2014

Forecasting Iowa gaming volume: A comparison of four time series forecasting methods

Hui Yu
Iowa State University

Follow this and additional works at: https://lib.dr.iastate.edu/etd

Part of the Quantitative, Qualitative, Comparative, and Historical Methodologies Commons

Recommended Citation
https://lib.dr.iastate.edu/etd/16534

This Thesis is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.
Forecasting Iowa gaming volume:
A comparison of four time series forecasting methods

by

Hui Yu

A thesis submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

Major: Hospitality Management

Program of Study Committee:
Tianshu Zheng, Major Professor
Thomas Schrier
Young-A Lee

Iowa State University
Ames, Iowa
2016

Copyright © Hui Yu, 2016. All rights reserved.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>LIST OF FIGURES</th>
<th>iv</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF TABLES</td>
<td>v</td>
</tr>
<tr>
<td>NOMENCLATURE</td>
<td>vi</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>vii</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>viii</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CHAPTER 1 INTRODUCTION</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Background of the Study</td>
<td>1</td>
</tr>
<tr>
<td>Statement of the Problem</td>
<td>2</td>
</tr>
<tr>
<td>Research Objectives</td>
<td>3</td>
</tr>
<tr>
<td>Research Questions</td>
<td>4</td>
</tr>
<tr>
<td>Significance of the Study</td>
<td>4</td>
</tr>
<tr>
<td>Definitions of Terms</td>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CHAPTER 2 LITERATURE REVIEW</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Times Series Forecasting</td>
<td>10</td>
</tr>
<tr>
<td>ARIMA/SARIMA Models with Box-Jenkins Procedure</td>
<td>11</td>
</tr>
<tr>
<td>Advantages and Applications of the ARIMA/SARIMA in the Gaming Field</td>
<td>15</td>
</tr>
<tr>
<td>Simple Moving Average and M-Competition</td>
<td>17</td>
</tr>
<tr>
<td>Artificial Neural Networks</td>
<td>18</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>Raw data plot of slot coin-in</td>
<td>26</td>
</tr>
<tr>
<td>Figure 2</td>
<td>Raw data plot of table drop</td>
<td>26</td>
</tr>
<tr>
<td>Figure 3</td>
<td>Identification plot for slot coin-in</td>
<td>28</td>
</tr>
<tr>
<td>Figure 4</td>
<td>Comparison of the tentative identification plots with the different gamma values for slot coin-in</td>
<td>29</td>
</tr>
<tr>
<td>Figure 5</td>
<td>Comparison of the tentative identification plots after the first order differencing for slot coin-in</td>
<td>30</td>
</tr>
<tr>
<td>Figure 6</td>
<td>Comparison among alternative models</td>
<td>32</td>
</tr>
<tr>
<td>Figure 7</td>
<td>Correlation of the time lag for slot coin-in</td>
<td>38</td>
</tr>
<tr>
<td>Figure 8</td>
<td>The architecture of dynamic neural network of slot coin-in</td>
<td>39</td>
</tr>
<tr>
<td>Figure 9</td>
<td>Performance of the artificial neural network model in slot coin-in</td>
<td>40</td>
</tr>
<tr>
<td>Figure 10</td>
<td>Comparisons among the four forecasting methods for slot coin-in</td>
<td>49</td>
</tr>
<tr>
<td>Figure 11</td>
<td>Comparisons among the four forecasting methods for table drop</td>
<td>50</td>
</tr>
<tr>
<td>Figure A1</td>
<td>The stationarity check of raw data plot of table drop</td>
<td>76</td>
</tr>
<tr>
<td>Figure A2</td>
<td>Identification plot for table drop after first order differencing</td>
<td>77</td>
</tr>
<tr>
<td>Figure A3</td>
<td>Comparison of the tentative identification plots with the different gamma values for table drop</td>
<td>78</td>
</tr>
<tr>
<td>Figure A4</td>
<td>Comparison of the tentative identification plots after the first order differencing for table drop</td>
<td>79</td>
</tr>
<tr>
<td>Figure A5</td>
<td>Estimation of SARIMA(2,1,0)(0,1,2)_12 on w= ( millions of dollars ) ^ 0.33 for Iowa monthly table drop</td>
<td>80</td>
</tr>
<tr>
<td>Figure A6</td>
<td>Estimation of SARIMA(2,1,0)(0,1,1)_12</td>
<td></td>
</tr>
</tbody>
</table>
on $w = (\text{millions of dollars})^{0.33}$ for Iowa monthly table drop

Figure A7  Feed-forward architecture of ANN for slot coin-in

Figure A8  Feed-forward architecture of ANN for table drop
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1</td>
<td>Characteristics of Theoretical ACF and PACF for Stationary Process</td>
<td>31</td>
</tr>
<tr>
<td>Table 2</td>
<td>Summary of the Comparison of Model Estimation Results</td>
<td>34</td>
</tr>
<tr>
<td>Table 3</td>
<td>Summary of Mean Squared Error (MSE) and Mean Absolute Percentage Error (MAPE) Values for Slot Coin-in</td>
<td>42</td>
</tr>
<tr>
<td>Table 4</td>
<td>Summary of Mean Squared Error (MSE) and Mean Absolute Percentage Error (MAPE) Values for Table Drop</td>
<td>43</td>
</tr>
<tr>
<td>Table 5</td>
<td>Summary of Parameter Estimates for Slot Coin-in</td>
<td>49</td>
</tr>
<tr>
<td>Table 6</td>
<td>Summary of Parameter Estimates for Table Drop</td>
<td>50</td>
</tr>
<tr>
<td>Table 7</td>
<td>Summary of Model Training and Testing for Slot Coin-in and Table Drop</td>
<td>51</td>
</tr>
<tr>
<td>Table A1</td>
<td>Results of the Parameter Estimation of SARIMA(2,1,0)(0,1,2)_{12} on $w = (\text{millions of dollars})^{0.33}$ for Iowa Monthly Coin-in</td>
<td>77</td>
</tr>
<tr>
<td>Table A2</td>
<td>Results of the Parameter Estimation of SARIMA(2,1,0)(0,1,1)_{12} on $w = (\text{millions of dollars})^{0.33}$ for Iowa Monthly Table Drop</td>
<td>78</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>AC</td>
<td>Autocorrelation</td>
<td></td>
</tr>
<tr>
<td>ACF</td>
<td>Autocorrelation function</td>
<td></td>
</tr>
<tr>
<td>AIC</td>
<td>Akaike’s information criterion</td>
<td></td>
</tr>
<tr>
<td>ANN</td>
<td>Artificial neural network</td>
<td></td>
</tr>
<tr>
<td>AR factor</td>
<td>Autoregressive factor</td>
<td></td>
</tr>
<tr>
<td>ARIMA</td>
<td>Autoregressive integrated moving average</td>
<td></td>
</tr>
<tr>
<td>MA factor</td>
<td>Moving average factor</td>
<td></td>
</tr>
<tr>
<td>MAPE</td>
<td>Mean absolute percent error</td>
<td></td>
</tr>
<tr>
<td>MSE</td>
<td>Mean squared error</td>
<td></td>
</tr>
<tr>
<td>PACF</td>
<td>Partial autocorrelation function</td>
<td></td>
</tr>
<tr>
<td>SARIMA</td>
<td>Seasonal autoregressive integrated moving average</td>
<td></td>
</tr>
<tr>
<td>SMA</td>
<td>Simple moving average</td>
<td></td>
</tr>
</tbody>
</table>
ACKNOWLEDGMENTS

I would like to thank my committee chair, Professor Tianshu Zheng, and my committee members, Drs. Thomas Schrier, and Young-A Lee, for their guidance and support throughout the course of this research.

In addition, I would also like to thank my friends, colleagues, and the department faculty and staff for making my time at Iowa State University a wonderful experience. I want to also offer my appreciation to those who were willing to spend the time to help in my research. This thesis would not have been possible without their help.

Finally, thanks to my family for their encouragement and support.
ABSTRACT

This study investigated the forecasting abilities of four forecasting techniques—(1) autoregressive integrated moving average (ARIMA), (2) artificial neural networks (ANNs), (3) simple moving average (SMA), and (4) the naïve method—as they apply to the Iowa monthly time series of slot coin-in and table drop. Mean squared error and mean absolute percentage error were adopted as evaluation criteria to compare the forecasting abilities of these various techniques. The results indicated that SMA outperformed the other three methods, which extends the conclusions of the M-competition to time series in the gaming field. Meanwhile, the ANN technique introduced without any modification was incapable of replacing the standing of ARIMA in the practice of gaming forecasting. This study is the first attempt to explore the forecasting abilities of four prevailing forecasting models based on gaming practices. It provides researchers and practitioners with a guide to and insights into the application of the ANN technique in gaming forecasting, selection of forecasting method, and effectiveness of the model for different horizons of gaming forecasting.
CHAPTER 1

INTRODUCTION

Background of the Study

Numerous existing gaming-related studies focused on destination casinos, such as Las Vegas, Atlantic City, etc. (Eisendrath, Bernhard, Lucas, & Murphy, 2008; Hunsaker, 2001; Suh & Lucas, 2011). However, as individual states have continually pursued the development of diverse economies, the local gaming markets of individual states have become increasingly valued due to their importance to local government finances. The gaming industry in Iowa, the first state in the country to legalize riverboat operations, also includes land-based and racetrack casinos with slots and table games, and is one example of an individual local gaming market. Iowa’s gross casino gaming revenue in 2012 was $1.424 billion, and the gaming tax revenue for 2012 was $321.53 million (American Gaming Association (1999–2013, 2013). Thus, Iowa’s gaming revenue, increasingly impacting its socioeconomic arena and fiscal benefits, has been prominent in Iowa’s economic growth and social life (Hsu, 1999). With this representativeness, the Iowa gaming industry has drawn wide attention in the practice of gaming research (Ahlgren, Dalbor, & Singh, 2009; Chhabra, 2007; Landers, 2008; Zheng, Farrish, Lee, & Yu, 2013).

In the meantime, market observers and analysts have expressed concern about gaming volume predictions due to the promising market prospect coming from the economic recovery and industrial upgrading (Horvath & Paap, 2011). The impacts of the economic recession or particular events on the gaming market have been discussed widely in the gaming literature (Eisendrath et al., 2008; Zheng et al., 2013). Thus, additional and more relevant novel
forecasting methods or models have been introduced from other research fields to the gaming literature for the improvement of forecasting accuracy.

Currently, the Box-Jenkins forecasting method in autoregressive integrated moving average (ARIMA) time series analyses, a kind of sophisticated forecasting method, is viewed as an effective approach for gaming volume predictions (Eisendrath et al., 2008). This method, applied extensively in gaming volume analysis, is based on three advantages: first, it is able to capture the seasonal and systematic trends in gaming data; second, it is competent in predicting gaming revenue (Cargill & Eadington, 1978); and third, it is parsimonious in predicting, compared with econometric models (Shonkwiler, 1992). However, artificial neural network (ANN) models, previously rarely used in the hospitality field, recently have been used for the analysis of seasonal data from the hospitality industry and even for gaming data. For instance, Balcilar, Gupta, Majumdar, and Miller (2010) adopted ANNs to predict gaming volume and indicated that nonlinear models (e.g., ANNs) generally outperform linear models (e.g., transition autoregressive models) in forecasting gross gaming revenue and tax sales in Nevada.

**Statement of the Problem**

Especially in reference to recent gaming literature, the results of Balcilar et al. (2010) challenged the prevailing perception mentioned earlier on the superiority of ARIMA in gaming studies. However, Kolarik and Rudorfer (1994) noted ANNs are unstable in forecasting seasonal time series. In other words, the question about whether the ANN technique can replace ARIMA to become a widely accepted approach in seasonal gaming time series forecasting is in need of serious discussion. As early as the 1980s, Makridakis et al. (1982) formally initiated a type of study known as the M-competition (Makridakis et al., 1982;
Makridakis & Hibon, 2000), that compared the performance of different methods in business time series. They proved that some simple techniques were as effective as many well-developed sophisticated methods. Therefore, the contradiction mentioned above might call for a new turn of competitions in gaming field studies. With this in mind, a less sophisticated method, the simple moving average (SMA), was taken into account due to its utilization in M-competition and its effectiveness in the empirical study of stock trading and business operations (Ellis & Parbery, 2005).

Therefore, this competition study was conducted, designed to focus on investigating the forecasting ability of the forecasting methods commonly used in the gaming field, in an attempt to discover the most appropriate forecasting methods with gaming data in different forecasting horizons.

**Research Objectives**

In time series forecasting, there is an attempt to apply forecasting methods, regardless their simplicity or complexity, to produce accurate results. However, forecasting performance of best techniques varies, corresponding to the attributes of different time series. To improve the accuracy of forecasting gaming time series, four forecasting methods—(1) ARIMA models, (2) ANN, (3) SMA, and (4) naïve method—based on the findings of the M-competition (Makridakis et al., 1982) and the trends reflected in current gaming literature, were employed in this study to forecast the monthly gaming volume of Iowa through a wide, persistent comparison in terms of mean squared error (MSE) and mean absolute percentage error (MAPE).
**Research Questions**

The following questions guided this study:

1. Can the ANN technique replace ARIMA as a more effective approach to seasonal gaming time series forecasting?

2. Do the conclusions of the M-competition, that the simple moving average performs as well as or even better than do sophisticated ones, still hold in this study?

3. What is the best model with regard to different forecasting horizons?

4. What is the difference between the forecasting abilities of each method given different variable indicators?

**Significance of the Study**

As more ANN methods are introduced in the literature and utilized in the tourism field (Song & Li, 2008), the irrereplaceable advantages of ANNs in forecasting are mentioned in hospitality literature. However, literature about the introduction of ANN techniques in gaming studies and their investigations are scarce regarding the advantages of ANNs in seasonal gaming data. In addition, to this author’s best knowledge, no study in gaming literature is found to have been conducted a comparison of the forecasting abilities of commonly used methods for multiple horizons. Therefore, this study serves to bridge the gap in the gaming literature about ANN methods, answer questions about the superiority of ARIMA techniques in gaming studies, verify conclusions about the M-competition in the gaming field, and provide
researchers and practitioners with a guide to and insights into seasonal data analyses, method selections in gaming forecasting, and the effectiveness of models in different horizons in gaming forecasting.

**Definitions of Terms**

The following terms used in this research are defined as follows:

*Akaike’s information criterion (AIC):* a criterion of the relative quality of a statistical model given a set of data. That is, given a collection of competing models for a set of data, AIC estimates the quality for each model, relative to each of the other models. It deals with the tradeoff between the goodness of fit ($-2 \log$ likelihood) and the complexity ($M$) of the model, using the expression is $\text{AIC} = -2 \log \text{likelihood} + 2M$. Where $\log \text{likelihood} \approx -n \log \sum_{t}^{T} (Z_t - \hat{Z}_t)^2$ and $M$ is the number of estimated parameters ($p + q$) in the ARIMA model (Bozdogan, 2000).

*Artificial neural network (ANN):* a parallel-distributed processor inspired by biological neural networks (in particular the brain) that capture information from among a large number of inputs, store it, and approximate functions with the experiential knowledge. It resembles the brain in two respects: first, knowledge is acquired by the network through a learning process and, second, the interneuron connection strength, known as synaptic weights, is used to store knowledge (Ripley, 1996).

*Autocorrelation (AC) criterion:* a general approach to determine the numbers of input variables for an ANN. Deciding the number of input nodes heavily impacts learning and prediction abilities of the network and, furthermore, of forecasting performance. AC is
performed between two values of the same variable at times \( t \) and \( t + k \) instead of a correlation between two different variables. Mathematically, given measurements, \( Z_1, Z_2, \ldots, Z_n \), the lag \( k \) AC function is defined as

\[
\gamma_k = \frac{\sum_{t=1}^{n-k}(Z_t - \bar{Z})(Z_{t+k} - \bar{Z})}{\sum_{t=1}^{n}(Z_t - \bar{Z})^2}.
\]

The AC criterion is according to the empirical results from Huang (2004). The inputs should not be correlated in ANN forecasting. Essentially, correlated input variables contain the same information in different forms and they degrade ANN performance by interacting with each other according to Huang. Generally, the continuous lag periods have higher degrees of correlation.

**Autoregressive integrated moving average (ARIMA) model:** generally referred to as an ARIMA(\( p, d, q \)) model, where parameters \( p, d, \) and \( q \) are nonnegative integers that refer to the order of the autoregressive, integrated, and moving average parts of the model, respectively. The ARIMA model’s equation is \( \phi_p(B)(1 - B)^dZ_t = \theta_q(B)a_t \), where \( \phi_p(B) \) denotes autoregressive (AR) factors; \( \theta_q(B) \) denotes moving average (MA) factors; \( B \) is the backshift operator (e.g., \( BZ_t = Z_{t-1} \) and \( (1-B)Z_t = Z_t - Z_{t-1} \)); \( Z_t \) denotes the observed value; \( a_t \) denotes observed error; and \( d, p, \) and \( q \) denote the number of differencing, autoregressive terms, and moving average terms, respectively (Wei, 2004).

**Mean squared error (MSE) and mean absolute percent error (MAPE):** the criteria used to compare the methods according to their equations:

\[
\text{MSE} = \frac{1}{T} \sum_{t=1}^{T}[(F - A)^2]
\]

and
MAPE = \frac{1}{T} \sum_{t=1}^{T} \left\{ \frac{|F_t - A_t|}{A_t} \right\} \times 100,

where \( F \) represents the forecasting value, \( A \) represents the actual value, and \( T \) represents the number of the time periods. MSE can be used to recognize the best forecasting method by magnifying the larger errors, whereas MAPE is a more objective measure because it is computed in relative percentages (Hamzaçebi, 2008).

Sample autocorrelation function (ACF): the functions used to identify the tentative model satisfying all statistical properties in a stationary state. Sample ACF measures the sample correlation of observations separated by \( k \) time lags. It can be expressed by

\[ \hat{\rho}_k = \frac{\sum_{t=1}^{n} \left( z_t - \bar{z} \right) \left( z_{t+k} - \bar{z} \right)}{\sum_{t=1}^{n} (z_t - \bar{z})^2} \]

where: \( k = 0, 1, 2, \ldots k \); \( \hat{\gamma}_k \) is the \( k \)th order autocovariance of \( z_t \) (Bowerman, O’Connell, & Koehler, 2005).

Sample partial autocorrelation function (PACF): the function used to measure the sample correlation of transformed observations separated by \( k \) time lag. For a time series, the partial AC between \( t \) and \( t + k \) is defined as the conditional correlation between \( t \) and \( t + k \), conditional on \( t + 1, \ldots, t + k - 1 \) between the time points \( t \) and \( t + k \). PACF \( \Phi_{k,k} \) can be computed through ACF using \( \Phi_{1,1} = \rho_1 \), if \( k = 1 \), and

\[ \Phi_{k,k} = \frac{\rho_k - \sum_{j=1}^{k-1} \Phi_{k-1,j} \rho_{k-j}}{1 - \sum_{j=1}^{k-1} \Phi_{k-1,j} \rho_j} \]

if \( k = 2, 3, \ldots k-1 \), where \( \Phi_{k,j} = \Phi_{k-1,j} - \Phi_{k,k} \Phi_{k-1,k-j} \) \( (k = 3, 4, \ldots; j = 1, 2, \ldots, k - 1) \)

(Bowerman et al., 2005).

Seasonal autoregressive integrated moving average (SARIMA) model: generally referred to as a SARIMA \((p, d, q)(P, D, Q)_s\) model, where parameters \( P, D, \) and \( Q \) are nonnegative integers that refer to the order of the seasonal autoregressive, integrated, and moving average parts of the model, respectively, and \( p, d, \) and \( q \) are nonnegative integers that
refer to the order of the nonseasonal ones, respectively. The SARIMA model equation is
\[ \phi_p(B^S) \phi_p(1 - B)^d(1 - B^S)^D Z_t = \Theta_Q(B^S)\theta_q(B)\alpha_t, \]
where \( \phi_p(B^S) \) denotes seasonal AR factors; \( \Theta_Q(B^S) \) denotes seasonal MA factors; and \( D, P, Q, \) and \( S \) denote the number of seasonal differencing, autoregressive terms, moving average terms, and the number of time periods within a seasonal cycle (Wei, 2004).

**Simple moving average (SMA):** a method that simply averages a certain number (denoted as \( n \)) of previous time series values to forecast the value for the next time period. The key to successful SMA forecasting is to identify the optimal value for \( n \) that leads to minimum forecasting error. This method was chosen because previous studies have shown less advanced forecasting methods (e.g., SMA, single exponential smoothing) perform even better than some complex techniques on time series in different forecasting horizons (Makridakis et al., 1982).

**Slot coin-in:** the amount of money wagered in slot machines, deemed the most appropriate gaming volume measurement (Eisendrath et al., 2008; Lucas, Dunn, & Kharitonova, 2006). Slot coin-in has increased to 50–80% of the entire casino revenue (Abarbanel, Lucas, & Singh, 2011; Ahlgren & Singh, 2011). Eisendrath et al. (2008) selected coin-in to investigate certain gaming volumes on the Las Vegas Strip after 9/11. Lucas (2010) used slot coin-in to predict untracked daily gaming volumes in a Las Vegas Strip hotel-casino. Abarbanel et al. (2011) consistently utilized slot coin-in to measure gaming volume. This wide employment of coin-in as an indicator shows the representativeness of coin-in for gaming volume. Slot coin-in and table drop were used to measure Iowa monthly gaming volume.
Table drop: Dollar amount wagered on table games and converted to chips for table games (Ahlgren et al., 2011). Suh and Lucas (2011) used daily coin-in, as well as table drop, for casino gaming volume, to determine the condition causing the volume increase. Another reason for using table drop in this study was to compare the efficiency of different prediction models. Technically, gaming volumes technically are not measured as revenues, because revenues are unstable in the short term; vary according to many different conditions such as winnings, taxes, fees, and so forth; and are unable to express the actual gaming demand (Kilby & Fox, 2005). Instead, slot coin-in and table drop are more appropriate for gaming volume measure because of their known stability (Lucas, 2010).
CHAPTER 2

LITERATURE REVIEW

Times Series Forecasting

Forecast accuracy is critical in major functional areas of business, especially as it relates to inventory management, demand forecasting, and financial performance in the hospitality practice, because management bases it decisions and plans, to a large extent, on an adequate forecast. However, it is usually difficult to obtain an accurate forecast with a backdrop of an uncertain economic environment and changing government policies. In practice and in research, models, which are simplified representation of reality, are used as a scientific approach for forecasting for a specific interest (Morlidge & Player, 2009). Generally, three types of models, judgmental, mathematical, and statistical models, and their combinations, are commonly used in applications. Judgment describes the process with an implicit mental model in an individual’s head. Judgmental techniques depend highly on the expertise, such as experience and knowledge, of managers and are commonly applied in practices with a majority of them reaching high levels of forecast accuracy. Mathematical models use mathematical equations and theories to simulate reality and empirically forecast future behaviors. They are time consuming and not able to deal with novelties and changes. Statistical models are employed to extrapolate a forecast if there are plenty of historical data, are good at revealing relationships, and are widely used in forecasting practices and research (Morlidge & Player, 2009).

In the literature, time series forecasting techniques are included in statistical approaches that extrapolate forecasts through modeling present and past values that might
have internal structures (such as trends and seasonal variations), based on the assumption that
the characteristics of previous observed values will recur in future values (Box & Jenkins,
1970). In other words, the model that fits the time series data best provides the most accurate
forecasting results. Since the 1970s, many studies have investigated seasonal time series
forecasting modeling with parameters and nonparameters. One of the most popular time
series models with parameters is the seasonal autoregressive integrated moving average
(SARIMA) and popular time series models with nonparameters include ANNs, specified as
generalized regression neural networks in their study; other useful time series models with
parameters, dealing with unpredictable and uncertainty factors in seasonal features data,
include fuzzy regression, least squares support vector regression, and some combination
models. In addition, some simple smoothing methods, such as SMA and so on, were verified as
being effective in forecasting RevPAR in the lodging area of hospitality (Zheng, Bloom, Wang, &
Schrier, 2012) and provide more insights into the collection of the competing time series
forecasting models.

**ARIMA/SARIMA Models with Box-Jenkins Procedure**

The ARIMA is also known as ARIMA(p,d,q) model, where parameters $p$, $d$, and $q$ are
nonnegative integers that refer to the order of the autoregressive, integrated, and moving
average parts of the model, respectively. The ARIMA model’s equation is

$$\phi_p(B)(1 - B)^d Z_t = \theta_q(B)\alpha_t,$$

where: $\phi_p(B)$ denotes AR factors;

$\theta_q(B)$ denotes MA factors;
B is the backshift operator;

\( Z_t \) denotes observed value;

\( a_t \) denotes observed error, \( BZ_t = Z_{t-1} \);

And \( (1 - B)Z_t = Z_t - Z_{t-1} \) (Wei, 2004).

The SARIMA model generally is referred to as a SARIMA\((p, d, q)(P, D, Q)\) s model, where parameters \( P, D, \) and \( Q \) are nonnegative integers that refer to the order of the seasonal autoregressive, integrated, and moving average parts of the model, respectively, and \( p, d, \) and \( q \) are nonnegative integers that refer to the order of the nonseasonal ones, respectively. The SARIMA model’s equation is

\[
\Phi_p (B^S) \phi_p (1 - B)^d (1 - B^S)^D Z_t = \Theta_Q (B^S) \theta_q (B) a_t,
\]

where: \( \Phi_p (B^S) \) denotes seasonal AR factors;

\( \Theta_Q (B^S) \) denotes seasonal MA factors;

\( D \) denotes the number of seasonal differencing;

\( P \) denotes the number of autoregressive terms;

\( Q \) denotes the number of moving average terms;

\( S \) denotes the number of the number of time periods within a seasonal cycle;

and \( (1 - B)(1 - B^{12})Z_t = (Z_t - Z_{t-1}) - (Z_{t-12} - Z_{t-13}) \) (Wei, 2004).

The Box-Jenkins procedure is a sophisticated technique that fits ARIMA models with time series data (Box & Jenkins, 1976). Autoregression is a process by where the time series value is the weighted average of previous values; Integration is a process that makes data
stationary by differencing; moving average (MA) is a process by which the time series value is the weighted average of forecasting errors of previous values. The Box-Jenkins procedure includes three important steps: (1) sequentially tentative model identification, (2) parameter estimation, and (3) diagnostic checking.

In the tentative identification stage, the data are transformed to a stationary condition through differencing. Regular or seasonal differencing (or both) is performed to remove trend and seasonality and to help transform the time series data into stationary data. Stationary data mean the statistical properties of the time series data, which include mean ($\mu = \mu_t$), variance ($\sigma^2 = \sigma^2_t$), and AC ($\rho_k$, depending only on $k$), are essentially constant over time (Bowerman et al., 2005). Sample autocorrelation function (ACF) and sample partial autocorrelation function (PACF) are used to identify the tentative model satisfying all statistical properties in the stationary state. ACF measures sample correlations of observations separated by $k$ time lag; its formula is

$$\rho_k = \frac{\sum_{t=b}^{n-b+k} (z_t - \bar{z})(z_{t+k} - \bar{z})}{\sum_{t=b}^{n-b} (z_t - \bar{z})^2},$$

where $\bar{z} = \frac{\sum_{t=b}^{n-b} z_t}{n - b + 1}$ and the range is $[-1,1]$. In other words, $\rho_k = \frac{\gamma_k}{\gamma_o}$, where $\gamma_k$ is the $k$th order autocovariance of $z_t$. PACF measures the correlation of transformed observations separated by $k$ time lag; its formula is $\phi_{1,1} = \rho_1$, if $k = 1$, and the partial AC of the $k$th order is defined as

$$\Phi_{k,k} = \frac{\rho_k - \sum_{j=1}^{k-1} \phi_{k-1,j} \rho_{k-j}}{1 - \sum_{j=1}^{k-1} \phi_{k-1,j} \rho_j},$$

if $k = 2, 3, \ldots$ (Bowerman et al., 2005). By examining the constructions of the ACF and the PACF plots, the number of model parameters can be determined (Balaguer, Palomares, Soria, & Martín-Guerrero, 2008); that is, a suitable model is then identified. A tentative identified model could be with a moving average (MA), an autoregressive factor (AR), or a combination.
The next stage is estimation. The most widely used estimation method is maximum likelihood estimation or nonlinear least-squares estimation. The model parameters are estimated through checking the sample’s ACF and PACF (Cryer & Chan, 2008), and the model fitting is according to experience and expertise and supported by significant estimation results calculated by a statistical package.

The last step is diagnostic checking. When there are multiple adequate models, the selection criterion is normally based on the summary statistics from residuals computed by a fitted model or on forecast errors computed from the out-sample forecasts. The general criterion is Akaike’s information criterion (AIC). Its expression is

\[ \text{AIC} = -2 \log (\text{likelihood}) + 2M. \]

Where \( \log (\text{likelihood}) \approx -n \log \sum_1^n (Z_t - \hat{Z}_t)^2 \) and \( M \) is the number of estimated parameters in the model (usually \( M = p + q \)). With the tentative model, the checking points lie in the following aspects: First, the residuals should be independent of each other and constant in mean and variance over time (checking the plots of the mean and variance of residuals over time, of residuals versus fitted values, and of normal probability of residuals (Q–Q plot for normal assumption and then performing a Ljung-Box test or checking the plots of ACF and PACF of these residuals). The Ljung-Box diagnostic check is based on residuals versus time. Its null hypothesis:

\[ \rho_1(a) = \rho_2(a) = \ldots = \rho_k(a) = 0. \]

Its alternative hypothesis:

at least one \( \rho_k(a) \neq 0. \)

Test statistics:
\[ Q = n(n + 2) \sum_{k=1}^{K} (n - k)^{-1} \rho_k^2, \]

where \( \rho_k^2 \) is the estimated AC of the time series at lag \( k \) and \( K \) is the number of lags being tested.

If \( Q > \chi^2_{1-\alpha; K-p-q} \) (chi-square), then reject its null hypothesis. That is, at least one \( \rho_k(a) \neq 0 \).

This indicates a better model is needed (Wei, 2004).

**Advantages and Applications of the ARIMA/SARIMA in the Gaming Field**

The ARIMA/SARIMA model is a widely applied time series model with significant advantages. First, it performs much better than an ordinary least squares (OLS) regression approach because, for liner regression, the adjacent error terms are correlated so that problems come forth when the mean of the time series varies as the time level increases. Furthermore, the standard errors for the OLS parameter estimates are biased when the covariance of error terms do not equal zero (McDowall, McCleary, Meidinger, & Hay, 1980). Second, an important advantage of ARIMA is its abilities to deal with nonstationary time series and to capture patterns, systematic trends, and seasonality. Specifically, it can convert a nonstationary time series to a stationary time series by taking the proper number of differencing (Ismail, Suhartono, Yahaya, & Efendi, 2009).

The ARIMA/SARIMA model has been used successfully in many forecasting fields such as business revenue (Dhahri & Chabchoub, 2007), tourism demand (Burger, Dohnal, Kathrada & Law, 2001; Goh & Law, 2002; Kulendran & Shan, 2002; Lin, Chen, & Lee, 2011), and rainfall in meteorology (Castellano-Méndez, González-Manteiga, Febrero-Bande, Prada-Sánchez, &
Lozano-Calderón, 2004; Valipour, Banihabib, & Behbahani, 2013). In addition, through a thorough review of the literature found the Box-Jenkins procedure to be the most commonly used time series analysis technique for forecasting gaming volume in related gaming studies. During the last decade, most researchers, such as Lucas et al. (2006) have extensively applied the ARIMA model to studies in the gaming field. They also have repeatedly highlighted the advantages of the ARIMA model on gaming data. As early as 1978, Cargill and Eadington (1978) pioneered the use of the ARIMA process to forecast quarterly gross gaming revenues for each of the three major tourist areas in Nevada. They found the performance of the ARIMA techniques is good or better than regression or other econometric techniques and concluded the main advantage of ARIMA is its ability to capture seasonal and systematic trends. Later, Shonkwiler (1992) also validated ARIMA could model seasonality and overall temporal trends of gaming data. In addition, by combining the ARIMA models with intervention analysis, Nichols (1998) investigated the effects of increased operating hours and expanded slot machine floor space for gaming demand in Atlantic City casinos. Following these studies, ARIMA models with intervention (special events or economic crisis) analyses abounded in the gaming literature. Moss, Ryan, and Parker (2004) used these models to compare the effects of 9/11 on casino gaming revenues in Las Vegas and Mississippi. In addition, Eisendrath et al. (2008) selected them to analyze the impacts of the events of 9/11 on gaming volumes on the Las Vegas Strip. They also indicated the ARIMA model developed in forecasting slot coin-in was a good fit for the data. Gounopoulos, Petmezas, and Santamaria (2012) employed ARIMA models to forecast tourist arrivals in Greece and to measure the impact of macroeconomic shocks. Their study concluded ARIMA outperformed the exponential smoothing models to forecast the directions from sample forecasts.
Simple Moving Average and M-Competition

Makridakis and Hibon (1979) were the first to compare a large number of major time series methods across multiple series. Makridakis et al. (1982) compared 24 various forecasting methods, using 1,001 economic time series, and concluded that the performance of the different forecasting methods varies for different forecasting horizons. Later, Makridakis et al. (1993) and Makridakis and Hibon (2000) expanded this research. They extended the number of time series to 3,003 and the forecasting fields into multiple areas. Makridakis and colleagues confirmed the previous conclusion that statistically sophisticated methods do not provide more accurate forecasts than simpler methods do and that the accuracy of the various methods depends upon the length of the forecasting horizon involved.

The SMA and naïve methods were chosen for this study because previous studies have shown that less advanced forecasting methods perform even better than some complex techniques using time series with certain forecasting horizons (Makridakis et al., 1982; Makridakis & Hibon, 2000; Zheng et al., 2012). Based on findings of M-competition, selecting simple methods has become one of the principles of method selection (Armstrong, 2001). In addition, simple methods such as naïve and SMA were most often used and account for 30.6 and 20.9%, respectively, in regularly used extrapolating methods, according to Daleymple’s (1987) survey. Therefore, naïve and SMA methods have comparable worth and practical implications. Thus, this study compared the performance of SMA with that of more sophisticated methods in different multiple forecasting horizons to examine the conclusions of M-competition. The forecasting period includes 24 months and was divided into eight time horizons. Then, the estimated values for each horizon were compared with the holdout sample data to evaluate forecasting abilities for each forecasting method.
Artificial Neural Networks

An artificial neural network is a parallel-distributed processor inspired by biological neural networks (in particular the brain) that capture information among a large number of inputs, store it, and approximate functions with the experiential knowledge gained. It resembles the brain in two respects. First, knowledge is acquired by the network through a learning process and second, the interneuron connection strength, known as synaptic weights, is used to store knowledge (Ripley, 1996). ANNs consist of three key factors—the input layer, the output layer, and one or more hidden layers. Each layer consists of multiple neurons. Through the interactions of these neurons, this model can capture the characteristics of the data. Input layers contain the predictors (input variables), output layers contain the responses, and hidden layers contain unobservable nodes and units. The value for each hidden unit is some function of the predictors. The exact form of the function depends, in part, upon the network type and, in part, upon user-controllable specifications. The value of output can be expressed as

\[
Z_{t+l} = \alpha_l + \sum_{j=1}^{m} wf \sum_{i=0}^{s-1} v Z_{t-i} + \theta_j.
\]

where: \(Z_{t+l}\) \((l = 1, 2, \ldots, s)\) represents the predictions for the future \(s\) periods;

\(Z_{t-i}\) \((i = 0, 1, 2, \ldots, s - 1)\) is the observations of the previous \(s\) periods;

\(v\) is the weight of the connections from the input layer neurons to the hidden layer neurons;

\(w\) is the weight of connections from hidden layer neurons to output layer neurons;
\( \alpha l \) and \( \theta_j \) are weights of bias connections;

and \( f \) is the activation function.

The ACF also was applied to the determination of input variables for Iowa gaming demand on slot coin-in and table drop, and analyzed the forecasting performance under AC lag periods. According to Huang's (2004) study, this approach has been demonstrated with a neural network trained with a back-propagation algorithm. Future research will attempt to demonstrate the effectiveness the approach holds for all learning neural network training algorithms (e.g., radial basis function, probabilistic, etc.) and is a general principle for time series modeling. Two broad types of model selection approaches are often commonly used in the ANN literature—cross-validation and sample model selection—based on certain criterion. A comprehensive examination of the effectiveness of a variety of models is commonly used in sample model selection criteria.

**Advantages and Disadvantages of ANN Compared with Other Techniques**

An ANN has three fundamental features—parallel processing, distributed memory, and adaptability—as advantages over other approaches. Furthermore, Gritta, Wang, Davalos, and Chow (2000) found one of the strongest features of an ANNs is its learning ability, specifically extracting inherent relationships from data. ANNs can deal with data that are incomplete, inconsistent, and ambiguous (Venugopal & Baets, 1994). Traditional methods, such as linear regression, also can acquire knowledge through the OLS method and store this information in regression coefficients. However, linear regression has a rigid model structure
and a set of assumptions imposed before learning from the data: whereas, a neural network determines the relationship through the learning process. Therefore, if a nonlinear relationship is identified, a neural network can automatically produce the correct model structure, which the linear regression has no capability of achieving.

There is no formal method for determining an optimal network (Goss & Ramchandani, 1995). Network constructions, such as the appropriate number of layers, hidden layer nodes, and appropriate learning and momentum rates, are determined through trial and error (Tsaur, Chiu, & Huang, 2002). In addition, there are no formulas developed to determine the sample size for developing ANNs to achieve the desired accuracy (Venugopal & Baets, 1994). Due to the lack of self-explanatory capabilities, the interpretation of ANN models requires more expertise from the user than traditional statistical models (Goss & Ramchandani, 1995; Metaxiotis & Psarras, 2004). The training of ANNs also requires significant amounts of time compared to other techniques, especially when both input and output variables are continuous (Etheridge, Sriram, & Hsu, 2000). Finally, ANN model developments also are needed for intensive computing, and the outcomes are sensitive to the selection of the learning rate (Salchenberger, Cinar, & Lash, 1992).

Applications of ANNs

Nevertheless, neural networks are the favorite tool for many predictive data mining applications because of their power, flexibility, and ease of use. Lately, they have attracted increasing interest. Predictive neural networks are particularly useful in applications in which the underlying process is complex. Neural networks used in predictive applications are supervised in the sense the model-predicted result can be compared against known values of
the target variables. They have been viewed as a promising alternative to traditional linear methods, due to their capability of capturing nonlinear relationships between input variables and output variables (Zhang, Patuwo, & Hu, 1998). ANNs have been applied to a diversity of complex problems across different disciplines such as psychology, computer science, engineering, medical diagnostics, target marketing, and market share prediction (Kuo & Reitsch, 1995) as well as marketing and forecasting (Hamzaçebi, 2008; Metaxiotis & Psarras, 2004; Valipour et al., 2013). Although ANNs are applicable to many different areas, the most frequently cited applications for ANNs are found in banking and finance (Metaxiotis & Psarras, 2004). ANNs have demonstrated successes in areas such as bond rating, credit applications, risk assessment of mortgages and loans, stock-market predictions, and financial forecasting and analysis (Odom & Sharda, 1990). In particular, ANNs have been used and proven efficient for modeling complex classification problems such as insolvency prediction in the insurance industry (Pawlik et al., 2005; Davalos, Gritta, & Chow, 1999; Goss & Ramchandani, 1995).

Specifically in the hospitality industry, Pattie and Snyder (1996) were pioneers in testing the predictive ability of ANN models in the area of hospitality and tourism. They found these artificial intelligence models generated the most accurate prediction results by comparing them to traditional statistical prediction models in forecasting tourist behaviors. Since then, ANN models have been widely used in forecasting tourism demand. Law and Au (1999) showed the outperformance of ANN models compared with multiple regressions, naïve, SMA, and exponential smoothing methods to forecast demand for travel to Hong Kong. Tsaur et al. (2002) utilized an ANN technique to analyze guest loyalty toward international tourist hotels. They incorporated guests’ assessments on eight service aspects as inputs and the loyalty measures as outputs. Then, they developed an ANN model to establish the functional relationship among these variables. In the meantime, they used logistic models using the same
dataset as a benchmark for comparison. Their results showed that the ANN model outperforms logistic models, due to the fact that ANN models have the capability of modeling nonlinear interactions between variables. By applying the ANN model to forecast tourist arrivals for Hong Kong’s hotel industry, Cho (2003) further found in his study that the ANN model was the best method for accurately forecasting visitor arrivals. Palmer, Sesé, and Montaño (2005) designed a neural network for tourism time series forecasting and concluded that ANNs are an effective, flexible statistical tool in the tourism forecasting field but, compared with the Box-Jenkins technique, they lack a theoretical background and a systematic procedure for model building.

Comparisons of the ARIMA and ANNs

The accuracy of a time series, forecasting model is fundamental to many decision processes. Hence, research for improving the forecasting modeling has never stopped. The ARIMA model has become one of the most popular methods in forecasting research and practice. Wide utilizations of it in tourism forecasting studies indicate the ARIMA model is a good forecasting model fit to data featured in seasonality and trends (Cargill & Eadington, 1978). Gounopoulos et al. (2012) employed ARIMA models to forecast tourist arrivals in Greece and concluded that ARIMA outperformed the exponential smoothing models to predict from sample forecasts. Nevertheless, ANN techniques lately have been viewed as a promising alternative to linear methods, due to their capability to decipher patterns and to capture nonlinear relationships between input variables and output variables (Zhang et al., 1998). The advantages of ANNs over ARIMA include that: (a) ANNs are very versatile and have universal
functionality for nonlinear relationship and (b) ANNs can tolerate chaotic components, as demonstrated by Masters (1995).

Comparisons of the performances of ARIMA, ANNs, and other methods have been developing in many fields, but there is no general view on forecasting accuracy. ANNs have been validated to outperform other techniques in psychology, economics, statistics, computers, game playing, and decision making, but sometimes some mixed results have been reported. Maier and Dandy (1996) mentioned that the ARIMA model is better suited for short-term forecasts, whereas the ANN is better suited for longer-term forecasts. In the tourism field, Burger et al. (2001) used ARIMA, ANNs, and other methods to model tourism demand in South Africa. They found that the error rate established by the ANN model is the lowest. Cho (2003) used monthly data with the ARIMA, ANN, and other methods to predict the number of tourists. He determined the ANN model had the best performance, too. Palmer et al. (2005) designed a neural network for tourism time series forecasting and concluded that ANNs are an effective, flexible statistical tool in the tourism field for forecasting. However, Palmer et al. (2005) also mentioned that ANNs, compared with the Box-Jenkins methodology, lack a theoretical background and a systematic procedure for model building. Also, the ANN models have more parameters to estimate and are difficult to interpret. Kolarik and Rudorfer (1994) noted that ANNs are not stable in forecasting seasonal time series. Finally, Lin et al. (2011) applied ARIMA and ANNs to a dataset of monthly visitors to forecast tourism demand and concluded, for their dataset, ARIMA outperformed ANNs in terms of root-mean-square error, mean absolute deviation, and MAPE.

Commonly used forecasting methods have always been compared during different forecasting competitions. Makridakis et al. (1982) conducted early, famous competitions.
Their results with 1,001 time series, including holdout data, are compatible with the results acquired from this study. Results from this study support that the best forecasting model is the simplest one. Foster, Collopy, and Ungar (1992), who did a comparison using data from M-competition. Comparing ANNs with traditional statistics forecasting models, in a study investigating the forecasting accuracy in different forecasting horizons, they found that ANN was inferior for time series of yearly data as well as for quarterly data and monthly data. These findings conflict with those from a study by Sharda and Patil (1992), which determined that an ANN produced results comparable to the Box-Jenkins model, showing that forecasting results will not improve when the forecasting period is broad. Sharda and Patil (1990) utilized 75 time series data to determine that an ANN performed as well as the Box-Jenkins procedure, a finding that does not agree with the results from the present study. The ANN did not outperform the Box-Jenkins procedure with 180 time series, including holdout data. Sample size may have contributed to the contradictory results. The notable aspect is that, in most cases, forecasters may not select the best neural network mode and the performance of neural networks may not be generalized. The data processing procedures impact the performance of neural networks in learning and generalizability. Detrending and dedeseasonalizing are both effective procedures to reduce model fitting and forecasting errors (Zhang & Qi, 2005).
CHAPTER 3

METHODOLOGY

Data Description

The data for this study were retrieved from statistical reports provided by the Iowa Racing and Gaming Commission. Statewide monthly aggregated coin-ins and table drops, which are reported in units of millions of dollars, spanned from January 1998 through December 2012 in a total of 180 months, were retrieved from monthly gaming revenue reports for the dataset. The data from January 1998 through December 2010 (156 months in total), a period of time encompassing the most recent recession from December 2008 to June 2009, were chosen as the in-sample data for model fitting. Data from January 2011 through December 2012 (24 months in total) were chosen as the holdout sample data for performance evaluation.

The slot coin-in series included nearly 13 periods and a total of 156 observations (no missing observations). The selection of 13 periods was based on the consideration that this dataset was complete and included a continuous increasing trend within two decades. The raw slot coin-in, shown in Figure 1, revealed that slot coin-in was increasing as the time level increased, except for a short drop in the period of 2008–2009, which reflects an economic recession. Since approximately 2007, the variances for the slot revenues increased greatly, although the slot revenues also increased quickly. However, the table drops series owns a different pattern from slot coin-in series. As shown in Figure 2, the growth rate of table drop is very slow and even decrease until around 2003. Since then, the demand of table drop increases speedily. Moreover, the growth rate of demand of the table drop is even higher than that of demand of the coin-in. Corresponded to the coin-in, the table drop also has quicker
decline in economic recession of the period of 2008–2009. It illustrates that table drop was more sensitive to the gaming market.

Figure 1. Raw data plot of slot coin-in (the slot coin-ins from 1998 to 2010 were scaled down to 1/1,000 versus time)

Figure 2. Raw data plot of table drop (slot drops from 1998 to 2010 were scaled down to 1/1000 versus time)
Study Design

As mentioned previously, four time series forecasting methods, that is, ARIMA, ANNs, SMA, and the naïve method, were applied to produce the forecasting models. The four models emerged through fitting the in-sample data adequately and then were selected to compete in the holdout sample, that is, the succeeding 24 months (January 2011 through December 2012). These 24 months were evenly divided into eight time horizons. The comparisons or competitions were completed in terms of two performance criteria: MSE and MAPE (Valipour, 2012).

Seasonal ARIMA Model with the Box-Jenkins Procedure

Tentative model identification, parameter estimation, and diagnostic checking are three key steps to attend sequentially for the Box-Jenkins procedure. R-Studio was utilized to conduct this procedure. The fundamental time series theorem implies that all stationary time series can be decomposed into an uncorrelated a purely deterministic component and a purely indeterministic component (Chatfield, 2004). Reaching stationarity of raw data is the first priority for sophisticated forecasting methods.

Tentative identification function was used from the packages of R-Studio to identify the model. As the range-mean plot shown in Figure 3 illustrates, the distribution of range-means plots seems to have a pattern. Therefore, the Box-Cox transformation was adopted to make the variance constant.
In 1964, Box and Cox developed an equation of transformations, which was widely applied in normalizing data or equalizing variance. Its expression is

$$Z_t = \begin{cases} \frac{(Z_t^* + m)^\gamma - 1}{\log(Z_t^* + m)} & \gamma \neq 0 \\ \log(Z_t^* + m) & \gamma = 0 \end{cases}$$

Where $\gamma$ = gamma and (or are called $\lambda$ = lambda in the other expression equations shown in Appendix B), and the quantity $m$ is typically chosen to be 0. Theoretically, transformation can be maximally calibrated to be effective in moving a variable toward normality and homogeneity (Osborne, 2010). In Box-Cox transformation,

Gamma = 0: natural log transformation;

0.33: cube root transformation;

0.5: square root transformation;

And 1: no transformation.
After trying gamma = 0.5, 0.333, and 0, it was found that when gamma = 0.33 was applied, the points of the range-mean plot distributed more randomly. Thus, 0.33 was selected as the gamma value for slot coin-in series. The result of each transformation is shown in Figure 4; And the results of table drop were included in Appendix A.

Figure 4. Comparison of the tentative identification plots with the different gamma values for slot coin-in
Concerned with the seasonality feature of the PACF, first order differencing was taken on the transformed data and the tentative model $d = 1, D = 0$ was identified when gamma = 0.33. It was found that the spikes of the PACF salient in lag 12, 24, and 36 showed strong seasonality. Then, the first seasonal differencing was taken on the transformed raw data, tentatively identifying the model $d = 0, D = 1$. However, the data still were not stationary. In addition, because there were four combinations for which $d = 0, 1$ and $D = 0, 1$, shown in Figure 4, the model $d = 1, D = 1$ was tried and found finally to make the data stationary. Usually, the number of applied orders of $p$ or $q$ is no more than 3 (3 is rare), according to experience (Wei, 2004). Therefore, based on the ACF and PACF plot with $d = 1, D = 1$ differencing shown in Figure 4 and the general knowledge shown in Table 1, models SARIMA (2,1,0)(0,0,2), SARIMA (0,1,1)(0,1,0), SARIMA (2,1,0)(0,1,3), and SARIMA (2,1,0)(0,1,2) were tentatively identified for diagnostic checking.
Figure 5. Comparison of the tentative identification plots after the first order differencing for slot coin-in

Table 1. Characteristics of Theoretical ACF and PACF for Stationary Process

<table>
<thead>
<tr>
<th>Process</th>
<th>AR (p)</th>
<th>MA (q)</th>
<th>ARIMA (p, q)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACF</td>
<td>Tails off</td>
<td>Cuts off after lag $q$</td>
<td>Tails off after lag $(p - q)$</td>
</tr>
<tr>
<td>PACF</td>
<td>Cuts off after lag $p$</td>
<td>Tails off</td>
<td>Tails off after lag $(p - q)$</td>
</tr>
</tbody>
</table>

Note. AR = autoregressive; MA = moving average; ARIMA = Autoregressive integrated moving average

Autoregressive integrated moving average; ACF = Autocorrelation function; PACF = Partial autocorrelation function.

Model estimation function was then used to acquire output statistics for model comparison, as shown in Figure 5. In viewing the comparisons, model 4, SARIMA $(2, 1, 0)(0, 1, 2)_{12}$, is better than others because of the smaller AIC and the not significant Ljung-Box statistics. Also, the model agrees with the principle of pursuing the lowest number of model parameters (Zheng & Gu, 2011). The random distribution of the residuals and normal
distribution of the Q–Q plot is shown in Figure 5. In addition, model 4 was chosen as the final forecasting model because (a) the AIC number did not decrease significantly after the addition of a MA parameter in Model 3, which indicates that Model 3 might be an overfit; and (b) Model 4 was easier to interpret, as it is parsimonious.

Model 1: SARIMA(2,1,0)(0,0,2)_{12}

Model 2: SARIMA(0,1,1)(0,1,0)_{12}

Model 3: SARIMA(2,1,0)(0,1,3)_{12}
Figure 6. Comparison among alternative models

Model 4: SARIMA(2,1,0)(0,1,2)_{12}
Table 2 Summary of the Comparison of Model Estimation Results

<table>
<thead>
<tr>
<th>Results</th>
<th>Model</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SARIMA (2,1,0) (0,0,2)</td>
<td>SARIMA (0,1,1) (0,1,0)</td>
<td>SARIMA (2,1,0) (0,1,3)</td>
<td>SARIMA (2,1,0)(0,1,2)</td>
<td></td>
</tr>
<tr>
<td>Box-Cox transformation parameter γ</td>
<td>0.33</td>
<td>0.33</td>
<td>0.33</td>
<td>0.33</td>
<td></td>
</tr>
<tr>
<td>Seasonal differences</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Regular differences</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Φ1</td>
<td>−0.528 (−6.65)</td>
<td>−0.633 (−7.63)</td>
<td>−0.612 (−7.56)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Φ2</td>
<td>−0.147 (−1.84)</td>
<td>−0.196 (−2.3)</td>
<td>−0.251 (−3.08)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Θ1</td>
<td>−0.646 (−9.41)</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td></td>
</tr>
<tr>
<td>Θ1(seasonal)</td>
<td>0.56 (7.29)</td>
<td>−0.519 (6.5)</td>
<td>−0.79 (−6.4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Θ2(seasonal)</td>
<td>0.297 (4.11)</td>
<td>−0.44 (5.3)</td>
<td>−0.209 (−2.42)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Θ3(seasonal)</td>
<td>−</td>
<td>−</td>
<td>0.362 (4.35)</td>
<td>−</td>
<td></td>
</tr>
<tr>
<td>Significant pk(a)</td>
<td>30, 33</td>
<td>3, 12, 30,</td>
<td>30</td>
<td>34, 35</td>
<td></td>
</tr>
<tr>
<td>$S = \sigma a$</td>
<td>0.184</td>
<td>0.195</td>
<td>0.0586</td>
<td>0.149</td>
<td></td>
</tr>
<tr>
<td>AIC</td>
<td>−69.144</td>
<td>−570.38</td>
<td>−543.06</td>
<td>−600.7</td>
<td></td>
</tr>
<tr>
<td>$-2(\log \text{likelihood})$</td>
<td>−79.144</td>
<td>−61.38</td>
<td>−555.06</td>
<td>−110.7</td>
<td></td>
</tr>
<tr>
<td>L − B $\chi^2$ (20 df)</td>
<td>38.767</td>
<td>50.051</td>
<td>44.53</td>
<td>37.48</td>
<td></td>
</tr>
<tr>
<td>p value for L − B</td>
<td>0.131</td>
<td>0.012</td>
<td>0.164</td>
<td>0.199</td>
<td></td>
</tr>
<tr>
<td>Number of parameters</td>
<td>4</td>
<td>1</td>
<td>5</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

Note: Values in parentheses are t ratios or p values. AIC = $-2(\log \text{likelihood}) + 2M$, where $M$ is the total number of parameters estimated, which helps correct for lack of parsimony; it is not possible to compare $S$, $-2 \log \text{likelihood}$, or AIC for different values of the Box-Cox parameter $γ$. Degrees of freedom $(df) = m − p − q$. The order of autoregressive factor plus the moving average factor is $p + q$. 
Artificial Neural Network Techniques

A dynamic multilayer perceptron neural network with feed-forward architecture and a back-propagation algorithm was employed because of its representativeness and wide usage (Sarle, 2002; Zhang et al., 1998). Dynamic neural networks feature feedback from the model output (Valipour et al., 2013), and, as Figure 5 shows, the output was also used as input.

In the process of building an autoregressive ANN, the first step is to determine the input variable. The input selection is crucial to later model forecasting performance. There are several criteria used to select the optimal number of inputs and time lag. AC criterion was used in this study, according to the verification of Huang (2004) who showed that AC criterion, a data driven approach, outperforms other criteria in times series forecasting, especially in financial areas. Based on the AC criterion, input variables should have a high degree of correlation to predictors and as low degree of correlation to each other as possible. Therefore, some algorithm steps to pursue optimal lag periods were programmed in MatLab to satisfy

\[(m + 1) = \text{Arg}_k \text{Max}[f(k)],\]

\[f(k) = \frac{\sum_{t=1}^{m} |y_{L-k}|}{\sum_{t=1}^{m} |y_{L-a(m)}|},\]

\[k = a(m) + 1, a(m) + 2, \ldots, N,\]

supposing a series of lag periods is \(a(1), a(2), \ldots, a(m)\).

In addition, in this study, according to Valipour et al.’s (2013) validation of data preparation, the monthly slot coin-in data were arranged as \(M_1\) to \(M_{180}\), and they were divided into an input portion and a target portion. The input portion comprised \(M_1\) to \(M_{132}\), and the target portion comprised \(M_{133}\) to \(M_{180}\). Because the number of input data points in both the
training and test set should be equal (Valipour et al., 2013), the input and output matrices were divided as follows:

**Input matrix for training:**
\[
\begin{bmatrix}
M_{37} & M_{61} \\
M_{38} & M_{62} \\
M_{39} & \ldots & M_{63} \\
M_{58} & M_{82} \\
M_{59} & M_{83} \\
M_{60} & M_{84}
\end{bmatrix}
\]
(24 x 2)

**Input matrix for testing:**
\[
\begin{bmatrix}
M_{85} & M_{109} \\
M_{86} & M_{110} \\
M_{87} & M_{111} \\
M_{106} & M_{130} \\
M_{107} & M_{131} \\
M_{108} & M_{132}
\end{bmatrix}
\]
(24 x 2)

**Target for training:**
\[
\begin{bmatrix}
M_{133} \\
M_{134} \\
\vdots \\
M_{155} \\
M_{156}
\end{bmatrix}
\]
(24 x 1)

**Target for testing:**
\[
\begin{bmatrix}
M_{157} \\
M_{158} \\
\vdots \\
M_{179} \\
M_{180}
\end{bmatrix}
\]
(24 x 1)
The seasonal effects were not removed before the model implementation, according to the findings of Hamzaçebi (2008) that ANNs are appropriate for time series with strong seasonality. Therefore, the monthly coin-in was divided into a training set, a validation set, and a test set. The training set was the largest generally accepted by researchers (West, Brockett, & Golden, 1997) and 70% by default set in the package of econometrics toolbox of MatLab. The learning rate can be set by trial and error (Tsaur et al., 2002). In this study, for the best performance, the distribution of the training set, validation set, and test set were 60%, 20%, and 20%, respectively.

The R output plot shown in Figure 6 reveals that lags 1–12 were significant (signifying that these pikes have a high degree of correlation to the predictor) and could be counted as the number of the time delay of the input variables, as shown in Figure 7. Also, lag 1, 2, and 12 were robust for better performance in trials. Zhang et al. (1998) proposed a guideline that one-half of the nodes of the input layer can be used as nodes for the hidden layer. Thus, the nodes for the hidden layer of the neural network on slot coin-in were set at 6, which provided a satisfying result through trial and error. Just one hidden layer of neurons for the neural networks was sufficient for this study with the least number of hidden neurons performing well and avoided overfitting problems (Masters, 1993)
For the output layer, the number of nodes corresponded to the forecasting horizon (Zhang et al., 1998). The number of output neurons was 1 when the forecasting horizon was 1. It is well known in ANN literature that determining the adequate number of hidden nodes usually takes a considerable amount of trial and error experimentation. This is because an inverse relationship exists between the network’s training performance and the network’s generalization ability. Having a higher number of hidden nodes may increase the model’s performance on a training dataset but at the expense of generalization, as shown by the model’s deteriorated performance on the test dataset (Jain & Nag, 1997). Therefore, one should test the model’s predictive ability not only on the training set but also on the test set before choosing the appropriate number of hidden nodes. For the sample of Iowa’s gaming slot coin-in, the ANN model performed the best with one hidden layer and six hidden nodes, as shown in Figure 8.
Figure 8. The architecture of dynamic neural network of slot coin-in (this information on weight (w) and bias (b) were stored in the neural network through running the data)

With regard to the activation function for the hidden layers, the hyperbolic tangent sigmoid function was selected; its equation is

\[ \gamma(x) = \tan(x) = (1 - e^{-2x})(1 + e^{-2x}). \]

The activation function between the hidden layer and output layer was the identity function; its equation is \( \gamma(c) = c \), and it takes real-value arguments. This is the default function for units in the output layer when an automatic architecture is selected. The adjusted normalized expression for scaling of the variable of the output layer was

\[ \{2*[(x - (\min - \epsilon))/((\max + \epsilon) - (\min - \epsilon))]\} - 1, \]

where \( \epsilon = 0 \) (a very small number). Values for both input and output variables were normalized within [-1, 1] to ensure each variable received equal treatment, regardless of the magnitude of its values. This process can simplify the learning process of ANNs and improve the results (Kim & Lee, 2004).
By training the data through the Levenberg-Marquart algorithm, the best ANN model was selected. The performance of the best ANN model is shown in Figure 8. As illustrated, the MSEs of the model were steadily decreasing during the training part (the model had good performance at the end of training), whereas they were high during both the validation and testing parts (the model did not perform well in either the validation and testing parts).

Figure 8. Performance of the artificial neural network model in slot coin-in

**Simple Moving Average Method and Naïve Method**

The SMA method uses the average of time series values from the most recent $n$ time periods to forecast the value for the next time period. It is the easiest and most straightforward smoothing method. This method can be demonstrated as

$$Y't = (Yt - 1 + Yt - 2 + \ldots + Yt - n) / n,$$

where $Yt - 1$ is the actual value in a time series at time period $t - 1$ and $Y't$ is the forecast of the time series for time period $t$. Microsoft Excel was used to calculate the SMA. Different time
series have a different optimal $n$, and the values of $n$ directly affect the accuracy of the forecasts (Anderson, Sweeney, & Williams, 2006).

Deseasonalization is the method used for data processing in SMA. Twelve seasonal indexes were calculated through the process of deseasonalizing. Finally, the deseasonalized data were transformed back to seasonal data. In the process of deseasonalizing the monthly Iowa gaming data, the foremost step for simple move average was to calculate 12-month indexes. First, the 12-month contentious moving average acquired was applied to the monthly Iowa slot coin-in and table drop. Second, the centered moving average was calculated after two moving averages were applied to the new 12-monthly moving average time series. Third, each value for the centered moving average was subtracted from the corresponding value of the monthly slot coin-in and table drop. Twelve-month indices were calculated by averaging all the same months of seasonal-irregular values after subtraction. Finally, the slot coin-in and table drop were deseasonalized by dividing the 12-month indices and used for the SMA forecasting method. In the process of using the SMA method, the number of time series observations, $n$, is critical for estimation. In this study, $n = 6$ for slot coin-in and $n = 2$ for table drop were established by virtue of their best fit to the data through trial and error. The identified best model from SMA for comparisons needed to have seasonality added, that is, be multiplied by the 12-month indices. The results, in terms of MSE and MAPE, are presented in chapter 4.

The naïve method, along with the SMA model, is not on par with the more sophisticated ARIMA and ANN methods in terms of complexity. The underlying assumption of the naïve method is that the value at the next level will be the same as that for the preceding level. As an effort-free method, the naïve approach was chosen by this study to identify whether it was
worthwhile to employ more computational resources and effort to forecast monthly Iowa gaming volume to reach the desired level of accuracy.
CHAPTER 4
RESULTS AND DISCUSSIONS

Results

The forecasting results of SMA for the years 2011 to 2012 had the lowest average MSE, at 2340.02, and MAPE, at 3.03%, for slot coin-in and the lowest average MSE, 14.12, and MAPE, at 6.77%, for table drop in the total 8 time periods, as shown in Tables 3 and 4. SMA outperformed other methods not only for the general period but also for all the different horizons. The SARIMA model showed the second best forecasting performance, which had the second lowest average and the lowest average MSE, 3,557.79 and MAPE, 3.88% for slot coin-in and second lowest average MSE, 26.57, and MAPE, at 9.57%, for table drop in the total 8 time periods. The ANN was the worst method, whether considering its ability to fit the data or the forecasting performance in any horizon. There were no differences for the ability of models at different forecasting horizons.

The SMA performances overshadowed those for the two sophisticated models. The actual data values were further away from the forecasting observations using the SARIMA and the ANN, as shown in Figures 9 and 10. However, in the comparison of the forecasting abilities between ARIMA and the ANN in this monthly gaming series, the SARIMA model produced using the Box-Jenkins procedure far outperformed the ANN. The MAPE values from the ANN for both slot coin-in and table drop were far beyond the general level and even higher than 10%.
<table>
<thead>
<tr>
<th>Time period</th>
<th>Method</th>
<th>MSE(^b)</th>
<th>MAPE (%)</th>
<th>MSE(^b)</th>
<th>MAPE (%)</th>
<th>MSE(^b)</th>
<th>MAPE (%)</th>
<th>MSE(^b)</th>
<th>MAPE (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1 Jan-Mar 2011 (month 01–month 03)</strong></td>
<td>Naive</td>
<td>17,303.24</td>
<td>11.16</td>
<td>156.81</td>
<td>0.87</td>
<td>811.91</td>
<td>2.02</td>
<td>209,025.66</td>
<td>39.59</td>
</tr>
<tr>
<td><strong>2 Apr-Jun 2011 (month 04–month 06)</strong></td>
<td>Simple moving average</td>
<td>17,174.18</td>
<td>10.89</td>
<td>486.94</td>
<td>1.54</td>
<td>736.80</td>
<td>1.81</td>
<td>229,347.83</td>
<td>40.08</td>
</tr>
<tr>
<td><strong>3 Jul-Sep 2011 (month 07–month 09)</strong></td>
<td>SARIMA (2,1,0)(0,1,2)(_{12})</td>
<td>22,014.01</td>
<td>11.73</td>
<td>1,385.29</td>
<td>2.98</td>
<td>1,710.72</td>
<td>3.04</td>
<td>11,995.95</td>
<td>8.08</td>
</tr>
<tr>
<td><strong>5 Jan-Mar 2012 (month 13–month 15)</strong></td>
<td>ANN</td>
<td>44,980.41</td>
<td>16.92</td>
<td>7,265.81</td>
<td>6.48</td>
<td>9,580.01</td>
<td>7.44</td>
<td>160,507.21</td>
<td>32.34</td>
</tr>
<tr>
<td><strong>6 Apr-Jun 2012 (month 16–month 18)</strong></td>
<td>ANN</td>
<td>27,568.62</td>
<td>13.46</td>
<td>1,605.54</td>
<td>2.77</td>
<td>2,669.53</td>
<td>3.81</td>
<td>176,609.84</td>
<td>33.81</td>
</tr>
<tr>
<td></td>
<td>Jul-Sep 2012 (month 19–month 21)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>----------------------------------</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>23,341.71</td>
<td>12.45</td>
<td>344.47</td>
<td>1.45</td>
<td>868.90</td>
<td>2.37</td>
<td>28,234.34</td>
<td>12.26</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>27,425.66</td>
<td>13.91</td>
<td>3,131.97</td>
<td>4.35</td>
<td>4,673.89</td>
<td>4.65</td>
<td>67,372.43</td>
<td>18.93</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Total average</strong></td>
<td><strong>26,345.32</strong></td>
<td><strong>13.13</strong></td>
<td><strong>2340.02</strong></td>
<td><strong>3.03</strong></td>
<td><strong>3,557.79</strong></td>
<td><strong>3.88</strong></td>
<td><strong>113,658.46</strong></td>
<td><strong>24.60</strong></td>
</tr>
</tbody>
</table>

*SARIMA = Seasonal autoregressive integrated moving average; ANN = Artificial neural network.*

*Units of measure are millions.*
Table 4 Summary of Mean Squared Error (MSE) and Mean Absolute Percentage Error (MAPE) Values for Table Drop

<table>
<thead>
<tr>
<th>Time period</th>
<th>Naïve</th>
<th>Simple moving average</th>
<th>SARIMA (2,1,0)(0,1,1)_{12}</th>
<th>ANN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MSE^b</td>
<td>MAPE (%)</td>
<td>MSE^b</td>
<td>MAPE (%)</td>
</tr>
<tr>
<td>1  Jan-Mar 2011 (month 01–month 03)</td>
<td>8.79</td>
<td>5.16</td>
<td>2.31</td>
<td>2.89</td>
</tr>
<tr>
<td>2  Apr-Jun 2011 (month 04–month 06)</td>
<td>29.37</td>
<td>10.76</td>
<td>8.85</td>
<td>5.41</td>
</tr>
<tr>
<td>3  Jul-Sep 2011 (month 07–month 09)</td>
<td>40.58</td>
<td>12.32</td>
<td>15.53</td>
<td>7.10</td>
</tr>
<tr>
<td>4  Oct-Dec 2011 (month 10–month 12)</td>
<td>55.89</td>
<td>14.65</td>
<td>25.32</td>
<td>9.53</td>
</tr>
<tr>
<td>6  Apr-Jun 2012 (month 16–month 18)</td>
<td>32.57</td>
<td>10.88</td>
<td>11.70</td>
<td>5.70</td>
</tr>
</tbody>
</table>
Table 4 (continued)

<table>
<thead>
<tr>
<th></th>
<th>Jul-Sep 2012 (month 19–month 21)</th>
<th>Oct-Dec 2012 (month 22–month 24)</th>
<th>Total average</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>35.52</td>
<td>45.98</td>
<td>37.21</td>
</tr>
<tr>
<td></td>
<td>12.06</td>
<td>13.51</td>
<td>11.62</td>
</tr>
<tr>
<td></td>
<td>11.32</td>
<td>18.77</td>
<td>14.12</td>
</tr>
<tr>
<td></td>
<td>6.80</td>
<td>8.33</td>
<td>6.77</td>
</tr>
<tr>
<td></td>
<td>19.98</td>
<td>35.01</td>
<td>26.57</td>
</tr>
<tr>
<td></td>
<td>9.06</td>
<td>11.45</td>
<td>9.52</td>
</tr>
<tr>
<td></td>
<td>239.10</td>
<td>249.13</td>
<td>256.88</td>
</tr>
</tbody>
</table>

aSARIMA = Seasonal autoregressive integrated moving average; ANN = Artificial neural network.

bUnits of measure are millions.
According to Hamzaçebi (2008), seasonal time series forecasting problems can be thought of as a function approximation problem. An ANN can learn seasonality in the data structure without removing the seasonal effect from the series. Using the $s$ parameter for determining the input and output neuron number may help to make better predictions. For monthly time series $s = 12$, representing the number of input and output neurons with parameter $s$ can increase the prediction performance of ANN in seasonal time series forecasting. However, as shown in Figures 9 and 10, although the results for ANN exactly illustrated the seasonality of forecasts, the ability to capture the pattern is obviously not as good as the ability of the others which had process of predeseasonalization.

In the process of analyzing input neurons of ANNs, one can observe which variables are important and contribute most to the final forecast. The numbers of neurons in the input layer, output layer, and hidden layer depend on the problem. If the number of hidden neurons is small, then the network may not have sufficient degrees of freedom to learn the process correctly (Karunanithi, Achenie, & Gani, 2004). If the number is too high, the training will take a long time and the network may overfit the data. The normalized importance is simply the importance value divided by the largest importance value and expressed as a percentage. Through the importance chart, sorted in descending value of importance, the ANN model, as configured, was shown to fit slot coin-in data better than it did table drop data in figure 9 and 10. Therefore, the ANN showed better accuracy in forecasting slot coin-in than it did table drop in this Iowa gaming forecast.
Figure 9. Comparisons among the four forecasting methods for slot coin-in
Figure 10. Comparisons among the four forecasting methods for table drop
However, the ANN did not produce the expected positive results that many studies mentioned nor was it able to develop its unique features over the other methods. As shown in Figures 9, 10, 18, and 19, the suitable performance of ANN models in training data accounts for the selection of the activation function of hyperbolic tangent being satisfactory. Also, the ANN model did not have lower forecasting errors for some data points from which trends and seasonal patterns emerged. Zimmerman (1994) concluded that ANNs predict well on the forecasting horizon beyond the first few periods ahead. However, this attribute was not found in this study. This phenomenon can be explained by the characteristics of ANNs; that is, ANNs are better fitted with long-term forecasts (Maier & Dandy, 1996). In-sample data were abundant for the validation of the ANN model. However, the ANN did not perform well in the gaming time series. The forecasting horizon of 24 months obviously was not appropriate for forecasting with the ANN. Some advantages for using an ANN for capturing a pattern of nonlinear time series were not demonstrated in the gaming series, which is one reason for the unsatisfactory performance.

<table>
<thead>
<tr>
<th>Table 5 Summary of Parameter Estimates for Slot Coin-in</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicted</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Hidden layer 1</td>
</tr>
<tr>
<td>Predictor</td>
</tr>
<tr>
<td>Input layer</td>
</tr>
<tr>
<td>(Bias)</td>
</tr>
<tr>
<td>Lag1</td>
</tr>
<tr>
<td>Lag2</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>0.344</td>
</tr>
<tr>
<td>0.057</td>
</tr>
<tr>
<td>0.175</td>
</tr>
<tr>
<td>0.343</td>
</tr>
<tr>
<td>-0.258</td>
</tr>
</tbody>
</table>

Hidden layer 1

<table>
<thead>
<tr>
<th>(Bias)</th>
<th>H(1:1)</th>
<th>H(1:2)</th>
<th>H(1:3)</th>
<th>H(1:4)</th>
<th>H(1:5)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.074</td>
<td>0.566</td>
<td>0.253</td>
<td>-0.532</td>
<td>-0.388</td>
</tr>
</tbody>
</table>
Table 6 Summary of Parameter Estimates for Table drop

<table>
<thead>
<tr>
<th>Predictor</th>
<th>H (1:1)</th>
<th>H (1:2)</th>
<th>H (1:3)</th>
<th>forecasts</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input layer</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Bias)</td>
<td>-0.201</td>
<td>-0.364</td>
<td>0.121</td>
<td></td>
</tr>
<tr>
<td>Lag1</td>
<td>-0.204</td>
<td>-0.344</td>
<td>-0.061</td>
<td></td>
</tr>
<tr>
<td>Lag2</td>
<td>-0.290</td>
<td>-0.256</td>
<td>-0.215</td>
<td></td>
</tr>
<tr>
<td>Lag3</td>
<td>-0.218</td>
<td>0.492</td>
<td>0.479</td>
<td></td>
</tr>
<tr>
<td>Lag4</td>
<td>0.495</td>
<td>0.162</td>
<td>-0.184</td>
<td></td>
</tr>
<tr>
<td>Lag5</td>
<td>-0.028</td>
<td>0.312</td>
<td>0.175</td>
<td></td>
</tr>
<tr>
<td>Lag6</td>
<td>0.310</td>
<td>0.390</td>
<td>0.065</td>
<td></td>
</tr>
<tr>
<td>Lag7</td>
<td>-0.427</td>
<td>-0.203</td>
<td>-0.270</td>
<td></td>
</tr>
<tr>
<td>Lag8</td>
<td>0.498</td>
<td>0.132</td>
<td>-0.270</td>
<td></td>
</tr>
<tr>
<td>Lag9</td>
<td>-0.363</td>
<td>-0.025</td>
<td>-0.216</td>
<td></td>
</tr>
<tr>
<td>Lag10</td>
<td>0.215</td>
<td>0.131</td>
<td>0.327</td>
<td></td>
</tr>
<tr>
<td>Lag11</td>
<td>-0.027</td>
<td>-0.064</td>
<td>0.483</td>
<td></td>
</tr>
<tr>
<td>Lag12</td>
<td>-0.289</td>
<td>0.420</td>
<td>0.193</td>
<td></td>
</tr>
<tr>
<td>Lag 13</td>
<td>0.549</td>
<td>0.475</td>
<td>0.283</td>
<td></td>
</tr>
<tr>
<td>Lag 14</td>
<td>0.497</td>
<td>-0.135</td>
<td>0.325</td>
<td></td>
</tr>
</tbody>
</table>

| **Hidden layer 1** |         |         |
| (Bias)             | 0.065   |         |
Each connection from input layer to hidden layer or from hidden layer to output layer, between nodes, has a weight attached to it. This weight corresponds to the degree of influence of one node on the other node (Davalos et al., 1999). The resulting weights of the input variables and hidden variables in the ANN model are depicted in Tables 5 and 6 depict. The training and testing sum of square errors of model A used to model slot coin-in and table drop are shown in Table 7.

### Table 7 Summary of Model Training and Testing for Slot Coin-in and Table drop

<table>
<thead>
<tr>
<th></th>
<th>Sum of square error</th>
<th>Relative error</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Slot coin-in</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Training</td>
<td>13.682</td>
<td>.288</td>
</tr>
<tr>
<td>Testing</td>
<td>11.723</td>
<td>.288</td>
</tr>
<tr>
<td><strong>Table drop</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Training</td>
<td>8.117</td>
<td>.171</td>
</tr>
<tr>
<td>Testing</td>
<td>1,809.075</td>
<td>.171</td>
</tr>
</tbody>
</table>

*Note.* Dependent variable: forecasts. Error computations are based on the testing sample.
As shown in Tables 5 and 6, some connections have positive values, whereas others have negative values. For ANNs, a connection can either strengthen or inhibit the link between two nodes based on its sign. A positive sign strengthens the link, whereas a negative one inhibits it.
Findings and Discussion

Models identified from in-sample data, in terms of each forecasting method, estimated the values in the succeeding 24 months. The succeeding months were narrowed into eight time horizons. Then, for each horizon, the estimated values were compared with the holdout sample data to evaluate the forecasting abilities of each forecasting method. These results varied in terms of different data and multiple horizons. For horizons, some studies validated that the forecasting ability of the various methods depends upon the length of the forecasting horizon (Zheng et al., 2012). This is also one of the findings of the M-competition (Makridakis et al., 1982; Makridakis & Hibon, 2000), whereby Makridakis and his colleagues chose various methods, using 1,001 time series early on and 3,003 time series later from various sources, and repeatedly validated their findings. The present study examined the four different methods using monthly Iowa slot coin-in and table drop with multiple forecasting time horizons and validated SMA outperformed the other methods with both of series for every forecasting horizon. The results once more provided empirical evidence from the gaming field supporting the findings of the M-competition. The SARIMA model showed the next best forecasting performance. The ANN was the worst method, whether considering its ability to fit the data or forecasting performance in any horizon.

As Figures 9 and 10 show, SMA and SARIMA performed better in forecasting slot coin-in than table drop. This implies that the revenues for slot coin-in are more resilient than those for table drop for extrinsic intervention, such as economic background and certain events. In a recession, people always tighten their budgets. However, the money expended on slot
machines as consistent entertainment expenditure is not impacted by many factors. The forecasting performance of all methods for slot coin-in was better than for table drop. Slot coin-in trends are more dynamic than are those for table drop. Therefore, the forecasting methods are inclined to capture the gradual and subtle variations and provide a better prediction. As Figures 9 and 10 demonstrate, the forecasting errors for SMA and SARIMA were stable for most of the time periods.

In addition, ANNs lack a theoretical background and a systematic procedure for model building, in contrast with classic approximations such as the Box–Jenkins methodology (Box & Jenkins, 1976). Thus, Hansen et al. (1999) suggested using time series methods to determine the number of lagged terms that should serve as input variables in a neural network. As a consequence, the model building stage involves an experimental selection of a large number of parameters through trial and error. This limitation is another reason for unsatisfactory performance.

This study focused on the most popular types of ANNs for forecasting: feed-forward multiplayer perceptron. Because a single hidden-layer network has been shown to be both theoretically and empirically capable of modeling any type of functional relationship, three layers of architecture are employed. Because only one-step-ahead forecasting was examined, only one output node was employed. There was a bias term associated with the output node and each hidden node. Generally speaking, too many nodes in the hidden layer (too many connections) produce a network that memorizes the input data and lacks the ability to generalize. In addition, the in-sample fit and the out-of-sample forecasting ability of neural networks are not very sensitive to the number of hidden nodes. According to Masters's (1993)
study, broader horizons in prediction will affect the performance of a neural network. However, Tang and Fishwick (1993) concluded that the performance of a neural network is not decreased by broader horizons. The present study validated Masters's (1993) conclusion.

A number of researchers (Hill, O'Connor, & Remus, 1996; Kolarik & Rudorfer, 1994; Nelson Hill, Remus, & O'Connor, 1999; Zhang & Qi, 2005) have stated that an ANN is not appropriate for modeling the seasonal effect directly. In agreement with the findings of Hill et al. (1996), the results from this study present that deseasonalized data are more effective for neural network models to fit linear discontinuous time series than are nondeseasonalized data. This study concurs with the conclusion mentioned by Nelson et al. (1999) that the forecasting horizon is an impact factor. Therefore, this could be the reason that the ANN did exhibit the expected performance.

Comparing traditional statistics forecasting models, Sharda and Patil (1992) found that ANN is inferior for time series of yearly data as well as for quarterly data. Their study was designed to investigate the forecasting accuracy in different forecasting horizons. The results showed that ANN is adequate in short period horizon in agreement with this study. These findings conflict with the findings of Sharda and Patil (1992), who determined that ANN produced results comparable to the Box-Jenkins model. Sharda and Patil (1990) utilized 75 time series data to determine that ANN performed as well as the Box-Jenkins procedure. Results from the present study totally conflicted with their findings. ANN did not outperform the Box-Jenkins procedure with 180 time series, including holdout data. Sample size may have contributed to the contradictory results. Both detrending and dedeseasonalization are effective procedures to reduce model fitting and forecasting errors (Zhang & Qi, 2005).
Comparison of the Forecasting Ability Between ARIMA and ANN

The ARIMA method is more suitable for short-term forecasts in comparison with ANN. It behaved much better than did ANN in this study, with a MSE and MAPE of 3,557.79 and 3.88%, respectively, for slot coin-in and with a MSE and MAPE of 326.57 and 9.52%, respectively, for table drop. Forecasting errors with the ARIMA method were close to those with the SMA method in slot coin-in forecasting. However, the techniques for identifying the correct model for ARIMA/SARIMA from the class of possible models are difficult to understand and usually computationally expensive. This process is also subjective, and the reliability of the chosen model can depend on the skill and experience of the forecaster. The underlying theoretical model and its structural relationships are not as distinct as are some simple forecasts models.

In this study, six and four ANN nodes were developed for slot coin-in and table drop, respectively. Palmer, Montaño, and Sesé (2006) suggested using the least number of nodes in selecting the structure of the neural network for the best performance of holdout data. The present study also showed that good in-sample fit has no direct relationship to the out-of-sample performance. This finding is in line with observations made by several other researchers. For example, Makridakis (1986) and Makridakis and Winkler (1989) found that the correlation between in-sample and out-of-sample forecasting is only about 0.2 based on a vast amount of empirical evidence. The low correlation between in-sample and out-of-sample performance measures is due to the model uncertainty that commonly occurs in statistical data analysis and particularly in time series analysis and forecasting (Chatfield, 2004). Chatfield
(2004) noted that model uncertainty comes from three main sources: model structure, parameter estimation, and data. The nonlinear nonparametric nature of ANNs may cause more uncertainties in neural network model building. That is, an ANN model can provide very good forecasts with in-sample data but poor forecasts out-of-sample. This learning and generalization dilemma has been studied extensively and is still an active research topic in the field.

Through time series forecasting, the findings show that the commonly used in-sample model selection criteria are unable to identify the best neural network model for out-of-sample prediction. Results clearly indicate the inconsistency between the best in-sample model selected by the popular model selection criteria and the best model out-of-sample. Furthermore, there is agreement between the best in-sample model and the best out-of-sample model, based on the same performance measures, such as MSE and MAPE. Therefore, neither model selection criteria nor performance measures based on in-sample data alone can serve as a reliable guide for choosing the model that has the best out-of-sample performance. This finding suggests that the popular in-sample selection criteria are not quite useful in neural network time series forecasting.

One of the most interesting findings from this study is the excellent performance of SMA on the gaming data. By extending the range of methods included in the work of Fildes and Makridakis (1995), SMA outperformed all other methods for almost all forecasting horizons and accuracy measures. This consistent superiority is directly related to the method designed to match the homogeneous structures for the time series used in this dataset (Grambsch &
Stahel, 1990). Even with more heterogeneous structures of time series, such as those for the M-competition, SMA performed as well as ARIMA models did.
CHAPTER 5

CONCLUSIONS

The performance of ARIMA models over longer horizons shows when identifying and extrapolating the trends in the data, the Box–Jenkins method is more appropriate than an ANN in differencing the data to be stationary in the mean. The forecasting results for ANN were unsatisfactory as some researcher concluded. Palmer et al. (2006) suggested using the least number of nodes in selecting the structure of neural network for the best performance of holdout data. The effects of selecting the proper activation function and the number of hidden layer neurons were identified. A suitable activation function and number of neurons produced good performance, but selection of the activation function and an inappropriate number of neurons significantly decreased forecasting accuracy. In this study, six and four ANN nodes for slot coin-in and table drop, respectively, were developed. It is notable that, in most cases, forecasters may not select the best neural network mode and the performance of neural networks may not be generalized. The data processing procedures used impact the performance of neural networks in learning and generalizability. This study concluded that the number of input nodes is almost half the number of lagged observations and provided more evidence about implications of the AC criterion approach applied to neural network input configuration in time series forecasting. However, ANN modeling was proven the least accurate for short-term gaming data. Therefore, ARIMA remains the dominant method in the gaming research field, even though SMA surpassed ARIMA by a narrow margin in the Iowa
gaming monthly time series, which was not able to accommodate data as good as ARIMA with exogenous interruption for long term.

Comparing commonly used forecasting methods is always conducted during forecasting competitions. Makridakis et al. (1982) completed early, famous competitions, and their results with 1,001 time series, including holdout data, are compatible with the results that emerged from this study. The results from this study support the observation that the best forecasting model is the simplest one. Therefore, the conclusion for M-competition still holds true with seasonal gaming data. Also, this study strengthened the findings of the M-competition for gaming data. First, in general, the complicated or statistically sophisticated methods requiring lots of forecasting experience and trial-and-error did not outperform simple methods. Second, the performance for different methods is determined by the specific accuracy measure used to evaluate the results. Finally, the performance of the various methods depended on the length of the forecasting horizon involved. The measures used in the M-competition and M3-competition can provide infinite or undefined values in commonly occurring situations. The scaled errors become the standard measure for forecast accuracy, whereby the forecast error is scaled using the naïve forecasting method. This is widely applicable and is always defined and finite, except in the irrelevant case in which all historical data are equal. Of course, there will be situations where some of the existing measures may still be preferred. When all series are on the same scale, MSE is preferred because it is simpler to explain. If all data are positive and much greater than zero, MAPE is preferred for reasons of simplicity. However, in situations where there are very different scales, including data with values close to zero or negative, MSE is the best available measure to forecast accuracy.
On a practical level, relatively accurate forecasts can be achieved with simple methods. This study indicates that SMA is the best model in every forecasting horizon. After implementation of the time series in the model, the findings showed that the adaptation of the moving average component improved the short-term results in holdout data. Also, the assumption referring to stochastic factors is crucial in modeling, and the simple short-term factor was insufficient for the model. This study concludes that the decrease of forecasting ability of the SMA is not caused only by the increase of volatility and uncommon pattern, but also by the own feature of the data. This leads that the forecasting ability of SMA is dependent with several conditions, such as market fluctuations.

All four methods performed better in forecasting slot coin-in than table drop. This implies that the forecasting ability for these methods are pattern and seasonality. The revenues for slot coin-in are more resilient than those for table drop for extrinsic intervention, such as economic background and certain events. There is no difference in the forecasting abilities for every forecasting horizon. This suggests that effective deseasonalization methods are the key point in the process of data transformation.

This study provided many experimental guides for model building, which are useful for practitioners to gain insights on model selection with different variables and in different horizons. This comparison showed that the SMA and SARIMA methods have lower prediction errors than does the ANN method. SARIMA is especially convenient when the seasonality in time series is strong; however, if the seasonality is weak, network structures may be more suitable.
Implications and Future Study

This study not only has theoretical implications but also has practical implications for gaming forecasting. First, this study fills a gap in the literature regarding methods for forecasting gaming time series in short and long horizons as well as provides evidence about the forecasting ability of ARIMA, ANN, and SMA methods for Iowa monthly slot coin-in and table drop time series in 24 time periods. Second, the results of this study provide practitioners with some insights on method selection for gaming time series data, especially with different time horizons.

The MSE and MAPE of the SMA and ARIMA from November 2011 through July 2012 are comparatively larger than those for the other sections. After the recession of 2008 and 2009 and the flattened economy of 2010, gaming revenue has gradually started to grow in those states with high tourist comfort levels. Future studies can investigate the relationship between gaming demand and tourists’ comfort levels.

AC was applied to the determination of input variables for Iowa gaming demand on slot coin-in and table drop, and was used to analyze the forecasting performance under AC lag periods. According to Huang’s (2004) study, this approach has been demonstrated with a neural network trained with a back-propagation algorithm. Future research should attempt to demonstrate the effectiveness the approach holds for all learning neural network training algorithms (e.g., radial basis function, probabilistic, etc.) and is a general principle for time series modeling.

The limitation of this study is that the comparisons of the forecasting models focused on the holdout data. Therefore, a neural network fit with long-term data may not achieve its
forecasting capability. Future research may broaden holdout data. In addition, the unpreprocessed gaming data probably impacted the accuracy of the ANN model. Therefore, in future studies with gaming data, preprocessing the data may be preferred before fitting ANN.
REFERENCES


APPENDIX A

FIGURES AND TABLES RELATED TO THE ANALYSIS OF TABLE DROP

Figure A1. The stationarity of raw data plot of table drop
Figure A2. Identification plot for table drop after first order differencing
Figure A3. Comparison of the tentative identification plots with the different gamma values. For table drop
Figure A4. Comparison of the tentative identification plots after the first order differencing for table drop
Figure A5. Estimation of SARIMA(2,1,0)(0,1,2)_12 on \( w = ( \text{millions of dollars} )^{0.33} \) for Iowa monthly table drop
Figure A6. Estimation of SARIMA(2,1,0)(0,1,0)_12 on w = (millions of dollars )^0.33 for Iowa monthly table drop.
Figure A7. Feed-forward architecture of ANN for slot coin-in
Figure A8. Feed-forward architecture of ANN for table drop
Table A1 The Results of Parameter Estimation of SARIMA(2,1,0)(0,1,2)_12 on w= ( millions of dollars ) ^ 0.33 for Iowa Monthly Coin-in

| Parameters statistics for SARIMA(2,1,0)(0,1,2)_12 |
|-----------------|-----------------|-----------------|-----------------|-----------------|
|                | MLE             | SE              | t.ratio         | 95% lower       | 95% upper       |
| AR1             | -0.6128         | 0.0811          | -7.5552         | -0.7718         | -0.4538         |
| AR2             | -0.2507         | 0.0814          | -3.0798         | -0.4102         | -0.0912         |
| SMA1            | -0.7906         | 0.1236          | -6.3969         | -1.0329         | -0.5484         |
| SMA2            | -0.2094         | 0.0863          | -2.4235         | -0.3786         | -0.0400         |

Table A2 The Results of Parameter Estimation of the SARIMA(2,1,0)(0,1,1)_12 on w= ( millions of dollars ) ^ 0.33 for Iowa Monthly Table Drop

| Parameters statistics for SARIMA(2,1,0)(0,1,1)_12 |
|-----------------|-----------------|-----------------|-----------------|-----------------|
|                | MLE             | SE              | t.ratio         | 95% lower       | 95% upper       |
| AR1             | -0.4054         | 0.0823          | -4.9231         | -0.5668         | -0.2439         |
| AR2             | -0.2071         | 0.0821          | -2.5219         | -0.3680         | -0.0461         |
| SMA1            | -0.8991         | 0.1531          | -5.8736         | -1.1992         | -0.5991         |
APPENDIX B
MATHEMATICAL EQUATIONS FOR ASSISTING CONCEPTS

Box-Cox transformation

The transformation equations by Box and Cox (1964) were expressed with lambda ($\lambda$):

$$y_t^\lambda = \left( y_t^\lambda - 1 \right) / \lambda, \text{ when } \lambda \neq 0;$$

$$y_t^\lambda = \log(y_t), \text{ when } \lambda = 0.$$

The values of lambda for different transformations:

$\lambda = 1.00$: no transformation needed;

$\lambda = 0.50$: square root transformation;

$\lambda = 0.33$: cube root transformation;

$\lambda = 0.25$: fourth root transformation;

$\lambda = 0.00$: natural log transformation;

$\lambda = -0.50$: reciprocal square root transformation;

$\lambda = -1.00$: reciprocal (inverse) transformation.

Maximum Likelihood Estimation

The expression equation of MLE:

$$f(y_1, y_2, y_3, y_n | w) = \prod f_{n}(y_m | w)$$
\( f(y_n \mid w) \) denotes the probability density function that specifies the probability of observing data vector \( y \) given the parameter \( w \). The parameter is a vector \( w = (w_1, w_2, \ldots, w_k) \) defined on a multi-dimensional parameter space.