysis, mean reversion mainly includes two aspects: identifying the price range of stock trading and calculating the average stock price. The Ornstein-Uhlenbeck process of mean reversion is a stochastic process. The process is a stationary Gauss-Markov process, which means that it is both a Gaussian and Markovian process, and is the only nontrivial process that satisfies these three conditions. The OU process with jumps is widely used in the pricing analysis of financial derivatives [9,10].

In this paper, the dynamic neural network model and the OU process with jumps are used to analyze the stock prices respectively, and the two models are compared by fitting and prediction performance.

## II. METHODOLOGY

### A. Dynamic Neural Network Model

Dynamic neural networks distinguish themselves from static neural networks in that they have at least one feedback loop. The feedback loops result in a nonlinear dynamical behavior of dynamic neural networks. Dynamic networks with feedback can be transformed between open-loop and closed-loop modes. Open-loop networks use for training, and closed-loop networks make forecasting. It is shown in Figure 1. A dynamic neural network structure that contains state feedback may provide more computational advantages than a static neural structure, which contains only a feed forward neural structure. Although dynamic networks can be trained by the same gradient-based algorithms used for static networks, the performance of the algorithms on dynamic networks can be quite different, and the gradient must be computed in a more complex way.

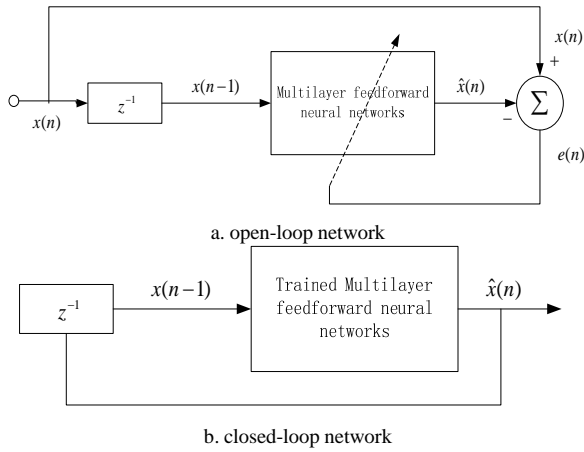


FIGURE I. DYNAMIC NEURAL NETWORK STRUCTURE

Typically, the errors of training sample are defined by:

$$\xi(n) = \frac{1}{2} \sum_j e_j^2(n) \quad (1)$$

where  $j$  is  $j$ th output unit, and

$$e_j^2(n) = d_j(n) - y_j(n) \quad (2)$$

The learning goal is to minimize the total error:

$$\xi_{total} = \sum_n \xi(n) \quad (3)$$

and its gradient is:

$$\frac{\partial \xi_{total}}{\partial w_{ji}} = \sum_n \frac{\partial \xi(n)}{\partial w_{ji}} \quad (4)$$

For (5), the method of temporal back-propagation learning gives:

$$\frac{\partial \xi_{total}}{\partial w_{ji}^l} = \sum_n \frac{\partial \xi_{total}}{\partial v_j^{l+1}(n)} \frac{\partial v_j^{l+1}(n)}{\partial w_{ji}^l(n)} \quad (5)$$

where  $v_j^{l+1}(n) = \sum_i w_{ji}^l x_i^l(n)$  is the net input of  $j$ th unit in layer  $l+1$ .

By the expanded form of (5), the formula of  $w(n)$  is

$$w_{ji}^l(n+1) = w_{ji}^l(n) - \eta \frac{\partial \xi_{total}}{\partial v_j^{l+1}(n)} \frac{\partial v_j^{l+1}(n)}{\partial w_{ji}^l(n)} \quad (6)$$

Making use of the fact that  $\frac{\partial v_j^{l+1}(n)}{\partial w_{ji}^l(n)} = x_i^l(n)$  for any unit  $j$ , we can define the local gradient

$$\delta_j^l(n) = - \frac{\partial \xi_{total}}{\partial v_j^l(n)} \quad (7)$$

Then (7) can be rewritten as

$$w_{ji}^l(n+1) = w_{ji}^l(n) - \eta \delta_j^{l+1}(n) x_i^l(n) \quad (8)$$

When  $j$ th unit is an output unit of network, there is

$$\delta_j^l(n) = - \frac{\partial \xi_{total}}{\partial v_j^l(n)} = - \frac{\partial \xi(n)}{\partial v_j^l(n)} = e_j(n) \phi'[v_j^l(n)] \quad (9)$$

With

$$\phi'[v_j^l(n)] = \frac{\partial \phi[v_j^l(n)]}{\partial v_j^l(n)} = \frac{\partial y_j(n)}{\partial v_j^l(n)} \quad (10)$$

When  $j$ th unit is an internal unit of network, there is

$$\delta_j^l(n) = \phi'[v_j^l(n)] \sum_{m \in A} \sum_{n=l}^M \delta_m^{l+1}(n+1) w_{mj}^l(n) \quad (11)$$

where  $A$  represents the set of all neurons receiving input from the  $j$ th unit,  $v_m^{l+1}(n)$  is the net input of  $m$ th unit and  $m \in A$ .

#### B. Ornstein-Uhlenbeck Process with Jumps

The Ornstein-Uhlenbeck process (OU process), is a stochastic process that, was proposed by Uhlenbeck and Ornstein in a physical modelling context. The process describes the velocity of a massive Brownian particle under the influence of friction. Since the original paper appeared, the model has been used in a wide variety of applications areas. In Finance, it is widely used to model interest rates, currency exchange rates, and commodity prices stochastically.

Usually, it is assumed that the stock price process is continuous. However, discrete changes in stock price should occur when significant new information is revealed. Individual stock prices often react abruptly to significant news like Government Work Reports. In addition, they have a prominent

seasonal component, along with reversion to mean levels. Therefore, these characteristics should be incorporated into a model of stock prices.

Usually, the stock price  $S_t$  driven by Poisson jump process is defined by

$$dS_t = u_t dt + \sigma_t dB_t + J_t dN_t \quad (12)$$

and assume that

$$U = f(S_t, t) \quad (13)$$

Then Ito's lemma is expressed as

$$\begin{aligned} dU &= \left( \frac{\partial U}{\partial t} + \frac{\partial U}{\partial S_t} u_t + \frac{1}{2} \frac{\partial^2 U}{\partial S_t^2} \sigma_t^2 \right) dt \\ &+ \frac{\partial U}{\partial S_t} \sigma_t dB_t + [f(S_t, t) - f(S_{t-}, t)] dN_t \\ &= \left( \frac{\partial U}{\partial t} + \frac{\partial U}{\partial S_t} u_t + \frac{1}{2} \frac{\partial^2 U}{\partial S_t^2} \sigma_t^2 \right) dt \\ &+ \frac{\partial U}{\partial S_t} \sigma_t dB_t + [f(S_{t-} + J_t, t) - f(S_{t-}, t)] dN_t \end{aligned} \quad (14)$$

In this paper, the stock price  $S_t$  is modeled as:

$$\log(S_t) = f(t) + X_t \quad (15)$$

The logarithmic model of stock price consists of two components: (1) the deterministic seasonal part  $f(t)$ , modeled as Trigonometric functions; (2) the stochastic part  $X_t$ : modeled as an Ornstein-Uhlenbeck process (mean-reverting) with jumps.

The deterministic seasonal part is modeled as

$$\begin{aligned} f(t) &= s_1 \sin(2\pi t) + s_2 \cos(2\pi t) + s_3 \sin(4\pi t) \\ &+ s_4 \cos(4\pi t) + s_5 \end{aligned} \quad (16)$$

where  $s_i, i = 1, \dots, 5$ , are constant parameters.

The stochastic part  $X_t$  is modeled as

$$dX_t = (\alpha - \kappa X_t) dt + \sigma dW_t + J(\mu_j, \sigma_j) d\Pi(\lambda) \quad (17)$$

The parameters  $\alpha$  and  $\kappa > 0$  are the mean-reversion parameters. Parameter  $\sigma$  is the volatility, and  $W_t$  is a standard

Brownian motion. The jump size is  $J(\mu_j, \sigma_j)$ , with normally distributed mean  $\mu_j$  and standard deviation  $\sigma_j$ .  $\Pi(\lambda)$  is a Poisson process with a jump intensity of  $\lambda$ .

### III. EXPERIMENTS AND RESULTS

In this paper, we use the data of Wuliangye stock (2013/7/1-2014/6/20) in Shenzhen stock market of China to test our proposed method. The testing process mainly choose daily closing price of Wuliangye stock as test data. The selected daily closing price data of Wuliangye stock is shown in Figure 2.

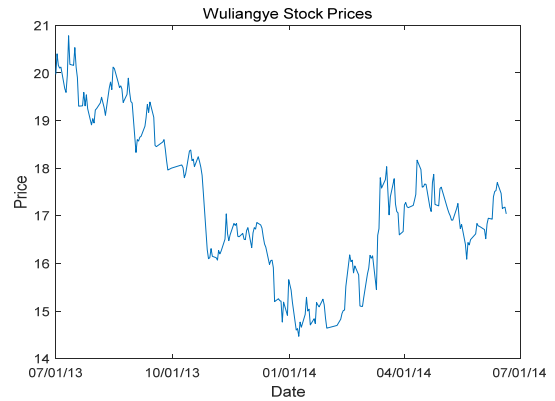


FIGURE II. THE DAILY CLOSING PRICES OF WULIANGYE STOCK

In the case of the dynamic neural network model, we use a two-layer neural network model to fit daily closing price data of Wuliangye stock. The data is randomly divided into three sets as follows: 70% is used for training, 15% is used to validate that the network is generalizing and to stop training before overfitting, and the last 15% is used as a completely independent test of network generalization. The number of hidden neurons is set to 10, and the number of delays is 4. The training uses Levenberg-Marquardt backpropagation, and the activation function uses hyperbolic tangent sigmoid transfer function. Maximum number of epochs to train is 1000, and the learning rate is 0.01.

Figure 3 shows the dynamic neural network model training performance. Figure 4 shows the actual prices and simulated prices of Wuliangye stock. The simulated prices are a multi-step-ahead prediction for 50 days.

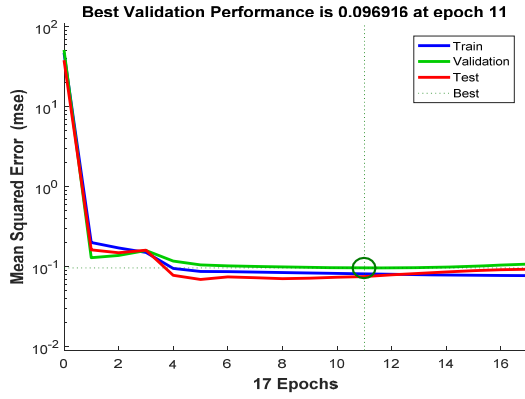


FIGURE III. DYNAMIC NEURAL NETWORK MODEL TRAINING PERFORMANCE

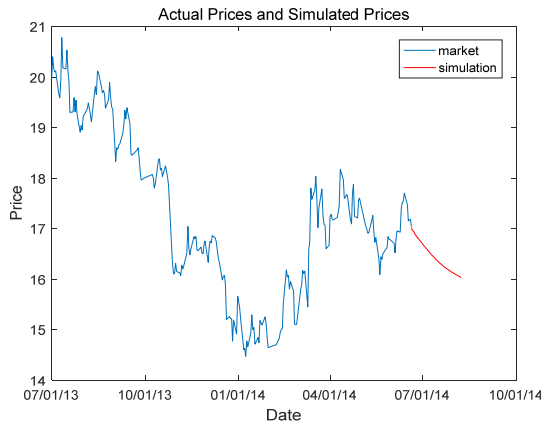


FIGURE IV. THE SIMULATION RESULT OF WULIANGYE STOCK BASED ON DYNAMIC NEURAL NETWORK MODEL

In the case of OU process with jumps, the parameters of deterministic seasonal part are calculated and shown in Table 1. The density function of the stochastic component  $X_t$  given  $X_{t-1}$  is

$$f(X_t | X_{t-1}) = (\lambda \Delta t) N_1(X_t | X_{t-1}) + (1 - \lambda \Delta t) N_2(X_t | X_{t-1}) \quad (18)$$

where

$$N_1(X_t | X_{t-1}) = \frac{1}{\sqrt{2\pi(\sigma^2 + \sigma_j^2)}} \exp\left(\frac{-(X_t - \alpha \Delta t - \phi X_{t-1} - \mu_j)^2}{2(\sigma^2 + \sigma_j^2)}\right)$$

$$N_2(X_t | X_{t-1}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(X_t - \alpha \Delta t - \phi X_{t-1})^2}{2\sigma^2}\right)$$

The parameters  $\alpha, \phi, \mu_j, \sigma^2, \sigma_j^2, \lambda$  can be solved by minimizing the negative log likelihood function:

$$\min - \sum_{t=1}^T \log(f(X_t | X_{t-1})) \quad (19)$$

$$\text{subject to } \phi < 1, \sigma^2 > 0, \sigma_j^2 > 0, 0 \leq \lambda \Delta t \leq 1$$

The results of the above maximum likelihood problem are shown in Table 2.

Figure 5 shows the log price and de-seasonalized log price of Wuliangye stock. Figure 6 shows the actual prices and simulated prices. The simulated prices are also a multi-step-ahead prediction for 50 days.

Comparison of Figure 4 and 6, it is shown that the neural network model, as the most representative machine learning algorithm in recent years, can fit the historical prices well, but cannot simulate the stochastic behaviour of prices in the prediction process while it gives the trend of prices. However, the OU process with jumps gives the prices prediction with stochastic characteristics. So, intuitively, we can say that the OU process with jumps is superior to the dynamic neural network for stochastic time series prediction.

TABLE I. THE PARAMETERS ESTIMATE OF DETERMINISTIC SEASONAL PART

Estimated Coefficients	$s_1$	$s_2$	$s_3$	$s_4$	$s_5$
Estimate	0.0887	-0.0151	-0.0307	-0.2088	2.9473

TABLE II. THE FITTED RESULTS OF THE TEST DATA

parameters	$\alpha$	$\phi$	$\mu_j$	$\sigma^2$	$\sigma_j^2$	$\lambda$
Estimate	0.0484	304.7	1.3265	0.3247	1.2594	0.0004

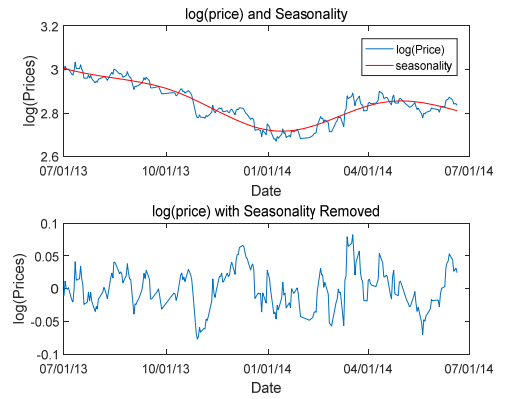


FIGURE V. THE LOG PRICE AND DE-SEASONALIZED LOG PRICE OF WULIANGYE STOCK BASED ON ORNSTEIN-UHLENBECK PROCOESS WITH JUMPS

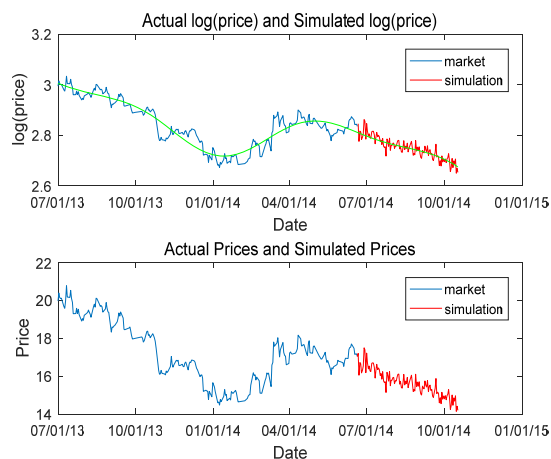


FIGURE VI. THE SIMULATION RESULT OF WULIANGYE STOCK BASED ON ORNSTEIN-UHLENBECK MODEL WITH JUMPS

#### IV. CONCLUSIONS

In this paper, we use dynamic neural network model and OU process with jumps to forecast the stock prices, respectively. The premise of this problem is to compare the prediction performance of the two models. The two models is used to analysis the real data of Wuliangye stock (2013/7/1-2014/6/20) in Shenzhen stock market of China. The experimental results show that the OU process with jumps is superior to the dynamic neural network for stochastic time series prediction.

#### REFERENCES

- [1] Box G, G Jenkins, and G C. Reinsel, Time series analysis: forecasting and control, 4th ed, John Wiley & Sons, Inc. 2008.
- [2] James W. Taylora, and Ralph D. Snyderb, "Forecasting intraday time series with multiple seasonal cycles using parsimonious seasonal exponential smoothing," Special Issue on Forecasting in Management Science, vol. 40(6), 2009, pp. 748-757, doi:10.1016/j.omega.2010.03.004.
- [3] Theodosiou, Marina, "Forecasting monthly and quarterly time series using STL decomposition," International Journal of Forecasting, vol. 27(4), 2011, pp. 1178-1195, doi:10.1016/j.ijforecast.2010.11.002.
- [4] Charles Wong and Massimiliano Versace, "CARTMAP: a neural network method for automated feature selection in financial time series forecasting," Neural Computing and Applications, vol. 21(5), 2012, pp. 969-977, doi:10.1007/s00521-012-0830-8.
- [5] Ivette Luna and Rosangela Ballini, "Top-down strategies based on adaptive fuzzy rule-based systems for daily time series forecasting," International Journal of Forecasting, vol. 27(3), 2011, pp. 708-724, doi:10.1016/j.ijforecast.2010.09.006.
- [6] Yi-Hui Liang, "Combining seasonal time series ARIMA method and neural networks with genetic algorithms for predicting the production value of the mechanical industry in Taiwan," Neural Computing and Applications, vol. 18(7), 2009, pp. 833-841, doi:10.1016/j.ijforecast.2010.09.006.
- [7] Guo, ZQ, Wang, HQ, and Liu, Q, "Financial time series forecasting using LPP and SVM optimized by PSO," SOFT COMPUTING, vol. 17(5), 2013, pp. 805-818, doi:10.1007/s00500-012-0953-y.
- [8] Tretyakov M V. Introductory course on financial mathematics[M]. London: Imperial College Press, 2013
- [9] Leung, Tim; Li, Xin. "Optimal Mean Reversion Trading with Transaction Costs and Stop-Loss Exit". International Journal of

Theoretical & Applied Finance. 18: 1550020. 2015. doi:10.1142/S021902491550020X.

- [10] Seifert, Jan, Uhrig-Homburg, Marliese. "Modelling Jumps in Electricity Prices: Theory and Empirical Evidence". Review of Derivatives Research, Vol. 10, pp 59-85, 2007.