EVALUATION OF FORECASTS PRODUCED BY GENETICALLY EVOLVED MODELS

M.A. Kaboudan*
MS&IS, Penn State Lehigh Valley, Fogelsville, PA 18051, USA
Email: mak7@psu.edu

Abstract

Genetic programming (GP) is a random search computer algorithm that parallels Darwin's theory of evolution and survival of the fittest. It finds application in pattern recognition and optimization problems in the natural sciences, engineering, business, and social sciences. This paper introduces GP and uses a GP computer program to evolve time-series models especially relevant for applied statisticians. Prediction models are evolved for simulated noise-free and noisy data as well as for real world Canadian lynx and sunspot numbers. Forecasts produced by the fittest of the genetically evolved models are evaluated and compared with available forecasts in prior studies.

KEY WORDS: Time-series prediction; Computational methods; Nonlinear regression; Canadian lynx data; Sunspot numbers.

Genetic programming (or GP) is a random search technique that emerged in the late 1980s and early 1990s. Koza (1992) was the first to introduce a formal description of the method. GP applies to many optimization areas. One of them is modeling time series and

* M.A. Kaboudan, Management Science & Information Systems, Penn State Lehigh Valley, 8400 Mohr Lane Fogelsville, PA 18051, USA. Tel: (610) 285-5115. Fax: (610) 285-5220. E-mail: mak7@psu.edu.

using those models in forecasting. Unlike other modeling techniques, GP is a computer program that 'searches' for a specification that replicates the dynamic behavior of observed series. To use GP, one provides mathematical operators and values of variables as input files. Upon its execution, the program randomly assembles equations with different specifications then identifies and reports that specification with the least sum of squared errors (or SSE) as output file. This process is an iterative evolution of successive generations consisting of thousands of assembled equations where the fittest within a generation survive and breed better equations also using random combinations until the best one is found. Clearly from this simple description, the method is based on heuristics and has no theoretical foundation. However, resulting final equations seem to produce reasonably accurate forecasts that compare favorably to forecasts from humanly conceived specifications. With encouraging results difficult to overlook or ignore, it is important to investigate GP as a forecasting methodology. This paper attempts to evaluate forecasts genetically evolved models produce for experimental data as well as real world time series. It is organized as follows: Section 1 contains an overview of genetically evolved models (GEMs). The reader will find an explanation of how models are evolved using genetic methodology as well as features that characterize GEMs as a modeling technique. Section 2 contains descriptions of simulated and real world data and their respective fittest identified GEMs. Simulated data were chosen to represent processes with different behavioral complexities, including linear, linear-stochastic, nonlinear, and nonlinear-stochastic. Real world data consist of two time series popular in analytical statistics: Canadian lynx data and sunspot numbers. This Section also contains evaluation of predictions of values used to evolve models. Section 3 presents single- and

multi-step-ahead forecasts produced using the fittest evolved models representing the different series, evaluation of the forecasts as well as comparisons with prior ones in studies of the Canadian lynx data and sunspot numbers. A brief discussion is in Section 4.

1. GENETICALLY EVOLVED MODELS

1.1 Basic Foundation

Only a brief background of GP and how it is used to evolve time series models is provided here. More complete theory can be found in Koza (1992) and Banzhaf et al. (1998). Its applications include fitting time series models in Sathyanarayan et al. (1999), predicting financial markets see Chen et al. (1998) and Iba and Sasaki (1999), and on electric power demand see Lee et al. (1997). A GP computer program or code is designed to optimize an objective function or specific task in a way that parallels Darwin's theory of natural selection and survival of the fittest. To find that fittest equation, a program randomly assembling a user defined initial number of equations first to use in breeding fitter equations. While breeding new ones, existing equations with lower sum of squared error (or SSE) have a better chance of surviving. Because the search process is totally random, the method involves no evaluation of the causal relationships between explanatory variables and the dependent variable, appropriateness of signs of coefficients, or equation stability conditions. The final fittest equation the program produces is often exceptionally long, difficult to interpret, and almost meaningless as a result. But it predicts well. Perhaps the gains are in the method's obvious ability to replicate a variable's dynamical behavior in the form of specifications beyond our human ability to construct. Given that time series models are designed to fit the serial correlation

properties of data and not to "explain" it, GP satisfies the goal time series should accomplish: finding a model which fits the data reasonably well.

The code that produces GEMs in this study is written in C++ and replicates an original one written by Koza (1992) in LISP. The modified computer code produces files that contain standard estimation statistics and forecast computations. It gets its instructions from an executable configuration file containing parameters the user furnishes for the program to follow while performing a desired specified optimization task. Upon execution, the program randomly combines variables and/or constants (also known as terminals) with functions (also known as operators) to evolve an initial population of "individuals" or equations. In this study, the operators used are: +, -, *, %, exp, sqrt, ln, sin, and cos, where %, exp, sqrt, and ln are protected operators. These protections follow standards commonly used in all GP software. They provide solutions to computational-problems that may halt execution during a search. The following protections prevent computational problems:

1. If in $(x \div y)$, $y = 0$, then $(x/y) = 1$.
2. If in $y^{1/2}$, $y < 0$, then $y^{1/2} = |y|^{1/2}$.
3. If in $\ln(y)$, $y < 0$, then $\ln(y) = 1$, where ln is the natural logarithm.
4. If in $\exp(y)$, $y > 10$, then $\exp(y) = \exp(10)$.

Before executing the program, its user determines and furnishes within the configuration file the list of operators, names of terminals, population size, and number of generations. Values of the dependent and independent variables are included in separate input files. Population size is the number of individual equations the program should

randomly produce and evaluate simultaneously. Typically the user would select a population size = 1000, 2000, or even 5000 individuals. (There is no agreement on an optimal population size or number of generations among GP researchers.)

The initial population is then used to breed a new generation or population containing the same number of equations. The new population breeds a succeeding one, and so on. To breed, the program ranks individuals in a population according to some fitness function. SSE is the one most commonly used. Naturally, the equation with the lowest SSE is ranked fittest. Fitter individuals are given a better chance to survive and breed. A new population contains fittest equations from the existing population and offspring created by a combination of self-reproduction, crossover, and mutation. The user decides on and provides rate or proportion of the next population to breed by self-reproduction, crossover, and mutation in the configuration file. A new population may contain say 10% produced by self-reproduction, 80% bred by crossover, and mutation produces the remaining 10%. The user also chooses one of several methods to use when selecting individuals to breed members of the next generation while favoring survival of the fittest individuals as well as their genes into the next generation. Tournament selection is the method most used to select members for breeding. In its simplest version, the user selects a small number (< 10) of individuals to participate in breeding. Assume the number chosen = 6. With larger weights assigned to the fittest individual, the program randomly and repeatedly draws six individuals at a time. In self-reproduction the fittest individual in each six is selected and passed on to the new generation. The six (or sometimes weakest three or four among the six) are replaced and can be reselected. This process is repeated until the proportion in the next population bred by self-reproduction is

satisfied. Once done, the program starts to breed using crossover. Crossover is sexual reproduction where the offspring inherit genes from their parents. Again six are randomly selected and the fittest two breed two offspring. The program randomly swaps part of one equations with a part of another to breed two offspring. The offspring are passed on to the next generation, the six are replaced, and the process is repeated. This also continues until the proportion bred by crossover is satisfied. Breeding by mutation then starts where only the fittest of each set of six is mutated to breed one individual for the next generation until that proportion is satisfied. Mutation is asexual where a part of an equation is randomly replaced or discarded. Finally, the program terminates and writes the best equation and associated results and statistics to an output file. A GP program terminates either by producing an equation with a minimum threshold error or by reaching a maximum number of generations specified in the configuration file.

1.2 GEMs in Perspective

Given that such models are produced by artificial means, a computer program, they possess their own stylized characteristics. The following characteristics are formulated after extensive experimentation with GP in modeling time series:

a. *A user has little control in dictating the final model specification*. The final model specification is the result of random selection. One may influence evolved equations only by adding or deleting operators or terminals. But the fittest final equation specification is always unknown before executing a GP program.
b. *GEMs are usually difficult to explain or justify even if they produce fairly good fits.* The difficulty is because equations are randomly assembled, usually exceptionally

long, and efforts to reduce the fittest one are hampered if it includes protected operators.

c. *Reproduction of the fittest models may be possible for data generated by simple processes but impossible for data generated by highly complex ones*. The probability of replicating the fittest equation is very small mainly because GEMs are random combinations of variables and operators only to replicate dynamics. Given nine operators and assuming only twelve explanatory variables, the number of possible combinations of individual expressions within a single equation is huge (= 9! * 12!). The odds of reproducing the same fittest individual are marginally better if one is evolving noise-free processes.

d. *GEMs produce residuals that may violate standard normality assumptions.* The assumption that a statistical or econometric model (linear or nonlinear) is judged adequate only if its residuals approximate Gaussian white noise may not hold when using GP to evolve models that replicate the dynamics of noise-free processes. If the process contains no noise, residuals must be neglected linearity or nonlinearity.

e. *To evolve a reliable model, it is necessary to produce a large number of equations.* During its search, GP produces equations with different quality. This occurs because it occasionally gets 'trapped' in local minima within the solution space while searching for the global minimum SSE. In a large number of runs (say 50 or 100), a GP program may produce a high percentage (50% or more) of final (fittest) equations with less than optimal SSE especially when modeling highly complex processes. As a result, the probability that GP finds a 'good' model increases when set to evolve a large number of equations.

2. TIME SERIES PREDICTION USING GEMS

This section presents descriptions of simulated and real-world series and reports their respective fittest GEMs. Selection of real-world series was simple. The Canadian lynx data and sunspot numbers make ideal candidates because of their popularity. Selection of models to simulate was more intricate given the availability of infinite processes to select from. To keep the number of series to evolve models for manageable, systems were grouped according to their complexity levels first. Linear and random processes were characterized as least and most complex systems, respectively. Nonlinear processes are relatively more complex than linear but lest than random systems. Linear-stochastic and nonlinear-stochastic possess complexities that vary according to their signal-to-noise ratios. (Signal-to-noise ratio = Signal variance / noise variance.) Generally however, they should be more complex than their noise-free component. Series are then selected to provide information that helps formulate generalization statements about GP's forecasting abilities of the different groups. Accordingly, seven representative processes were selected. They include one linear, two linear-stochastic with different signal-to-noise ratios, two nonlinear, and two nonlinear-stochastic also with different signal-to-noise ratios. One would expect GP to forecast linear more accurately than it would linear-stochastic with low noise and the latter more accurately than series with high noise. The linear process introduced is rather unique. It produces a series that remains stable long enough to have its variance measured. This helps simulate linear-stochastic series with controlled, measurable and different signal-to-noise ratios. Nonlinear and nonlinear-stochastic processes are four rather than three because popular nonlinear chaotic

functions (such as the Henon (1976) and logistic (May, 1976) maps) produce unstable solutions when dynamically dithered with noise. It is possible to dither those series with static noise, however. In dynamic dithering, $Y_t = f(Y_{t-k}, \varepsilon_t)$ where ε_t are independent identically distributed normal deviates with variance σ^2_ε. In static dithering noise affects only current period outcome, or $Y_t = X_t + \varepsilon_t$ where $X_t = f(X_{t-k})$. Processes dynamically affected by noise are more commonly investigated and are the focus here. (Analysis of static dithering of chaotic processes are in Kaboudan, 1999.) To simulate nonlinear-stochastic processes with measurable signal-to-noise ratios, a new nonlinear process that remains stable after dynamically adding noise is introduced below and investigated in addition to the Henon map - a known chaotic process.

To evolve models, each process was assumed as $Y_t = f(Y_{t-1},\ldots, Y_{t-12})$ and the best of fifty equations from fifty different program executions per series is reported as the fittest model specification below. Executing the program 50 times per series increases the chance of obtaining that global minimum SSE as explained earlier. For each execution, the number of generations = 100 and population size = 1,000 which translate into 100,000 equations assembled to produce the fittest equation per execution and half a million assembled equations to produce one, the fittest of the fittest fifty. Simulated samples are 132 observations per series. The first twelve are lost degrees of freedom for lags. Observations t = 13, …, 112 are used to evolve models. Twenty observations t_f = 113, …, 132 are used to evaluate forecasts. For the Canadian lynx where only 114 observations are available, t = 13, …, 100 are used to evolve models and forecasts are over t_f = 101, …, 114 which is consistent with what prior studies of this series used. To

be consistent with statistical literature as well, 221 sunspot numbers starting in the year 1700 are used to evolve that series' fittest model which is then used to forecast the next 35 observations.

Evolved models are evaluated according to sample mean square error (s_e^2), the normalized mean square error (NMSE), an α-statistic, mean absolute percent error (MAPE), and (1-β)-statistic. NMSE = $s_e^2 / \hat{\sigma}_Y^2 = T^{-1}\sum(Y_t - \hat{Y}_t)^2 / (T-1)^{-1}\sum(Y_t - \bar{Y})^2$, where $\hat{\sigma}_Y^2$ is the variance of the dependent variable Y. The α-statistic = $s_e^2 / s_{e\_RW}^2$ = $\sum(Y_t - \hat{Y}_t)^2 / \sum(Y_t - Y_{t-1})^2$, where the summation is over the same training period t = 13,…,T, and where $s_{e\_RW}^2$ = MSE of the naïve random walk model. In a naïve random walk model predicted $Y_t = \hat{Y}_t = Y_{t-1}$. By design $\alpha \geq 0$. An $\alpha = 0$, 1, or >1 suggests perfect prediction by the evolved model, the model is no better than a naïve random walk model, or it is worse than the naïve random walk model, respectively. Benchmarking model prediction (or forecast as presented in Section 3) to random walk prediction (or forecast) provides a more meaningful and stringent measure of model performance and forecast accuracy. MSE is sensitive to the variable's unit of measurement and cannot be compared for two variables. NMSE uses $\bar{Y}$ as the benchmark. It measures the model's performance relative to the historic series mean as predicted value for all periods. The α-statistic is more stringent because it measures performance of prediction exceeding that of a unit root first order autocorrelation model. Alternatively, if asymptotically $\bar{Y} = 0$, then NMSE < α. MAPE is a reasonable measure but can be easily affected by outliers. To

demonstrate, assume that a model predicts a value = 2.01 when the actual value for this particular period is 0.01 but predicts all other actual values accurately. For a prediction sample = 100, MAPE > 200% and eliminating this outlier reduces MAPE (by 200%). If a different model specification of the same data succeeds in predicting that outlier, then eliminating the outlier cannot be an acceptable solution. Computing (1-β)-statistic seems to dilute the effect of this problem. It is a measure of the models' predictive power over random walk prediction. It is a simple calculation where $(1- \beta) = 1 - (MAPE_M / MAPE_{RW})$. The numerator is MAPE the current model produces while the denominator is MAPE the random walk model produces. Given that random walk model prediction may outperform prediction produced by a different model, $(1- \beta) \leq 1$. If $(1- \beta) = 0$, the evaluated model has no power over random walk prediction. If $(1- \beta) = 1$, the evaluated model produces perfect prediction. If $(1- \beta) < 0$, the evaluated model is worthless. The measure is useful in evaluating forecast performance as well.

2.1 Artificially Simulated Series

For each system, the model used to simulate actual data, the equation evolved, and plots of predicted versus actual values are presented. Table 1 has a summary of the statistics of all evolved equations.

Table 1. Values of MSE, NMSE, α, MAPE, and $1-\beta$ (Historic Simulated Data)

	LDM	**LDM_LN**	**LDM_HN**	**Henon**	**JSM**	**JSM_LN**	**JSM_HN**
s_e^2	0.000	0.0423	0.169	0.001	0.007	0.050	0.478
NMSE	0.000	0.012	0.133	0.001	0.002	0.015	0.118
α	0.000	0.009	0.105	0.000	0.001	0.006	0.052
MAPE	0.012	0.432	4.749	0.397	0.016	0.041	0.132
1-β	0.996	0.912	0.702	0.973	0.973	0.935	0.802

Here are the models and evolved equations:

a. *Linear Diffusion Map (or LDM):*This is a system described by the following linear relationship:

$$Y_t = 1.359141 + 0.67957 \;\; Y_{t-1} - 1.019356 \;\; Y_{t-2}.$$

It produces sufficiently lengthy oscillatory growth series using starting values of 0.3638 and 0.5630, respectively. Figure 1a depicts successive pairs of the series' values (a phase diagram) to show its non-random behavior and why this is a *diffusion* map. The best genetically evolved model (GEM) found is:

$$Y_t = Y_{t-5} - 0.04674 \;\; Y_{t-9} - 0.02064 \;\; Y_{t-4} - 0.01984 \;\; Y_{t-3} - 0.008264(Y_{t-6} - 94^{-1} Y_{t-11})(Y_{t-2} + Y_{t-6} + Y_{t-9} + Y_{t-11}).$$

The statistics in Table 1 show that GP performed well. An easy to detect and specify linear regression model with $Y_t = f(Y_{t-1}, Y_{t-2})$ produces MAPE = 0.0 and $(1 - \beta) = 1$ with exact specification, however. Figure 1b shows complete overlap between the actual and predicted values. However, since this model is perfectly approximated using standard statistical techniques, and because GP evolved a nonlinear system to represent a linear one, GP cannot be considered superior if the process to model is linear.

b. *Linear Diffusion Map with Low-Noise (or LDM_LN):* This linear-stochastic system is a second-order autoregressive process of LDM with noise added such that the resulting signal-to-noise ratio is 40. LDM has a variance $\sigma^2_Y = 1.071719$. Therefore, adding low noise (ε_{tL}) generated such that $\varepsilon_{tL} \sim N(0, \sigma^2_{\varepsilon L} = \sigma^2_Y / 40)$ to LDM produces a linear-stochastic process with the desired signal-to-noise ratio. Alternatively,

$$Y_t = 1.359141 + 0.67957 \;\; Y_{t-1} - 1.019356 \;\; Y_{t-2} + \varepsilon_{tL}.$$

The resulting GEM is:

$$Y_t = Y_{t-5} - 0.30824 \;\; Y_{t-6} + 0.32047 \;\; Y_{t-1} - 0.070635 \;\; Y_{t-3} + 0.05522 \;\; Y_{t-11.}$$

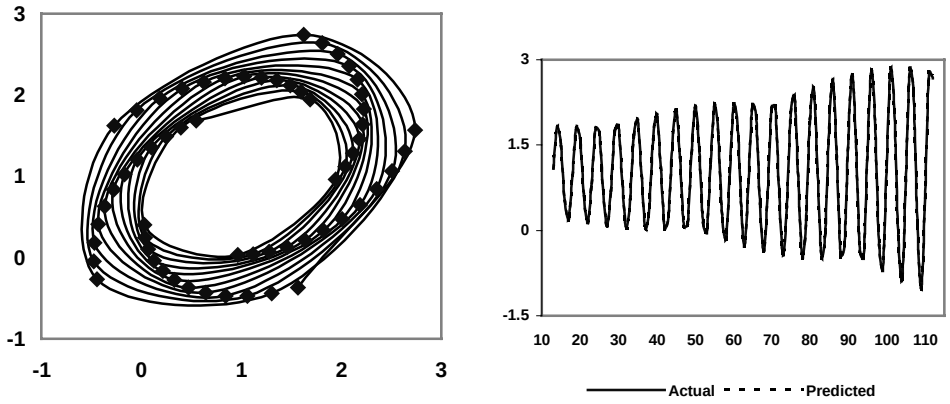

Figure 1: (a) LDM's Phase Diagram

(b) Its Actual & Predicted

MAPE is 91% lower than that of a random walk model (or 1– β = 0.91) as indicated in Table 1 along with other statistics. Figure 1c portrays the series' actual and predicted plots. Fitting the data to a multiple regression model produced the correct specification with lower s_e^2 = 0.02, NMSE = 0.006, α = 0.004 and (1– β) = 0.978, however. This makes GP prediction less efficient than standard statistical methods when fitting low-noise linear-stochastic processes.

c. *Linear Diffusion Map with High-Noise (or LDM_HN):* This is also a LDM but with louder noise added. The signal-to-noise ratio is 10 or $\varepsilon_{\tau H} \sim N(0, 0.3724^2)$. The fittest GEM is:

$$Y_t = 0.5726 + 0.75\ Y_{t-5} - 0.08333\ Y_{t-2} - 0.25\ Y_{t-3} - 0.0137\ Y_{t-5}^{-1} - 0.0092\ Y_{t-9} - 0.0137\ Y_{t-11}^{-1}.$$

As shown in Table 1, predicting this equation was more difficult. MAPE explodes to 4.749 (an example of the outlier effect explained earlier) and (1– β) = 0.702. Fitting this data to a multiple regression model also begets correct specification with MAPE = 0.239, and (1– β) = 0.985 that are superior to GP's statistics. Figure 1d shows actual versus GP-predicted values. As one may expect, increased noise affected GP's predictive ability negatively. More importantly, while the graph documents GP's success in capturing most turning points, its prediction remains less efficient than prediction of the regression model.

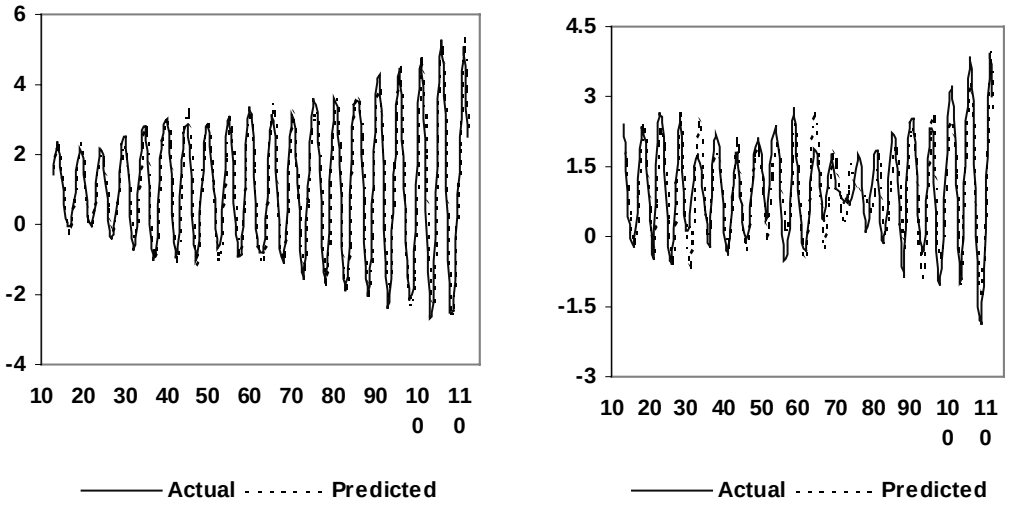

Figure 1: (c) Actual & Predicted LDM_LN (d) Actual & Predicted LDM_HN

d. *The Henon Map:* This is a nonlinear chaotic system introduced by Henon (1976). It is described by the following nonlinear relationship:

$$Y_t = 1 - 1.4\ Y_{t-1}^2 + 0.3\ Y_{t-2}.$$

Figure 2a portrays its dynamics in a phase map. The fittest GEM found is:

$$Y_t = \cos(Y_{t-1}) \div \{\cos(\ln(\cos(\cos(\ln(\cos(Y_{t-3})))))\} - X1_t,$$

where

$$X1_t = (105^{-1}\ Y_{t-1} + [118^{-1}\ \{\cos(\cos(\ln(\ln(Y_{t-3}))))\}] + \{116^{-1}\ (\cos\ (Y_{t-8}))\} + 64^{-1}\ Y_{t-5} + X2_t,$$

$$X2_t = (Y_{t-1})^2 + 24^{-1}\ Y_{t-5} + 9^{-1}\ [\{(\cos(\cos(Y_{t-5}))) \div (\cos(Y_{t-3}))\} - \{(Y_{t-1})^2 + (\cos(\ln(\cos(Y_{t-3}))))\} + \cos(Y_{t-5})] + X3_t,$$

and

$$X3_t = 35^{-1}\ [\{\cos(Y_{t-1}) / \cos(Y_{t-3})\} - \{\ln(\ln(\cos(\cos(\ln(\cos(Y_{t-3})))))) + \ln(\ln(Y_{t-3})\}]$$

GP evolved an acceptable model with 97.3% lower MAPE than residuals of a random walk model and according to the other statistics in Table 1. Figure 2b shows actual versus predicted values of the process. Although GP predicts this nonlinear chaotic map rather well, Stern (1996) reports RMSE = 0.0047 (GP's RMSE = 0.019) using an artificial neural network (or ANN) algorithm to predict the same data.

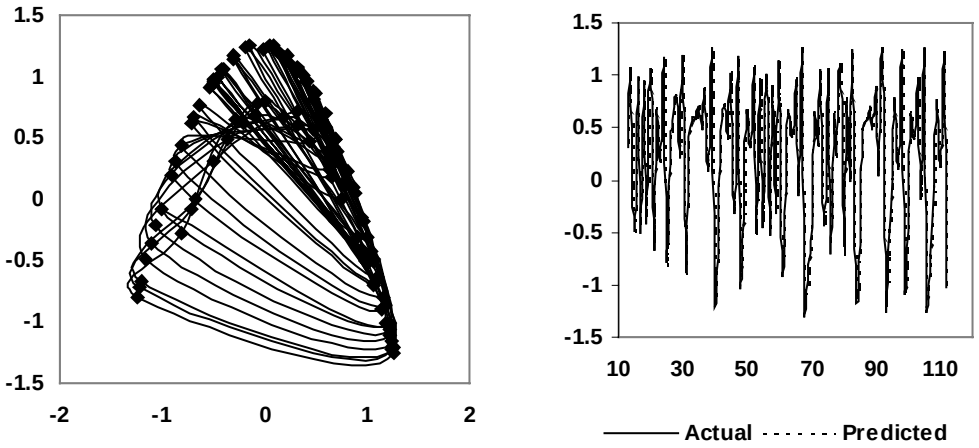

Figure 2. (a) Henon Map Phase Diagram. (b) Its Actual and Predicted Values.

e. *The Jet Ski Map (or JSM):* This is a nonlinear function (introduced here for the first time) that produces the phase diagram in Figure 3a with a shape similar to that of a jet ski. It is described by this nonlinear relationship:

$$Y_t = 0.5 - 2.5 \sin Y_{t-1} + \cos Y_{t-2} - 0.8\ (\ln Y_{t-1} / \exp Y_{t-2}).$$

Unlike the Henon map, it is possible to dither this one dynamically as mentioned earlier. The fittest GEM found is:

$$Y_t = \ln (3*Y_{t-2} + 124 + Y_{t-9}) - \sin(\sin(\sin(\sin(Y_{t-1})))) - \sin(Y_{t-1}) - \sin(\sin(\sin(Y_{t-1}))) + \cos(Y_{t-2})$$

GP's ability to fit this model is rather impressive according to the statistics in Table 1. MAPE is 97.3% lower than that of a random walk model. Figure 3b shows actual versus predicted values. No attempt was made to fit this map using a different method since a linear fit is inappropriate and one would only be fishing for a nonlinear fit otherwise. It would be interesting to evaluate prediction of this series using an artificial neural network algorithm (ANN).

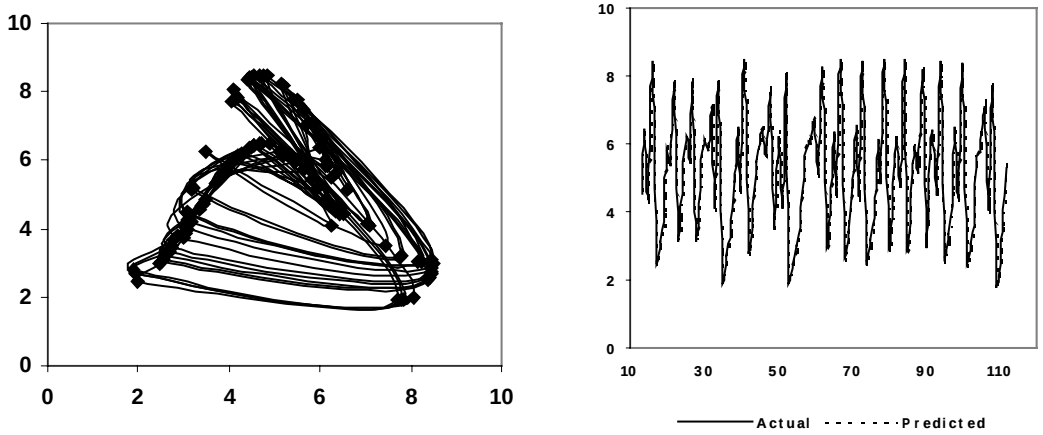

Figure 3. (a) JSM's Phase Diagram. (b) Its Actual and Predicted Values.

f. *Jet Ski Map with Low-Noise (or JSM_LN):* This is the same nonlinear jet ski map with low noise added. The process has signal-to-noise ratio ≈ 68 with ε_{tL} ~N(0, 0.2325^2). The fittest GEM is:

$$Y_t = \cos [\sin\{(\ln(Y_{t-7}))^{1/2}\} + \cos \{\sin(\ln(Y_{t-1})) + Y_{t-1}\}] + \cos [\{\sin (\ln(Y_{t-6}))^{1/2}\} + \{\cos(\sin\{(\sin(Y_{t-1}))^{1/2}\}^{1/2}) + Y_{t-1}] + \cos (Y_{t-2}) + [\cos\{\sin(Y_{t-1})^{1/2}\} + \sin\{\cos(\sin(Y_{t-11})) + Y_{t-1}\}^{1/2}]^{1/2} + \ln(86)$$

The statistics in Table 1 show low α-statistic and high $(1-\beta)$-statistic. Figure 3c shows actual versus predicted.

g. *Jet Ski Map with High-Noise (or JSM_HN):* This is the same process but with signal-to-noise ratio ≈ 6.8 or $\varepsilon_{tH} \sim N(0, 0.7352^2)$. The fittest GEM found is:

$$Y_t = 5 + (\ln(Y_{t-1}) / Y_{t-9}) + (\cos(Y_{t-5}) / Y_{t-8}) - 2\sin(Y_{t-1}) - (\sin(Y_{t-1}) / (Y_{t-8})^{1/2})$$

$$+ (\sin(Y_{t-6}) / Y_{t-7}) + \ln[\cos\{\sin(\cos(\exp(Y_{t-6})))\}] + \cos(Y_{t-2})$$

This model's MAPE is 80.2% less than that produced by a random walk model as shown in Table 1. Figure 3d shows actual versus predicted. No other forecasting method was attempted here either.

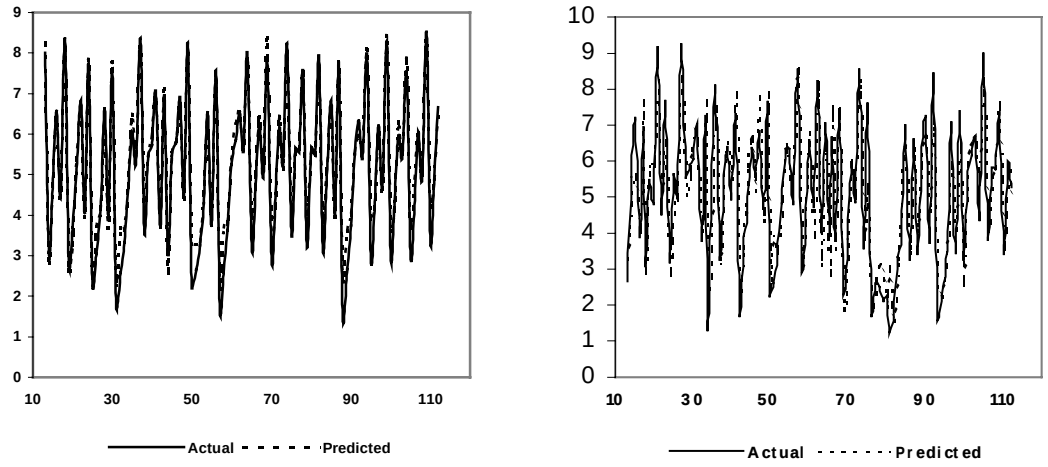

Figure 3. (c) JSM_LN, Actual and Predicted.

(d) JSM_HN, Actual and Predicted.

2.2 Real-World Series

For each series, a review of its history and prior studies precedes presenting plots of its dynamics and predicted versus actual values.

a. Canadian Lynx Data

This classic time series contains annual records of the numbers of Canadian lynx trapped in the MacKenzie river district of North-West Canada for the period 1821-1934. Elton and Nicholson (1942) were first to report it, and Moran (1953) was first to analyze the data statistically. Studies that analyzed modeling of this data include Tong (1977), Campbell and Walker (1977), Gabr and Rao (1981), Haggan and Ozaki (1981), Tsay (1989), Tong (1990), and Terasvirta (1994) among others. Following Moran (1953) as well as succeeding studies and to make the series more symmetric, the original series is transformed by $\log^{10}$ first. Figure 4a shows the phase diagram of the transformed series.

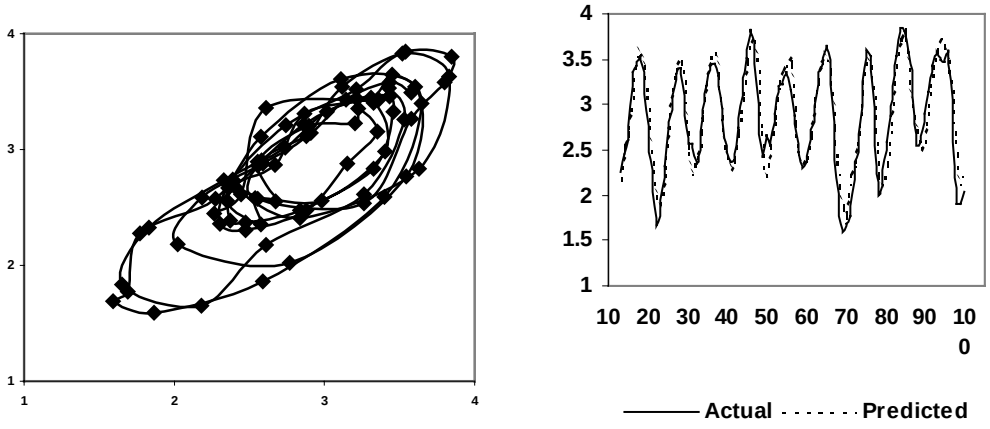

Figure 4. (a) Lynx Data Phase Diagram. (b) Its Actual and Predicted.

The fittest GEM found is:

$$Y_t = \left[Y_{t-1} * \{ Y_{t-9} * (X1_t / X2_t)^{1/2} \}^{1/2} \right]^{1/2}$$

where

$$X1_t = Y_{t-1} + \cos\left[Y_{t-4} * \{ Y_{t-9} * (Y_{t-12} * Y_{t-9})^{1/2} \}^{1/2} \right]$$

and

$$X2_t = \left[Y_{t-1} + \{\cos (Y_{t-3} * (Y_{t-11} - Y_{t-9}))^{1/2}\}\right] - \left[\cos\{Y_{t-9} + \cos (Y_{t-4} * (Y_{t-4} - Y_{t-9})^{1/2}) - \cos(Y_{t-2} - Y_{t-1})\}\right].$$

Table 2. Values of MSE, NMSE, and α (Historic Canadian Lynx Data)

	PADD	SETAR1	FAR	SAR	SETAR2	SBL	GP
s_e^2	0.046	0.042	0.036	0.038	0.042	0.022	0.028
NMSE	0.137	0.125	0.107	0.113	0.125	0.067	0.084
α	0.329	0.300	0.294	0.310	0.341	0.183	0.231

Table 2 compares GP's performance with prior studies. Figure 4b shows the actual versus predicted. Predicted values from the fittest GEM compares well with values from prior studies. Although Lin and Pourahmadi (1998) were able to obtain a PADD model with s_e^2 = 0.0194 using the same sample, adjusted for data mining effects s_e^2 = 0.038. Data mining which involves data snooping and exploratory analysis may improve prediction but with known practical problems. Breiman (1995) and Chatfield (1995) discuss such problems and suggest solutions. Gabr and Rao (1981) (SBL) were able to obtain the lowest s_e^2 = 0.022 using a bilinear fit of the series. The GP model's s_e^2 = 0.028 < 0.0358, 0.0378, or 0.0415 produced by AR subset (or SAR), full AR (or FAR), and SETAR(2; 6, 3) models. SETAR(2; 6, 3) is referred to as SETAR2 in the Table and was reproduced from Tong and Lim (1980) in Gabr and Rao (1981, p. 166). SETAR1 is a SETAR(2; 5, 2) model reproduced by Lin and Pourahmadi (1998) from Tong (1990) because of its successful forecasting. The statistics reported in Table 2 make GP a promising prediction algorithm. It was not possible to compute (1- β) for all compared models, but GP's MAPE was only 56% that of the random walk model.

b. Sunspot Numbers

Sunspots are dark blotches on the sun that sometimes exceed the earth in size. Sunspot numbers are representations of sunspot activity since 1700 in the form of an index. Waldmeier (1961) reported monthly means of sunspot numbers from 1700 to 1960. Tong (1990) provided a review of statistical models applied to this data. Weigend et al. (1990) explain that sunspot appearances are usually in pairs affected by the sun's magnetic field and therefore follow a magnetic cycle. Most statistical analysis and modeling of sunspot numbers fit models to the first 221 observations starting 1700 and forecast 35 observations for the period 1921-1956. Therefore, it is essential for comparison to evolve GEMs using the same 221 observations then forecast the same 35 periods using the resulting GEM. Figure 5a shows the phase diagram of the series.

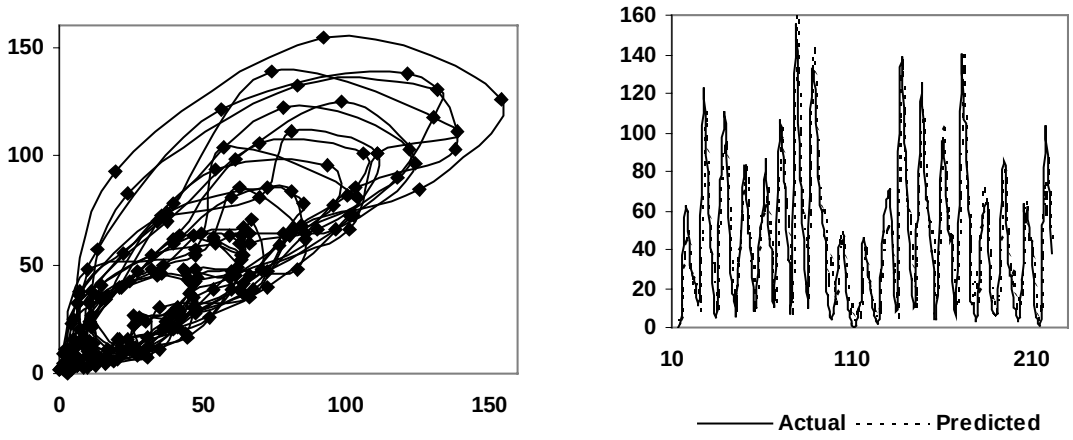

Figure 5. (a) Sunspot Numbers Phase Diagram. (b) Their Actual & Predicted Values.

The fittest GEM found is:

$$Y_t = -X1_t + X2_t + X3_t + X4_t$$

where

$X1_t = Y_{t-1} - \{(Y_{t-2} - (Y_{t-11})^{1/2})/ (17/Y_{t-3}) - Y_{t-11}\}^{1/2}$,

$X2_t = \mathbf{[}Y_{t-2} - \mathbf{\{}* [Y_{t-2} - (Y_{t-1} - \{(Y_{t-2} - (Y_{t-1} - ((Y_{t-2} - \ln(Y_{t-3})) - (Y_{t-3})^{1/2}) / (16/Y_{t-3}))) /$
$((Y_{t-11})^{1/2} / Y_{t-3})\} / (2 / Y_{t-4})\mathbf{\}} \mathbf{]}^{1/2}$,

$X3_t = [Y_{t-1} - \{(Y_{t-1} - (Y_{t-8})^{1/2} - (Y_{t-12})^{1/2}) / (12/Y_{t-3})\}]^{1/2}$,

and

$X4_t = [Y_{t-1} - \{(Y_{t-2} - (Y_{t-12})^{1/2} / (11/Y_{t-3})\}]^{1/2}$.

Table 3. Values of MSE, NMSE, and α (Historic Sunspot Numbers)

	ANN	**FAR**	**SAR**	**SETAR**	**SBL**	**GP**
s_e^2	97.47	199.27	203.21	153.71	124.33	111.75
NMSE	0.082	0.171	0.175	0.132	0.107	0.106
α	0.217	0.444	0.453	0.343	0.277	0.249

The results and comparison with other models are summarized in Table 3. Figure 5b shows the actual versus predicted. The fittest GEM produced prediction statistics ranking second among existing studies. As shown in Table 3, GP's s_e^2 = 111.75 is above training s_e^2 = 97.47 in Weigend et al. (1990) obtained using ANN algorithm. However, the produced GEM compares favorably to s_e^2 = 124.33 reported by Gabr and Rao (1981) using a subset bilinear model (SBL), s_e^2 = 153.71 produced by a SETAR(2, 4, 12) model by Tong and Lim (1980), the Full AR(9) (or FAR) model with s_e^2 = 198.33, and the Subset AR (or SAR) model with s_e^2 = 202.27 reported in Gabr and Rao (1981, p. 162). The α-statistics were much higher than expected for all models. The best model (ANN) produced mean squared error that is 21.7% that of a random walk model. The (1- β)-statistic for the GP model = 0.282 suggesting poor improvement over random walk prediction. It was not possible to compute (1- β) for the other studies.

3. EVALUATION OF FORECASTS

This section contains the main result of this paper: testing GP's integrity in forecasting time series. It is important to analyze single-step-ahead as well as multi-step-ahead forecasts and when available compare those to existing forecasts. As the name suggests, in single-step-ahead the model forecasts only one period ahead using actual lagged values of the data and not model predicted values. In multi-step-ahead forecasts the model is used to forecast a number of future values using predicted values for lags in periods beyond data used to evolve the model. Clearly, multi-step-ahead forecasts are more useful but, as one may suspect, forecast-errors increase as one forecasts periods further out into the future. With a few modifications, the same statistics used to evaluate historic prediction apply when evaluating single- and multi-step-ahead forecasts. NMSE is redefined such that NMSE = $F^{-1}\sum(Y_{t_f} - \hat{Y}_{t_f})^2 / T^{-1}\sum(Y_t - \bar{Y})^2$, where $t_f = 1,\ldots,F$ forecast periods and $\bar{Y}$ is the mean of historic values. The modified α-statistic is only for the specified period, or $\alpha = s_e^2 / s_{e\_RW}^2 = F^{-1}\sum(Y_{t_f} - \hat{Y}_{t_f})^2 / F^{-1}\sum(Y_{t_f} - Y_{t_f-1})^2$. Computations of forecast statistics are in Table 4 for the simulated data, Table 5 for the Canadian lynx data, and Table 6 for sunspot number. The number of multi-step-ahead to forecast is denoted by *h*. For example, a one-step-ahead forecast is identified by $h = 1$, two-step-ahead forecast by $h = 2$, and so on.

3.1 Artificially Simulated Series

Since there are no prior forecasts for these series except for the Henon map (Stern, 1996) and alternative forecasts only for linear and linear-stochastic series, GEMs' single-

and multi-step-ahead forecasts are presented with comparison made only where available. Brief comments on forecasts of simulated data follow.

Table 4. Values of MSE(h), α(h), MAPE(h), and 1-β(h) (Forecasts of Simulated Data)

	LDM	**LDM_LN**	**LDM_HN**	**Henon**	**JSM**	**JSM_LN**	**JSM_HN**
$s_e^2(1)$	0.000	0.104	0.425	0.001	0.007	0.236	0.655
$s_e^2(2)$	0.000	0.133	0.425	0.002	0.025	1.370	2.126
$s_e^2(3)$	0.000	0.130	0.461	0.006	0.084	2.441	2.835
$s_e^2(4)$	0.000	0.133	0.588	0.021	0.197	3.145	4.789
$s_e^2(5)$	0.000	0.135	0.589	0.050	0.696	8.145	5.874
$s_e^2(6)$	0.000	0.173	0.991	0.075	2.411	7.357	6.181
$\alpha(1)$	0.000	0.009	0.034	0.000	0.001	0.031	0.086
$\alpha(2)$	0.000	0.004	0.012	0.003	0.004	0.210	0.352
$\alpha(3)$	0.000	0.004	0.013	0.003	0.007	0.304	0.291
$\alpha(4)$	0.000	0.009	0.038	0.026	0.019	0.327	0.925
$\alpha(5)$	0.002	0.423	0.612	0.036	0.055	1.486	0.806
$\alpha(6)$	0.000	0.019	0.105	0.064	0.254	0.990	1.091
MAPE(1)	0.019	0.288	0.371	0.055	0.017	0.097	0.139
MAPE(2)	0.019	0.260	0.371	0.078	0.028	0.230	0.291
MAPE(3)	0.019	0.253	0.361	0.108	0.044	0.315	0.315
MAPE(4)	0.019	0.260	0.390	0.269	0.061	0.348	0.423
MAPE(5)	0.019	0.268	0.389	0.399	0.095	0.521	0.422
MAPE(6)	0.032	0.346	0.373	0.409	0.186	0.458	0.440
$1-\beta(1)$	0.992	0.916	0.885	0.980	0.972	0.816	0.707
$1-\beta(2)$	0.992	0.904	0.885	0.928	0.950	0.557	0.426
$1-\beta(3)$	0.995	0.940	0.888	0.951	0.939	0.488	0.459
$1-\beta(4)$	0.994	0.943	0.880	0.757	0.904	0.439	-0.258
$1-\beta(5)$	0.953	0.564	0.880	0.788	0.839	-0.335	-0.041
$1-\beta(6)$	0.984	0.909	0.885	0.756	0.615	-0.025	-0.274

a. *Linear Diffusion Map (LDM):* LDM statistics in Table 4 indicate GP's ability to produce reasonably acceptable single- and multi-step-ahead forecasts for up to six

periods. However, forecasts generated by the linear regression model produce perfect multi-step-ahead forecasts even when $h = 20$ with $\alpha(20) = 0.0$ and $1 - \beta(20) = 1.0$. GP's forecast are less impressive with $\alpha(20) = 0.0002$ and $1 - \beta(20) = 0.984$.

Table 5. Values of MSE(h), α(h), MAPE(h), and 1-β(h)-(Lynx Forecasts)

	PADD	**SETAR1**	**FAR**	**SAR**	**SETAR2**	**SBL**	**GP**
$s_e^2(1)$	0.008	0.014	0.025	0.022	0.014	0.013	0.020
$s_e^2(2)$	-	-	0.074	0.072	0.026	0.044	0.043
$s_e^2(3)$	-	-	0.116	0.120	0.033	0.063	0.058
$s_e^2(4)$	-	-	0.161	0.169	0.037	0.077	0.067
$s_e^2(5)$	-	-	0.185	0.202	0.048	0.086	0.061
$s_e^2(6)$	-	-	0.186	0.207	0.123	0.076	0.057
$s_e^2(14)$	0.010	0.018	-	-	-	-	0.051
$\alpha(1)$	0.123	0.199	0.371	0.325	0.211	0.194	0.294
$\alpha(2)$	-	-	0.304	0.298	0.107	0.183	0.191
$\alpha(3)$	-	-	0.247	0.255	0.070	0.134	0.128
$\alpha(4)$	-	-	0.255	0.267	0.059	0.121	0.107
$\alpha(5)$	-	-	0.287	0.314	0.075	0.133	0.094
$\alpha(6)$	-	-	0.389	0.433	0.257	0.158	0.117
$\alpha(14)$	0.151	0.269	-	-	-	-	0.746
MAPE(1)	0.024	0.039	0.041	0.038	-	0.032	0.037
MAPE(14)	0.024	0.039	-	-	-	-	0.067
$1-\beta(1)$	0.681	0.600	0.475	0.505	-	0.592	0.529
$1-\beta(14)$	0.688	0.500	-	-	-	-	0.135
Ranking:							
$s_e^2(1)$	1	3	7	6	4	2	5
$s_e^2(2)$	-	-	5	4	1	2	3
$s_e^2(3)$	-	-	4	5	1	3	2
$s_e^2(4)$	-	-	4	5	1	3	2
$s_e^2(5)$	-	-	4	5	1	3	2
$s_e^2(6)$	-	-	4	5	3	2	1
$\alpha(1)$	1	3	7	6	4	2	5
$\alpha(2)$	-	-	5	4	1	3	2
$\alpha(3)$	-	-	4	5	1	3	2
$\alpha(4)$	-	-	4	5	1	3	2
$\alpha(5)$	-	-	4	5	1	3	2
$\alpha(6)$	-	-	4	5	3	2	1

Table 6. Values of MSE(h), α(h), MAPE(h), and 1-β(h) (Forecasts of Sunspot Numbers)

	ANN	FAR	SAR	SETAR2	SBL	GP
$s_e^2(1)$	102.23	190.89	214.10	148.21	123.77	126.06
$s_e^2(2)$	-	414.83	421.40	383.90	337.54	353.27
$s_e^2(3)$	-	652.21	660.38	675.59	569.79	789.54
$s_e^2(4)$	-	725.85	716.08	773.51	659.05	1037.5
$s_e^2(5)$	-	771.04	756.39	784.27	718.87	1257.6
$\alpha(1)$	0.160	0.299	0.335	0.232	0.194	0.197
$\alpha(2)$	-	0.209	0.212	0.193	0.170	0.178
$\alpha(3)$	-	0.178	0.180	0.184	0.155	0.215
$\alpha(4)$	-	0.144	0.143	0.154	0.131	0.206
$\alpha(5)$	-	0.132	0.130	0.135	0.123	0.216
MAPE(1)	-	0.271	0.337	-	0.328	0.264
$1-\beta(1)$	-	0.556	0.448	-	0.462	0.568
Ranking:						
$s_e^2(1)$	1	5	4	6	2	3
$s_e^2(2)$	-	4	5	3	1	2
$s_e^2(3)$	-	3	4	2	1	5
$s_e^2(4)$	-	3	2	4	1	5
$s_e^2(5)$	-	3	2	4	1	5
$\alpha(1)$	1	5	4	6	2	3
$\alpha(2)$	-	4	5	3	1	2
$\alpha(3)$	-	2	3	4	1	5
$\alpha(4)$	-	3	2	4	1	5
$\alpha(5)$	-	3	2	4	1	5

b. *Linear Diffusion Map with Low-Noise (LDM_LN):* GP produced one- and multi-step-ahead forecasts almost statistically equivalent to those a linear fit. In one-step-ahead forecasting, both methods yielded $\alpha(1) = 0.009$ but the linear model yielded $1 - \beta(1) = 0.954 > 0.916$ for GP. GP's multi-step-ahead forecast was acceptable up to four periods ahead where the linear regression model produced better forecasts at $h = 5$. However,

$\alpha(20) = 0.20$ and $1 - \beta(20) = 0.896$ for the linear regression while $\alpha(20) = 0.28$ and $1 - \beta(20) = 0.905$ for GP's forecasts.

c. Linear Diffusion Map with High-Noise (LDM_HN): GP's forecasts of this series were rather acceptable. One-step-ahead of linear model $\alpha(1) = 0.01$ and $1 - \beta(1) = 0.977$ and multi-step-ahead $\alpha(20) = 0.147$ and $1 - \beta(20) = 0.959$. GP's one-step-ahead $\alpha(1) = 0.034$ and $1 - \beta(1) = 0.885$ and multi-step-ahead $\alpha(20) = 0.112$ and $1 - \beta(20) = 0.965$. These results suggest that GP performs worse in one-step-ahead forecast but produces statistics that are competitive in multi-step-ahead forecasting.

d. *The Henon Map:* GP forecasting of this data was rather successful as the data in Table 4 reveal. However, when compared with ANN one-step-ahead forecast (Stern, 1996), it appears that neural network architecture is more suited for noise free nonlinear time series. Using provided statistics, Stern's MSE = 0.0 and $\alpha(1) = 0.00002$. Multi-step-ahead GP forecasts were lower at lower *h* steps-ahead. (Stern (1996) did not report multi-step-ahead forecasts and thus there is no comparison to make here.)

e. *Jet Ski and Noisy Jet Ski Maps (JSM, JSM_LN, and JSM_HN):* These three are grouped together since there are no available forecasts to compare. The last three columns of Table 4 show that GP was successful in forecasting the noise-free data, and that its performance worsened as louder noise was added. It also shows that multi-step-ahead forecasts were worse at higher *h* periods ahead.

3.2 Real-World Series

Availability of many prior forecasts of Canadian lynx data and sunspot numbers provides a foundation for a better comparison of GP's relative forecasting ability of those series. Since prior forecasts are from different sources and for different multi-step periods ahead, a separate discussion of each is presented.

a. *Canadian Lynx Data:* Table 5 contains this series' forecast statistics for single- and multi-step-ahead periods. Gabr and Rao (1981, p. 166) present three one-step-ahead forecasts the 14 periods as well as the mean square errors for h = 1, …, 6. This explains why the selection of h = 1, …, 6. One of their models, SBL, produced the second best one-step-ahead forecast statistics found in this survey. Lin and Pourahmadi (1998, p. 189) reported the best one-step-ahead forecast statistics for the lynx data. Table 5 has all available statistics in its upper half. The lower half contains ranking of those statistics. A ranking of 1 indicates that it is the best of the forecasts reported for that specific number of h-steps-ahead. SETAR2's forecast ranks first and GP second when forecasting for $1 < h < 6$. At h = 6, GP's forecast ranks first and SBL second. Table 5 also shows that Lin and Pourahmadi's (1998) PADD model forecast was much better than GP's forecast at h = 14-step-ahead. Figure 8a displays forecasts for h = 1-6 produced by GEMs. A zero identifies actual values, and > 0 identifies the forecast plot representing those for h = 1-6.

b. *Sunspot Numbers:* Table 6 has the comparison of forecast statistics of sunspot numbers and their ranking according to h = 1-5 periods ahead following Gar and Rao (1981, p. 162). It is easy to understand why $h < 6$ given the fast deterioration in forecast quality as h increased. The best single-step-ahead forecast was that produced by Weigend

et al. (1990) using ANN, followed by SBL, then GP. For multi-step-ahead forecasting, SBL ranked first and GP second at $h = 2$, SBL first and SETAR2 second at $h = 3$, and SBL first and SAR second at $h > 3$. GP came in last at $h > 2$. Figure 8b displays plots of actual (0) versus forecasts at $h = 1$-6.

4. DISCUSSION

This study used genetic programming to evolve models for and forecast future values of (a) seven simulated data sets with different structures and (b) two popular time series: Canadian lynx and sunspot numbers. The results show that for linear and linear-stochastic systems GP may not be of much value and current standard statistical methods are superior. Statistical theory furnish established techniques to detect linearity that when used produce better specifications and at least equally reliable to GEMs' single- and multi-step-ahead forecasts. For nonlinear processes (with or without noise) little comparison was possible and perhaps ANN produces better forecasts than GP. GP provides an obvious advantage over conventional statistical methods that furnish no technique for detecting the type of nonlinearity needed for modeling such processes. In forecasting Canadian lynx data, GP performed relatively well, coming in second among the seven studies reporting single-step-ahead forecasts, and second among five studies reporting multi-step-ahead forecasts. GP's forecast of sunspot numbers ranked third among six studies reporting single-step-ahead forecasts, second among five reporting two-step-ahead forecasts, but last among five reporting forecasts for more than two periods ahead.

Genetic programming modeling and forecasting is a new and promising methodology, which has only been peripherally investigated. Whether GP can reach levels beyond any method known is difficult to predict at this time. ANN seems to produce better forecasts, albeit from a black box. GP modeling and forecasting does have some merit. It produces model specifications without any human input. Once a user identifies an appropriate set of explanatory variables, GP will search for the best specification. While comforting, these equations are rarely ever the correct specification but predict and forecast well. Using this method to predict time-series is logical since one's main interest is limited to forecasting and not understanding a functional relationship. Using GP to evolve behavioral models may be a bit more intricate but remains promising and will surely soon capture the attention of many researchers. Analyses presented in this study invite much needed further investigation of the potential GP may have. Applied statisticians in disciplines such as physics, chemistry, biology, and economics may find the new method logical, easy to use, and useful.

REFERENCES

Banzhaf, W., Nordin, P., Keller, R., and Francone, F. (1998), *Genetic Programming: An Introduction*, San Francisco: Morgan Kaufmann Publisher and Heidelberg: dpunkt.verlag.

Breiman, L. (1995), "Better Subset Regression Using the Nonnegative Garrote," *Technometrics*, 37, 373-383.

Campbell, M. and Walker, A. (1977), "A Survey of Statistical Work on the MacKenzie River Series of Annual Canadian Lynx Trappings from the Year 1821-1934 and a New Analysis," *Journal of Royal Statistical Society A*, 140, 411-431.

Ceciliano, J. and Nieva, R. (1999), "Transmission Network Planning Using Evolutionary Programming," in *Proceedings of the 1999 Congress of Evolutionary Computation*, July 6-9, Washington, D.C., Piscataway, NJ: IEEE, 1796-1803.

Chatfield, C. (1995), "Model Uncertainty, Data Mining and Statistical Inference," *Journal of the Royal Statistical Society A*, 158, 419-466.

Chen, S., Yeh, C., and Lee, W. (1998), "Option Pricing with Genetic Programming," in J. Koza, W. Banzhaf, K. Chellapilla, K. Deb, M. Dorigo, D. Fogel, M. Garzon, D. Goldberg, H., Iba, and R. Riolo, eds., *Genetic programming 1998: Proceedings of the Third Annual Conference*, San Francisco, CA: Morgan Kaufmann, 32-37.

Downing, K. (1998), "Combining Genetic Programming and Genetic Algorithms for Ecological Simulation," in J. Koza, W. Banzhaf, K. Chellapilla, K. Deb, M. Dorigo, D. Fogel, M. Garzon, D. Goldberg, H., Iba, and R. Riolo, eds., *Genetic Programming 1998: Proceedings of the Third Annual Conference*, San Francisco, CA: Morgan Kaufmann, 48-53.

Elton, C. and Nicholson, M. (1942), "The ten-Year Cycle in Numbers of the Lynx in Canada," *Journal of Animal Ecology*, 11, 215-244.

Fukunaga, A. and Stechert, A. (1998), "Evolving Nonlinear Predictive Models for Lossless Image Compression with Genetic Programming," in J. Koza, W. Banzhaf, K. Chellapilla, K. Deb, M. Dorigo, D. Fogel, M. Garzon, D. Goldberg, H., Iba, and R.

Riolo, eds., *Genetic Programming 1998: Proceedings of the Third Annual Conference*, San Francisco, CA: Morgan Kaufmann, 95-102.

Gabr, M. and Rao, S. (1981), "The Estimation and Prediction of Subset Bilinear Time Series with Applications," *Journal of Time Series Analysis*, 2, 155-171.

Haggan, V. and Ozaki, T. (1981), "Modeling Nonlinear Random Vibrations Using Amplitude-Dependent Autoregressive Time Series Model," *Biometrika*, 68, 189-196.

Henon, M. (1976), "A Two-Dimensional Mapping with a Strange Attractor," *Comm. Math. Phys.*, 50, 69.

Hinterding, R. (1999), "Representation, Constraint Satisfaction and the Knapsack Problem," in *Proceedings of the 1999 Congress of Evolutionary Computation*, July 6-9, Washington, D.C., Piscataway, NJ: IEEE, 1286-1292.

Hu, Y. (1998), "Biopattern Discovery by Genetic Programming," in J. Koza, W. Banzhaf, K. Chellapilla, K. Deb, M. Dorigo, D. Fogel, M. Garzon, D. Goldberg, H., Iba, and R. Riolo, eds., *Genetic Programming 1998: Proceedings of the Third Annual Conference*, San Francisco, CA: Morgan Kaufmann, 152-157.

Iba, H. and Sasaki, T. (1999), "Using Genetic Programming to Predict Financial Data," in *Proceedings of the 1999 Congress of Evolutionary Computation*, July 6-9, Washington, D.C., Piscataway, NJ: IEEE, 244-251.

Kaboudan, M. (1999), "Diagnosing Chaos by a Fuzzy Classifier," *Fuzzy Sets and Systems*, 108, 1-10.

Koza, J., (1992), *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, Cambridge, MA: The MIT Press.

Lee, D., Lee, B., and Chang, S. (1997), "Genetic Programming Model for Long-Term Forecasting of Electric Power Demand," *Electric Power Systems Research,* 40, 17-22.

Lin, t. and Pourahmadi, M. (1998), "Nonparametric and Non-Linear Models and Data Mining in Time Series: A Case-Study on the Canadian Lynx Data," *Applied Statistics,* 47, Part 2, 187-201.

May, R. (1976), "Simple Mathematical Models with Very Complicated Dynamics," *Nature*, 261, 459-467.

McKay, B. Willis, M., and Barton, G. (1997), "Steady-State Modelling of Chemical Process Systems Using Genetic Programming," *Computers and Chemical Engineering*, 21, 981-996.

Moran, P. (1953), "The Statistical Analysis of the Canadian Lynx Cycle," *Australian Journal of Zoology*, 1, 163-173.

Paredis, J. (1995), "Coevolutionary Computation," *Artificial Life,* 2, 355-375.

Sathyanarayan, S., Birru, H., and Chellapilla, K. (1999), "Evolving Nonlinear Time-Series Models Using Evolutionary Programming," in *Proceedings of the 1999 Congress of Evolutionary Computation*, July 6-9, Washington, D.C., Piscataway, NJ: IEEE, 236-243.

Singleton, A. (1994), http://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/genetic /gp/systems/gpquick.

Stern, H. (1996), "Neural Network in Applied Statistics," *Technometrics*, 38, 208-214.

Stoica, A., Klimeck, G., Salazar-Lazaro, C., Keymeulen, D., and Thakoor, A., (1999), "Evolutionary Design of Electronic Devices and Circuits," in *Proceedings of the*

1999 Congress of Evolutionary Computation, July 6-9, Washington, D.C., Piscataway, NJ: IEEE, 1271-1278.

Terasvirta, T. (1994), "Specification, Estimation and Evaluation of Smooth Transition Autoregressive Models," *Journal of the American Statistical Association*, 89, 208-218.

Tong, H. (1977), "Some Comments on the Canadian Lynx Data – with Discussion," *Journal of the Royal Statistical Society A*, 140, 432-435.

Tong, H. (1990), *Nonlinear Time Series Analysis: A Dynamical System Approach*, Oxford: Oxford University Press.

Tong, H., and Lim, K. (1980), "Threshold Autoregressive, Limit Cycle and Cyclical Data," *Journal of Royal Statistical Society B,* 42, 245-292.

Tsay, R. (1989), "Testing and Modeling Threshold Autoregressive Processes," *Journal of the American Statistical Association*, 84, 231-240.

Waldmeier, M. (1961), "The Sunspot Activity in the Years 1610-1960," Zurich, Swiss Federal Observatory.

Weigend, A., Huberman, B., and Rumelhart, D. (1990), "Predicting the Future: A Connectionist Approach," *International Journal of Neural Systems*, 1, 193-209.