Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/120604

Temperature Forecasting: A Comparison between Parametric and Non-Parametric Models

Alaa Sheta[1,*], *Ajay Katangur*[2] *and Abdelkarim Baareh*[3]

[1] Department of Computing Sciences, Texas A&M University-Corpus Christi, Corpus Christi, TX 78412, USA
[2] Computer Science Department, Missouri State University, Springfield, MO 65897, USA
[3] Computer Science Department, Ajlune College, Al-Balqa Applied University, Ajlune, Jordan

Received: 23 Jun. 2018, Revised: 20 Sep. 2018, Accepted: 24 Sep. 2018
Published online: 1 Nov. 2018

Abstract: The development of accurate temperature prediction models is essential for not only human life but also for agricultural, animal life, tourism, and many others. Power consumption and achieving energy efficiency in buildings also depends on temperature. Although modeling-based regression is one of the most popular approaches, it still suffers from many difficulties related to the number of available measurements, the order of the model and the non-linearity of the data. In this paper, we provide a comparison between parametric and non-parametric models for temperature forecasting. We propose three-model structures to estimate the temperature in Mumbai, the business capital of India. They are parametric (i.e. Linear Regression (LR), Multi-gene Genetic Programming (MG-GP)) and non-parametric (i.e. Artificial Neural Networks (ANN)) models. These models are tested on data collected in Mumbai for the year of 2009. The results show that multi-gene GP model performs relatively well in predicting the temperature with a high degree of accuracy compared to the LR and ANN techniques.

Keywords: Temperature forecasting, Regression, Artificial Neural Networks, Multi-gene Genetic Programming

1 Introduction

Weather forecasting is now one of the most important fields of research due to its impact on environment, industry, and agriculture. Weather forecasting is the process of estimating future weather conditions based on a number of meteorological parameters. The development of forecasting models for a number of decades concentrated mainly on the linear model. The model is used to be static or dynamic based on the model variables and model type. A number of linear methods such as least square estimation (LSE) are used to estimate model parameters, such that this model can be used for prediction or forecasting purposes. Unfortunately, the results in many cases were not satisfactory. The models were unable to coop with environmental changes and utilization of model parameters was with limited success. Nonlinear process modeling presented a solution to forecasting problems in specific cases; that is when the nonlinear model is quantified before the parameters estimation is completed. This was also a challenge.

Recently, soft computing (SC) techniques were presented to handle the weather forecasting problem [1,2]. The main features that represent soft computing are the allowance of uncertainty, tolerance of imprecision, and partial truth. Many of the well-known SC techniques are ANN, Fuzzy Logic (FL), Evolutionary Computation (EC) (i.e., genetic algorithms and genetic programming) and Machine Learning (ML) techniques. Many modeling and identification problems were solved efficiently using SC techniques in river flow forecasting [3], flood prediction [4,5] and rainfall forecasting [6,7,8].

Temperature forecasting is important for many reasons. Agriculture, animal life, tourism, achieving energy efficiency, for example, are highly affected by temperature. Forecasting temperature also allows us to identify patterns of vegetation zones [9] and modeling hourly-diffuse solar radiation [10,11]. In the past, many research articles explored the

* Corresponding author e-mail: alaa.sheta@tamucc.edu

modeling of temperature-based meteorology, hydrology and agro-hydrology methods. Temperature prediction of a molten salt collector tube based on the Back-Propagation (BP) Neural Networks was presented in [12]. In [13], authors presented three models for the daily cycle of air temperature for nine years at five locations in North Carolina. Recently, forecasting ozone concentrations in the east of Croatia using nonparametric ANN models was presented in [14]. Two models-based Feed Forward (FF) and Radial Basis Function (RBF) network were provided [14].

This paper is organized as follows. In Section 2 we present the three adopted modeling techniques for temperature forecasting. They are the LR, ANN and MG-GP models. The method to model acceptance and evaluation of the proposed model structures are presented in Section 3. The developed results along with tables and figures are shown in Section 5. Finally, a conclusion and future works are discussed.

2 Proposed Model Structures

In this section, we provide a description of the three adopted models to handle the temperature forecasting problem. They are the simple linear regression, artificial neural network, and the multi-gene Genetic Programming models. The simple linear regression is a model which has set of parameters to be estimated using least square estimation (LSE) technique. This model is a linear model in the parameters. ANN can model any dynamic relationship and produce a nonlinear function of a system. It was successfully used to solve many system identification and modeling problems. MG-GP is a relatively new concept. It was explored in the diversity of modeling problem. MG-GP is capable of providing a mathematical equation model that represents the relationship between the model input variables to give an approximation to a particular output. We plan to discuss the advantages and disadvantages of each approach.

2.1 Linear Regression

Assume we have a single input single output (SISO) system that has input x and an output y. Our goal is to build a model for that system using n collected measurements $(x_i, y_i), i = 1, \ldots, n$. The system Equation 1 represents the relationship between x and y.

$$y = \alpha + \beta x \tag{1}$$

where α is the y-intercept and β is the slope of the line. The estimated output based on the developed model $\hat{y}$ (see Equation 2). Given that ϵ_i is the error difference between the actual system output y and the estimated output $\hat{y}$.

$$\hat{y} = \alpha + \beta x + \epsilon_i \tag{2}$$

Thus, we have to minimize the optimization function $Min\ L$ (see Equation 3) such that:

$$\begin{aligned} Min\ L &= \sum_{i=1}^{n} \epsilon_i^2 \\ &= \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \\ &= \sum_{i=1}^{n} (y_i - \beta - \alpha x_i)^2 \end{aligned} \tag{3}$$

To find the optimal values of the parameters α and β, we need to calculate the differentiation for the functions:

$$\frac{\partial L}{\partial \alpha} = \frac{\partial L}{\partial \beta} = 0 \tag{4}$$

Differentiating Equation 3 with respect to the model parameters α and β will produce Equations 4. Equating Equations 4 to zero produces two equations. In the above-described examples, we get two equation since we only have two unknowns α and β). Solving these two equations will result in an estimate for the two parameters. Thus, we will have a linear model that describes the function $f(x)$ for y. This model can be used to estimate new values of y_i based values of x_i.

2.2 Artificial Neural Networks

A neural network shows significant advantages in modeling and simulation of nonlinear and complex systems. They can capture the input/output relationships of the system by modifying their weights to reach the modeling goal with minimum error difference between the actual system response and the estimated one [15]. ANN is similar to the human brain such that they can obtain knowledge through learning and store this knowledge in their weights. ANNs have been found very useful in solving problems such as improving the production quality of a hot-rolling industrial process [16], predicting air pollution parameters [17,18], estimating software effort [19], rainfall prediction [7], wind power forecasting [20], forecasting ozone concentration [21] and many others.

ANN consists of hundreds or thousands of processing nodes called neurons. The ANN-based learning algorithm can be used to model a dynamical function between input and output variables. This function could be a complex nonlinear one. ANN shows significant success in solving various modeling problems in science and engineering. Yet, there are progress and improvement every day in their learning algorithms and application domains. Using a learning algorithm, these neurons are capable of creating a function that can map a relationship between inputs and output training examples.

An example of a fully-connected network ANN is shown in Figure 1. All inputs/units in one layer are connected to all units in the following layer, the first layer is known as the hidden layer h and the second layer is the output layer. The depicted network consists of three inputs, two hidden units, and two outputs.

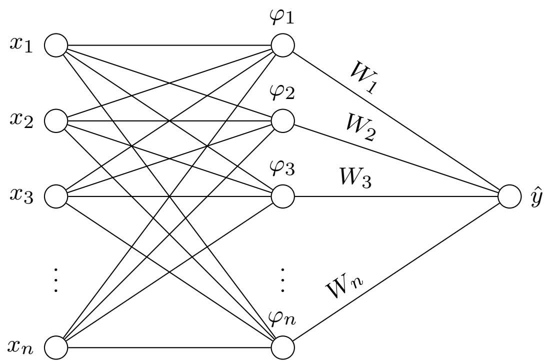

Fig. 1: Architecture of ANN

The system modeling process-based ANN is presented in Figure 2 and adopted from [22]. A common type of ANN is the FeedForward (FF) network. This type of system consists of at least three layers: an input layer, at least one intermediate hidden layer, and an output layer.

Assume we have an n set of input-output training examples $(x_1,t_1),...,(x,t_n)$ to be used for the BP learning algorithm of the FF neural network. x is the input and t_i is the target output, $i=1,\ldots,n$. Our objective is to develop a FF network such that the input pattern x_i when presented to this network, it produces an output o_i that is closely related to the target t_i. The goal is always to minimize an error criteria such that the ANN-produced behavior will be as close as possible to the actual one. An error criteria could come as in the form of Equation 5.

$$Min\ E = \frac{1}{2}\sum_{i=1}^{n}||o_i - t_i|| \tag{5}$$

The proposed BP ANNs always initialize its weight randomly. The gradient of the error function is computed and used to modify these initial weights recursively till best output results are reached. The

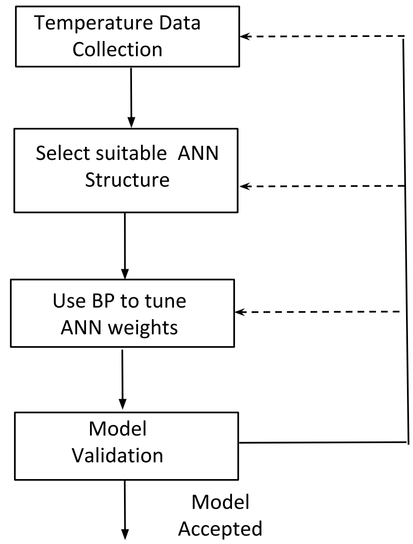

Fig. 2: Steps of the modeling process using ANN

BP algorithm is used to find a local minimum of the error function. The function E can be minimized using iterative process-based gradient descent as a function of the ANN weights $w_1,\ldots,w_j; j=1,\ldots,m$. m is the total number of weights in the ANN. The weights are updated with amount Δ computed as given in Equation 6.

$$\Delta E = \Big(\frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \ldots, \frac{\partial E}{\partial w_m}\Big) \tag{6}$$

Each weight is updated using the increment Δw_i as given in Equation 7.

$$\Delta w_i = -\gamma \frac{\Delta E}{\Delta w_i}, i = 1\ldots,m \tag{7}$$

where γ represents learning constant, i.e., a proportionality parameter which defines the step length of each iteration in the negative gradient direction.

2.3 Genetic Programming

GP is an evolutionary algorithm which is inspired by the principles of Darwinian evolution theory and natural selection [23,24]. GP helps developing a mathematical model of nonlinear systems by the optimization of a tree structure. The proposed tree structure always with variable length depending on the complexity of the function to be modeled. The tree consists of a set of nodes. These nodes are with different types. For example, it could be terminal or functional nodes. Other nodes are located inside the tree structure with various characteristics. An example of a tree structure of GP is shown Figure 3.

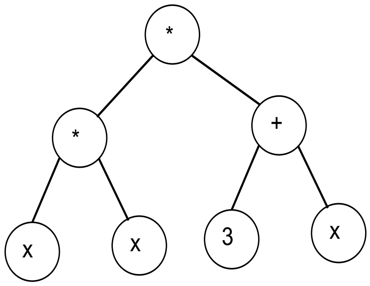

Fig. 3: Example of basic tree representation in GP.

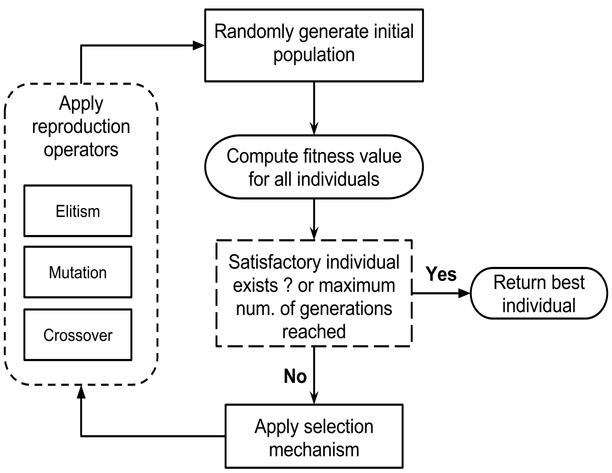

Fig. 4: Main loop of the GP [16]

GP algorithm as any evolutionary computation algorithm has a population of size n, which is chosen arbitrary based on the problem at hand. In Figure 4, we show the evolutionary process of GP. GP evolves computer programs that model complex nonlinear systems. Traditionally, J. Koza suggested LISP programs. LISP makes GP more flexible to deal with various data and structures for model manipulation. GP was used to solve diversity of problems in manufacture process modeling [25,26,27,16], identification of chemical processes [28], detection of diabetes [29], prediction of stock exchange [30,31], river flow prediction [3], surface ozone prediction [32], software effort estimation [33] and many others.

Symbolic regression is one of the most famous modeling techniques which are similar to traditional regression modeling. Multi-gene symbolic regression GP is unlike traditional regression analysis in which the user needs to decide upon the required model structure specification and then estimate the parameters from the input-output data. MG-GP has the ability to automatically evolve both the model structure and the parameters of the model parameters. One of the most interesting features of the MG-GP model is that it can include both linear and non-linear terms. MG-GP also use symbolic regression technique to estimate the model parameters. MG-GP approach used multiple tree structure to evolve a model in the format presented in Figure 5. The prediction of the $\hat{y}$ training data is given by:

$$\hat{y} = b_0 + b_1 T_1 + \cdots + b_G T_G \tag{8}$$

where T_G is the (N) vector of outputs from the $i^t h$ tree/gene comprising a multi-gene individual.

3 Model Acceptance and Evaluation

In order to check the prediction output performance of the developed models a number of criteria are adopted as follows:

– R^2 - goodness of fit (coefficient of determination)

$$R^2 = \frac{\sum_{i=1}^{n}(\hat{y}_i - \overline{y_i})^2}{\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \overline{y_i})^2 + \sum_{i=1}^{n}(\hat{y}_i - y_i)^2}} \tag{9}$$

– RMSE - root mean-squared error

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y - \hat{y})^2} \tag{10}$$

– MAE - mean-absolute error

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y - \hat{y}| \tag{11}$$

– SSE - sum of squared errors

$$SSE = \sum_{i=1}^{n}(y - \hat{y})^2 \tag{12}$$

– MSE - mean squared error

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y - \hat{y})^2 \tag{13}$$

where t is the actual temperature, $\hat{y}$ is the predicted temperature based on n measurements.

4 Materials and Method

This study is based on data recorded by the air temperature system at Weather Underground [35]. This dataset contains the real-time observation of the weather temperature in Mumbai, India (map

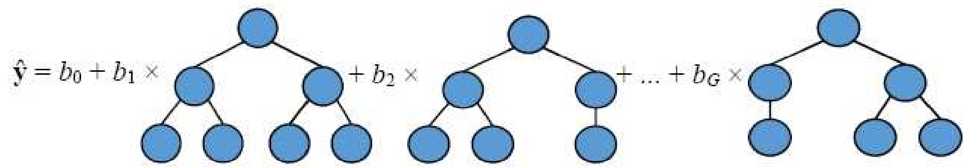

Fig. 5: Multigene symbolic regression model [34]

shown in Figure 6) from January 2009 to December 2009. The dataset contains many attributes such as Temperature (T), Dew Point (C), Humidity (%), Sea Level Pressure (hPa), Visibility (km), Wind Speed (km/h), and Precipitation (cm). In this study, we specify the proposed input and output variables used to develop the temperature prediction models. We used 365 measurements to develop the models, 255 were used as training data and 109 as testing data set. The selected attributes (i.e., inputs) are given in Table 1. A statistical measurement of the selected variables input and outputs are given in Table 2.

Fig. 6: Mumbai on Indian Map

5 Experimental Results

In this section, we plan to provide the results of three developed models for estimating the temperature in Mumbai, India.

Table 1: Inputs and output variables for the *Temperature* model

Inputs	Dew Point (C)	x_1
	Humidity (%)	x_2
	Sea Level Press. (hPa)	x_3
	Visibility (km)	x_4
	Wind Speed (km/h)	x_5
Output	Temp. (T)	y

5.1 Prediction-based Regression Model

The values of the regression model parameters are estimated using LSE method to produce the values of the parameters as given in Equation 14. The observed and estimated temperature using regression model in both training and testing cases are shown in Figure 7.

$$y = 0.9571x_1 - 0.28068x_2 - 0.12688x_3 - 0.18474x_4 - 0.031773x_5 - 156.19 \quad (14)$$

5.2 Prediction-based ANN Model

A Feed Forward ANN model is trained using BP learning algorithm to develop a temperature model. The network consists of one input layer, one hidden layer, and one output layer, respectively. The input layer has five neurons, the hidden layer has 15 neurons and the output layer has one neuron. The observed and estimated output temperature in both training and testing cases are presented in Figure 8. The convergence of the ANN over 300 epochs is shown in Figure 9.

5.3 Prediction-based MG-GP Model

To develop our multi-gene GP model, some parameters have to be tuned first before we start the evolutionary process. These parameters include the population size, selection mechanism, crossover and

Table 2: Statistical measurement of the selected variables

	Min	Max	Mean	Std
Dew Point	11	27	21.244	4.2066
Humidity	31	98	67.608	15.136
Sea Level Press	998	1017	1008.3	3.5781
Visibility	2	6	3.4932	1.0126
Wind	3	24	10.592	4.1172
Temperature	23	34	28.603	1.9396

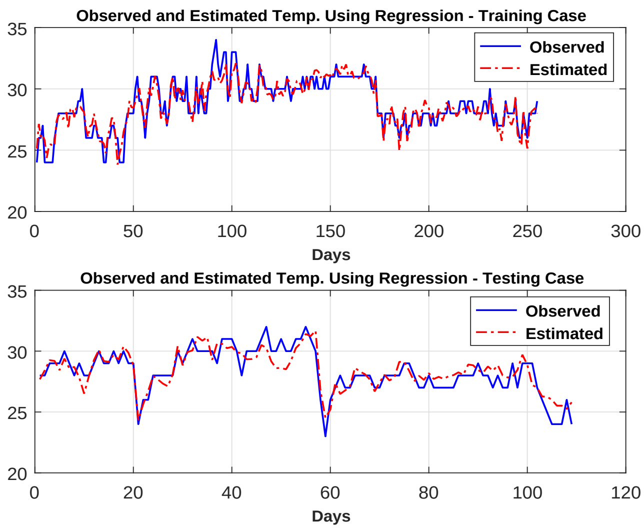

Fig. 7: Observed and Estimated Temperature Using Regression Model

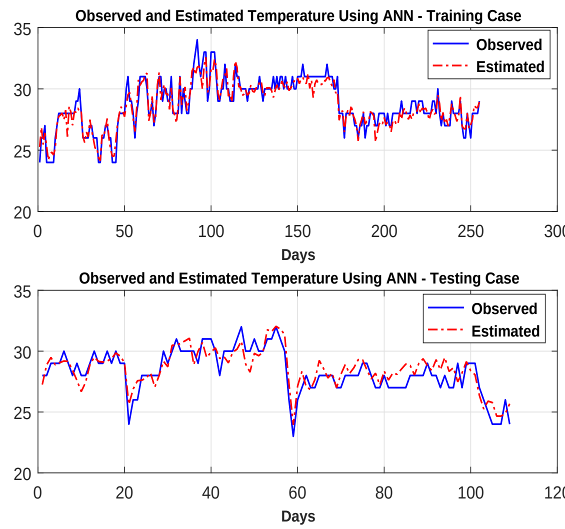

Fig. 8: Observed and Estimated Temperature Using ANN Model

mutation probabilities, the maximum number of genes allowed to constitute the multi-gene and many others. User has to setup the maximum number of genes G_{max} where a model is allowed to have. These setup parameters are presented in Table 3. MG-GP is using a tree structure to represent the temperature model. For the tree we need to setup a maximum depth D_{max}. Although this depth restricts the domain of search for the best model, it is important such that the MG-GP model will not be very complex for the user. The adopted function set to develop the GP model is shown in Table 3.

Table 3: GP Tuning Parameters

Population size	250
Number of generations	150
Selection mechanism	Tournament
Tournament size	25
Max. tree depth	4
Probability of crossover	0.84
Probability of mutation	0.14
Number of inputs	5
Max. genes	5
Function set	$\times$, +, -
Constants range	[-10 10]

5.3.1 GPTIPS Toolbox

In this research, we adopt a MATLAB software toolbox called Genetic Programming & Symbolic Regression for MATLAB (GPTIPS) toolbox [34] to develop our results. GPTIPS is an open source genetic programming toolbox for multi-gene symbolic regression. This software tool offers a number of suitable functions for exploring the population of possible models, thus examining model behavior, post-run a model simplification function and export the model to a number of formats, such as graphics file, LaTex expression, symbolic math object or standalone MATLAB file [34]. One of the main characteristics of GPTIPS is that it can be

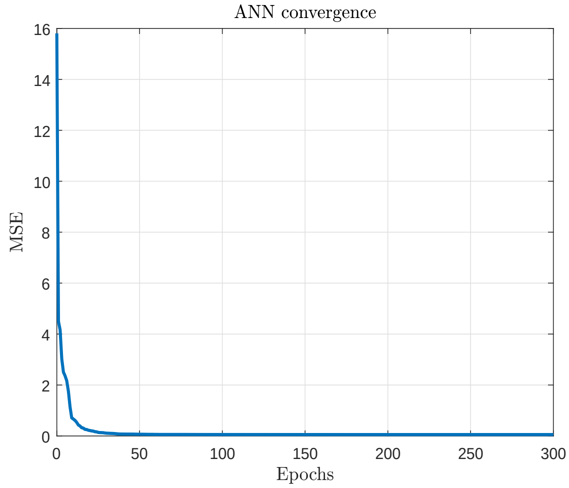

Fig. 9: Convergence of ANN for Temperature Forecasting

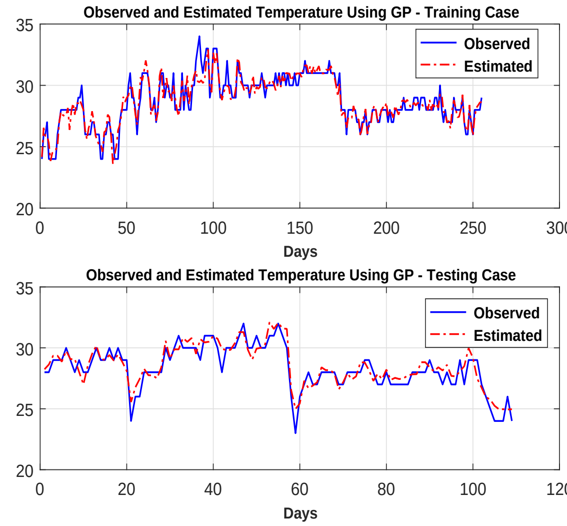

Fig. 10: Observed and Estimated Temperature Using Multi-gene GP Model

configured to evolve multi-gene individuals. Some features can be summarized as follows [34]:

–Multiple tree (multi-gene) individuals.
–Tournament selection & lexicographic tournament selection.
–Standard sub-tree crossover operator.
–Elitism.
–Early run termination criterion.
–Graphical population browser is showing best and non-dominated individuals (fitness & complexity).
–Graphical summary of fitness over GP run.
–Six different mutation operators.

The generated five individual genes/model terms are given in Table 4. Each gene includes its weighting coefficient.

In Equation 15, we show the mathematical equation evolved for index prediction using multi-gene GP. Figure 10 shows the observed and estimated temperature values based on the developed GP model in both training and testing cases. The performance measurements for the model is computed and summarized in Table 5.

$$y = 0.014x_1x_3 - 0.578x_2 - 0.332x_3 - 0.00722x_1x_2 - 12.7x_1 + 0.00338x_2^2 + 367.0 \tag{15}$$

6 Some Observations

From the results obtained as shown in Figures 7, 8, 10 we can clearly see that the multi-gene GP model is better at forecasting the temperature accurately during the testing phase. This is also evident from the results seen in Table 5. The advantage of this model is that we can use Equation 15 to calculate the temperatures which make it convenient for the user to predict temperatures in the future. The R^2 value for multi-gene GP model is higher than the other two models which are good as this model exhibits a higher degree of variability of the predicted temperature data around its mean value. The RMSE, MAE, SSE, and MSE values are much lower for the multi-gene GP model which clearly proves that this model is predicting the temperatures with a high degree of accuracy.

7 Conclusion and Future Work

In this paper, temperature forecasting of Mumbai, India is presented. We explored the use of parametric and non-parametric techniques to solve the temperature modeling problem. ANN is used to build a nonlinear structure of the temperature model. Meanwhile, Multi-gene Genetic Programming is used to derive a mathematical formula for the temperature. A comparison between these two models and the traditional linear regression model is presented. Multi-gene GP model clearly outperforms the other two techniques. We plan to extend this research to cover other trends of soft computing methods.

Table 4: The generated five individual genes/model

Term	Value
Bias	367.0
Gene 1	$-0.343x_3 - 0.685$
Gene 2	$0.00676x_1x_2 - 0.00338x_3 + 0.00338x_2^2$
Gene 3	$-0.564x_2$
Gene 4	$-12.6x_1$
Gene 5	$0.014x_3 - 0.014x_2 - 0.014x_1 - 0.014x_1x_2 + 0.014x_1x_3$

Table 5: Evaluation results of the developed Regression, ANN and Multi-gene GP Models

	Regression		ANN		GP	
	Training	Testing	Training	Testing	Training	Testing
R^2	0.84128	0.76612	0.87204	0.73796	0.86841	0.83262
RMSE	0.791	0.8692	0.71023	0.92005	0.72023	0.73533
MAE	0.61668	0.69809	0.54692	0.7739	0.52862	0.58582
SSE	159.55	82.35	128.63	92.267	132.27	58.937
Max. Absolute Error	3.2616	2.3797	2.2445	1.9349	3.7404	1.989
MSE	0.62569	0.75551	0.50442	0.84649	0.51873	0.54071
No. of samples	255	109	255	109	255	109

References

[1] K. Abhishek, A. Khairwa, T. Pratap, and S. Prakash, A stock market prediction model using artificial neural network, Proceedings of the Third International Conference on Computing Communication and Networking Technologies, 1-5, 26-28 July 2012.

[2] S. Khajure and S. Mohod, Future weather forecasting using soft computing techniques, 1st International Conference on Information Security and Privacy, **78**, 402-407, 2016.

[3] A. F. Sheta and A. Mahmoud, Forecasting using genetic programming, *Proceedings of the 33rd Southeastern Symposium on System Theory*, Mar 2001, 343-347.

[4] F. A. Ruslan, A. M. Samad, Z. M. Zain, and R. Adnan, Modelling flood prediction using radial basis function neural network (RBFNN) and inverse model: A comparative study, Proceedings of the 2013 IEEE International Conference on Control System, Computing and Engineering, Nov 2013, 577-581.

[5] S. H. Elsafi, Artificial neural networks (ANNs) for flood forecasting at Dongola station in the river Nile, Sudan, Alexandria Engineering Journal, **53**, no. 3, 655-662, 2014.

[6] D. R. Nayak, A. Mahapatra, and P. Mishra, A survey on rainfall prediction using artificial neural network, International Journal of Computer Applications, **72**, no. 16, 32-40, June 2013.

[7] K. Namitha, A. Jayapriya, and G. S. Kumar, Rainfall prediction using artificial neural network on map-reduce framework, Proceedings of the Third International Symposium on Women in Computing and Informatics, New York, NY, USA: ACM, 2015, 492-495.

[8] Mislan, Haviluddin, S. Hardwinarto, Sumaryono, and M. Aipassa, Rainfall monthly prediction based on artificial neural network: A case study in tenggarong station, east kalimantan - indonesia, Proceedings of the international Conference on Computer Science and Computational Intelligence (ICCSCI 2015), **59**, 142-151, 2015.

[9] M. Akbari Fard, H. Hadadi, and A. Tavakoli Targhi, Fruits and vegetables calorie counter using convolutional neural networks, Proceedings of the 6th International Conference on Digital Health Conference, 121-122, 2016.

[10] A. Mellit, Artificial intelligence technique for modelling and forecasting of solar radiation data: A review, Int. J. Artif. Intell. Soft Comput., **1**, no. 1, 52-76, Nov. 2008.

[11] G. Notton, C. Paoli, and S. Diaf, Estimation of tilted solar irradiation using artificial neural networks, Energy Procedia, Proceedings of International Conference mediterranean Green Energy Forum (MGEF-13), **42**, 33-42, 2013.

[12] T. Ren, S. Liu, G. Yan, and H. Mu, Temperature prediction of the molten salt collector tube using bp neural network, IET Renewable Power Generation, **10**, no. 2, 212-220, 2016.

[13] M. Wann, D. Yen, and H. J. Gold, Evaluation and calibration of three models for daily cycle of air temperature, Agricultural and Forest Meteorology, **34**, no. 2, 121 - 128, 1985.

[14] E. Kovač-Andrić, A. Sheta, H. Faris, and M. Š. Gajdošik, Forecasting ozone concentrations in the east of Croatia using nonparametric Neural Network

Models, Journal of Earth System Science, **125**, 997-1006, July 2016.

[15] J. Zurada, Introduction to Artificial Neural Systems. St. Paul, MN, USA: West Publishing Co., 1992.

[16] A. Sheta and H. Faris, Improving production quality of a hot rolling industrial process via genetic programming model, International Journal of Computer Applications in Technology, **49**, no. 3/4, 2014, special Issue on: "Computational Optimisation and Engineering Applications".

[17] M. Alkasassbeh, A. F. Sheta, H. Faris, and H. Turabieh, Prediction of PM10 and TSP air pollution parameters using artificial neural network autoregressive, external input models: A case study in salt, jordan, Middle-East Journal of Scientific Research, **14**, no. 7, pp. 999-1009, 2013.

[18] S. T. A. Niaki and S. Hoseinzade, Forecasting s&p 500 index using artificial neural networks and design of experiments, Journal of Industrial Engineering International, **9**, no. 1, 1-9, 2013.

[19] A. F. Sheta, D. Rine, and S. Kassaymeh, Software effort and function points estimation models based radial basis function and feedforward artificial neural networks, International Journal of Next-Generation Computing (IJNGC), **6**, no. 3, 2015.

[20] M. Zakerinia and S. F. Ghaderi, Short term wind power forecasting using time series neural networks, Proceedings of the 2011 Emerging M&S Applications in Industry and Academia Symposium, San Diego, CA, USA: Society for Computer Simulation International, 17-22, 2011.

[21] A. Sheta, H. Faris, A. Rodan, E. Kovač-Andrić, and A. M. Al-Zoubi, Cycle reservoir with regular jumps for forecasting ozone concentrations: two real cases from the east of croatia, Air Quality, Atmosphere & Health, **11**, no. 5, 559-569, Jun 2018.

[22] L.Ljung, System Identification - Theory For The User. Prentice Hall, 1999.

[23] J. Koza, Evolving a computer program to generate random numbers using the genetic programm paradigm, Proceedings of the Fourth Internatic Conference on Genetic Algorithms. Mor Kaufmann, La Jolla,CA, 1991.

[24] J. R. Koza, Genetic Programming: On Programming of Computers by Means of Nat Selection. Cambridge, MA, USA: MIT Press, 199

[25] A. E.-A. Hussian, A. Sheta, M. Kamel, M. Telba and A. Abdelwahab, Modeling of a winding macl using genetic programming, in Proceedings of the 2 Congress on Evolutionary Computation. La J Marriott Hotel La Jolla, California, USA: IEEE Pı 6-9 July 2000, 398-402.

[26] A. F. Sheta, P. Rausch, and A. S. Al-Afeef, A monitoring and control framework for lost foam casting manufacturing processes using genetic programming, Int. J. Bio-Inspired Comput., **4**, no. 2, pp. 111-118, Jun. 2012.

[27] H. Faris, A. Sheta, and E. Öznergiz, Modelling hot rolling manufacturing process using soft computing techniques, International Journal of Computer Integrated Manufacturing, **26**, no. 8, 762-771, 2013.

[28] H. Faris and A. F. Sheta, Identification of the tennessee eastman chemical process reactor using genetic programming, International Journal of Advanced Science and Technology, **50**, 121-140, 2013.

[29] A. A. Sharief and A. Sheta, Developing a mathematical model to detect diabetes using multigene genetic programming, International Journal of Advanced Research in Artificial Intelligence, **3**, no. 10, 54-59, 2014.

[30] A. F. Sheta, H. Faris, and M. Alkasassbeh, A genetic programming model for S&P 500 stock market prediction, International Journal of Control and Automation, **6**, no. 5, 303-314, 2012.

[31] A. Sheta, H. Faris, and M. Alkasassbeh, A genetic programming model for S&P 500 stock market prediction, International Journal of Control and Automation, **6**, 303-314, 2013.

[32] A. F. Sheta and H. Faris, Influence of nitrogen-di-oxide, temperature and relative humidity on surface ozone modeling process using multigene symbolic regression genetic programming, International Journal of Advanced Computer Science and Applications, **6**, no. 6, 2015.

[33] S. Aljahdali and A. F. Sheta, Evolving software effort estimation models using multigene symbolic regression genetic programming, International Journal of Advanced Research in Artificial Intelligence, **2**, no. 12, 52-57, 2013.

[34] D. P. Searson, D. E. Leahy, and M. J. Willis, GPTIPS: an open source genetic programming toolbox for multigene symbolic regression, Proceedings of the International Multiconference of Engineers and Computer Scientists, Hong Kong, 17-19 March, 77-80.

[35] S. Baboo and I. K. Shereef, An efficient weather forecasting system using artificial neural network, International Journal of Environmental Science and Development, **1**, no. 4, 321-326, 2010.

Alaa Sheta is Assistant Professor at the Department of Computing Sciences, Texas A&M University-Corpus Christi, TX, USA. He received his B.E., M.Sc. degrees in Electronics and Communication Engineering from the Faculty of Engineering, Cairo University in 1988 and 1994, respectively. He received his Ph.D. from the Computer Science Department, School of Information Technology and Engineering, George Mason University, Fairfax, VA, the USA in 1997. He received the Best Poster Award from the SGAI International Conference on Artificial Intelligence, Cambridge, the UK in December 2011 for his publication on Quality Management of Manufacturing Processes. He also published two books in the area of Landmine Detection and Classification and Image Reconstruction of a Manufacturing Process by LAP LAMBERT Academic Publishing. He is also the co-editor of the book entitled,"Business Intelligence and Performance Management - Theory, Systems, and Industrial Applications" by Springer Verlag, United Kingdom, published in March 2013. Dr. Sheta published more than 130 journal and conference papers. His main focus is on machine learning and its application in the domain of industrial process modeling, software reliability modeling, software cost estimation, automation, medical/non-medical imagining, and biomedical applications.

Ajay Katangur is a Professor and the Computer Science Department Chair at Missouri State University, USA. Katangur worked at Texas A&M University-Corpus Christi, USA for 14 years before joining Missouri State University in Fall 2018. Dr. Katangur received his MS and Ph.D. degrees in computer science from Georgia State University, USA. Katangur's current research focuses is on computer networks, security, cloud computing, and unmanned aerial systems. He has published several journal and conference papers in these areas. He currently serves as editor for two journal publications and is a reviewer for several journal and international conferences. Katangur's research has been primarily funded by National Science Foundation (NSF) and TRDF Program Grants for Innovative Research, TAMUCC, TX.

Abdel Karim Baareh is an associated professor with the Computer Science Department, Al-Balqa Applied University, Ajloun College, Jordan. He is the Chairman of the Applied Science Department at Ajloun College since 2009. He received his B.Sc degree in Science from Mysore University India in 1992, completed a Post Graduate Diploma in Computer Application from Mysore University India in 1993. He received his Master in Computer Application from Bangalore University India in 1999. He received his Ph.D. in Informatics from Damascus University, Syria in 2008. His research interest includes Neural Networks, Fuzzy Logic, Image Processing and Genetic Algorithms.