이학석사학위논문

Nonlinear ARMA-GARCH Forecasting for S&P500 Index based on Recurrent Neural Networks

순환신경망 기반 비선형 ARMA-GARCH 모형을 이용한 S&P500 지수 예측

2019 년 8 월

서울대학교 대학원

통계학과

정 용 진

이학석사학위논문

Nonlinear ARMA-GARCH Forecasting for S&P500 Index based on Recurrent Neural Networks

순환신경망 기반 비선형 ARMA-GARCH 모형을 이용한 S&P500 지수 예측

2019 년 8 월

서울대학교 대학원

통계학과

정 용 진

Nonlinear ARMA-GARCH Forecasting for S&P500 Index based on Recurrent Neural Networks

순환신경망 기반 비선형 ARMA-GARCH 모형을 이용한 S&P500 지수 예측

지도교수 이 상 열

이 논문을 이학석사 학위논문으로 제출함

2019 년 6 월

서울대학교 대학원

통계학과

정 용 진

정용진의 이학석사 학위논문을 인준함

2019 년 6 월

위 원 장	이 영 조	(인)
부위원장	이 상 열	(인)
위 원	장 원 철	(인)

Abstract

A nonlinear ARMA-GARCH model is proposed for forecasting daily stock market returns. The only difference from the linear ARMA-GARCH is the conditional mean component. Two parameters are added and the hyperbolic tangent function is utilized to give a nonlinearity. The nonlinear ARMA-GARCH is solved by the recurrent neural network concept. In order to show the practical applicability of the proposed nonlinear ARMA-GARCH model, daily algorithmic trading is carried out with historical S&P500 daily closing index from 1950 to 2018. It is shown that the proposed nonlinear ARMA-GARCH model outperforms the linear ARMA-GARCH model in terms of financial and statistical measures.

Keywords: Nonlinear ARMA-GARCH, recurrent neural networks, financial time series forecasting, S&P500.

Student Number: 2017-29033

Contents

List of Figures

List of Tables

Chapter 1

Introduction

For several decades, forecasting stock price has been considered as the most important work by personal or organization investors: see Croston (1972); Kim (2003); Wang et al. (2009). However, forecasting stock price has difficulties. In order to get a reliable model, we need a large number of data sets, but observational time series data is highly limited. If we increase observation frequency, then we can have a lot of data. However, the increased frequency makes high volatility and volatility persistence: see Enders (2010). Because of the trade-off between the number of data and the volatility, researchers have suggested remedies. A crucial remedy is to concurrently model conditional mean and variance by statistical models. Box and Jenkins (1994) suggested Autoregressive Moving Average (ARMA) model for modeling conditional mean and Engle (1982); Bollerslev (1986) suggested Generalized Autoregressive Conditional Heteroscedasticity (GARCH) model for modeling conditional variance. Finally, Li et al. (2002) suggested the ARMA-GARCH model for modeling conditional mean and variance simultaneously. The ARMA-GARCH model has been con-

sidered to conduct the daily stock price forecasting: see Karanasos (2002).
Nowadays, deep neural networks successfully substitute statistical methods in many fields, especially image classification and language translation: see He et al. (2016); Szegedy et al. (2015); Cho et al. (2014). Moreover, there have been lots of researches for time series forecasting with Recurrent Neural Networks (RNN): see Hochreiter and Schmidhuber (1997); Zhang et al. (1998); Gamboa (2017); Petneházi (2019). Without considering properties of data, however, an imprudent application of deep neural networks may not give a proper result. In time series forecasting, for instance, if a model does not satisfy stationarity, the forecasting would be nothing but an extrapolation of the model. Well known RNN kernels could adapt several concepts for time series forecasting, e.g. lag and seasonality: see Petneházi (2019). However, a volatility persistency which is crucial characteristics for financial time series data cannot be modeled by the ordinary RNN kernels since they only model conditional mean component. In order to appropriately model financial time series data, therefore, it is essential to consider the conditional mean and variance simultaneously.
The relationship between the conditional mean or variance, and the previous information could be linear or nonlinear: see Tong (1983); Fan and Yao (2003); Wang (2008). It has been shown that one model could not dominate the other regarding the linear and nonlinear models. This paper suggests a nonlinear ARMA-GARCH model that is slightly modified from a linear ARMA-GARCH model but retains the parsimonious property and stationarity.
Chapter 2 contains the properties of the S&P500 daily closing index used to verify the suggested model. Chapter 3 explains the specific formulation of the suggested model and trading strategies. Chapter 4 shows the algorithmic trading result. Finally, Chapter 5 briefly reviews the result of this paper and discusses the potentials and limitations of this study.

Chapter 2

Data Description

S&P500 daily closing index is obtained from yahoo finance web site. Figure 2.1 shows the closing index of S&P500 from Jan. 1950 to Dec. 2018. It is possible to see a trend in Figure 2.1. In time series analysis, the stationarity is a very import assumption: see Box and Jenkins (1994). The stationarity is that mean, variance and autocorrelation structures are all constant over time. If there is a trend in data, it is obvious that the mean is not constant over time. Then, it is impossible to apply time series models because most statistical forecasting algorithms are based on the stationarity condition. Figure 2.1 shows the logarithmic returns. Equation 2.1 shows the relationship between original data, return and logarithmic return.

$$r_t = \log(\frac{y_t}{y_{t-1}}) = \log(1 + \frac{y_t - y_{t-1}}{y_{t-1}}) \approx \frac{y_t - y_{t-1}}{y_{t-1}} = R_t, \tag{2.1}$$

where r_t is the logarithmic return and R_t is the return. The approximation in Equation 2.1 is feasible when $R_t \approx 0$. The first differencing or return is widely

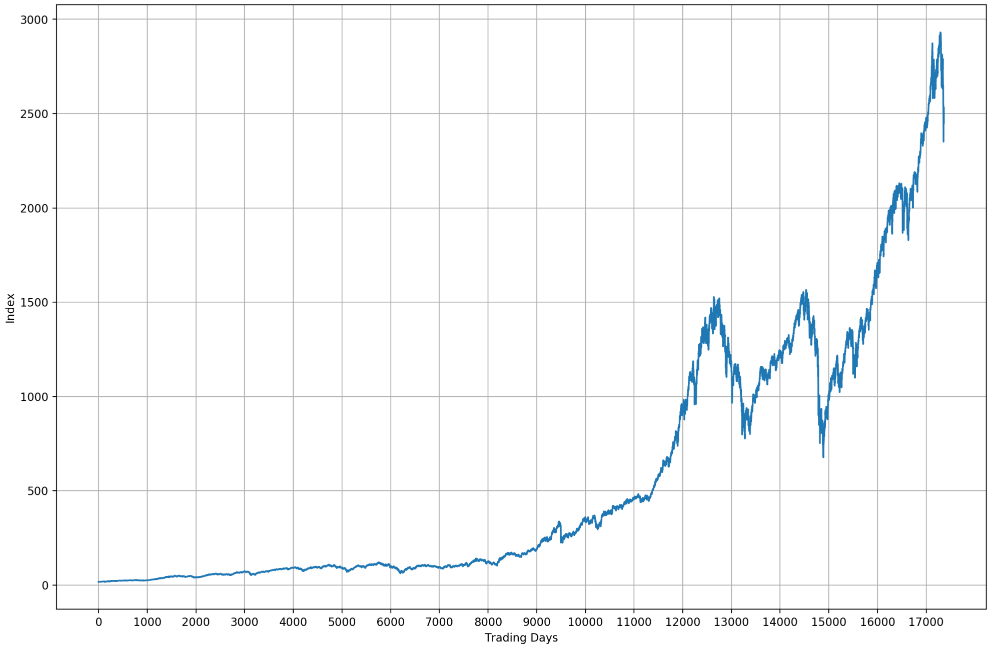

Figure 2.1 Daily closing prices of S&P500

used to detrend original data. However, ordinary stock price data has a gradual increasing tendency, so the first differencing or return is usually negatively skewed. The logarithmic return r_t is always less than R_t and almost the same when $R_t \approx 0$ thus the sample mean of r_t is almost zero. Because of that property, the logarithmic return is widely used to forecast stock price data.

Figure 2.2 is the logarithmic return from the S&P500 closing index in Figure 2.1. The trend is removed in Figure 2.2. Although the logarithmic returns could be assumed to be stationary, a single statistical time series model may not explain the whole data. To remedy the problems, the rolling window concept is introduced in Section 3.3.

Furthermore, it is evident that the daily logarithmic return is highly heteroscedastic. Therefore, the ARMA-GARCH model is considered in this paper.

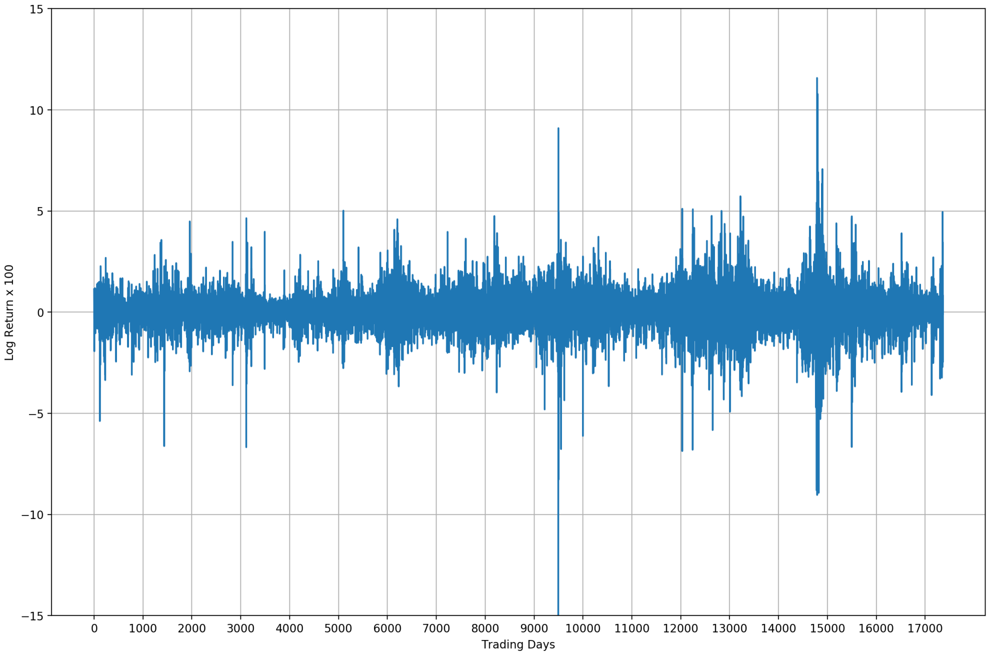

Figure 2.2 Logarithmic returns of S&P500

Chapter 3

Model Description

In this chapter, the proposed nonlinear ARMA-GARCH model is introduced and a corresponding RNN structure is explained. Furthermore, trading algorithms for backtesting are defined. Equation 3.1 denotes the linear ARMA(m, n)-GARCH(p, q) model.

$$
\begin{aligned}
y_t &= \mu_t + \epsilon_t,\ \ \epsilon_t = \sigma_t \xi_t,\ \ \xi_t \sim \mathrm{WN}(0,1), \\
\mu_t &= \mu + \sum_{i=1}^{m} \phi_i (y_{t-i} - \mu) + \sum_{j=1}^{n} \theta_j \epsilon_{t-j}, \\
\sigma_t^2 &= \omega + \sum_{i=1}^{q} \alpha_i \epsilon_{t-i}^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2,
\end{aligned}
\tag{3.1}
$$

where y_t is 100 times the logarithmic return, which is equivalent to r_t in Equation 2.1, μ is an unconditional mean, μ_t is a conditional mean, σ_t is a conditional standard deviation and $\omega, \alpha_i' s, \beta_i' s > 0$.

3.1 Nonlinear ARMA-GARCH Model

For the univariate time series model, observation y_t is response variables as well as explanatory variables. Because the linear model is highly affected from outliers in explanatory variables, a nonlinear ARMA(m, n)-GARCH(p, q) model is suggested as follows:

$$\begin{aligned} y_t &= \mu_t + \epsilon_t,\ \epsilon_t = \sigma_t \xi_t,\ \xi_t \sim \mathrm{WN}(0,1), \\ \mu_t &= \mu + \sum_{i=1}^{m} \phi_i \tanh(\Phi(y_{t-i} - \mu)) + \sum_{j=1}^{n} \theta_j \tanh(\Theta \epsilon_{t-j}), \\ \sigma_t^2 &= \omega + \sum_{i=1}^{q} \alpha_i \epsilon_{t-i}^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2, \end{aligned} \tag{3.2}$$

where the meaning of the parameters is the same as Equation 3.1. Note that an analytic formulation of the proposed nonlinear ARMA-GARCH model is almost analogous to the original linear ARMA-GARCH model. The only difference is that two more parameters, Φ and Θ, and nonlinear function by hyperbolic tangent function. The two parameters control the degree of boundedness by the hyperbolic tangent function. One might wonder why the two parameters, Φ and Θ, are applied before the hyperbolic tangent transformation. The reason is that the unconditional variance of y_t and ϵ_t are different as in Equation 3.3 and 3.4 so the two parameters are added to AR part and MA part, respectively.

$$\begin{aligned} \mathrm{Var}(y_t) &= E\,\mathrm{Var}(y_t | I_{t-1}) + \mathrm{Var}(E(y_t | I_{t-1})) \\ &= E\sigma_t^2 + \mathrm{Var}(\mu_t) \\ &= \sigma^2 + \mathrm{Var}(\mu_t), \end{aligned} \tag{3.3}$$

where $I_t = \{y_t, y_{t-1}, \cdots, \epsilon_t, \epsilon_{t-1}, \cdots\}$, so called previous information.

$$\begin{aligned} \text{Var}(\epsilon_t) &= E\,\text{Var}(\epsilon_t | I_{t-1}) + \text{Var}(E(\epsilon_t | I_{t-1})) \\ &= E\sigma_t^2 \\ &= \sigma^2. \end{aligned} \tag{3.4}$$

It is also noted that the linear and nonlinear ARMA-GARCH models give similar results when Φ and Θ are closed to zero since $\tanh(x) \approx x$ when $x \ll 1$. Even though the nonlinear ARMA-GARCH model is modified from the linear ARMA-GARCH model, the proposed nonlinear model has similar properties with the linear model and is still parsimonious. That transformation could be applied to conditional variance but Miah and Rahman (2016) showed that the GARCH(1,1) is enough to explain the volatility of financial data. For that reason, an order of all GARCH part considered in that paper is set as $p = 1, q = 1$.

3.2 Recurrent Neural Networks Structure

Figure 3.1 shows the RNN structure for the proposed nonlinear ARMA(m, n)-GARCH(p, q) model. L_t is a conditional Gaussian likelihood in Equation 3.5 and the definition of other variables is the same as in Equation 3.2.

$$L_t = \frac{1}{\sqrt{2\pi\sigma_t^2}} \exp\{-\frac{1}{2}\frac{(y_t - \mu_t)^2}{\sigma_t^2}\}. \tag{3.5}$$

By using the conditional likelihood, L_t, a cost function for the RNN is set as follows:

$$C = -\frac{2}{T}\sum_{t=1}^{T} \log(L_t) = \log(2\pi) + \frac{1}{T}\sum_{t=1}^{T} \log(\sigma_t^2) + \frac{(y_t - \mu_t)^2}{\sigma_t^2}, \tag{3.6}$$

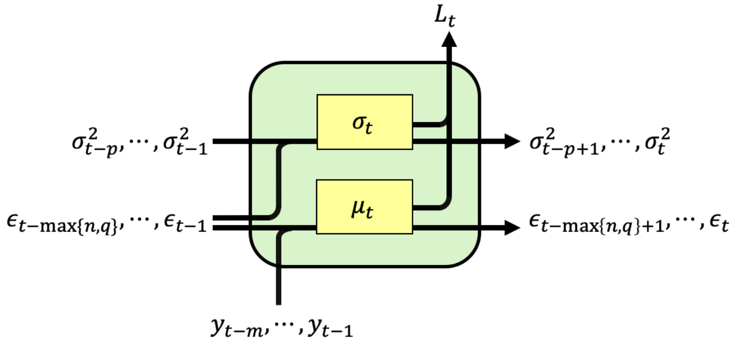

Figure 3.1 Schematic recurrent neural network structure for the nonlinear ARMA(m, n)-GARCH(p, q) model

where T is sample size and C is the negative conditional log-likelihood and is minimized by the established RNN structure in Figure 3.1 with respect to parameters, $\mu, \phi_1, \cdots, \phi_m, \Phi, \theta_1, \cdots, \theta_n, \Theta, \omega, \alpha_1, \cdots, \alpha_q, \beta_1, \cdots, \beta_p$.

The recurrent graph is customized by a Tensorflow library since an intrinsic function in deep neural network libraries is not comparable with the structure in Figure 3.1. In order to ensure the stationarity, the GARCH parameters $\omega, \alpha_1, \beta_1$ should have values in $(0, 1)$ and satisfy $\alpha_1 + \beta_1 < 1$. This restriction, however, causes serious trouble in computation, and thereby, an optimal solution is found by the transformed parameters $x_\omega, x_{\alpha_1}, x_{\beta_1}$ as follows:

$$\begin{aligned} \omega &= sigmoid(x_\omega) \\ \alpha_1 &= sigmoid(x_{\alpha_1}) \\ \beta_1 &= sigmoid(x_{\beta_1}), \end{aligned} \tag{3.7}$$

where $sigmoid(x) = \frac{1}{1+\exp(-x)} \in (0, 1)$.

3.3 Model Selection

As mentioned in Chapter 2, it is hard to explain 17,000 trading days data only with a single statistical time series model. Thus, the rolling window concept is adapted in this analysis: see Zivot and Wang (2006). An optimum model is selected for every rolling window among a constrained function class in Equation 3.8, based on Cross-Validation (CV) or Akaike Information Criterion (AIC): see Geisser (1993); Akaike (1974). Each window size is 500 and one day ahead out of the window is forecasted by the selected model.

$$\begin{aligned}\mathcal{F}_l^{M,N} &= \{\mu_t^{m,n} : \mu_t^{m,n} = \mu + \sum_{i=1}^{m} \phi_i (y_{t-i} - \mu) + \sum_{j=1}^{n} \theta_j \epsilon_{t-j},\ 0 \le m \le M, 0 \le n \le N\},\\ \mathcal{F}_n^{M,N} &= \{\mu_t^{m,n} : \mu_t^{m,n} = \mu + \sum_{i=1}^{m} \phi_i \tanh(\Phi(y_{t-i} - \mu)) + \sum_{j=1}^{n} \theta_j \tanh(\Theta \epsilon_{t-j}),\\ &\qquad 0 \le m \le M, 0 \le n \le N\},\end{aligned} \tag{3.8}$$

where M and N are the maximum orders of ARMA(m, n) and $\mathcal{F}_l^{M,N}$ is the linear function class and $\mathcal{F}_n^{M,N}$ is the proposed nonlinear function class. If $\mu_t \in \mathcal{F}_l^{M,N}$, it is equivalent to the original ARMA-GARCH model, and if $\mu_t \in \mathcal{F}_n^{M,N}$, it is as to the nonlinear ARMA-GARCH model.

When the model selection criterion is CV, the initial 400 time points are used for training, the next 100 time points are used for validation, and the orders of ARMA that give minimum validation cost, $-2\sum_{t=401}^{500} \log(L_t)$, are selected. Finally, all 500 time points are used to fit the nonlinear ARMA-GARCH with the selected orders and that forecasts one day ahead out of window daily return. When the model selection criterion is AIC, all 500 time points are used in training and the model that gives minimum AIC, $-2\sum_{t=1}^{500} \log(L_t) + 2|M|$, is selected for forecasting one day ahead out of window daily return. Note that $|M|$ is

Algorithm 1 Forecasting Algorithm with CV

for $w = 1$ to 16860 **do** ▷ 16860 rolling windows
  $TR_w = \{y_w, \cdots, y_{w+399}\}$ ▷ Training set in wth rolling window
  $VA_w = \{y_{w+400}, \cdots, y_{w+499}\}$ ▷ Validation set in wth rolling window
  $TE_w = y_{w+500}$ ▷ Test set in wth rolling window
  for $\mu_t \in \mathcal{F}^{3,3}$ **do** ▷ $\mathcal{F}^{3,3} = \mathcal{F}_l^{3,3}$ for linear and $\mathcal{F}^{3,3} = \mathcal{F}_n^{3,3}$ for nonlinear
    Fit by RNN with μ_t for TR_w
    Get validation cost for VA_w
  Select μ_t that minimizes validation cost
  Fit by RNN with the selected μ_t for $TR_w \cup VA_w$
  Forecast TE_w

Algorithm 2 Forecasting Algorithm with AIC

for $w = 1$ to 16860 **do** ▷ 16860 rolling windows
  $TR_w = \{y_w, \cdots, y_{w+499}\}$ ▷ Training set in wth rolling window
  $TE_w = y_{w+500}$ ▷ Test set in wth rolling window
  for $\mu_t \in \mathcal{F}^{3,3}$ **do** ▷ $\mathcal{F}^{3,3} = \mathcal{F}_l^{3,3}$ for linear and $\mathcal{F}^{3,3} = \mathcal{F}_n^{3,3}$ for nonlinear
    Fit by RNN with μ_t for TR_w
    Get AIC for TR_w
  Select μ_t that minimizes AIC
  Forecast TE_w

the number of estimated parameters. For the linear ARMA(m, n)-GARCH(1,1) model, $|M| = 4 + m + n$, and for the nonlinear ARMA(m, n)-GARCH(1,1) model, $|M| = 4 + m + n + I(m > 0) + I(n > 0)$, where $I(x)$ is an indicator function. The indicator function is for the parameters, Φ and Θ.

The function classes for the conditional mean component is in Equation 3.8. The maximum order of the linear and nonlinear ARMA-GARCH models is set to $M = 3$ and $N = 3$. Algorithm 1 and 2 denote the model selection and forecasting scheme based on CV and AIC, respectively.

3.4 Trading Strategies

For the backtest, it is assumed that there is no trading delay nor commission. Therefore, the performance achieved in real trading would be less than this research. One naive strategy (buy-and-hold) and four algorithmic long-short trading strategies are investigated: see Jacobs et al. (1999). The buy-and-hold strategy is literally to buy stock and hold it forever. The 1st algorithmic strategy, called a linear-CV strategy, is $\mu_t \in \mathcal{F}_l^{3,3}$ with CV, the 2nd one, called a linear-AIC strategy, is $\mu_t \in \mathcal{F}_l^{3,3}$ with AIC, the 3rd one, called a nonlinear-CV strategy, is $\mu_t \in \mathcal{F}_n^{3,3}$ with CV, the 4th one, called a nonlinear-AIC strategy, is $\mu_t \in \mathcal{F}_n^{3,3}$ with AIC. For each rolling window, the best μ_t in the function class corresponding to the strategy is used to forecast a one day ahead out of window daily return. If the forecasting result is negative, the stock is shorted at the previous close, while if it is positive, it is longed. In order words, if the algorithmic strategies forecast the future price to go up, an invested money depends on the future price. On the other hand, the algorithmic strategies forecast the future price to go down, the invested money oppositely depends on the future price. It is clear that if the algorithms correctly forecast to increase or decrease, then the equity increases, but if the algorithms incorrectly forecast to increase or decline, then the equity decreases.

Chapter 4

Results

The linear ARMA-GARCH model is solved by the rugarch R package: see Ghalanos (2019). The proposed nonlinear ARMA-GARCH model is solved by the RNN kernel customized by Tensorflow library in Python: see Abadi et al. (2015).

Figure 4.1 shows the equity curves for the algorithmic backtest trading result. The starting point of the forecasting is Jan. 1952 and the day is a control point with equity 1. It is noted that 250 trading days are almost equivalent to a year. The final cumulative equity is 100 for buy-and-hold, 10,000 for linear-AIC, 50,000 for linear-CV, 1,500,000 for nonlinear-CV, 2,000,000 for nonlinear-AIC. The proposed nonlinear ARMA-GARCH with AIC is 20,000 times profitable than the buy-and-hold. Furthermore, the nonlinear ARMA-GARCH model outperforms the linear ARMA-GARCH model in terms of the equity curves.

Before the about 7500th trading day (about Jan.1980), the equity to investment of the 4 algorithmic trading strategies dominate the naive, or buy-and-hold strategy. After the 7500th trading day, however, the two linear ARMA-GARCH based strategies are worse than the buy-and-hold strategy and the two nonlin-

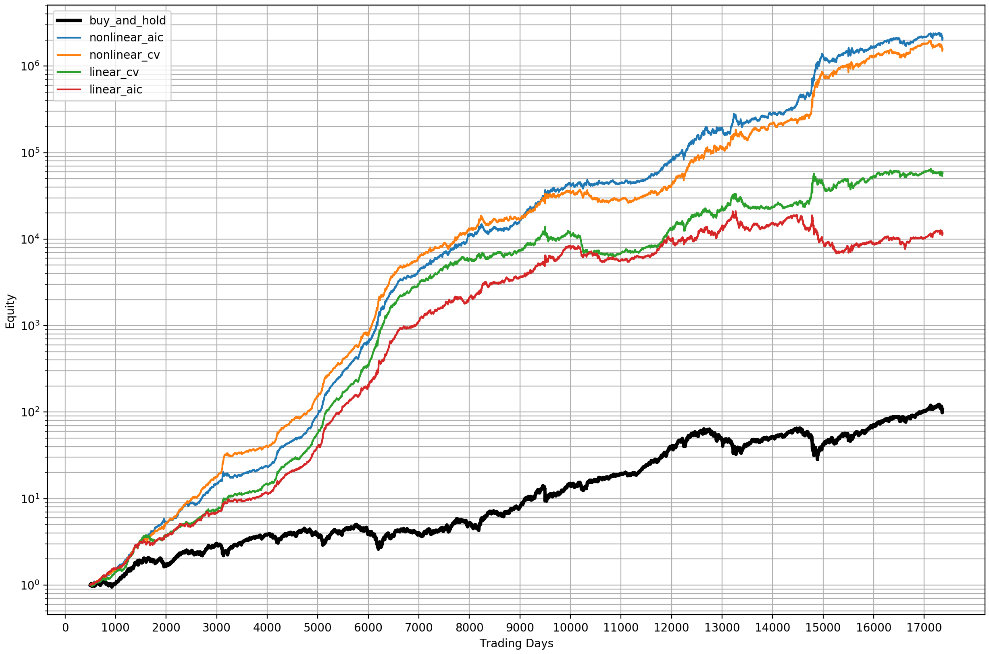

Figure 4.1 Equity curves of buy-and-hold and algorithmic trading strategies

ear ARMA-GARCH based strategies are better than buy-and-hold strategy. It is possible to identify that logarithmic returns after the 7500th trading day have more outliers and fat-tailed distribution than before the 7500th trading day in Figure 2.2. Presumably, the property makes the equity curves of the linear model based trading strategies worse than the equity curve of buy-and-hold strategy after the 7500th trading day.

Table 4.1 contains hit rates of 4 algorithmic trading strategies corresponding to true return intervals. For a designated return interval, if a forecasted return has the same sign, it is regarded as a hit. Note that if the true return, y_t, is zero, the long-short trading strategy does not affect equity on no condition. It is interesting to note that hit rates for positive returns are much higher than those for negative returns. It is because the logarithmic returns are still skew-

Table 4.1 Forecasting hit rates of algorithmic trading strategies

Return (%)	Proportion	Accuracy			
		Nonlinear		Linear	
		CV	AIC	CV	AIC
$y_t < -2$	0.022	0.396	0.388	0.369	0.402
$-2 < y_t < -1$	0.077	0.418	0.414	0.397	0.403
$-1 < y_t < -0.5$	0.119	0.440	0.450	0.391	0.408
$-0.5 < y_t < 0$	0.247	0.333	0.331	0.310	0.358
$y_t < 0$	0.465	0.369	0.365	0.344	0.379
$0 < y_t < 0.5$	0.282	0.707	0.715	0.723	0.688
$0.5 < y_t < 1$	0.144	0.719	0.724	0.735	0.687
$1 < y_t < 2$	0.081	0.716	0.729	0.714	0.702
$2 < y_t$	0.022	0.712	0.701	0.673	0.646
$y_t > 0$	0.529	0.712	0.719	0.723	0.688
$y_t \neq 0$	0.994	0.551	0.553	0.546	0.544

negative. The rank of the final cumulative equity in Figure 4.1 is in accordance with the rank of hit rate in the last row in Table 4.1.

Table 4.2 contains the measure of the accuracy of the 4 algorithmic trading strategies in terms of Mean Absolute Percentage Error (MAPE) and Mean Squared Error (MSE): see Myttenaere et al. (2016). The following equations denote MAPE and MSE, respectively.

$$MAPE = \frac{100}{T}\sum_{t=1}^{T}\left|\frac{y_t - \hat{y}_t}{y_t}\right|, \tag{4.1}$$

where y_t is the true return, $\hat{y}_t$ is the forecasted return, T is the number of forecasted returns, 16860 in this analysis.

$$MSE = \frac{1}{T}\sum_{t=1}^{T}(y_t - \hat{y}_t)^2. \tag{4.2}$$

Table 4.2 Forecasting accuracies of algorithmic trading strategies

Measure of accuracy	Nonlinear		Linear	
	CV	AIC	CV	AIC
MAPE	146.3	146.7	158.0	163.3
MSE	0.9414	0.9398	0.9557	0.9618

The rank of the final cumulative equity in Figure 4.1 is in accordance with the rank of MSE in Table 4.2 but the rank for MAPE is slightly different. However, the nonlinear ARMA-GARCH based algorithmic trading algorithms are still better than the linear ones.

Chapter 5

Concluding Remarks

It is shown that the practical applicability of the proposed RNN based nonlinear ARMA-GARCH model for historical S&P500 daily return forecasting. The linear ARMA-GARCH model has parsimoniousness and stationarity and RNN has nonlinearity. The proposed nonlinear ARMA-GARCH model is constructed to have the characteristics of the linear model and RNN simultaneously. The financial measure (equity curve) and the statistical measure (MAPE and MSE) show that the proposed nonlinear model outperforms the linear model.

The proposed nonlinear ARMA-GARCH model preliminarily combines the theories of statistics and practicality of neural networks. There could be a lot of variations in the nonlinear structure. Moreover, the RNN kernel could be more elaborated with exogenous variables, e.g. volume, fundamental data, news data and etc. Definitely, the trading strategy considered in this paper could be diversified by combining several stocks simultaneously. Then the diversified trading strategy would be more close to the real algorithmic trading system.

Bibliography

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3):307–327.

Box, G. E. P. and Jenkins, G. (1994). *Time series analysis: forecasting and control*. Prentice Hall, New Jersey.

Cho, K., van Merriënboer, B., Gulcehre, C., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder–decoder for statistical machine translation. *Proceedings of the Con-*

ference on Empirical Methods in Natural Language Processing, pages 1724–1734.

Croston, J. D. (1972). Forecasting and stock control for intermittent demands. *Journal of the Operational Research Society*, 23(3):289–303.

Enders, W. (2010). *Applied time series analysis.* Wiley, New York.

Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econometrica*, 50(4):987–1007.

Fan, J. and Yao, Q. (2003). *Nonlinear time series: nonparametric and parametric methods.* Springer-Verlag, New York.

Gamboa, J. C. B. (2017). Deep learning for time-series analysis. *arXiv preprint.*

Geisser, S. (1993). *Predictive Inference.* Chapman and Hall, New York.

Ghalanos, A. (2019). *rugarch: Univariate GARCH models.* R package version 1.4-1.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Jacobs, B. I., Levy, K. N., and Starer, D. (1999). Long-short portfolio management: An integrated approach. *American Scientific Research Journal for Engineering,Technology, and Sciences*, 25(2):23–32.

Karanasos, M. (2002). Prediction in ARMA models with GARCH in mean effects. *Journal of Time Series Analysis*, 22(5):555–576.

Kim, K. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, 55(1–2):307–319.

Li, W., Ling, S., and McAleer, M. (2002). Recent theoretical results for time series models with garch errors. *J. Econ. Surv.*, 16(3):245–269.

Miah, M. and Rahman, A. (2016). Modelling volatility of daily stock returns: Is garch(1,1) enough? *American Scientific Research Journal for Engineering,Technology, and Sciences*, 18(1):29–39.

Myttenaere, A., Golden, B., Grand, B. L., and Rossi, F. (2016). Mean absolute percentage error for regression models. *Neurocomputing*, 192(5):38–48.

Petneházi, G. (2019). Recurrent neural networks for time series forecasting. *arXiv preprint*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., and Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9.

Tong, H. (1983). *Threshold models in nonlinear time series analysis*. Springer-Verlag, New York.

Wang, H. (2008). Nonlinear arma models with functional ma coefficients. *Journal of Time Series Analysis*, 29(6):1032–1056.

Wang, W., Guo, Y., Niu, Z., and Cao, Y. (2009). Stock indices analysis based on arma-garch model. *Proceedings of the IEEE Conference on Industrial Engineering and Engineering Management*, pages 2143–2147.

Zhang, G., Patuwo, B. E., and Hu, M. Y. (1998). Forecasting with artificial neural networks:: The state of the art. *International Journal of Forecasting*, 14(1):35–62.

Zivot, E. and Wang, J. (2006). *Modeling Financial Time Series with S-PLUS®*. Springer-Verlag, Berlin, Heidelberg.

국문초록

일별 주가 예측을 위한 순환신경망 기반 비선형 ARMA-GARCH 모형이 제안되었다. 기본적인 선형 ARMA-GARCH 모형에 두 개의 모수가 더해지고 쌍곡탄젠트함수를 이용하여 비선형성이 추가된 모형이다. 제안된 비선형 ARMA-GARCH 모형의 해는 순환신경망 개념을 이용하여 얻었다. 제안된 모형의 현실적 적용 가능성을 보이기 위하여 1950년부터 2018년까지 S&P500 지수의 일별 종가를 이용하여 알고리즘 기반 거래를 수행하였다. 금융 및 통계적 측도로 비교하였을 때 제안된 비선형 ARMA-GARCH 모형이 기존의 선형 ARMA-GARCH 모형보다 뛰어남을 보였다.

주요어: 비선형 ARMA-GARCH 모형, 순환신경망, 재정시계열자료 예측, S&P500.

학번: 2017-29033