# 

Machine Learning for Time Series Prediction, Comparative analysis

Olu Salami

A Thesis in the Field of Computational Finance

for the Degree of Master of Science in Computational Finance

University of Greenwich

January 2019

Copyright 2018 [David Salami]

Abstract

Autonomous and Automated decision making in the area of trading securities

Keywords: Machine learning; Convolutional Neural Networks; Recurrent Neural Networks; LTSM; Reinforcement learning; Machine Learning; Forex; Algorithmic Trading; Portfolio Management; Quantitative Finance; Q-learning; Markov Decision Process;

## Dedication (optional)

To delete this or any other unwanted section, select it in its entirety, including the title and the Section Break, and press Backspace or Delete.

## Acknowledgments (optional)

To delete this or any other unwanted section, select it in its entirety, including the title and the Section Break, and press Backspace or Delete. To see the Section Break, show formatting symbols by clicking the ¶ button in the Paragraph section

Table of Contents

Dedication (optional) iv

Acknowledgments (optional) v

List of Tables (optional) xi

List of Figures (optional) xii

Chapter I. Introduction 1

Introduction 1

Problem Background 1

Problem Statement 1

Purpose of Study 1

Project Objectives 2

Scope of Study 2

The Significant of Study 2

Organization of Report 2

Chapter II. Literature Review 3

Introduction 3

Phishing 3

Existing Anti-Phishing Approaches 3

Design of Classifiers 3

Hybrid System 3

Lookup System 3

Classifier System 3

Ensemble System 3

Normalization 3

Related Work 3

Summary 3

Chapter III. Research Methodology 4

Introduction 4

Research Framework 4

Research Design 4

Phase 1: Dataset Processing and Feature Extraction 4

Phase 2: Evaluation of Individual Classifier 4

Classification Background 4

Classifier Performance 4

Phase 3a: Evaluation of Classifier Ensemble 4

Phase 3b: Comparison of Individual vs. Ensemble Technique 4

Dataset 4

Summary 4

Chapter IV. Classical Methods 5

Vector Auto Regression 5

Auto Regressive Integrated Moving Average 5

Chapter IV. Deep Learning LTSM 5

Mathematical Formulation 5

Experiment 5

Dataset Division 5

Building The Model 5

Training Model 5

Validation Technique 5

Tuning The Model 5

Summary 5

Chapter V. Comparative Analysis 6

Introduction 6

ARIMA 6

VAR 6

LTSM 6

Summary 6

Chapter VI. Conclusion 7

Concluding Remarks 7

Research Contribution 7

Research Implications 7

Recommendations for Future Research 7

Closing Note 7

Chapter VII. References 8

End 9

Heading 3, SectionTitle/A-Head 9

Heading 4, B-Head 10

B-Head Style 11

B-Head Style 11

C-Head Style and Accompanying Paragraph 11

Body Text with C-Head 11

Numbered List Style 12

Bulleted List Style 13

Block Quote Style 13

Figures 14

Figure Title Style. 14

Figure Description Style. 14

Tables 15

Table Title Style. 15

Table Cell Style. 15

Table Description Style. 16

Styles Used in Other Parts of the Document 16

Title Style 16

Author Style 16

Centered Text Style 16

Harvard,Month,Year Style 16

Copyright Style 17

No-TOC-Heading Style 17

Bibliography Entry Style 17

Appendix 1. Title 18

Appendix 2. Title 19

[Bibliography/References/Works Cited.] 20

## List of Tables (optional)

[Table 1. Table Title. 2](#_upglbi)

[Table 2. Table Title 2](#_3ep43zb)

## List of Figures (optional)

[Figure 1. Figure Title. 2](#_1tuee74)

[Figure 2. Figure Title 2](#_4du1wux)

## Chapter I. Introduction

There are two ways to add a new chapter. First, you can simply type the name of

**Research question,**

1. How well does LTSM perform with relatively small dataset in a machine learning context. The commodity Gold was selected for its low volatility.
2. 2) In contract too existing forecasting model, how well does LSTM perform. My hypothesis is that LSTM will outperform existing (non ML) models

### 

### Introduction

### Problem Background

### Problem Statement

### Purpose of Study

### Project Objectives

### Scope of Study

### The Significant of Study

### Organization of Report

.

## Chapter II. Literature Review

### Introduction

### Phishing

### Existing Anti-Phishing Approaches

### Design of Classifiers

### Hybrid System

### Lookup System

### Classifier System

### Ensemble System

### Normalization

### Related Work

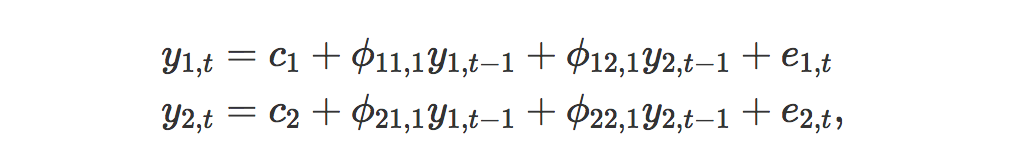
### Summary

## Chapter III. Research Methodology

### Introduction

### Vector Auto Regression

A VAR model is a generalisation of the univariate autoregressive model for forecasting a vector of time series. It comprises one equation per variable in the system. The right hand side of each equation includes a constant and lags of all of the variables in the system. To keep it simple, we will consider a two variable VAR with one lag. We write a 2-dimensional VAR(1) as Eq1.1

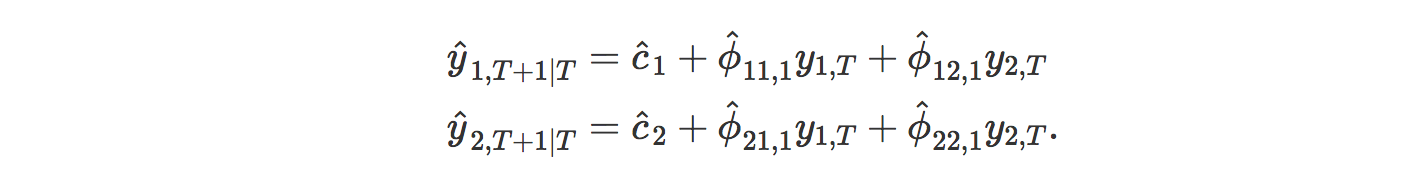


where e1,t and e2,t are white noise processes that may be contemporaneously correlated. The coefficient ϕii,ℓ captures the influence of the ℓth lag of variable yi on itself, while the coefficient ϕij,ℓ captures the influence of the ℓth lag of variable yj on yi.

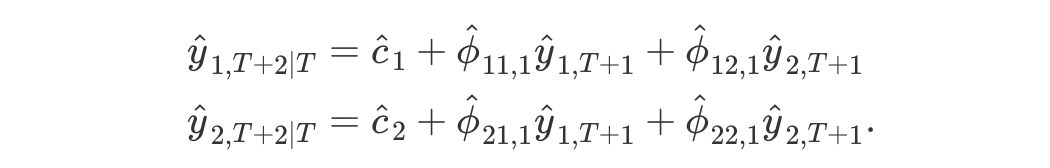
If the series are stationary, we forecast them by fitting a VAR to the data directly (known as a “VAR in levels”). If the series are non-stationary, we take differences of the data in order to make them stationary, then fit a VAR model (known as a “VAR in differences”). In both cases, the models are estimated equation by equation using the principle of least squares. For each equation, the parameters are estimated by minimising the sum of squared ei,t values.

The other possibility, which is beyond the scope of this book and therefore we do not explore here, is that the series may be non-stationary but cointegrated, which means that there exists a linear combination of them that is stationary. In this case, a VAR specification that includes an error correction mechanism (usually referred to as a vector error correction model) should be included, and alternative estimation methods to least squares estimation should be used.

Forecasts are generated from a VAR in a recursive manner. The VAR generates forecasts for *each* variable included in the system. To illustrate the process, assume that we have fitted the 2-dimensional VAR(1) described in Equations (1.1) –(1.2) for all observations up to time Then the one-step-ahead forecasts are generated by



except that the errors have been set to zero and parameters have been replaced with their estimates. For h=2, the forecasts are given by



Again, this is the same form as [(1.1)](https://otexts.org/fpp2/VAR.html#eq:var1a)–[(1.2)](https://otexts.org/fpp2/VAR.html#eq:var1b), except that the errors have been set to zero, the parameters have been replaced with their estimates, and the unknown values of y1 and y2 have been replaced with their forecasts. The process can be iterated in this manner for all future time periods.

There are two decisions one has to make when using a VAR to forecast, namely how many variables (denoted by K) and how many lags (denoted by p) should be included in the system. The number of coefficients to be estimated in a VAR is equal to K+pK2 (or 1+pK per equation). For example, for a VAR with K=5 variables and p=3 lags, there are 16 coefficients per equation, giving a total of 80 coefficients to be estimated. The more coefficients that need to be estimated, the larger the estimation error entering the forecast.

In practice, it is usual to keep K small and include only variables that are correlated with each other, and therefore useful in forecasting each other. Information criteria are commonly used to select the number of lags to be included.

Vector autoregressions (VARs) constitute a special case of the more general class of VARMA models. In essence, a VAR model is a fairly unrestricted (flexible) approximation to the reduced form of a wide variety of dynamic econometric models. VAR models can be specified in a number of ways. Funke (1990) presented five different VAR specifications and compared their forecasting performance using monthly industrial production series. Dhrymes and Thomakos (1998) discussed issues regarding the identification of structural VARs. Hafer and Sheehan (1989) showed the effect on VAR forecasts of changes in the model structure. Explicit expressions for VAR forecasts in levels are provided by Arin˜o and Franses (2000); see also Wieringa and Horva´th (2005). Hansson, Jansson, and Lo¨ f (2005) used a dynamic factor model as a starting point to obtain forecasts from parsimoniously parametrized VARs. In general, VAR models tend to suffer from doverfittingT with too many free insignificant parameters. As a result, these models can provide poor outof-sample forecasts, even though within-sample fitting is good; see, e.g., Liu, Gerlow, and Irwin (1994) and Simkins (1995). Instead of restricting some of the parameters in the usual way, Litterman (1986) and others imposed a prior distribution on the parameters, expressing the belief that many economic variables behave like a random walk. BVAR models have been chiefly used for macroeconomic forecasting (Artis & Zhang, 1990; Ashley, 1988; Holden & Broomhead, 1990; Kunst & Neusser, 1986), for forecasting market shares (Ribeiro Ramos, 2003), for labor market forecasting (LeSage & Magura, 1991), for business forecasting (Spencer, 1993), or for local economic forecasting (LeSage, 1989). Kling and Bessler (1985) compared out-of-sample forecasts of several thenknown multivariate time series methods, including Litterman’s BVAR model. The Engle and Granger (1987) concept of cointegration has raised various interesting questions regarding the forecasting ability of error correction models (ECMs) over unrestricted VARs and BVARs. Shoesmith (1992), Shoesmith (1995), Tegene and Kuchler (1994), and Wang and Bessler (2004) provided empirical evidence to suggest that ECMs outperform VARs in levels, particularly over longer J.G. De Gooijer, R.J. Hyndman / International Journal of Forecasting 22 (2006) 443–473 449 forecast horizons. Shoesmith (1995), and later Villani (2001), also showed how Litterman’s (1986) Bayesian approach can improve forecasting with cointegrated VARs. Reimers (1997) studied the forecasting performance of seasonally cointegrated vector time series processes using an ECM in fourth differences. Poskitt (2003) discussed the specification of cointegrated VARMA systems. Chevillon and Hendry (2005) analyzed the relationship between direct multi-step estimation of stationary and nonstationary VARs and forecast accuracy.

### Autoregressive Integrated Moving Average Model

It explicitly caters to a suite of standard structures in time series data, and as such provides a simple yet powerful method for making skillful time series forecasts.

ARIMA is an acronym that stands for AutoRegressive Integrated Moving Average. It is a generalization of the simpler AutoRegressive Moving Average and adds the notion of integration.

This acronym is descriptive, capturing the key aspects of the model itself. Briefly, they are:

* **AR**: *Autoregression*. A model that uses the dependent relationship between an observation and some number of lagged observations.
* **I**: *Integrated*. The use of differencing of raw observations (e.g. subtracting an observation from an observation at the previous time step) in order to make the time series stationary.
* **MA**: *Moving Average*. A model that uses the dependency between an observation and a residual error from a moving average model applied to lagged observations.

Each of these components are explicitly specified in the model as a parameter. A standard notation is used of ARIMA(p,d,q) where the parameters are substituted with integer values to quickly indicate the specific ARIMA model being used.

The parameters of the ARIMA model are defined as follows:

* **p**: The number of lag observations included in the model, also called the lag order.
* **d**: The number of times that the raw observations are differenced, also called the degree of differencing.
* **q**: The size of the moving average window, also called the order of moving average.

A linear regression model is constructed including the specified number and type of terms, and the data is prepared by a degree of differencing in order to make it stationary, i.e. to remove trend and seasonal structures that negatively affect the regression model.

A value of 0 can be used for a parameter, which indicates to not use that element of the model. This way, the ARIMA model can be configured to perform the function of an ARMA model, and even a simple AR, I, or MA model.

Adopting an ARIMA model for a time series assumes that the underlying process that generated the observations is an ARIMA process. This may seem obvious, but helps to motivate the need to confirm the assumptions of the model in the raw observations and in the residual errors of forecasts from the model.

Next, let’s take a look at how we can use the ARIMA model in Python. We will start with loading a simple univariate time series.

3.1. Preamble Early attempts to study time series, particularly in the 19th century, were generally characterized by the idea of a deterministic world. It was the major contribution of Yule (1927) which launched the notion of stochasticity in time series by postulating that every time series can be regarded as the realization of a stochastic process. Based on this simple idea, a number of time series methods have been developed since then. Workers such as Slutsky, Walker, Yaglom, and Yule first formulated the concept of autoregressive (AR) and moving average (MA) models. Wold’s decomposition theorem led to the formulation and solution of the linear forecasting problem of Kolmogorov (1941). Since then, a considerable body of literature has appeared in the area of time series, dealing with parameter estimation, identification, model checking, and forecasting; see, e.g., Newbold (1983) for an early survey. The publication Time Series Analysis: Forecasting and Control by Box and Jenkins (1970)3 integrated the existing knowledge. Moreover, these authors developed a coherent, versatile three-stage iterative

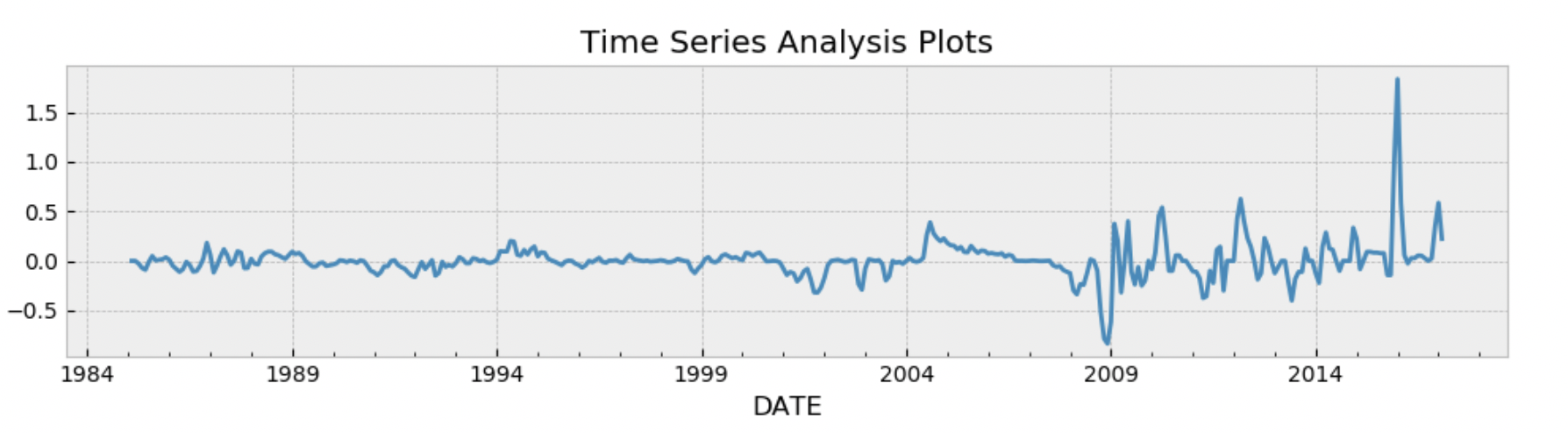
cycle for time series identification, estimation, and verification (rightly known as the Box–Jenkins approach). The book has had an enormous impact on the theory and practice of modern time series analysis and forecasting. With the advent of the computer, it popularized the use of autoregressive integrated moving average (ARIMA) models and their extensions in many areas of science. Indeed, forecasting discrete time series processes through univariate ARIMA models, transfer function (dynamic regression) models, and multivariate (vector) ARIMA models has generated quite a few IJF papers. Often these studies were of an empirical nature, using one or more benchmark methods/models as a comparison. Without pretending to be complete, Table 1 gives a list of these studies. Naturally, some of these studies are more successful than others. In all cases, the forecasting experiences reported are valuable. They have also been the key to new developments, which may be summarized as follows. 3.2. Univariate The success of the Box–Jenkins methodology is founded on the fact that the various models can, between them, mimic the behaviour of diverse types of series—and do so adequately without usually requiring very many parameters to be estimated in the final choice of the model. However, in the midsixties, the selection of a model was very much a matter of the researcher’s judgment; there was no algorithm to specify a model uniquely. Since then, many techniques and methods have been suggested to add mathematical rigour to the search process of an ARMA model, including Akaike’s information criterion (AIC), Akaike’s final prediction error (FPE), and the Bayes information criterion (BIC). Often these criteria come down to minimizing (in-sample) onestep-ahead forecast errors, with a penalty term for overfitting. FPE has also been generalized for multistep-ahead forecasting (see, e.g., Bhansali, 1996, 1999), but this generalization has not been utilized by applied workers. This also seems to be the case with criteria based on cross-validation and splitsample validation (see, e.g., West, 1996) principles, making use of genuine out-of-sample forecast errors; see Pen˜a and Sa´nchez (2005) for a related approach worth considering. There are a number of methods (cf. Box et al., 1994) for estimating the parameters of an ARMA model. Although these methods are equivalent asymptotically, in the sense that estimates tend to the same normal distribution, there are large differences in finite sample properties. In a comparative study of software packages, Newbold, Agiakloglou, and Miller (1994) showed that this difference can be quite substantial and, as a consequence, may influence forecasts. They recommended the use of full maximum likelihood. The effect of parameter estimation errors on the probability limits of the forecasts was also noticed by Zellner (1971). He used a Bayesian analysis and derived the predictive distribution of future observations by treating the parameters in the ARMA model as random variables. More recently, Kim (2003) considered parameter estimation and forecasting of AR models in small samples. He found that (bootstrap) bias-corrected parameter estimators produce more accurate forecasts than the least squares estimator. Landsman and Damodaran (1989) presented evidence that the James-Stein ARIMA parameter estimator improves forecast accuracy relative to other methods, under an MSE loss criterion. If a time series is known to follow a univariate ARIMA model, forecasts using disaggregated observations are, in terms of MSE, at least as good as forecasts using aggregated observations. However, in practical applications, there are other factors to be considered, such as missing values in disaggregated series. Both Ledolter (1989) and Hotta (1993) analyzed the effect of an additive outlier on the forecast intervals when the ARIMA model parameters are estimated. When the model is stationary, Hotta and Cardoso Neto (1993) showed that the loss of efficiency using aggregated data is not large, even if the model is not known. Thus, prediction could be done by either disaggregated or aggregated models. The problem of incorporating external (prior) information in the univariate ARIMA forecasts has been considered by Cholette (1982), Guerrero (1991), and de Alba (1993). As an alternative to the univariate ARIMA methodology, Parzen (1982) proposed the ARARMA methodology. The key idea is that a time series is transformed from a long-memory AR filter to a shortmemory filter, thus avoiding the bharsherQ differencing operator. In addition, a different approach to the dconventionalT Box–Jenkins identification step is used. In the M-competition (Makridakis et al., 1982), the ARARMA models achieved the lowest MAPE for longer forecast horizons. Hence, it is surprising to find that, apart from the paper by Meade and Smith (1985), the ARARMA methodology has not really taken off in applied work. Its ultimate value may perhaps be better judged by assessing the study by Meade (2000) who compared the forecasting performance of an automated and non-automated ARARMA method. Automatic univariate ARIMA modelling has been shown to produce one-step-ahead forecasts as accurate as those produced by competent modellers (Hill & Fildes, 1984; Libert, 1984; Poulos, Kvanli, & Pavur, 1987; Texter & Ord, 1989). Several software vendors have implemented automated time series forecasting methods (including multivariate methods); see, e.g., Geriner and Ord (1991), Tashman and Leach (1991), and Tashman (2000). Often these methods act as black boxes. The technology of expert systems (Me´lard & Pasteels, 2000) can be used to avoid this problem. Some guidelines on the choice of an automatic forecasting method are provided by Chatfield (1988). Rather than adopting a single AR model for all forecast horizons, Kang (2003) empirically investigated the case of using a multi-step-ahead forecasting AR model selected separately for each horizon. The forecasting performance of the multi-step-ahead procedure appears to depend on, among other things,

### Summary

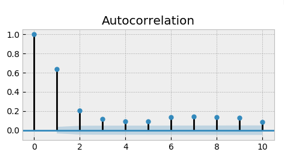
## Chapter IV. Classical Methods

In this section a limited analysis of two classic linear predictor is presented. The commodity selected for analysis is **Gold**. The commodity was selected for the following reasons, historic and reliable price quotes for are monthly (not) is you wish to go back a long time historically Gold as an asset relative to other financial instrument is illiquid and relatively stable. In comparison to other asset Gold is not as volatile, thus not prone to arbitrary jumps. Forecasting gold prices with machine learning is one of the motivations for this paper. There is a common conception that deep learning algorithm perform poorly with small datasets. In this experiment the commodity Gold, with monthly dataset for over 30 years will be applied to various predictive models, the objective is to compare the relative performance of classic models, including VAR and ARIMA, then contrast the predictive performance with a machine learning algorithm LSTM.

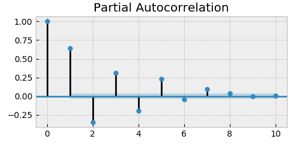
### Exploratory Data Analysis



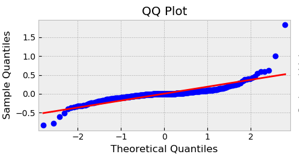
Autocorrelation



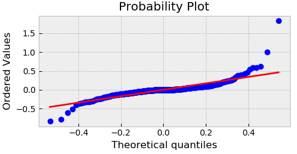
Partial Autocorrelation



QQ Plot



Probability Plot



Descriptive Statistics

Distribution of returns

Test for stationarity

|  |  |
| --- | --- |
| Dicky fuller Test | Box-Ljung test |
|  |  |

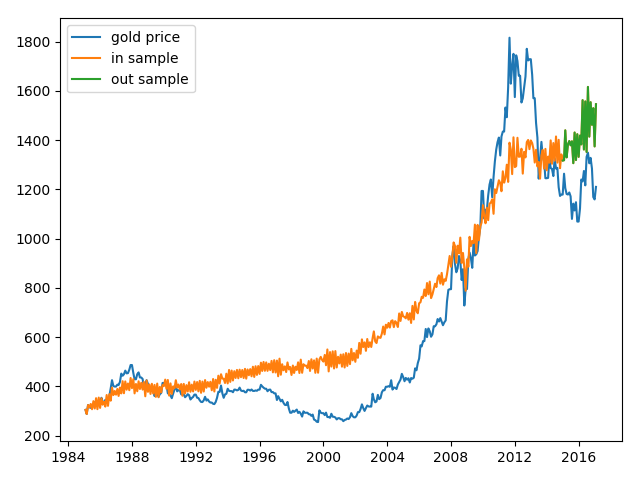
VAR Model

Model Summary

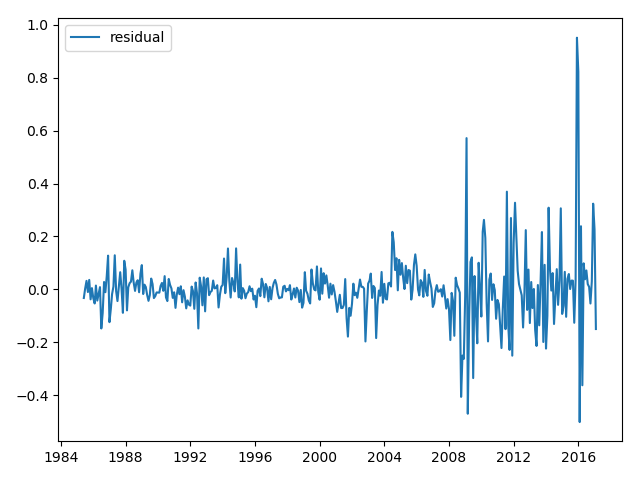
|  |  |  |  |
| --- | --- | --- | --- |
| Model: | VAR |  |  |
| Method: | OLS |  |  |
| No. of Equations: | 4 | BIC: | -27.3445 |
| Nobs: | 381 | HQIC: | -27.7690 |
| Log likelihood: | 3248.71 | FPE: | 6.59078e-13 |
| AIC: | -28.0482 | Det(Omega\_mle): | 5.53483e-13 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | coefficient | std. error | t-stat | prob |
| const | 0.008636 | 0.004346 | 1.987 | 0.047 |
| L1.Gold\_ret | 0.650066 | 0.051679 | 12.579 | 0.000 |
| L2.Gold\_ret | -0.499789 | 0.059456 | -8.406 | 0.000 |
| L3.Gold\_ret | 0.31606 | 0.059173 | 5.341 | 0.000 |
| L4.Gold\_ret | -0.14064 | 0.051525 | -2.73 | 0.006 |

VAR Gold



Residual plot



Error Metrics

rmse 238.01521006034054

mse 56651.24022006804

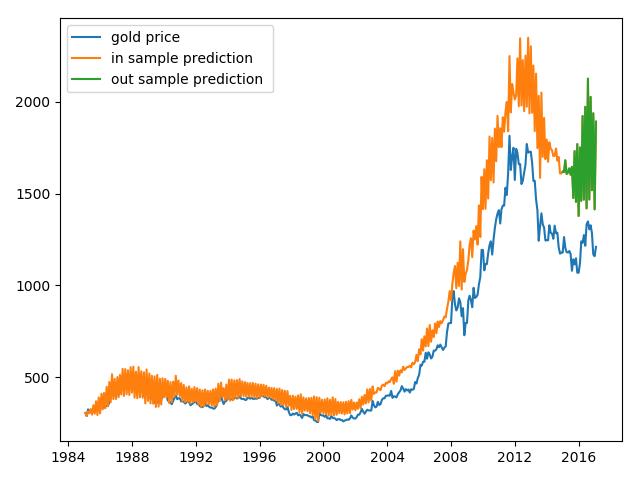
mse\_log 0.03317009614943625

rsquared -6.330709837631952

mae 218.71415906552355

### Auto Regressive Integrated Moving Average

ARIMA Gold



Residual plot

## Chapter IV. LSTM

### Mathematical Formulation

### Experiment

### Dataset Division

### Building The Model

### Training Model

### Validation Technique

### Tuning The Model

### Summary

## Chapter V. Comparative Analysis

### Introduction

### ARIMA

### VAR

### LTSM

### Summary

## Chapter VI. Conclusion

### Concluding Remarks

In this paper, we have revisited the evolutionary path of the nowadays deep learning models.

We revisited the paths for three major families of deep learning models: the deep generative model family, convolutional neural network family, and recurrent neural network family as well as some topics for optimization techniques.

This paper could serve two goals: 1) First, it documents the major milestones in the

science history that have impacted the current development of deep learning. These mile-

stones are not limited to the development in computer science fields. 2) More importantly, by revisiting the evolutionary path of the major milestone, this paper should be able to suggest the readers that how these remarkable works are developed among thousands of other contemporaneous publications. Here we briefly summarize three directions that many of these milestones pursue:

• Occam’s razor:

While it seems that part of the society tends to favor more complex

models by layering up one architecture onto another and hoping backpropagation can

find the optimal parameters, history says that masterminds tend to think simple:

Dropout is widely recognized not only because of its performance, but more because

of its simplicity in implementation and intuitive (tentative) reasoning. From Hopfield

Network to Restricted Boltzmann Machine, models are simplified along the iterations

until when RBM is ready to be piled-up.

• Be ambitious:

If a model is proposed with substantially more parameters than

contemporaneous ones, it must solve a problem that no others can solve nicely to be

remarkable. LSTM is much more complex than traditional RNN, but it bypasses the

vanishing gradient problem nicely. Deep Belief Network is famous not due to the fact

the they are the first one to come up with the idea of putting one RBM onto another,

but due to that they come up an algorithm that allow deep architectures to be trained

effectively.

• Widely read:

Many models are inspired by domain knowledge outside of machine

learning or statistics field. Human visual cortex has greatly inspired the development

of convolutional neural networks. Even the recent popular Residual Networks can find

corresponding mechanism in human visual cortex. Generative Adversarial Network

can also find some connection with game theory, which was developed fifty years ago.

We hope these directions can help some readers to impact more on current society. More

directions should also be able to be summarized through our revisit of these milestones by

readers.

### Research Contribution

### Research Implications

### Recommendations for Future Research

### Closing Note

## Chapter VII. References

## End

## [Bibliography/References/Works Cited.]

Last Name, Full First Name. *Basic MLA Style Only*. City: Publisher, Year. Print.

Last Name, First and Middle Initials (Year). *Basic APA Style Only.* City, State Abbreviation: Publisher.

Last Name, First Name. *Basic CMS Style Only, Requires Footnotes or Endnotes.* Place of publication: Publisher, Year.