Testing for Unit Roots Using Artificial Neural Networks

Rukman Ekanayake[1], V.A. Samaranayake[2]
[1,2]Department of Mathematics & Statistics, Missouri University of Science and Technology, Rolla Building, 400 W. 12th Street, Rolla, MO 65401

Abstract
Since the seminal paper by David Dickey and Wayne Fuller in 1979, there has been a continued interest in developing tests to detect unit roots in the ARMA formulation of empirical time series. Both asymptotic distribution-based as well as bootstrap-based tests have been developed with each method exhibiting both strengths and weaknesses. The use of artificial neural networks (ANNs) for forecasting empirical time series has also grown over the last quarter century, but there has been no serious attempt to develop an ANN-based methodology for unit root testing. Results of an initial attempt, to establish the proof of concept that an ANNs can be trained to detect the presence of a unit root in time series, is presented in this paper. Comparison with the Augmented Dickey-Fuller (ADF) test via Monte-Carlo simulations show the ANN outperforming the ADF for all parameter combinations studied, except for some exceptions for small sample sizes. Overall, results show promise in the use of ANNs to test for unit roots, but several issues such as the control of Type I error, optimal number of input nodes, hidden nodes, and hidden layers, have to be resolved prior to recommending this methodology as a viable alternative to existing test for unit roots.

Key Words: ANN, Time Series, Non-stationarity, Dickey-Fuller Tests, ADF Test

1. Introduction

Over the last few decades, there has been considerable interest in developing tests to detect the presence of a unit root in the autoregressive polynomial of time series modeled using the Autoregressive Moving Average (ARMA) formulation. David Dickey & Wayne Fuller were the first to introduce a test to detect the presence of a unit root in Autoregressive (AR) processes (Dickey and Fuller, 1979). This procedure, known as the Dickey-Fuller (DF) test, was applicable to AR processes of order one. A test for higher order AR processes was introduced by Dickey and Fuller (1981) and a more general test applicable to the general ARMA model was introduced by Said and Dickey (1984). This latter test is known as the Augmented Dickey Fuller (ADF) test. More recent years have seen the development of bootstrap-based unit root tests, for an example Ferretti and Romo (1996) developed a bootstrap based test for first order AR models and later Chang and Park (2003) used sieve bootstrap method to test the unit root in ARMA time series, but all these tests are rooted in classical statistical testing procedures. An interesting question one can raise is whether machine-learning algorithms such as Artificial Neural Networks (ANNs) can be adopted to detect the presence of unit roots in an empirical time series, which can be assumed to have an ARMA model as its underlying data generation process. Such techniques have shown to produce very promising results in areas such as classification and forecasting of time series. Therefore, it is a natural question to ask whether ANNs can be utilized to detect unit roots.

Multilayer Feedforward Neural Networks are universal function approximators (Hornik *et al.* 1989), which are capable of approximating any function to any desired degree of accuracy. ANNs are used in many practical situations for pattern recognition, image classification, forecasting and classification. There has been extensive research on the topic of forecasting time series using ANNs. An early example in the effective use of ANNs for time series forecasting are as follows. Szkuta *et al.* (1999), who employed a three-layered feed forward ANN to forecast electricity price. A more nuanced study is that of Butler and Kazakov (2011) who investigated the characteristics of non-stationarity in financial time series and its effect on forecasts based on ANNs. Chapter 14 of the Handbook of Natural Computing (2012) is devoted to the topic of neural networks for time series forecasting and the Chapter author Zhang reports twenty three publications on the topic over the limited period from 2005 to 2009. Another use of ANNs in time series is clustering. Fawaz et al. (2019) used deep neural network (DNN) methods to classify hundreds of time series. In spite of this growth in the use of ANNs for time series forecasting and investigations on their use on clustering, the utilization of ANNs in testing for unit roots in time series has not been attempted up to date. This study is an attempt to demonstrate that ANNs can be utilized to test for unit roots in empirical time series generated through an ARMA process. It is an attempt at proof of concept rather than to develop an exhaustive and optimal testing procedure.

2. Methodology

In this section, we first introduce the ARMA model and explain how the ADF test can be used to test for unit roots in this formulation. Then in Section 2.2 we introduce ANN based testing for unit roots in ARMA processes described in Section 2.1.

2.1 Decomposition of a general time series

Let $\{Y_t\}$ a time series which satisfies the model

$$Y_t = \rho Y_{t-1} + Z_t \quad (t=1,2,...) \tag{2.1}$$

$$Z_t = \alpha Z_{t-1} + \beta e_{t-1} + e_t, \ t = \mathbb{Z},$$

where, $|\alpha| < 1, |\beta| < 1$ and $\{e_t\}$ is a sequence of independent and identically distributed normal random variables. The time series $\{Y_t\}$ is a stationary ARMA (2, 1) if $|\rho| < 1$. If the $|\rho| = 1$ time series is an Autoregressive Integrated Moving Average (ARIMA (1, 1, 1)) process and has a unit root. It is important in empirical time series analysis to know whether or not $|\rho| = 1$. In the standard unit roots testing scenario, the null and alternative hypotheses are as follows:

$$H_0 : \rho = 1 \ vs \ H_1 : \rho \neq 1.$$

The Augmented Dickey Fuller (ADF) formulation of (2.1), can be written as

$$\Delta Y_t = (\rho - 1) Y_{t-1} + \sum_{i=1}^{\infty} c_i \Delta Y_{t-i} + e_t, \tag{2.2}$$

where, the coefficients c_i, $i \in \mathbb{N}$, are function of the parameters $\{\alpha, \beta\}$, and $\Delta Y_t = Y_t - Y_{t-1}$, the first difference of Y_t. The true order of the autoregression in (2.1) is infinite when $\beta \neq 0$. In practice, Said and Dickey (1984) suggested approximating the infinite autoregression in (2.2) by a truncated version of order *k*, which is a function of the number of observations, *T*, so that one can write

$$\Delta Y_t = \gamma Y_{t-1} + \sum_{i=1}^{k} c_i \Delta Y_{t-i} + u_t, \tag{2.3}$$

where, $\gamma = \rho - 1$. Note that u_t is not independent and identically distributed in (2.3). The ordinary least square (OLS) method is used to estimate the parameters of (2.3) and they are defined as $\hat{\gamma} = \hat{\rho} - 1$ and $\hat{c}_i$ $(i = 1,2, \dots, k)$. Now, the test hypotheses above can be written as

$$H_0 : \gamma = 0 \ vs \ H_1 : \gamma \neq 0.$$

The ADF formation for more general ARMA (p, q) formulations can also be written as in (2.3). Note that in all ARMA formulations we assume the invertiblility of the MA component and that there are no common roots for the AR and MA polynomials. The test statistic used for testing a presence of a unit root for more general time-series such as ARMA (p, q) error distribution is defined as

$$\hat{\tau} = \frac{\hat{\gamma}}{se(\hat{\gamma})}, \tag{2.4}$$

where $\hat{\tau}$ follows a Dickey Fuller distribution under the null hypothesis.

2.2 Use of Artificial Neural Network (ANN) for Unit Root Testing

A naïve approach to using ANNs to test for unit roots would be to train the ANN on a sample of ARMA time series with and without unit roots and use the actual time series values, say the last 200 values, as input to the network. Such an approach would mislead the ANN to identify spurious features, such as the variance of the time series, as the identifying characteristic for the presence of a unit root. For example, a zero mean time series with a unit roots wanders around zero much more than a zero mean time series without a unit root. Therefore, we incorporate following preprocessing to the time series before training the ANN. First, subtracted the mean of the time series and then divided the time series by its absolute maximum value.

The topology of the ANN used in this study is shown in Figure 1, where we have an input layer, a hidden layer and an output layer. The main objective is to classify whether the time series possesses a unit root or do not posses such a root. Therefore, we used a binary output layer with two nodes. In this exercise, sample sizes of 50, 100, and 250 were employed. We selected 50 as the number of hidden nodes when training on time series of lengths 100 and 250, and 20 number of hidden nodes for the time series of length 50. The number of input nodes corresponded to the length of the time series. The hidden nodes were assigned rectified linear unit (ReLU) activation functions (see Nwankpa *et.al.* (2018)) and the two output nodes have softmax activation function (see Nwankpa *et.al.* (2018)). The weights for the ANN were updated by the back propagation algorithm. We used a modified binary cross-entropy loss function as the loss function in this method.

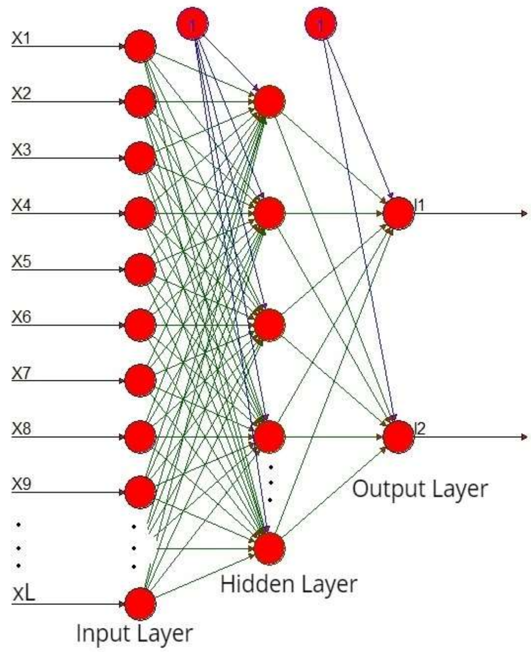

Figure 1: The feed forward ANN topology used in this study.

2.2.1 Controlling for Type I error rate

The binary cross-entropy loss function was used in the primary attempt at training the Neural Network. It returned a large Type I error rate in this configuration. The nature of the loss function is to minimize the error and maximize the overall classification accuracy. However, in order to develop a statistical test comparable to the ADF tests, we need to control the Type I error rate. Therefore, we used weighted cross-entropy loss function (WCE) as our loss function. This has been notably used by Ronneberger *et.al.* (2015). We have,

$$WCE = -\frac{1}{N}\sum w_1 x_i \log(p(x_i)) + w_2(1-x_i)\log(1-p(x_i)),$$

$$\text{where } x_i = \begin{cases} 1; \textit{If the time series in unit root class} \\ 0; \textit{If the time series is not in unit root class} \end{cases}$$

with w_1 and w_2 defined as the weights of the loss function. Note that N is the total number of time series samples, and $p(x_i)$ is the predicted probability that the time series is in the unit root class.

3. Results

Samples of time series of length T = 50, 100, 250 were generated according to the equations (2.1), with and without unit roots. Initial value of time series, y_0, was set to zero. For all generated samples α in Equation (2.1) was set to zero. The parameter combinations used for the simulation are given in the Table 1 below.

Table 1: Parameter Combination used in the simulation.

β	0.99	0.95	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
ρ	1	0.99	0.95	0.9	0.5	0.2					

Five thousand samples each were simulated for all parameter values of β with $\rho = 1$ for the unit root cohort. Non-unit root cohort contained 1,000 samples each for parameter combination of ρ ($\rho \neq 1$) and β. Altogether, we simulated 60,000 time series each for the unit root cohort and the non-unit root cohort for training. In the initial training of the ANN by setting w_1 and w_2 to the value of unity in the weighted cross-entropy loss function we got large Type I error rates (>0.05) for all the time series lengths considered.

In order to fix the Type I error rate at 0.05 we needed to find suitable values for the weights w_1 and w_2 in the WCE loss function. Therefore, we trained multiple ANNs using simulated time series while fixing the w_1 fixed at unity, and changing the w_2. Then we calculated the corresponding Type I error rates. The plotted Type I error rates are shown in Figure 2 for sample size 100. Then by fitting a polynomial regression model to the resulting data, we estimated the value for w_2 when Type I error is 0.05. The estimated w_2's are given in Table 2 for length 50, 100, and 250.

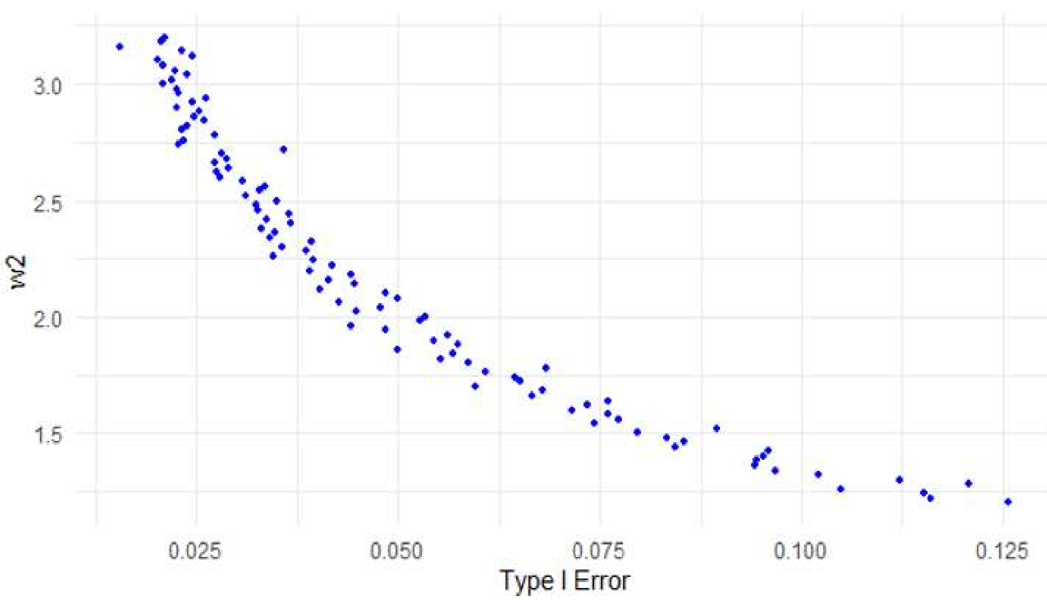

Figure 2: w_2 vs Type I error rate calculated for time series length 100

Table 2: Estimated value of weight (w_2) for the weighted loss function.

Length of time series	**50**	**100**	**250**
w_1	1	1	1
w_2	2.44	1.99	1.94
Type I error rate	0.048	0.05	0.47

After estimating the weights for the loss function, we trained three ANNs for different time series lengths of 50, 100 and 250 using data samples simulated according parameter combinations given in Table 1. This constitute the training samples. Then we generated 1,000 samples for every parameter combination for testing. The test results were compared with those from the ADF test. The ADF procedure employed the General to Specific (G-t-S) sequential t-test procedure as used in Ng and Perron (1995) to determine the $\hat{\tau}$ as in (2.4). The maximum lag parameter (k) in the ADF model given in the Equation (2.3) was fixed at 10 for all time series lengths. Then, 10% significance level was used for the G-t-S sequential t-tests because the size distortion associated in G-t-S procedure with the 5% significant level is slightly higher than for the 10% significant level when $\beta \neq 0$ (Ng and Perron, 1995; and Patterson, 2011). Results of Type I error obtained for the training dataset for are listed in the Table 3.

These error rates were observed for the training samples using the estimated weights. Results show that in the training sample, Type I error rate was kept below 0.05 for almost all parameter combinations except few cases.

Table 3: Type I error rate for training samples in NN-based method.

$\boldsymbol{\beta}$	**Type I Error**		
	T=250	**T=100**	**T=50**
0.99	0.044	0.045	0.034
0.95	0.048	0.041	0.034
0.9	0.046	0.041	0.037
0.8	0.049	0.042	0.036
0.7	0.047	0.044	0.031
0.6	0.045	0.043	0.043
0.5	0.046	0.043	0.044
0.4	0.050	0.041	0.043
0.3	0.050	0.051	0.048
0.2	0.041	0.045	0.060
0.1	0.047	0.057	0.078
0	0.052	0.063	0.084

Tables 4a, 4b, 5, and 6 give the Type I error rates and power yielded by the ANN-based unit roots tests based on the test data sample.

Table 4a: Size of Unit Root Tests based on Test Data Set (T =50).

$\boldsymbol{\rho}$	$\boldsymbol{\beta}$	**ANN P(Reject H0) x100**	**ADF P(Reject H0) x 100**	**Test Property**
1	0.9	4.9	5.8	Type I Error
1	0.6	4.2	6.9	
1	0.3	5.8	8.2	
1	0	9.8	7.2	

Table 4b: Power of Unit Root Tests based on Test Data Set (T =50).

ρ	β	ANN P(Reject H0) x100	ADF P(Reject H0) x 100	Test Property
0.95	0.9	9.2	12.7	Power
0.95	0.6	10.5	12.8	
0.95	0.3	14.6	10.9	
0.95	0	22.6	12.5	
0.9	0.9	19.4	17.5	
0.9	0.6	21.9	19.1	
0.9	0.3	22.9	17.7	
0.9	0	37.4	16.8	
0.8	0.9	36.9	20.4	
0.8	0.6	42.7	30.1	
0.8	0.3	46.6	30.5	
0.8	0	60.1	32.8	

Table 5: Size and Power of Unit Root Tests based on Test Data Set (T =100)

ρ	β	ANN P(Reject H0) x100	ADF P(Reject H0) x 100	Test Property
1	0.9	5.5	5.0	Type I Error
1	0.6	6.1	6.4	
1	0.3	5.2	6.3	
1	0	6.8	5.8	
0.95	0.9	24.9	11.9	Power
0.95	0.6	24.9	15.8	
0.95	0.3	27.4	14.7	
0.95	0	33.7	17.3	
0.9	0.9	52.9	20.2	
0.9	0.6	54.5	27.1	
0.9	0.3	56.4	31.1	
0.9	0	61.6	31.0	
0.8	0.9	84.9	33.9	
0.8	0.6	84.0	49.6	
0.8	0.3	87.0	61.9	
0.8	0	84.9	67.4	

The calculated Type I error rate for the time series of length 100 and 250 is close to 5% in both methods. The higher variability in the Type I error rate can be seen in the ANN based method over ADF method for time series of length 50 with a range of 4.9 and 1.4

respectively. However, the Type I error of ANN based method is less than that of ADF method when the moving average parameter takes values of 0.9, 0.6 and 0.3.

It is clear that the power of the ANN based method exceeds the power of the ADF method by a considerable amount for each ρ = 0.95, 0.90 and 0.8 and β = 0.9, 0.6, 0.3 and 0 combinations, for sample sizes 100 and 250. Even for the case with sample size 50, apart from few cases when ρ = 0.95, the power of ANN based method is considerably higher than that of the of ADF method.

While the above results suggest the feasibility of using ANNs to test for unit roots, some additional testing not reported here have shown that time series that are stationary over most of the time span but with one or more occasional spikes and other such features can trick the trained ANN to classify such series as having unit roots. Methods to preprocess and standardize such series have to be developed if the proposed method is to be used for testing empirical time series. In addition, the simulation results reported herein are for a small set of parameters combinations. Additional studies must be conducted to determine the feasibility of this method for more general ARMA processes.

Table 6: Size and Power of Unit Root Tests based on Test Data Set (T =250)

ρ	β	ANN P(Reject H0) x100	ADF P(Reject H0) x 100	Test Property
1	0.9	4.9	4.1	Type I Error
1	0.6	5.1	5.7	
1	0.3	6.4	5.2	
1	0	5.7	5.0	
0.95	0.9	68.2	29.4	Power
0.95	0.6	66.6	37.2	
0.95	0.3	64.4	39.7	
0.95	0	64.8	38.5	
0.9	0.9	92.0	60.0	
0.9	0.6	94.4	76.8	
0.9	0.3	91.7	81.4	
0.9	0	92.4	83.3	
0.8	0.9	99.5	90.6	
0.8	0.6	99.3	95.0	
0.8	0.3	99.3	96.4	
0.8	0	98.7	97.1	

4. Conclusions

We introduce a new approach to testing for unit roots in the ARMA formulation by using artificial neural networks. Based on the simulation results, neural network based unit root testing outperforms the traditional Augmented Dickey Fuller test for the time series considered in this study. Further, we address the issue of controlling the Type I error rate when using ANN by selecting appropriate weights for the binary cross entropy loss function. However, additional work in this area is needed to be done. This study utilized a feedforward ANN with one hidden layer with number of nodes varying with sample size. The number of hidden layers and the number of nodes in it were selected based on initial investigations but further studies are needed to conduct to optimize these structural features. Investigating the type of neural networks and estimation methods optimal for unit root testing is another aspect that should be investigated. In addition more innovative methods of standardizing the input series is warranted to avoid spurious features of the time series effecting the test outcomes.

References

Butler, M., & Kazakov, D. (2011). The effects of variable stationarity in a financial time-series on Artificial Neural Networks. *2011 IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CIFEr)*.

Chang, Y. and Park, J.Y. (2003). A Sieve Bootstrap for the Test of a Unit Root, *Journal of Time Series Analysis, 24,* 379–400.

Dickey, D., & Fuller, W. (1979). Distribution of the Estimators for Autoregressive Time Series with a Unit Root. *Journal of the American Statistical Association, 74*(366), 427-431.

Dickey, D., & Fuller, W. (1981). Likelihood Ratio Statistics for Autoregressive Time Series with a Unit Root. *Econometrica, 49*(4), 1057-1072.

Said E. Said, & Dickey, D. (1984). Testing for Unit Roots in Autoregressive-Moving Average Models of Unknown Order. *Biometrika, 71*(3), 599-607.

Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., and Muller, P. (2019) "Deep learning for time series classification: a review," *Data Mining and Knowledge Discovery.*

Ferretti, N. and Romo, J. (1996) Unit root bootstrap tests for AR(1) models. *Biometrika 83*, 849–60.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks, 2*, 359-366.

Ng, S., & Perron, P. (1995). Unit Root Tests in ARMA Models with Data-Dependent Methods for the Selection of the Truncation Lag. *Journal of the American Statistical Association, 90*(429), 268-281.

Nwankpa, C., Ijomah, W., & Gachagan, A., and Marshall, S. Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv*:1811.03378.

Patterson, K. D. (2011). *Unit root tests in time series*. New York, NY: Palgrave Macmillan.

Szkuta, B., Sanabria, L., & Dillon, T. (1999). Electricity price short-term forecasting using artificial neural networks. *IEEE Transactions on Power Systems, 14*(3), 851-857.

Zhang, G. (2012). Neural Networks for Time-Series Forecasting. In G. Rozenberg, T. Back and J.N. Kok, *Handbook of Natural Computing* (pp. 461-478). New York, NY: Springer.