Predicting Stock Abundance of the Barents Sea Capelin Using Genetic Programming

Hossam Faris[1], Mouhammd Alkasassbeh[2], Alaa Sheta[3]

[1.] Department of Business Information Systems Department, University of Jordan, Amman, 11942, Jordan
[2.] Department of Computer Science, Mutah University, Mutah, Jordan
[3.] Department of Computers and Systems, The Electronics Research Institute (ERI), Cairo, Egypt
7ossam@gmail.com

Abstract: Genetic Programming (GP) has been used significantly to solve many problems in modeling and prediction for dynamical systems. In this paper we explore the use of GP in solving the prediction problem for stock abundance of the Barents Sea capelin. The Barents Sea at the Northwest Atlantic, Capelin plays an important role of the forage and commercial fish. The distribution of the capelin fish has shown dramatic changes in its biology during the 1990s. This change affected the major ground fish stocks. Many soft computing techniques were used to predict the Capelin stock distribution. We propose the GP technique for modeling the Capelin stock problem in of the Barents Sea. The proposed GP model is compared with two other models developed using well known techniques; the Artificial Neural Network (ANNs) and the Multiple Linear model Regression (MLR) model. The proposed GP model shows higher capability and accuracy in prediction the Capelin stock distribution.

[Hossam Faris, Mouhammd Alkasassbeh, Alaa Sheta. **Predicting Stock Abundance of the Barents Sea Capelin Using Genetic Programming.** *Biomedicine and Nursing* 2018;4(2): 62-71]. ISSN 2379-8211 (print); ISSN 2379-8203 (online). http://www.nbmedicine.org. 12. doi:10.7537/marsbnj040218.12.

Keywords: Genetic Programming, Modeling, Capelin stock distribution, Barents Sea, Prediction.

Introduction

Prediction is an important subject since ancient times and attracted a lot of talent researchers in various areas. Prediction is about analyzing information or knowledge using certain methodology such these models can be built to predict the future events. Many predicting applications were reported. They include water prediction [1], [2], river flow prediction [3], [4] and stock market prediction [5–7].

One of the areas which have always been in the main concern of the researchers is the water flow prediction and management. In [8], authors provided a solution to the forecasting problem of the river flow for two well known rivers in USA. They are the Black Water River and the Gila River.

A comparison between the Feed-forward Neural Network and the Linear Auto-Regressive (AR) models were provided.

NN model showed a better modeling capability compared to the AR model. An adaptive network-based fuzzy inference system (ANFIS) approach was used to construct a River flow forecasting system in [9]. The advantages of ANFIS as an estimation model for river's flow were investigated in details. The developed results showed that the ANFIS can be applied successfully to solve the river's flow forecasting problem and provide reliable results. River's flow forecasting using ANN was discussed in [10].

Earlier, two models for forecasting the Nile River flow have been developed in [11]. A traditional linear autoregressive (AR) model and a feed forward neural networks (NNs) model are presented. Various NNs models with a variable number of neurons in the hidden layer were developed. The performance of both the AR and NNs models was tested using a set of measurements recorded at Dongola station in Egypt. A significant improvement of the error when using NNs model was achieved. Other investigation on predicting the river Nile using ANN, FL and GP were presented in [11–15].

Barents Sea

The Barents Sea is a part of the Arctic Ocean located north of Norway and Russia. It is named after the Dutch navigator Willem Barents. In the Barents Sea at the Northwest Atlantic, Capelin plays an important role of the forage and commercial fish. The capelin fish has shown dramatic changes in its biology during the 1990s, which affected the major ground fish stocks [16] (see Figure 1). The capelin has a northerly circumpolar distribution, and it plays a key role in the arctic food. Since 1979, the Barents Sea capelin fishery has been regulated by a bilateral fishery management agreement between Russia and Norway [17], [18].

The Capelin stock in the Barents Sea is the largest in the world and has maintained a fishery with annual catches of up to 3 million tons [19]. The joint Norwegian-Russian survey on pelagic fish in the Barents Sea, the capelin stock was estimated at only 530 thousand tones, compared to 2.21 million tones in 2002. Historical stock levels and the rate of removals from most stocks are in most cases obtained from analyses of commercial fisheries data by Virtual

Population Analysis (VPA) or other stock number at age based models [20].

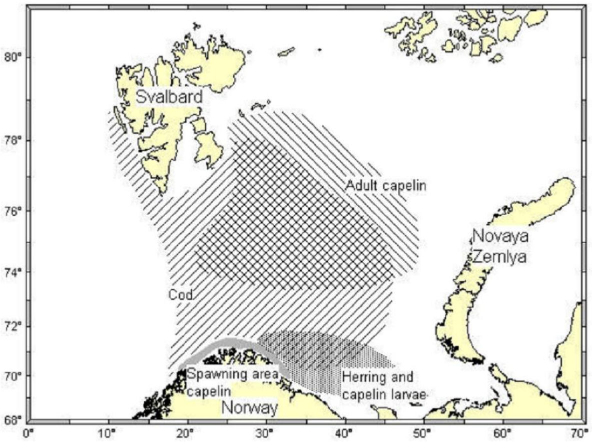


Figure 1. Map of the Barents Sea with the main features of the distribution of various age groups of capelin as well as cod and herring. Predation by juvenile herring (1-3 years old) on larval capelin seems to have great impact on capelin recruitment at times when herring is abundant in the Barents Sea. Cod predation infers high mortality on the adult capelin, especially during capelin spawning in March [17].

Likewise, the marine fish population was a hot area to be investigated, for example, the work of [19]; they used differential equation model and a frequency transfer function model of the stock properties. Their results showed that the capelin stock dynamics is adapted to the 18.6 year. Also, An artificial neural network (ANN) approach for predicting stock abundance of the Barents Sea capelin based on training ANN with genetic algorithm was presented at [17]. Their model was tested for its ability to predict capelin abundance in single years, using the remaining time series for training.

The results were promising for the predictions, and the ANN method gives higher predictive ability than a simple fisheries assessment model [21].

In [22] authors tackled the problem by using two different approaches first by ANN adapted using the Genetic Algorithm (ANN-GA) model and secondly by Multiple Linear model Regression (MLR) model. The results of the developed two models were compared with the observed real values. The work showed that the ANN-GA model can have better overall accuracy over (MLR). In this paper, we compare the results obtained the GP approached with their results in order to assist its performance.

What Makes Gp Suitable For Prediction?

Genetic programming (GP) [23], [24] is one of the most well known evolutionary computation techniques, which inspired by the biological evolution and developed by J. R. Koza at Stanford University. GP has many advantages in modeling the dynamic and complex nonlinear systems in a wide range of different domains [25–27]. Mainly, GP has the following advantages:

1. GP generates mathematical models with interpretable structure, relating input and output variables from a data set without preprocessing and identifying key parameters.
2. GP does not need any a priori knowledge about the internal structure of the system and can adapt with various constraints.
3. As part of GP successful evolutionary process; they can estimate the required model parameters to fit the GP model with the collected system measurements [28].
4. GP has a high explanation power. Models developed using GP can give an insight into the hidden relationships between model variables. Moreover, these models are sometimes less complex than models which can be developed using conventional modeling approaches [29].

Gp Evolutionary Process

Genetic Programming (GP) is an evolutionary algorithm based methodology for automatically solving problems in inspired by biological evolution. GP has been applied successfully to a large number of complex problems like industrial modeling, electronic circuitry, pattern recognition, computational finance, and picture generation. The GP evolutionary cycle is described as a flow chart in Figure 2. The details of the flow chart can be summarized in the following four steps:

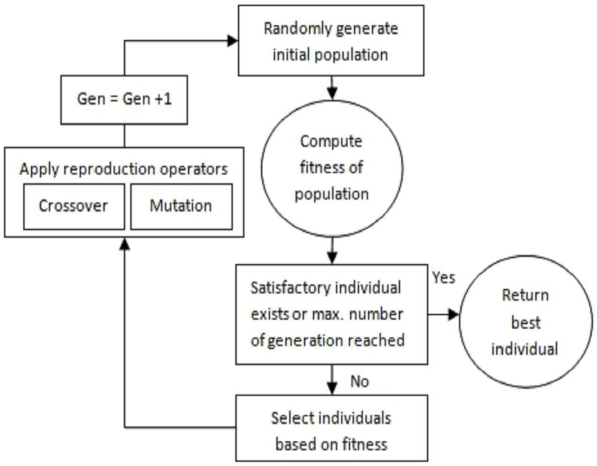

Randomly generate initial population
Gen = Gen +1
Compute fitness of population
Apply reproduction operators
Crossover
Mutation
Satisfactory individual exists or max. number of generation reached
Yes
Return best individual
No
Select individuals based on fitness

Figure 2. Flow chart of the evolutionary process based GP.

1. Initialization: Genetic programming evolutionary cycle starts by randomly generating a set of individuals which form a population. The number

of individuals in one population is referred to as population size. Each individual is a computer program and can be represented as a tree or as LISP expression. In Figure 3, we show a simple GP tree representation of the system (with output z) as given in Equation 1.

$$z = \frac{(X*2)+5}{\sin(Y)} \qquad (1)$$

2. Fitness Evaluation: The second step is called fitness evaluation. In this step, each individual is evaluated using a specific measurement. In this research, we use the difference between the values obtained in this step and the desired output value. By this process, the accuracy of the generated individuals in predicting is determined.

3. Reproduction: Genetic operations applied on probabilistically selected individuals from the population based on fitness their fitness value (known as selection mechanism). The higher value the individual has the more probable to be selected. Genetic operations include:

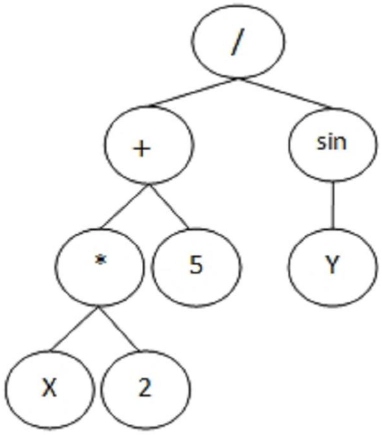


Figure 3. A simple GP Tree Representation.

a) Crossover: this is the most important genetic operator. In this operator two computer programs are randomly selected then a randomly chosen part of the first individual is replaced by another randomly chosen part from the second computer program. The same operation is performed on the second program in the same manner. The operation is illustrated in Figure 4.

b) Mutation: This operation is applied on single individual by randomly choosing a point in the tree representation of the computer program and replacing it with another randomly generated subtree as shown in Figure 5. Usually, the probability of the mutation operator is much less than the crossover one. After applying genetic operations iteratively, the new generated populations replace the old one.

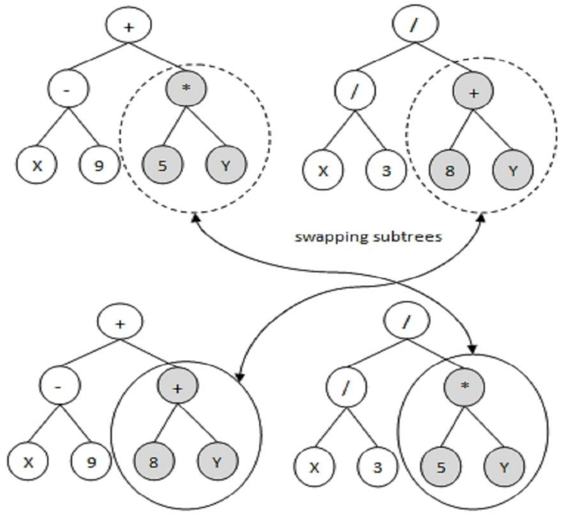


Figure 4. Example of GP crossover operator.

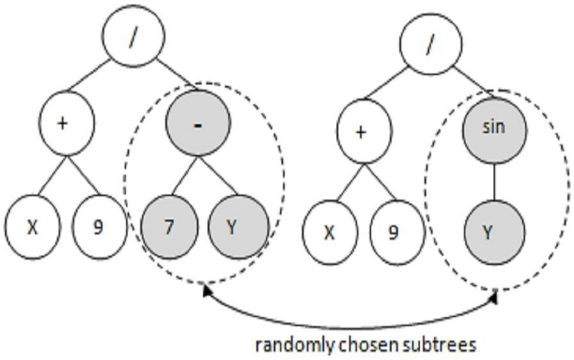


Figure 5. Example of GP mutation operator.

4. Termination condition: New populations are generated iteratively by the last process until one of the following conditions is met;

- Number of generations is reached. Which is a predetermined number specified by the user to end the iterative process after a number of loops.
- An individual with a specific fitness value is reached. Finally, the best-so-far individual is chosen to be the solution of the problem.

Models For Dynamic Systems

When we deal with a system, we need to get some idea on how its variables relate to each other. Thus, in a general point of view, we can call such an assumed relationship among observed signals a system model. Clearly, models may come n various structures and be phrased with varying degree of mathematical forms. The intended use determines the degree of sophistication that is required to make the model meaningful [30]. Modeling and identification of nonlinear systems are quite application dependent problem and often have their roots in tradition and specific techniques in the application area in question.

In many advanced applications, it is necessary to use models that describe the relationships among the system variables in terms of mathematical expressions like difference or differential equations. These models are called mathematical models. Mathematical models may be further characterized by a number of adjectives (linear or nonlinear) signifying the type of difference or differential equation used.

The use of mathematical models is inherent in all fields of engineering. In fact, a major part of the engineering field deals with how to make good designs based on mathematical models. A dynamic system can be described by two types of models: input-output models and state-space models [31]. In the following section, we describe a few model structures linear and nonlinear systems.

A. Input-Output Models

An input-output model describes a dynamic system based on input and output data. In the discrete-time domain, an input-output model can be of the ARMA type or the parametric Hammerstein model [32]. An input-output model assumes that the system output can be predicted by the past inputs and outputs of the system. If the system is further supposed to be deterministic, time invariant, single-input single-output (SISO), the input-output model becomes:

$$y(k) = f(y(k-1), y(k-2), \dots, y(k-n), u(k-1), u(k-2), \dots, u(k-m)) \qquad (2)$$

Where u (k), y (k) represents the input-output pairs of the system at time k. Positive integers n and m are respectively the number of past outputs (also called the order of the system) and the number of past inputs. In practice m is usually smaller than or equal to n. f can be a static nonlinear function which maps the past inputs and outputs to a new output.

B. Proposed GP Model

The capelin biomass values starting from 1979 until 1999 were obtained from a data set published in [17]. The data set was divided into two equal arts.

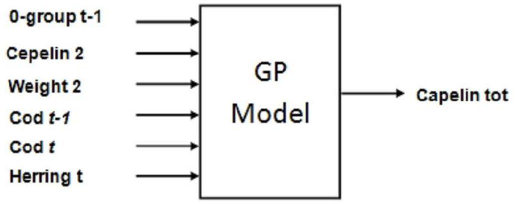


Figure 6. The inputs and output of the proposed GP model.

The first 50% of the data (i.e. 13 years) was used in the training phase of the GP modeling process while the next 13 years were used for the testing phase. Six variables were chosen as input variables. Input variables are 0-group t-1, Capelin 2 t, Weight 2 t, Cod t, Cod t-1 and Herring t while Capelin tot. is the predicted output variable. The proposed GP model architecture based the inputs and the output variables is shown in Figure 6. The data set used for developing our GP model is given in Table I.

C. Fitness Evaluation

In order to check the performance of the developed regression model and compare the results obtained with previous works, the Variance-Accounted-For (VAF) performance criterion is assessed to measure how close the measured values to the values developed using the genetic programming approach. The VAF is computed as:

$$VAF = \left[1 - \frac{var(y-\hat{y})}{var(y)}\right] . 100\% \qquad (3)$$

where y, $\hat{y}$ are the real actual output and the model estimated output, respectively.

I. Hueristiclab framework

HueristicLab framework[1] was used to apply the GP approach and the experiments designed in this research. HeuristicLab is a flexible and extensible graphical user interface software environment for heuristic optimization based on Microsoft. Net and C# [33], [34]. HeuristicLab consists of a large number of plugins (95 plugins in HeuristicLab 3.3.5). In this paper, we are using ”Genetic Programming – Symbolic Regression” from the list of algorithms provided by the package. The data set is then imported easily as a CSV file format. Input/Output variables are determined then we tune the GP parameters from a special dedicated tab as specified in Table II.

A snapshot of HeuristicLab environment at the end of the run is shown in Figure 7. The snapshot shows the best GP individual obtained is represented as a tree graph on the right side while the statistical results are shown on the left.

Experimental Results

The data set described in the previous section was loaded into Heuristiclab framework then a symbolic regression via GP was applied with parameters set as shown in Table IV. The cross validation was tuned to 50% for training and 50% for testing. After a run of 13 generations GP converged to the best model shown in equation 4. The GP best individual obtained was able to model the Capelin biomass from the year 1974 to 1986 with a VAF value of 94.2%, while it was capable of predicting for the years 1987 through 1999 with a VAF value of 80.4%.

The predicted values based GP model were compared with results obtained by other two different

[1] HeuristicLab is a framework for heuristic and evolutionary algorithms that is developed by members of the Heuristic and Evolutionary Algorithms Laboratory (HEAL). http://dev.heuristiclab.com

models from the literature [22]; the Artificial Neural Network - Genetic Algorithm (ANN-GP) model and the Multiple Linear Regression (MLR) model [22]. In Table II, we show the VAF for the GP model along with other models reported in the literature [22].

Table 1. VAF for the GP model and other reported models.

	Training	Testing
GP	94.20%	80.40%
NN-GA	81%	77%
MLR	86%	56%

The best generated GP model is represented in Equation 4 where X_1 is 0-group t-1, X_2 is Capelin 2 t, X_3 is Weight 2 t and X_4 is Cod t. The model parameters are c_0 = 1.92, c_1 = 2.20, c_2 = 1.28, c_3 = 1.01, c_4 = 0.79, c_5 = 2.20, c_6 = 1.28, c_7 = -17.31, c_8 = -8.16, c_9 = 0:0004

$$\hat{y} = ((C_0.X_2 + C_1.X_1.C_2.X_4).((C_3.X_2 + C_4.X_4 + C_5.X_1.C_6.X_4).C_7.C_8 + C_9) \quad (4)$$

GP shows a superior prediction power compared to the NNGA and MLR approaches. Moreover, GP has an important advantage compared to NN-GA; GP generates a mathematical model which gives an insight into the interaction between the variables, in contrast to neural network models that work as black-box input/output models, which is hard to explain. The tuning parameters of GP evolutionary process are shown in Table IV. In Figure 8 and Figure 9, we show the actual and predicted Caplin biomass curves in both the training and testing cases, respectively. The convergence of the GP evolutionary process is shown in Figure 10.

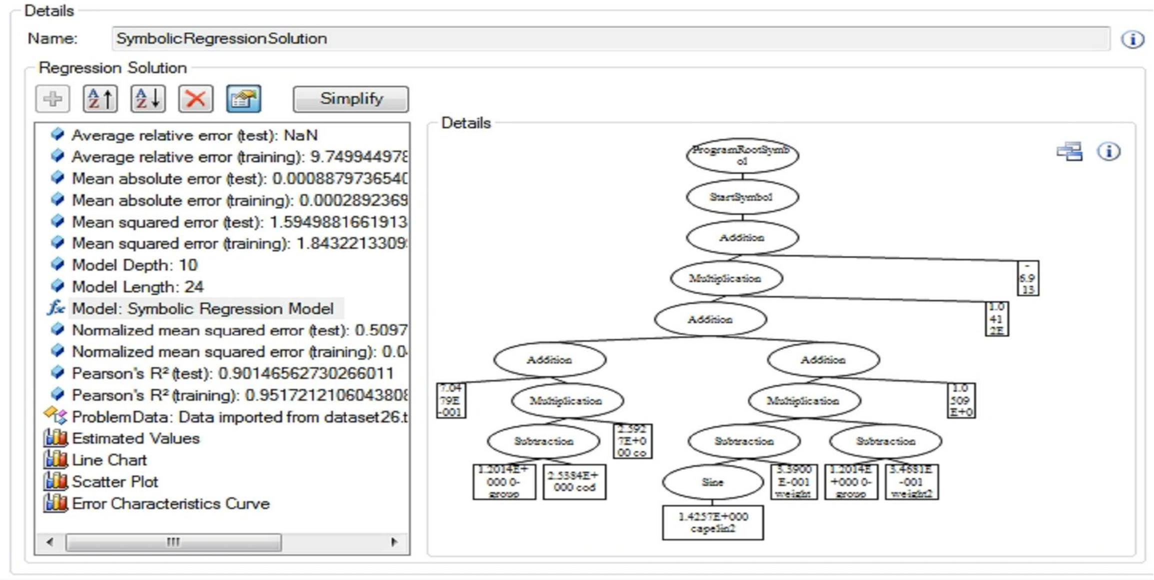
Table 2. Parameters used during the evolutionary process of GP.

Parameter	Value
Mutation probability	15%
Population size	1000
Maximum generations	50
Maximum Tree Depth	12
Maximum Tree Length	20
Selection mechanism	Tournament selector
Elites	1
$F_{\text{Capelin (t+1)}}$	{+,-, *}

The scatter plot of the observed and estimated values is represented in Figure 11. The figure shows how the training scatters (in orange color) of both values are highly concentrate in the vicinity of the identity line while the testing scatters (in red color) are little bit less concentrating.

Conclusions

In this work, a genetic programming approach was proposed for modeling the Capelin stock problem in of the Barents Sea. GP was applied on a data set of biomass values starting from 1979 until 1999 which collected from VBA for technical analysis of Capelin's biomass. The results were compared with two other techniques used for modeling the same problem; Artificial Neural Network (ANNs) and Multiple Linear model Regression (MLR) model. GP showed significant improvement in the prediction accuracy and higher explanation power.



Details

Name: SymbolicRegressionSolution

Regression Solution

- Average relative error (test): NaN
- Average relative error (training): 9.749944978
- Mean absolute error (test): 0.000887973654(
- Mean absolute error (training): 0.0002892369
- Mean squared error (test): 1.5949881661913
- Mean squared error (training): 1.843221330S
- Model Depth: 10
- Model Length: 24
- Model: Symbolic Regression Model
- Normalized mean squared error (test): 0.5097
- Normalized mean squared error (training): 0.0
- Pearson's R² (test): 0.90146562730266011
- Pearson's R² (training): 0.951721210604380
- ProblemData: Data imported from dataset26.t
- Estimated Values
- Line Chart
- Scatter Plot
- Error Characteristics Curve

Figure 7. A snapshot from HeuristicLab environment at the end of the run.

Table 1. Input/Ouput data for the GP model.

Year	0-group	Cap. 2	Cap. tot.	Weight 2	Cod	Herring
1974	359	3.1	4.8	5.6	3.1	0
1975	320	2.5	7.3	6.8	2.5	0
1976	281	2	5.8	8.2	2.55	0
1977	194	1.5	4.2	8.1	2.15	0
1978	40	2.5	4.5	6.7	1.8	0
1979	660	2.5	4.1	7.4	1.5	0
1980	502	1.9	5.5	9.4	1.2	0
1981	570	1.8	3	9.4	1.2	0
1982	393	1.3	2.5	9	1.05	0
1983	589	1.9	2.6	9.5	0.8	0
1984	320	1.4	2.4	7.4	0.85	0.98
1985	110	0.4	0.7	8.2	0.95	1.84
1986	125	0.04	0.08	11.7	1.15	0.26
1987	55	0.02	0.02	12.3	1	0
1988	187	0.4	0.4	12.2	0.85	0
1989	1300	0.2	0.3	12.4	0.9	0.02
1990	324	2.7	3.2	15.3	0.95	0.05
1991	241	5	5.6	8.7	1.5	0.49
1992	26	1.7	3.9	8.6	1.85	1.67
1993	43	0.5	0.8	9	2.5	1.52
1994	58	0	0.1	11.2	2.3	2.86
1995	43	0.1	0.15	13.8	2	0.63
1996	291	0.2	0.26	18.6	1.9	0.1
1997	522	0.5	0.49	11.5	1.6	0.01
1998	428	1	1.25	13.4	1.6	0.15
1999	650	1.3	2.12	13.6	1.4	0.33

Table 3. Target and estimated Results of the GP developed model.

Year	Target Capelin t+1 (y)	Estimated Capelin t+1 ($\hat{y}$)
1974	4.8	5.129
1975	7.3	7.739
1976	5.8	5.827
1977	4.2	3.692
1978	4.5	4.464
1979	4.1	3.148
1980	5.5	4.674
1981	3	3.68
1982	2.5	2.802
1983	2.6	2.623
1984	2.4	2.334
1985	0.7	0.904
1986	0.1	0.475
1987	0	0.463
1988	0.4	0.553
1989	0.3	0.575
1990	3.2	9.452
1991	5.6	12.27
1992	3.9	3.328
1993	0.8	0.838
1994	0.1	0.489
1995	0.1	0.554
1996	0.3	0.574
1997	0.5	1.402
1998	1.3	3.222
1999	2.1	2.853

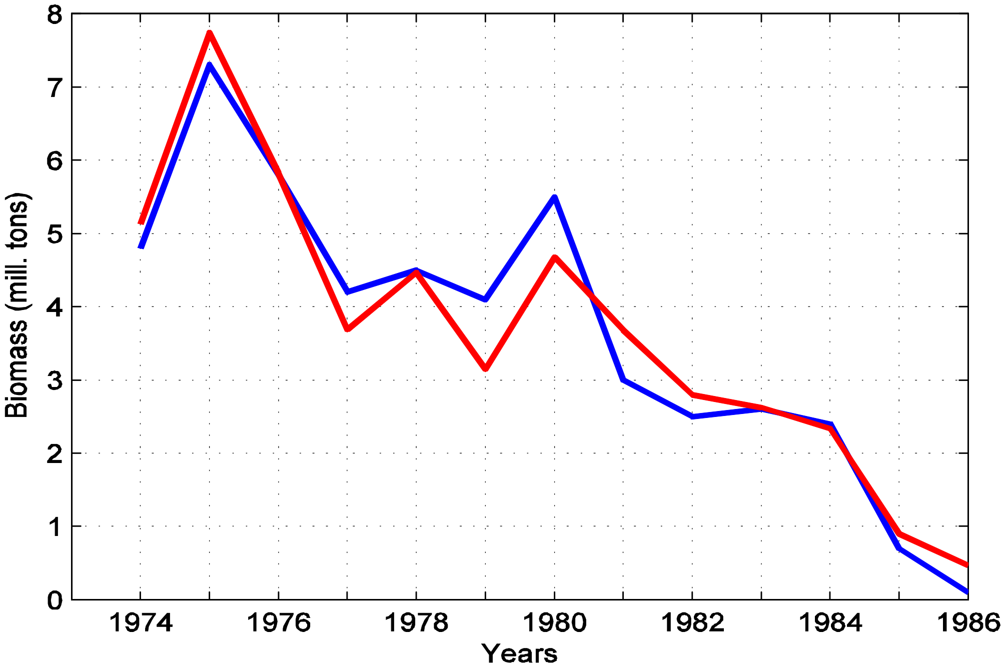


Figure 8. Actual and predicted Capelin biomass based GP model-Training.

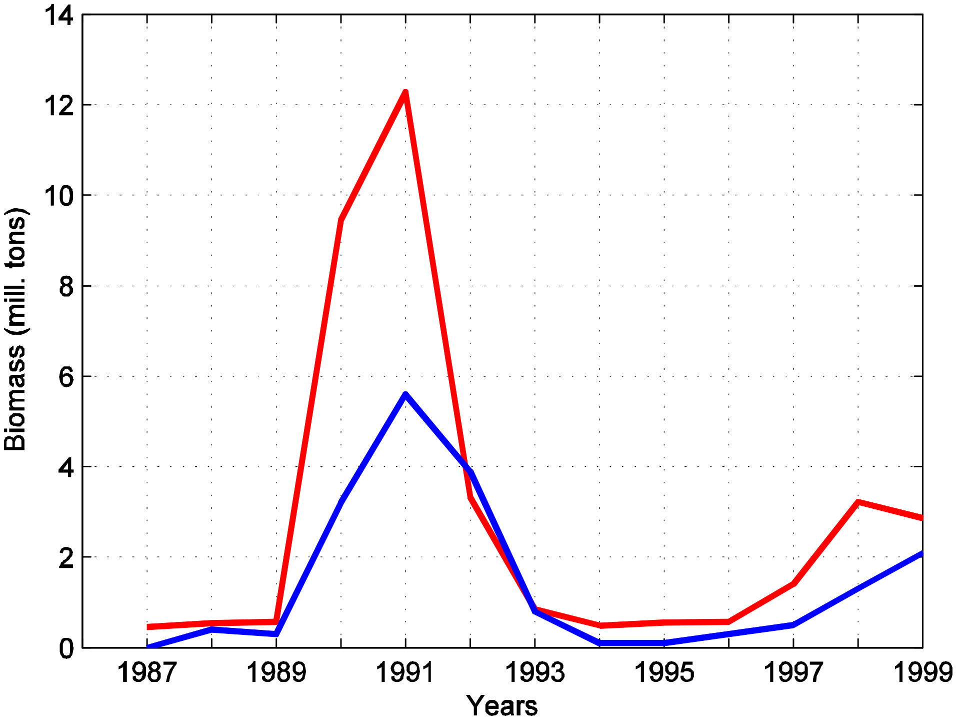


Figure 9. Actual and predicted Capelin biomass based GP model-Testing Case.

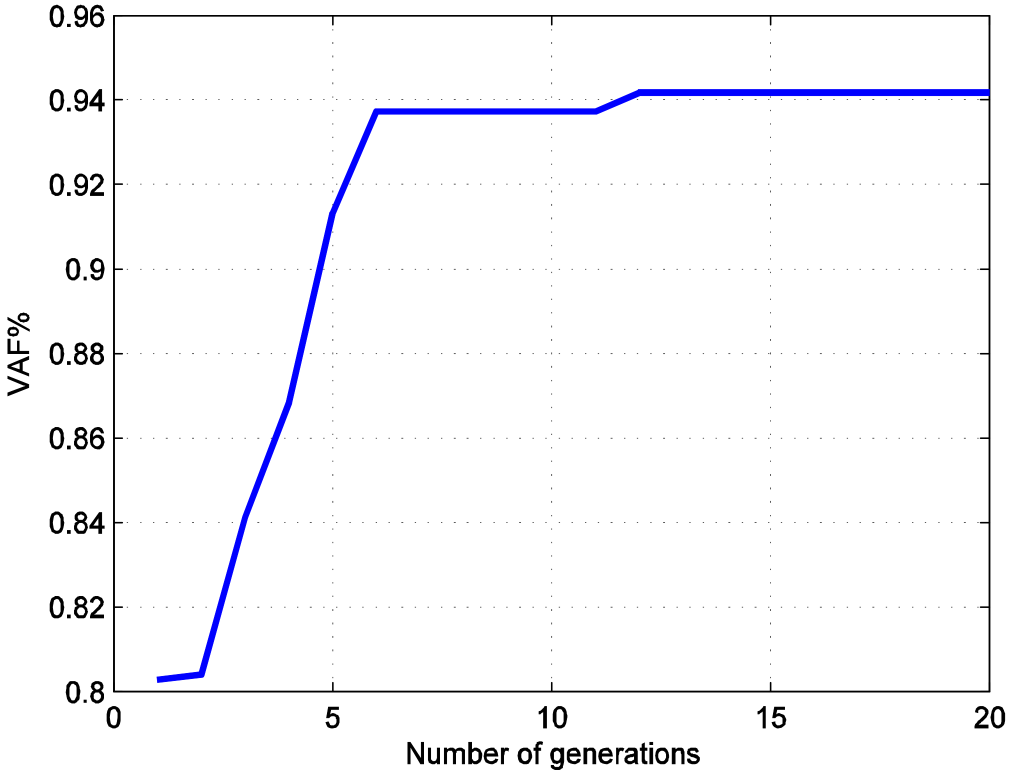


Figure 10. GP best so far for the Capelin prediction model.

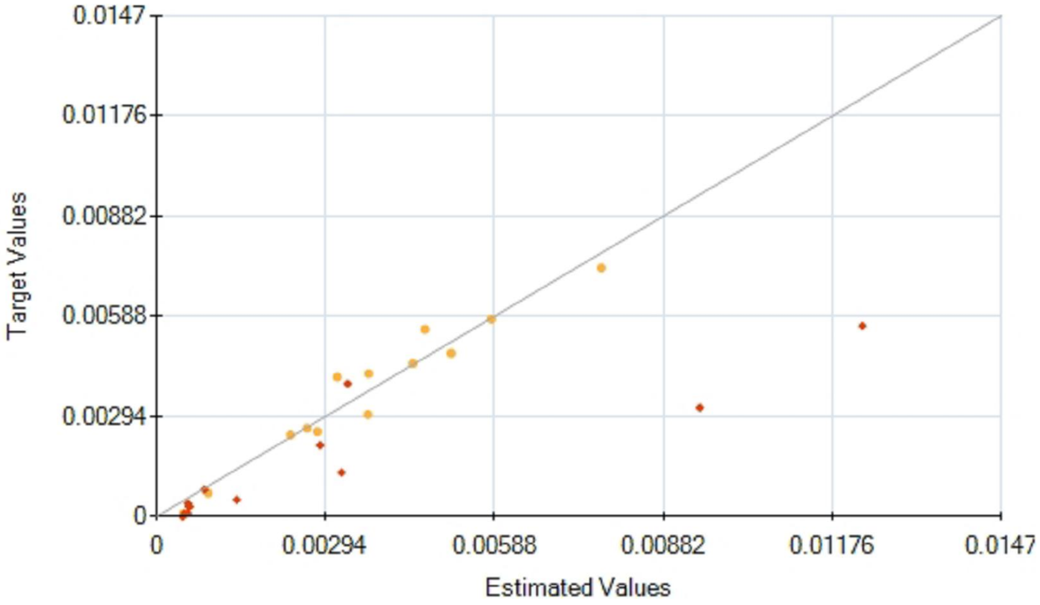


Figure 11. Scatter chart for observed Capelin biomass against predicted vales

References

1. P. Lynch, “The origins of computer weather prediction and climate modeling,” J. Comput. Phys., vol. 227, pp. 3431–3444, Mar. 2008.
2. K. Y. Wong, C. L. Yip, and P. W. Li, “Automatic identification of weather systems from numerical weather prediction data using genetic algorithm,” Expert Syst. Appl., vol. 35, pp. 542–555, July 2008.
3. L. See and S. Openshaw, “Applying soft computing approaches to river level forecasting,” Hydrol. Sci. J, vol. 44, pp. 763–778, 1999.

4. D. N. Kumar, K. S. Raju, and T. Sathish, “River flow forecasting using recurrent neural networks,” Water Resour. Manage, vol. 18, p. 2004, 2004.
5. Z. Rahamneh, M. Reyalat, A. Sheta, and S. Aljahdali, “Forecasting stock exchange using soft computing techniques,” Computer Systems and Applications, ACS/IEEE International Conference on, vol. 0, pp. 1–5, 2010.
6. M. A. Boyacioglu and D. Avci, “An adaptive network-based fuzzy inference system (anfis) for the prediction of stock market return: The case of the istanbul stock exchange,” Expert Syst. Appl., vol. 37, pp. 7908–7912, Dec. 2010.
7. W. Dai, J.-Y. Wu, and C.-J. Lu, “Combining nonlinear independent component analysis and neural network for the prediction of asian stock market indexes,” Expert Syst. Appl., vol. 39, pp. 4444–4452, Mar. 2012.
8. A . K. M. Baareh, A. F. Sheta, and K. A. Khnaifes, “Forecasting river flow in the usa: a comparison between auto-regression and neural network non-parametric models,” in Proceedings of the 6th WSEAS International Conference on Simulation, Modelling and Optimization, SMO’06, (Stevens Point, Wisconsin, USA), pp. 7–12, World Scientific and Engineering Academy and Society (WSEAS), 2006.
9. Firat and M. G¨ung¨or, “River flow estimation using adaptive neuro fuzzy inference system,” Math. Comput. Simul., vol. 75, pp. 87–96, July 2007.
10. Valenc¸a, T. Ludermir, and A. Valenc¸a, “River flow forecasting with constructive neural network,” in Proceedings of the 18th Australian Joint conference on Advances in Artificial Intelligence, AI’05, (Berlin, Heidelberg), pp. 1031–1036, Springer-Verlag, 2005.
11. A . Sheta and M. El-Sherif, “Optimal prediction of the nile river flow using neural networks,” in Proceeding of the IEEE International Joint Conference on Neural Networks, vol. 5 of IJCNN ’99, pp. 3438–3441, 1999.
12. B. Whitcher, S. D. Byers, P. Guttorp, and D. B. Percival, “Testing for homogeneity of variance in time series: Long memory, wavelets and the nile river,” tech. rep., Water Resour. Res, 1998.
13. A . Atiya, S. Member, S. M. El-shoura, S. I. Shaheen, and M. S. Elsherif, “A comparison between neural-network forecasting techniques - case study: River flow forecasting,” IEEE Transactions on Neural Networks, vol. 10, pp. 402–409, 1999.
14. A . Sheta and A. Mahmoud, “Forecasting using genetic programming,” in Proceedings of the 33rd Southeastern Symposium on System Theory, vol. 5, pp. 3438–3441, 1999.
15. Y. Al-Zubi, A. Sheta, and J. Al-Zubi, “Nile river flow forecasting based takagi-sugeno fuzzy model,” Journal of Applied Sciences, vol. 10, no. 4, pp. 284–290, 2010.
16. J. E. Carscadden, K. T. Frank, and W. C. Leggett, “Ecosystem changes and the effects on capelin (mallotus villosus), a major forage species,” Canadian Journal of Fisheries and Aquatic Sciences, vol. 58, no. 1, pp. 73–85, 2001.
17. G. Huse and H. Gjster, “A neural network approach for predicting stock abundance of the barents sea capelin,” Sarsia: a Norwegian journal of marine biology, vol. 84, pp. 479–482, 1999.
18. G. Rose, “Capelin (mallotus villosus) distribution and climate: a sea canary for marine ecosystem change,” ICES Journal of Marine Science: Journal du Conseil, vol. 62, no. 7, pp. 1524–1530, 2005.
19. H. Yndestad and A. Stene, “System dynamics of the barents sea capelin,” ICES Journal of Marine Science: Journal du Conseil, vol. 59, no. 6, pp. 1155–1166, 2002.
20. P. Sparre, “Introduction to multispecies virtual population analysis,” ICES mar. Sci. Symp, vol. 193, pp. 12–21, 2002.
21. J. Brodziak, P. Rago,, and R. Conser, “A general approach for making short-term stochastic projections from an age-structured fisheries assessment model.”
22. K. M. Eghnam, S. Bani-Ahmad, and A. F. Sheta, “Article: Solving capelin time series ecosystem problem using hybrid artificial neural networks-genetic algorithms model,” International Journal of Computer Applications, vol. 19, pp. 8–12, April 2011. Published by Foundation of Computer Science.
23. J. R. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection. The MIT Press, 1992.
24. J. R. Koza, Genetic Programming II. MIT Press, 1994. [25] A. H. Gandomi and A. H. Alavi, “A new multi-gene genetic programming approach to nonlinear system modeling. part i: materials and structural engineering problems,” Neural Comput. Appl., vol. 21, pp. 171–187, Feb. 2012.
25. J. Liu and A. Wu, “Modeling gene regulatory network based on genetic programming,” in Proceedings of the 2010 International Conference on Electrical and Control Engineering, ICECE ’10, (Washington, DC, USA), pp. 5341–5344, IEEE Computer Society, 2010.
26. E. F. Georgopoulos, G. P. Zarogiannis, A. V. Adamopoulos, A. P. Vassilopoulos, and S. D. Likothanassis, “A genetic programming environment for system modeling,” in Proceedings of the 5th Hellenic conference on Artificial Intelligence: Theories, Models and

Applications, SETN ’08, (Berlin, Heidelberg), pp. 85–96, Springer-Verlag, 2008.

27. N. Muttil and K.-W. Chau, “Neural network and genetic programming for modelling coastal algal blooms,” International Journal of Environment and Pollution, vol. 28, pp. 223–238, 6 November 2006.
28. A . Hussian, A. Sheta, M. Kamel, M. Telbaney, and A. Abdelwahab, “Modeling of a winding machine using genetic programming,” in Proceedings of the 2000 Congress on Evolutionary Computation CEC00, (La Jolla Marriott Hotel La Jolla, California, USA), pp. 398–402, 2000.
29. L. Lennart, System Identification: Theory for the User. New Jersey: Prientice Hall, 1987.
30. D. Pham and X. Liu, Neural Networks for Identification, Prediction and Control. New York: Springer, 1995.
31. R. Isermann, K. Lachmann, and D. Matko, Adaptive Control Systems. New York: Printicce Hall, 1992.
32. S. Wagner and M. Affenzeller, “The heuristiclab optimization environment,” tech. rep., Johannes Kepler University Linz, Austria, 2004.
33. A . Beham, M. Affenzeller, S. Wagner, and G. K. Kronberger, “Simulation optimization with heuristiclab,” in The 20th European Modeling and Simulation Symposium (EMSS2008), pp. 75–80, 2008.

6/25/2018