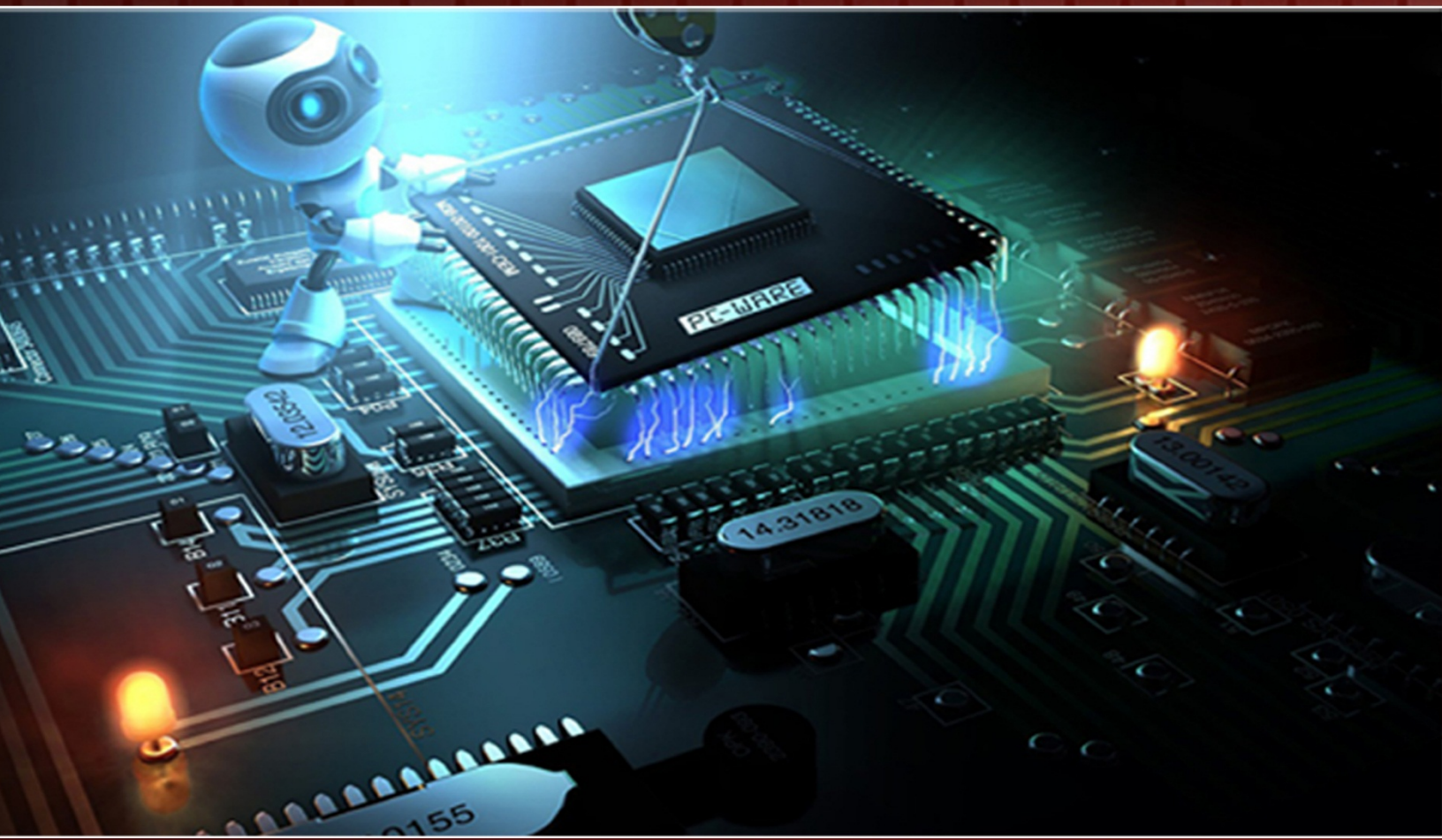
ISSN: 2054-7390

Volume 5 Issue 5, Oct 2017

Transactions on Machine Learning and Artificial Intelligence



Society for Science and Education
UNITED KINGDOM

TABLE OF CONTENTS

EDITORIAL ADVISORY BOARD

DISCLAIMER

All the contributions are published in good faith and intentions to promote and encourage research activities around the globe. The contributions are property of their respective authors/owners and the journal is not responsible for any content that hurts someone's views or feelings etc.

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM
TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE
ISSN 2054-7390

Optimization of Fuzzy Neural Networks using Mine Blast Algorithm for Classification Problem

Mohd Najib Mohd Salleh, Kashif Hussain

Faculty of Computer Science and Information Technology
Universiti Tun Hussein Onn Malaysia Parit Raja, Batu Pahat, Johor, Malaysia.
najib@uthm.edu.my, hi130019@siswa.uthm.edu.my

ABSTRACT

The integration of Fuzzy Neural Networks (FNNs) with optimization techniques has not only solved the issues "black box" in Artificial Neural Networks (ANNs) but also has been effective in a wide variety of real-world applications. Adaptive Neuro-Fuzzy Inference System (ANFIS) still needs effective parameter training and rules optimization methods to perform efficiently when the number of inputs increases. ANFIS accuracy depends on the parameters it is trained with and the drawbacks of gradients based learning of ANFIS using gradient descent and least square methods in two-pass learning algorithm. Many researchers have trained ANFIS parameters using metaheuristic, however, very few have considered optimizing the ANFIS rule-base. We propose an effective technique for optimizing ANFIS rule-base and training the network parameters using newly Accelerated modified MBA (AMBA) to convergence the speed during exploitation phase. The AMBA optimized ANFIS was tested on real-world benchmark classification problems like Breast Cancer, Iris, and Glass. The AMBA optimized ANFIS has also been employed to model real datasets. The performance of the proposed AMBA optimized ANFIS model was compared with the ones optimized by Genetic Algorithm (GA), Particle Swarm Optimization (PSO), MBA and Improved MBA (IMBA), respectively. The results show that the proposed AMBA optimized ANFIS achieved better accuracy with optimized rule-set in less number of function evaluations. Moreover, the results also indicate that AMBA converges earlier than its other counterparts.

Keywords: ANFIS; neuro-fuzzy; fuzzy system; Mine Blast Algorithm (MBA); optimization

1 Introduction

Various optimization techniques and learning algorithms have been used with Fuzzy Neural Network (FNN) to reduce the cost of learning, and achieving higher accuracy at the same time. Most of these algorithms require substantial gradient information, and may become difficult or unstable when the objective function and the constraints have multiple or sharp peaks. To improve learning capability of FNN, many researchers have optimized the training process by various metaheuristic algorithms like Genetic Algorithm (GA), Particle Swarm Optimization (PSO) or Ant Colonies, which are modeled on swarm intelligence [1, 2].

DOI: 10.14738/tmlai.55.3464
Publication Date: 26th July, 2017
URL: http://dx.doi.org/10.14738/tmlai.55.3464

In recent years, Adaptive Neuro-Fuzzy Inference System (ANFIS) has gained more attraction than other types of fuzzy expert systems. This is because the results obtained from ANFIS are sturdier than other fuzzy systems [3]. After designing and testing the ANFIS systems, Neshat *et. al.,* [4] found that ANFIS results were comparatively better than other fuzzy expert systems. However, when designing ANFIS based models, the major concern of researchers is to train its parameters efficiently so that enhanced accuracy can be achieved. On the other hand, Liu, Leng [2] and Petković *et. al.,* [5] also agree that tuning membership function (MF) parameters is more complex than the consequent parameters.

Mine Blast Algorithm (MBA) is recently introduced by Sadollah *et. al.,* [6], which has outperformed GA, PSO, and their variants in terms of convergence speed and better optimal solutions. Sadollah *et. al.,* [7] improved MBA and called it Improved MBA (IMBA). This paper further accelerates its convergence speed by modifying exploitation phase and calling the new variant as Accelerated MBA (AMBA).

The following section briefly explains ANFIS and its learning mechanism. Section 3 presents MBA algorithm and the proposed AMBA is introduced, followed by ANFIS training using AMBA in Section 4. The experimental results are given in Section 5. Section 6 makes conclusion of this study.

2 The Concept of ANFIS

Jang introduced ANFIS architecture in 1993 [8], which can approximate every plant with adequate number of rules using adaptive technique to assist learning and adaptation [2, 9]. Figure 1 shows five layer ANFIS architecture:

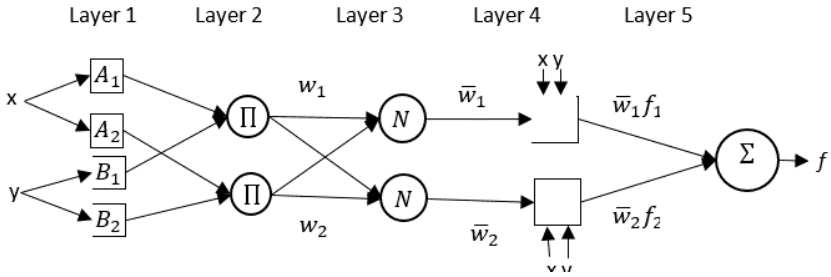


Figure 1. ANFIS Architecture (M. A. Shoorehdeli et al., 2009)

Layer 1: Every node i in this layer is adaptive MF, i.e., Triangle, Trapezoidal,

Gaussian, or generalized Bell function.

$$O_{1,i} = \mu_{A_i}(x), \qquad i = 1,2 \tag{1}$$

$$O_{1,i} = \mu_{B_{i-2}}(y), \qquad i = 3,4 \tag{2}$$

Layer 2: These nodes are fixed and represent simple product ∏ to calculate firing

strength of a rule.

$$O_{2,i} = w_i = \mu_{A_i}(x)\mu_{B_i}(y), \qquad i = 1,2 \tag{3}$$

In this paper, the rules are generated using grid partitioning. The number of rules is m^n where m is the number of MFs in each input variable and n is the number of inputs to ANFIS.

Layer 3: Each node is fixed and represented as N in Figure 1. It normalizes firing strength of a rule from previous layer by calculating the ratio of the ith rule's firing strength to the sum of all rules' firing strength.

$$O_{3,i} = \overline{w}_i = \frac{w_i}{w_1 + w_2}, \qquad i = 1,2. \tag{4}$$

where $\overline{w}$ is referred to as normalized firing strength of a rule.

Layer 4: These are consequent nodes which are identified during training. Each node has node function

$$f_i = p_i x + q_i y + r_i$$

$$O_{4,i} = \overline{w}_i f_i = \overline{w}_i (p_i x + q_i y + r_i), \qquad i = 1,2. \tag{5}$$

Layer 5: It is output node which does the summation of rules output

$$O_{5,i} = \sum_{i=1}^{2} \overline{w}_i f_i = \frac{\sum_{i=1}^{2} \overline{w}_i f_i}{w_1 + w_2} \tag{6}$$

ANFIS learns by adjusting all modifiable parameters using gradient descent (GD) and least squares estimator (LSE). The parameter update process uses a two pass learning algorithm as presented in Table 1.

Table 1. Two Pass Hybrid Learning Algorithm for ANFIS

	Forward Pass	Backward Pass
Antecedent Parameters	Fixed	GD
Consequent Parameters	LSE	Fixed
Signals	Node Outputs	Error Signals

In forward pass, consequent parameters are updated by LSE, and in backward pass, the premise parameters are updated using GD. Backward pass is influenced by back propagation (BP) algorithm of ANN which has the drawback to be likely trapped in local minima [9]. This paper explores the applicability of MBA after modifying its exploitation phase; calling it Accelerated MBA (AMBA). This paper presents an overview of MBA in the following section. The proposed AMBA is explained in the next section.

3 Mine Blast Algorithm – MBA

Sadollah et. al., [6] recently developed MBA as an optimization technique for handling complex optimization problems. This method is derived from the idea of explosion of mines, and thrown shrapnel pieces explode other mines by colliding with them. The most explosive mine (min or max f(x)) located at the optimal point X* is considered as optimal solution. The solution individuals in a population are shrapnel pieces (Ns).

The initial population is created by first shot point, represented by X_0^f

$$X_0 = LB + rand \times (UB - LB) \tag{7}$$

where X0, LB and UB are the generated first shot point, lower, and upper bounds of the problem, respectively. rand is a uniformly distributed random number between 0 and 1. Explosion of a landmine generates shrapnel pieces Ns which collide with other landmine at location X_{n+1}. The user of MBA can decide to start with multiple first shot points.

$$X^f_{n+1} = X^f_{e(n+1)} + \exp\left(-\sqrt{\frac{m^f_{n+1}}{d^f_{n+1}}}\right) X^f_n, \qquad n = 0,2,3,\ldots \tag{8}$$

$$X^f_{e(n+1)} = d^f_n \times rand \times \cos(\theta), \quad n = 0,2,3,\ldots \tag{9}$$

$$d^f_{n+1} = \sqrt{(X^f_{n+1} - X^f_n)^2 + (F^f_{n+1} - F^f_n)^2} \tag{10}$$

$$m^f_{n+1} = \frac{F^f_{n+1} - F^f_n}{X^f_{n+1} - X^f_n}, \qquad n = 0,2,3,\ldots \tag{11}$$

where $X^f_{e(n+1)}$, d^f_{n+1} and m^f_{n+1} are the location of exploding mine, the distance and the direction of the thrown shrapnel pieces in each iteration, respectively. In (9), θ is the angle of the shrapnel pieces. F is the objective function value for the point X in (10) and (11).

In MBA, the user defined parameter, called exploration factor μ, allows to randomly search for optimal solutions at small and large distances using (12) and (13).

$$d^f_{n+1} = d^f_n \times (|randn|)^2, n = 0,2,3,\ldots \tag{12}$$

$$X^f_{e(n+1)} = d^f_{n+1} \times \cos(\theta), \qquad n = 0,2,3,\ldots \tag{13}$$

Sadollah, Yoo [7] improved MBA to find optimal cost design for water distribution systems. The exploitation phase, defined in MBA, is modified by IMBA, which focuses on the solution closest to the best one so far. IMBA modifies (8) as below:

$$X^f_{n+1} = X^f_{e(n+1)} + \exp\left(-\sqrt{\frac{1}{D}}\right) \times \{rand\} \otimes \{X_{Best} - X_{best-1}\}, \; n = 0,1,3,\ldots \tag{14}$$

In (14), the perception of direction is replaced by moving to the best solution. The exponential term in this equation improves the obtained exploded point by including information from current best solution X_{Best} and previous best solution X_{best-1} and Euclidean distances between them in m dimensions.

$$D = \left[\sum_{i=1}^{m} (X_{iBest} - X_{i(best-1)})^2\right]^{1/2} \tag{15}$$

Unlike MBA, distance between shrapnel pieces are reduced by (16), only when there is no change in the value of the cost function.

$$d_n^f = \frac{d_{n-1}^f}{\exp\left(\frac{k}{\alpha}\right)}, \qquad n = 1,2,3,\ldots \tag{16}$$

where α and k are reduction constants and iteration number index, respectively.

3.1 Accelerated Mine Blast Algorithm – AMBA

The modification in exploitation equation by IMBA improves results, more quicker results could be achieved by having distance between current exploded point $X_{e(n+1)}$ and current best solution so far. The proposed modifications in (14) and (15) are illustrated below:

$$X_{n+1}^f = X_{e(n+1)} + \exp\left(-\sqrt{\frac{1}{D}}\right) \times \{rand\} \quad \otimes\{X_{Best} - X_{e(n+1)}\}, \quad n = 1,2,3,\ldots \tag{17}$$

$$\mathrm{D} = \left[\sum_{i=1}^{m} (\mathrm{X_{iBest}} - \mathrm{X_{e(n+1)}})^2\right]^{1/2} \tag{18}$$

where D represents Euclidean distances between current best solution $\mathrm{X_{iBest}}$ and current point of explosion $X_{e(n+1)}$ in m dimensions. The proposed approach did not use information of previous best location; therefore it accelerated the convergence of the algorithm. Hence, this new variant of MBA is referred to as Accelerated MBA (AMBA). To validate its performance, when training ANFIS network on benchmark classification problems, the results are compared with MBA and IMBA.

3.2 ANFIS Training using AMBA

In this paper, AMBA is employed to tune premise and consequent parameters of ANFIS. Each shrapnel piece of mine in AMBA represents a set of parameters comprising of both the MF parameters and the consequent part of the fuzzy rule. The performance validation criterion mainly focused on three measures: optimized rule-set of ANFIS, accuracy of ANFIS, and convergence speed of optimization algorithms. Optimized rule-set consisted of the potentially contributing rules which were extracted from the overall knowledge-base of ANFIS. The accuracy was measured in terms of Mean Square Error (MSE) between actual and the desired output. The speed of convergence of optimization algorithms was measured in number of iterations.

The fitness is defined as mean squared error (MSE) between actual output and the desired output, it can expressed as:

$$MSE = \frac{\sum_{i=1}^{m} (O - O_m^t)^2}{m}, \tag{19}$$

where MSE, O, $\mathrm{O_m^t}$, and m are mean square error, ANFIS output, target output of mth training pair, and the size of training dataset, respectively.

The ANFIS network trained by AMBA algorithm is outlined as below Figure 2:

ANFIS Training using AMBA

[1]. **begin**
02: Initialize AMBA: N_s, $\mu, \alpha, maxItr$

02: Randomly initialize first shot point

03: **while**($maxItr$, or the stopping criterion is not met)

5: **for** n=1 to N_s (Each shrapnel contains all ANFIS parameters)

1 **if** $n < \mu$ **then** (Exploration phase)
○ Calculate the updated position of shrapnel pieces using (12) and (13)
8: **else** (Exploitation phase)
9: Calculate the position of exploded mines using (9)

10: Calculate the Euclidean distance between mines using (18)
11: Update the position of shrapnel pieces using (17)
12: **end if**
13: Evaluate function value of shrapnel pieces and update *CurrentBest* using (19)
14: Update θ

15: **next**
16: Reduce the distance of the shrapnel pieces adaptively using (16)
17: **next** explosion until stopping criterion
18: **end**

Figure 2. AMBA Algorithm

The datasets are partitioned into two sets: training and testing set. The partitioning of is performed randomly such that 75% reserved for training and 25% for testing purpose. For training ANFIS with standard MBA, IMBA and the proposed AMBA, Table 2 presents the initialization values.

Table 2. Specification of ANFIS and AMBA Algorithm MBA, IMBA, AMBA

Parameters	Value	ANFIS Parameters	Value
Number of first shot points ( )	1	Number of inputs	As per datasets mentioned in Table 2
Number shrapnel pieces ( )	15	Number of MFs for each input	3
Exploration Factor (		Type of MF	Guassian
Reduction Factor ( ) Maximum Iterations			
Objective Function	Mean Square Error (MSE)		

4 Experimental Results

This section provides the analysis and discussion on the performance of the proposed Accelerated Mine Blast Algorithm (AMBA) optimized ANFIS. The performance of the optimized ANFIS was evaluated in terms of accuracy while AMBA was evaluated based on convergence rate. The efficiency was evaluated using three real world classification problem datasets which were breast cancer, iris, and glass. These datasets were taken from University California Irvine Machine Learning Repository (UCIMLR) at Center for Machine Learning and Intelligent Systems. To validate superiority over other optimization methods, the proposed AMBA optimized ANFIS was compared with the one optimized by GA, PSO, MBA, and IMBA, respectively. For further evaluation, the ANFIS-based AMBA model was first trained and then tested on industry data acquired from SME Corporation Malaysia. The data was taken from SCORE (SME Competitiveness Rating for Enhancement), a software diagnostic system developed by SME Corporation Malaysia for ranking SMEs. The summarization drawn from the results is presented as follow.

Table 3 proves that the proposed AMBA outperformed other optimization algorithms in the list. AMBA optimized rule-set from 16 rules to 7 for achieving accuracy of 99.777%. Even though, the optimized rule-set of AMBA and IMBA was equal but AMBA achieved this rule-set in just 16 iterations as compared to IMBA did in 30. Moreover, the accuracy of IMBA was lesser than the proposed algorithm. This shows the proposed AMBA exhibited more speed of convergence than other algorithms. Here, MBA and PSO obtained 5 rules each in their optimized rule-set but the accuracy of PSO was better than MBA due to better membership functions parameters identified by PSO. GA performed least in this case as well. It was able to bring only 4 rules therefore losing accuracy.

Table 3: Summary of optimization algorithms' performances in Iris classification problem

CRITERIA	GA	PSO	MBA	IMBA	AMBA
Optimized Rule-Set	4	5	5	7	7
MSE	0.15533	0.089008	0.13505	0.034896	0.0044512
Accuracy %	92.233%	95.5496%	93.248%	98.255%	99.777%
Iterations To Converge	30	30	30	30	16

From the results shown in Table 4, it is worth noticing that AMBA performed better than GA, PSO, MBA, and IMBA. AMBA reduced the number of rules in the total rule-set of 512 rules to 58 and achieved 98.433% accuracy. It achieved highest accuracy among others optimization algorithms in just 9out of 30 iterations. PSO attained second best accuracy among others which was 97.636% with 24 rules. It consumed 23 iterations out of 30 to reach the target error for overall ANFIS output. MBA and IMBA consumed all 30 iterations and could not reach the target error. They demonstrated 97.347% and 97.086% accuracies with 59 and 62 rules, respectively. GA was once again lowest performer with 96.679% accuracy in this classification problem as well.

Table 4: Optimization algorithms' performances in Breast Cancer classification problem

CRITERIA	GA	PSO	MBA	IMBA	AMBA
Optimized Rule-Set	25	24	59	62	58
MSE	0.14643	0.047274	0.053058	0.058273	0.03134
Accuracy %	92.679%	97.636%	97.347%	97.086%	98.433%
Iterations To Converge	30	23	30	30	9

In Table 5, the performance of AMBA is compared with GA, PSO, MBA, and IMBA while training and optimizing ANFIS network. Even though AMBA and IMBA took same number of iterations (6) to meet target error tolerance (0.01) and both the algorithms brought 21 out of 512 rules in their optimized rule-set, but AMBA was able to achieve better accuracy than IMBA. MBA also reached error tolerance in 8 iterations unlike PSO and GA which consumed maximum iterations limit of 30. MBA, PSO, and GA reached the accuracy of 99.524%, 97.935%, and 97.837%, respectively, whereas they optimized the rule-set to 20, 13, and 7, respectively.

Table 5: Optimization algorithms' performances in Glass classification problem

CRITERIA	GA	PSO	MBA	IMBA	AMBA
Optimized Rule-Set	7	13	20	21	21
MSE	0.043267	0.041296	0.0095231	0.0087719	0.0027582
Accuracy %	97.837%	97.935%	99.524%	99.561%	99.862%
Iterations To Converge	30	30	8	6	6

As shown by the results in Table 6 below, the proposed AMBA optimized ANFIS network effectively than other optimization algorithms. It obtained optimum number of rules with higher accuracy in less number of iterations as compared to the original MBA, its variant IMBA, and GA and PSO as well. Table 4.3 reveals that AMBA retrieved 98 rules out of total 128 in forward pass with rule tolerance 0.0001. It achieved 99.764% which is higher than original MBA and its variant IMBA. As compared to the proposed AMBA, MBA and IMBA achieved less accuracy with less number of rules in more iterations; 99.681% with 96 rules and 99.613% with 91 rules, respectively, with 30 iterations each. However, they performed better than GA and PSO. GA performed least in this problem and could only reach to 42.251% with 34 rules. PSO showed better accuracy than GA, reaching 90.735% with 58 rules. Both the optimizers spent30 iterations.

Table 6: Optimization algorithms' performances in SME Classification problem

CRITERIA	GA	PSO	MBA	IMBA	AMBA
Optimized Rule-Set	29	58	96	91	98
MSE	0.4321	0.1853	0.0063873	0.0077464	0.0047279
Accuracy %	78.395%	90.735%	99.681%	99.613%	99.764%
Iterations To Converge	30	30	30	30	16

5 Conclusion

A new variant of MBA, so called AMBA, has been proposed in this paper. AMBA is integrated with ANFIS for training the premise and consequent parameters to achieve minimum error difference between the desired and actual output. The findings from the results, obtained from several experiments conducted on real-world benchmark problems, indicate that the proposed AMBA can efficiently train ANFIS network. The proposed AMBA reduces the computational cost by eliminating the cost of maintaining the previous best solution. It only uses current best solution and the available candidate solution. Because of this modification, AMBA shows the ability of converging quicker as compared to the standard MBA, and the improved variant IMBA. Even though, MBA is a potential optimization algorithm, it can still be improved by modifying exploitation phase and distance reduction policy.

ACKNOWLEDGEMENTS

The authors would like to thank Universiti Tun Hussein Onn Malaysia (UTHM) for supporting this research under Postgraduate Incentive Research Grant Vote No.U62.

REFERENCES

[1] H. M. I. Pousinho, V. M. F. Mendes and J. P. S. Catalão, *A hybrid PSO–ANFIS approach for short-term wind power prediction in Portuga*l, Energy Conversion and Management, vol. 52, no. 1, 2011, pp. 397-402.

[2] P. Liu, W. Leng and W. Fang, *Training anfis model with an improved quantum-behaved particle swarm optimization algorithm*, Mathematical Problems in Engineering, 2013.

[3] O. Taylan and B. Karagözoğlu, *An adaptive neuro-fuzzy model for prediction of student's academic performance*, Computers & Industrial Engineering, vol. 57, no. 3, 2009, pp. 732-741.

[4] M. Neshat, A. Adeli, A. Masoumi and M. Sargolzae, "*A Comparative Study on ANFIS and Fuzzy Expert System Models for Concrete Mix Design*", 2011.

[5] D. Petković, *Adaptive neuro-fuzzy prediction of grasping object weight for passively compliant gripper*, Applied Soft Computing, 2014.

[6] A.Sadollah, A. Bahreininejad, H. Eskandar and M. Hamdi, *Mine blast algorithm for optimization of truss structures with discrete variables*, Computers & Structures, vol. 102, 2012, pp. 49-63.

[7] A. Sadollah, D. G. Yoo and J. H. Kim, *Improved mine blast algorithm for optimal cost design of water distribution systems* , Engineering Optimization, 2014 (ahead-of-print): pp. 1-17.

[8] J. S. R. Jang, *ANFIS: adaptive-network-based fuzzy inference system*, Systems, Man and Cybernetics, IEEE Transactions on, vol. 23, no. 3, 1993, pp. 665-685.

[9] K. Ananda and M. Punithavalli, *Efficient cancer classification using fast adaptive neuro -fuzzy inference system (FANFIS) based on statistical techniques*, IJACSA) International Journal of Advanced Computer science and Applications, Special Issue on Artificial Intelligence, 2011, pp. 132-137.

[10] J. Derrac, S. GARCia, L. Sanchez and F. Herrera, *KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework*, 2015.

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM
TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE
ISSN 2054-7390

Theory Of Dynamic Interactions: Synthesis

Gabriel Barceló
Advanced Dynamics C. B.
gabarce@iiies.es

ABSTRACT

In this text, we carry out a brief summary of the Theory of Dynamic Interactions developed by the author in the new book: New Paradigm in Physics. Certain keys are provided to better understand the dynamic hypotheses proposed, conclusions are drawn from the studies carried out and ideas for future development are proposed in this area. The author believes that with this new model proposed, the understanding of our observational universe will be facilitated, as well as that of the physical phenomena we notice.

This paper is based on ideas and excerpts from other texts by the author or his team, which are mentioned in the references. Nevertheless, some paragraphs may not be duly referenced.

Keywords: Paradigm in Physics; Rotational Motion; Rotational Dynamics; Dynamic Interactions; Inertial Fields; Generalization of mechanics .

1 Keys to understand the Theory of Dynamic Interactions

Taking as a starting point the aporia between rotation and orbit, a Theory of Dynamic Interactions, has been developed by Advanced Dynamics team (See figure 1), that I explain on the book: New Paradigm in Physics[1]. The writing of this book is described by Professor Merino[2], and it incorporates a prologue of great interest of Francisco Dalby[3].

This theory is based on the inertial incapacity of matter to vector add, under certain conditions, the resultant angular momenta and, in general the angular magnitudes of the rotating bodies. The dynamics of rigid solid bodies is not a closed discipline, particularly in the field of rotational dynamics. From the observation of bodies with intrinsic rotation in our universe, our research group proposed new dynamic hypotheses that explain the behavior observed when these bodies are subject to new simultaneous non-coaxial rotations.

The findings of the Bernoulli's, Riccati and especially D´Alembert and Euler, followed by that of Lagrange, Laplace and Hamilton, meant that from the 19th century Mechanics could be considered a mathematically defined and fully modelled science. However, if we analyses Rotational Dynamics specifically, we cannot be satisfied or share that same approach.

It was in fact Euler who established the equations of motion of rotating solid bodies4. His studies on rotational dynamics culminated in the publication of his work Theoria motus corporum solidorum seu rigidorum5. In said work, he expresses the rotation of the main axes of the body in relation to the other

DOI: 10.14738/tmlai.55.3344
Publication Date: 8th August, 2017
URL: http://dx.doi.org/10.14738/tmlai.55.3344

three fixed axes, through the use of three variable angles, which determine new angular coordinates, and through very similar formulas to those currently known.

The orientation of a rigid solid body can be determined from Euler's angular coordinates. If these coordinates are known depending on time, we will be able to deduce the temporal evolution of its orientation. These are Euler's equations, based on which we should be able to determine the trajectory of a body subjected to multiple momentums.Euler's equations are to rotational dynamics what Newton's second law is to translational dynamics. The problem is that those equations, which are consistent and formally correct, do not allow a general solution apparently both from the physical and purely mathematical point of view, because they generate complex equations that in most cases can only be solved approximately.

Figure 1. Advanced Dynamics Team in 2008: From left to right: F. Dalby, M. Cano, R. Gómez-Olea (†), G. Barceló, J. Cano, A. Álvarez and E. López.

Even if we analyze the trajectories calculated with these equations, we note that not correspond to observable reality. In figure 2 we see estimated trajectory with the formulation of Classical Mechanics, and on figure 3 the real trajectory observed.

By observing in nature the constancy of the relationship between orbiting and intrinsic rotation, Gabriel Barceló deduced the principle that: Everything that orbits, rotates7; or rather, everybody that moves through an orbit simultaneously rotates on an intrinsic axis. He deduced this principle from observing the planetary system, the rings of Saturn and also the behavior of the spinning top.

He understood, however, the need for empirical checks to confirm or rectify the new dynamic hypotheses deduced from the aforementioned principle and, where appropriate, to be able to explain that behavior by formulating a new dynamic theory that would simultaneously resolve other Rotational Dynamics phenomena and generalize inertial phenomena. The Theory of Dynamic Interactions allows developing a specific dynamic for rotating solids, submitted to successive torques in which the sequence of the forces' action and its behavior do not coincide exactly with the laws of classic mechanics.

The statement of behavior laws of mobiles in space, and therefore the development of the Theory of Dynamic Interactions, has been carried out after experimentally verifying the previsions of this theory and the real inertial behavior of the rotating matter.

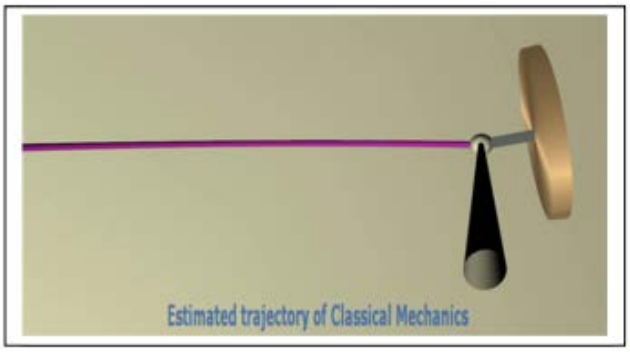


Figure 2. Estimated trajectory of Classical Mechanics.

The incorporation of this theory in the field of rational mechanics is achieved, not only by speculative and mathematical reasons, as well as being a result of the realized experimental tests, and even through the extrapolation of the experimental behavior with specific tests.

In line with these previous experiments we have been able to observe that the knowledge of these behavior laws11 will allow the development of new dynamic technologies, unknown to date.

Through this model different results are obtained, for certain assumptions, basing ourselves exclusively on the new interpretation of the coupling concept. We propose new criteria in the composition or superposition of the motions originated by the acting forces.

I strongly believe that this exposure corresponds to the real dynamic behavior of bodies subjected to acceleration by rotation.

Up to now, I have not been refuted by any logical argument or experimental evidence that allows us to suppose that my proposal is wrong.

I proposed that from a relativistic point of view, an intrinsic rotation can be seen as a fixed mobile and a rotation of the space of events which contains it. In this assumption, different experiments can be carried out, and the resulting observations be noted. By way of illustration we could

propose:

1 Multiple rotations. A mobile exposed to successive non coaxial torques might react with two simultaneous non coaxial rotations.
2 Inertial reactions of the mass (Gyroscopic momentum). The successive action of noncoaxial torques, generates an inertial reaction, which does not correspond to Newton's laws, and that is not structured in classical mechanics.

From these experimental references, we can reiterate that we infer the existence of a different rotational dynamics, non-Newtonian, necessary for the identification of the behavior of rotating bodies, when exposed to new non-coaxial stimulations, and the behavior of which, in many cases, nowadays is considered to be anomalous, paradoxical or chaotic, because the laws that we have at our disposal do not allow to identify and predetermine it. Based on the Principle of Conservation of the Momentum, we can infer that the field of inertial reactions generated in the rotating space by a new non coaxial momentum, upon a moving body with a rotational movement ω and an inertial momentum I upon that rotation axis, and thus with an angular momentum *L,* will oblige the moving body to acquire a orbitation rate Ω.

This orbitation rate Ω can be observed simultaneously to the initial ω, which stays constant within the body. Further, and as discriminating hypothesis, in the case of transfer movement of the body, we propose the dynamic hypothesis of the linear speed field, coupling to the anisotropic field of inertial speeds created by the second non-coaxial momentum, obtaining as resultant movement, a simultaneous orbiting with the intrinsic rotation of the moving body. This new orbiting movement, generated by a non-coaxial momentum, defines itself through the rotation of the speed vector, the latter being kept constant in module.

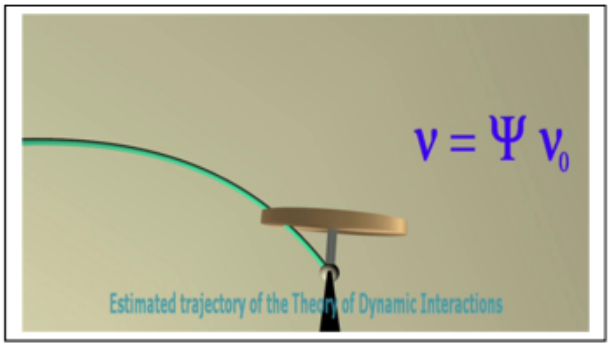


Figure 3. Real trajectory of the Pendulum of Dynamic Interactions.14

As a result of this analysis we obtained as motion equation:

$$\vec{v} = \vec{\vec{\Psi}}\ \vec{V}_0 = \begin{pmatrix} \cos M' t/I\omega & -\text{sen}\, M' t/I\omega & 0 \\ \text{sen}\, M' t/I\omega & \cos M' t/I\omega & 0 \\ 0 & 0 & 1 \end{pmatrix} \vec{V}_0$$

The rotation operator $\Rightarrow_{\Psi}$ with angular displacement, transforms the initial velocity $\underset{V_0}{\rightarrow}$ into $\underset{\nu}{\rightarrow}$, both of which are situated on the same plane. We find that the rotation operator is $\Rightarrow_{\Psi}$ perpendicular to the velocity (See Figure 4.) and a function of the sine or cosine Ωt, which clearly indicates the relationship between the angular velocity of the orbit Ω, torque $\underset{M}{\rightarrow}$ and the initial angular velocity $\underset{\nu}{\rightarrow}$ (…)

Accordingly, we have obtained a simple mathematical relationship between the initial angular velocity of the body and its translation velocity . (…) Dynamic effects can be associated with velocity and a clear mathematical correlation between rotation and translation. This mathematical coupling enables us to identify a physical relationship between the transfers of rotational kinetic energy and translational kinetic energy and vice versa.

With this analysis, we believe to have given a full answer to our initial aporia between constant turn and orbital movement, being justified the coincidence in nature where mobiles are rotating and orbiting simultaneously, based on the peculiar inertial behavior of mass.

Throughout our exposition, we have tried to avoid the concept of inertial force and we have substituted it by inertial reactions. No existence of real forces can be inferred from the observation of matter. Although we can infer the existence of the non-homogeneous ddistributions of velocities whose derivative

generates an inertial field of accelerations that neither is homogeneous, but can be interpreted as a field of inertial forces.

Quite a number of examples can be thought of for checking these dynamic hypotheses, which would allow us to interpret many assumptions in nature, which still remain unexplained.

This new non-inertial rotational dynamics based on the Theory of Dynamic Interactions, we have developed in laws and corollaries, allowing a number of new, unknown scientific and technological applications. The Theory of Dynamic Interactions was first exposed at the XXX Physics Biennale held in Orense (Spain) in 2005,16 and published in the book: The flight of the Boomerang, with prologue of Professor Garcia Moliner17, and analyzed later by Almudena Martin.

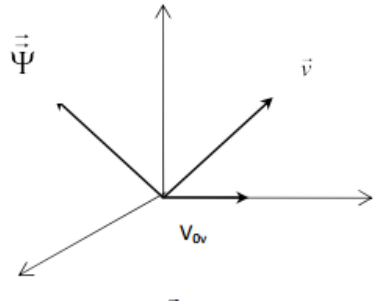


Figure 4. The rotation operator $\underset{\Psi}{\Rightarrow}$ will be perpendicular to the successive velocities $\underset{v}{\rightarrow}$, all of equal module and tangential to the path of the moving object.

2 Generalization of mechanics

We live in a physical world based on the fact that everything rotates, though for the simple observer it seems as something is just moving there in the sky. In reality, we are living in a physical world based on the intrinsically movement on axes of symmetry: The planets rotate on their axis and orbit, as well as their satellites, the planetary systems and galaxies.

As we expressed, the classical mechanics has been formulated for inertial reference frames, and not for rotating spaces of events. Nevertheless, it is possible to think of a new mechanics for any type of space, adding their inertial reactions and defining a dynamics of inertial fields. Thus, the not inertial phenomena would also be structured, and get incorporated in a unified mechanics.

In order to incorporate the inertial phenomena into the structure of physical knowledge, it is necessary to analyze the motion in non-holonomous coordinates and the resulting axial reactions, in the understanding that a mechanics, as the classical mechanics, based on holonomous coordinates and polar reactions, will only represent a limited and partial view of nature.

We have already said that the proposed generalization does not say that classical mechanics is obsolete or wrong, but simply that it is partial and limited, as it refers to the specific assumption of inertial systems. We wish and are able to be more ambitious, looking for more general dynamic laws, which establish the behavior of moving bodies when rotating, or even when they are exposed to multiple non-coaxial rotations of the space of events.

The Theory of Dynamic Interactions generalizes the concept of gyroscopic momentum, and of other inertial phenomena, incorporating them into the unified structure of a new non inertial rotational dynamics.

According to the defended Theory of Dynamic Interactions (See figure 5 and video referred), we can conceive a universe in a constant dynamic balance, in which a force momentum, with a zero resultant, will generate, as long as it works, a movement of constant orbiting, within a closed path.

The importance of this mathematical model is obvious, we have already said that it is not only the forces leading players, but also the momentums of forces which, while staying constant, will generate orbiting and constantly recurrent movements, generating a system in dynamic balance, and not in unlimited expansion.

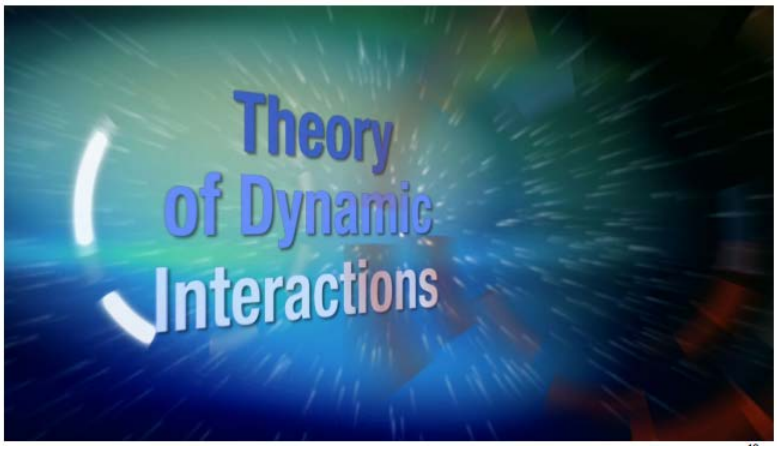


Figure 5. Numerous experimental tests can be observed in videos and can be easily repeated

At the 1930 annual meeting of the Society of German Scientists and Physicians, the Czech mathematician Kurt Gödel tentatively announced the first expression of his incompleteness theorem: There are things we know are true, but we cannot prove.

In my opinion, in our case: There are things we know are true and can prove!!!

3 Justification

The theory justifies and explains many scientific unknowns that could not be understood using classical mechanics. TDI fully justifies the flat orbits Kepler proposed: Bodies provided with intrinsic angular momentum and submitted to one single torque will not orbit in a space formed by possible spherical paths, but in a plane containing the torque.

But this feature is not stated, but it is implicit in Kepler's first law: The orbiting motion is configured in a plane. On the contrary, they are not expressly specified in Newton´ laws.

However, our theory specifically proposed: The mobile's path will be situated on the plane determined by the vector of the initial velocity $\overrightarrow{V}_0$ and an axis parallel to the momentum $\overrightarrow{M}$

And also on the third corollary of the tenth Law, in absence of nutation: Rigid solid bodies equipped with intrinsic angular momentum and translation speed, subject to momenta of constant force, will describe

closed paths in a plane determined by the vector of initial velocity and that will possess an axis parallel to the acting momentum.

The resultant movement, even if we are situated in a three dimensional space, will be a flat orbit, and all the interaction movements resultant of the same torque, will be situated on a plane containing the torque.

This means that, if the variables are kept stable, the orbital movement will be flat. But neither Kepler and Newton laws mentioned in the coincidence of the rotation and orbiting in our universe.

In the laws of classical mechanics, there does not exist a mathematical correlation which relates the movements of orbit and rotation of the planets around the Sun. However, the question of theexistence of a physical relation between both movements, which has to date not been mathematically shown can be raised.

Also to justify how rings with multiple satellites such as the ones of Saturn22 (See figure 6), are formed many times in our solar system. Newton neither explains the reason for the rings of Saturn nor many systems of flat rings in our solar system like the asteroid belt, the Kuiper belt or the diffused disc.

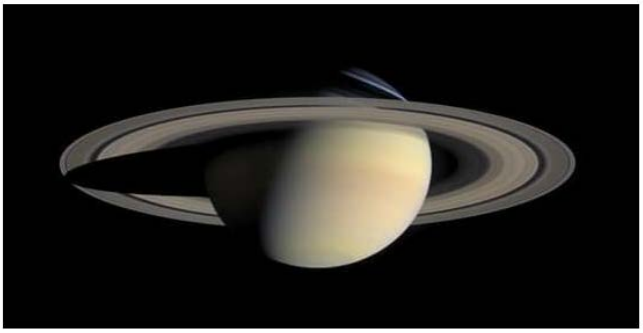


Figure 6. In accordance with the Theory of Dynamic Interactions, the ecliptic or Saturn's discs trace flat orbits.

Our theory may also justify the ripples that occur in the rings of Saturn, and in general in all the ring systems: may be due to changes or disturbances moment acting. For example, by fluctuations of

gravity of Saturn itself, in its rotation, or due to the masses of other planets or satellites affecting

the movement of the rings.

It also justifies the configuration of spiral galaxies, whose arms must consist of celestial bodies in rotation and accelerated traslation speed (See figure 7).

According to the General Theory of Relativity, we can estimate that the mass of the Earth distorts space-time in its surroundings. In this case, we can assume the analogy that the Moon makes a rolling movement on the curved surface of the space-time deformed by the Earth, generating a new rotation of the satellite, which we can suppose is not coaxial to the intrinsic rotation that it already has. In this case, the dynamic interactions predicted by the TDI would be generated, resulting in the closed and flat orbit of the Moon that we see. In this way, we justify the behavior of the celestial bodies, in accordance with the criteria of relativity, without needing torques or forces.

In this same area, the second Law of Kepler can also be justified, since, in the case of an elliptical orbit, it must have a cause according to the TID, in a variation of the orbital velocity, which is consistent with the greater distortion of space-time in the vicinity of the central mass.

Also, the same or analog reasoning could be applied to understand the behavior of so many rotating solid elements like the boomerang, the hoop or the wheel.

4 New paradigm

New Paradigm in Physics, and the accompanying videos, provides only a brief summary of the works and studies carried out over the last thirty-five years, to propose a Rotational Dynamics of Interactions applicable to bodies subjected to multiple successive non coaxial torques. Theinitial hypotheses are based on new criteria about speed coupling and rotational inertia, and have been confirmed by experiments and by a mathematical model allowing the simulation of the real behavior of bodies submitted to these excitations. In this study, I found a clear correlation between the initial speculations, original hypotheses, simulation model, deduced physical laws, the realized experiments, and mathematical models corresponding to the equations of motion that result from the proposed dynamics laws.

Figure 7. The spiral galaxy whose arms, in accordance with what the TDI predicted, will be constituted by celestial bodies in rotation and accelerated traslation.

As a result of this dynamic investigation work we can propose the following conclusions:

1 There is a wide subject area not yet developed in rotational dynamics inasmuch as rigid bodies are subjected to accelerations caused by simultaneous non coaxial rotations.
2 This area of knowledge can be analyzed under relativistic and non-relativistic mechanics. Hypotheses are based on new criteria about speed coupling and rotational inertia.
3 In the exposed experimental non relativistic tests carried out, we have concluded that new general laws of behavior can be obtained, based in the analysis of the dynamics fields created.
4 We have obtained an equation of motion for rigid bodies in translational motion with intrinsic angular momentum, when subjected to non-coaxial pairs, which defines the dynamic behavior of rigid bodies in these cases.
5 We find a clear mathematical correlation between rotation and translation. This mathematical connection allows us to identify a physical relation between transfers of rotational kinetic energy to translational kinetic energy and vice versa.

6 The mathematical model implies that it would be possible that moving bodies subjected to successive non coaxial torques would initiate orbital motion as a result of inertial dynamic interactions.

7 While maintaining constant initial angular momentum and the second torque constant, the center of mass of the moving bodies would follow a closed orbit without requiring any centripetal force.

8 The theory also allows to give an answer to an initial aporia: to be aware and to understand the physical and mathematical correlation between orbitation and intrinsic rotation.

The result of this project is the conception of an innovative dynamic, and also the demonstration of a rational field theory, that gives a new understanding of the behavior of matter. In my opinion, the application of these dynamic hypotheses to astrophysics, astronautics, and other fields of physics and technology will allow new surprising, and stimulating advances in investigation and in the innovation of an unprecedented Rotational Dynamics of Interactions.

Also has numerous and significant scientific and technological applications, especially in orbital dynamics, orbit determination, and orbit control. For instance:

- Variation of the affecting torque, arises when subjecting intrinsic angular momentum bodies to new non-coaxial momentums.
- To conceive an intrinsic rotating mobile solid, which could be exclusively controlled due to Dynamic Interactions.
- To calculate the trajectory of any intrinsic angular momentum solid in space.
- To propose a new steering system independent from a rudder or any other external element.

We can suggest advances in the studies and application related to orbital mechanics, guidance, navigation, and control of single or multi-spacecraft systems as well as space robotics and rockets.

This theory has also had numerous technological applications26 in the control of moving objects, in astronautics27, in nuclear fusion plants or for interpreting climate phenomena with rotating fluid masses, such as typhoons or tornadoes.28 Indeed, we will even consider numerous technological applications, for example, dynamic confinement in nuclear fusion reactors to generate clean electricity, which has been described in two already published articles.

We would like to note that in our deductive reasoning, we have introduced a discriminating hypothesis, in the case of the body's translation movement, when we propose that the field of translation speeds will be coupled to the anisotropic field of inertial speeds created by the second non-coaxial torque, with an orbit simultaneous to the initial intrinsic rotation of the mobile as the resulting movement. This new orbital movement generated by a non-coaxial momentum, will be defined by the rotation of the velocity vector of translation, the latter kept constant in module.

In previous texts, we have proposed that, through this analysis, the nature of any movement in space can be determined and predicted, defining its relativity. The movement equation that is proposed, and the laws that are formulated, permit the initiation of the structuring of a rational mechanics and of a rotational dynamics based on principles and axioms, for bodies submitted to accelerations by rotations, clearly differentiated from classical mechanics.

In this new rational structure, phenomena that are paradoxical or alien to the main structure should not be present as happens in classical mechanics with the so called gyroscopic torque or fictitious forces.

The Theory of Dynamic Interactions is a logical-deductive system constituted from some dynamic hypotheses. By means of the observation of nature, the establishing of some initial hypotheses, and starting from axioms and postulates, we have constructed a structure of knowledge in relation to rigid solid bodies, when submitted to successive accelerations by rotation. The physical-mathematical model obtained allows us to interpret the observable behavior of these bodies, subject to successive non coaxial torques, according to deduced laws, as well as to extract new consequences, inferences and predictions. For example, the theory allows justifying the deviation that undergoes the horizontal curvilinear trajectory of a ball, when it is submitted to non-coaxial moments (See figure 8).

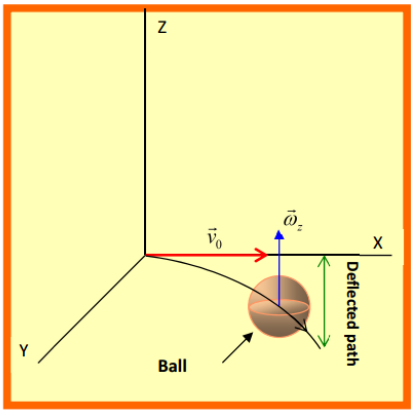


Figure 8. Horizontal curvilinear trajectory of a ball with effect, whose deviation can be justified by the Theory of Dynamic Interactions.

This theory has been checked and confirmed by experimental tests. This text does not pretend to challenge the laws of Newton; what has been developed is a conceptual structure complementary to classical mechanics for systems accelerated by rotations. We propose a theory based on a specific rotational algebra for non-inertial environments where the starting hypotheses that the laws of translational classical mechanics are based on, are not respected. We propose the exploration of a new niche of knowledge for some very specific, but not trivial dynamic conditions that are repeated in our universe.

Also noteworthy is that, through the development of these studies, full coherence has been obtained between the hypotheses of the beginning, the applied principles and axioms, the developed physical-mathematical model, the obtained movement equation, the deduced laws, the reached simulation models and the conducted experimental tests. We also have referred to examples in nature which support the Theory of Dynamic Interactions, all endorsing the laws proposed in this text.

It is necessary to admit the existence of a rotational dynamics of interactions with real results and which modifies the behavior of bodies in accordance with some specific and universal dynamic Laws.

This research can be extended with the Field Theory and a relativistic deep analysis, and may allow the physical knowledge of new space systems and brings potential applications for the future, along with numerous relevant technology developments.

We want to suggest that interest should arise in physics in the exploration of non-inertial accelerated systems, and also to express a call for the need to develop scientific investigation projects for their evaluation and analysis, as well as technological projects based on these hypotheses. In our opinion, these hypotheses suggest new keys to understanding the dynamics of our environment and the harmony of the universe. A universe composed not only of forces, but also of their momentums; and when these act constantly upon rigid rotating bodies, with an also constant translation speed, the result is a closed orbiting movement, thus a system which is moving, but within a dynamic equilibrium.

The application of these dynamic hypotheses to astrophysics, astronautics and to other fields of physics and technology possibly allows new and stimulating advances in investigation.

The result of this project is the conception of an innovative dynamic theory, which specifically applies to rigid rotating physical systems and which has numerous and significant scientific and technological applications, The Theory of Dynamic Interactions establishes new conceptual criteria, of a more general description, to understand the behavior of nature, meaning that the current laws of dynamics could

be considered special and specific cases of this theory. For example, Newton's laws would apply to the case of a physical model of behavior, without force momentums.

5 Other suggestions

The Theory of Dynamic Interactions has led to suggestions that the Theory of Relativity should be reviewed and doubts have been raised over Einstein's Equivalence Principle. This theory should give rise to a review of the current concepts of astrophysics. There should be special analysis of the fascinating history of human knowledge of the universe, while also proposing, developing and explaining the application of the Theory of Dynamic Interactions to afford us a better understanding of the dynamic enigmas that surround us.

The Theory of Dynamic Interactions It is fully described in the book New paradigm in physics (See figure 9).

We can remember how Max Planck used the term "quanta" in his thermodynamic studies. He proposed the concept of the quantization of radiation in 1900 by studying the emission of blackbody radiation by suggesting that energy can only be absorbed or released into discrete packets, which he called elements of energy. Planck also deduced the numerical value of h, later named after him as Planck's constant, being understood in physics as the smallest amount of energy that can be transmitted.

In 1905, Albert Einstein suggested generalizing this concept for energy radiation, and even for electromagnetic radiation, proposing the existence of light quanta. This concept was generalized until it was accepted as the minimum value that any physical quantity can take. In this way, it turns out to be the least possible variation of any physical quantity.

This constant is still assumed today, but its nature or justification is unknown. Nevertheless, this Quanta or minimum value of physical magnitudes, could have its origin in atomic physics, and in particular, in atomic particles, being related to their spin, and therefore, with the possible angular momentum of the particle.

We suggest exploring and analyzing this possibility, and determining if there maybe a correlation between the atomic particle spin and the Planck constant, based on the criteria established by the Theory of

Dynamic Interactions. On the other hand, before proceeding with the structuring of a physics developed from logical configurations resulting from mathematical deductions, which have not been experimentally contrasted, and therefore with no observational result, such as black holes, axions, dark matter,

dark energy, etc ..., we reiterate the need to incorporate the postulates of the theory we propose, ina review of the Theory of Relativity, that allows us to reach a Theory of everything that is consistent with physical reality.

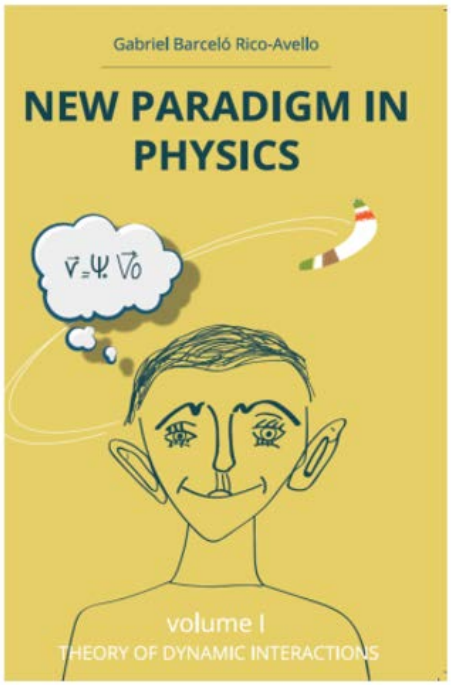


Figure 9. New paradigm in physics

We believe that with this new model that we propose, the understanding we have of our observational universe will be facilitated, as well as that of the physical phenomena we notice in it.

To end, we can remember Isaac Newton: We are to admit no more causes of natural things than such as are both true and sufficient to explain their appearances.

6 Final note

According to the information we have, only our team of Advanced Dynamics has investigated actually in this field, and has published on these researches.I have presented here a synthesis of our Theory of Dynamic Interactions, after more than forty years of research. Perhaps I have to repeat concepts and arguments from previous texts.

Our practice of careful elaboration and possible repetition of our own texts is explained by our desire to overcome, as efficiently as possible, difficulties in spreading our ideas and also, to avoid possible misinterpretations.

The text refers to other papers published by the same author, or other texts of his research team, understanding that this repetition, in a synthesis text, is necessary and convenient for clarity in the dissemination of the results obtained.

REFERENCES

[1] Barceló, Gabriel: *New Paradigm in Physics, Volume I: Theory of Dynamics Interactions.* Amazon, 2017.

[2] Merino, J. (2017) *The Works and Days of Gabriel Barceló*. World Journal of Mechanics, Volume 7, 43-45. Number 3, March 2017 (Special Issue on Rotational Dynamics: Theory of Dynamic Interactions). Doi: 10.4236/wjm.2017.73006.
http://www.scirp.org/Journal/PaperInformation.aspx?PaperID=74664

[3] Dalby, F. (2017) *Rolling Over into the Age of Algorithm*. World Journal of Mechanics, Volume 7, 39-42. Number 3, March 2017 (Special Issue on Rotational Dynamics: Theory of Dynamic Interactions). Doi: 10.4236/wjm.2017.73005.
http://www.scirp.org/Journal/PaperInformation.aspx?PaperID=74663

[4] Euler, Leonard: *Die Entdeckungeines neuen Prinzips der Mechanik.* History of the Royal Academy of Berlin, 1750, pp. 185-217.

[5] Euler, Leonard: *Theoria motus corporum solidorum seu rigidorum.* Rostock, 1763. Other edit., Greifswald, 1790, p. 301.

[6] Cano, J.: *The Pendulum of Dynamic Interactions*. Journal of Applied Mathematics and Physics, Vol.3 No.9, September 2015, 1186-1198. Published Online: DOI: 10.4236/jamp.2015.39146 http://www.scirp.org/journal/jamp

[7] Barceló, G.: *The Flight of the Boomerang.* (El vuelo del bumerán). Ed. Marcombo: Barcelona, 2006. http://www.dinamicafundacion.com/

[8] Cano, J.: *The Pendulum of Dynamic Interactions*. Journal of Applied Mathematics and Physics, Vol.3 No.9, September 2015, 1186-1198. Published Online: DOI: 10.4236/jamp.2015.39146 http://www.scirp.org/journal/jamp

[9] Barceló Aristoy. Veronica: *A scientific legacy: Theory of Dynamics Interactions.* World Journal of Mechanics. Special issue: Rotational Dynamics: Theory of Dynamic Interactions. March, 2017. http://www.scirp.org/Journal/Home.aspx?IssueID=9235#74661

[10] Pérez, L. A. *The Pendulum of Dynamic Interactions.* Video. 2015. www.advanceddynamics.net/the-pendulum-video.
https://www.dropbox.com/s/rrjb1786ub75a8h/PIDing_m.mp4?dl=0

[11] Barceló, G.: *Theory of Dynamic Interactions: Laws of Motion*. World Journal of Mechanics, 3, 328-338. (2013) http://dx.doi.org/10.4236/wjm.2013.39036

[12] Alvarez Martínez, Alejandro: *Theory of dynamic interactions: innovations.* World Journal of Mechanics. Special issue: Rotational Dynamics: Theory of Dynamic Interactions. March, 2017. http://www.scirp.org/Journal/Home.aspx?IssueID=9235#74661

[13] Barceló, G.: *On the equivalence principle.* 61st International Astronautical Congress, American Institute of Aeronautics and Astronautics, Prague, CZ. 2010.

[14] Pérez, L. A. *The Pendulum of Dynamic Interactions.* Video. 2015. www.advanceddynamics.net/the-pendulum-video. https://www.dropbox.com/s/rrjb1786ub75a8h/PIDing_m.mp4?dl=0

[15] Barceló, Gabriel: *New Paradigm in Physics, Volume I: Theory of Dynamics Interactions.* Amazon, 2017. Section: 5.4.0- Equation deduction.

[16] Barceló, G.: *Theory of Dynamic Interactions.* The Free Thinker, 2012. http://www.ellibrepensador.com/2012/07/06/teoria-de-interacciones-dinamicas-por-gabriel-barcelo/

[17] Garcia-Moliner, F. (2017) *Physico-Mathematical Models in Rotational Motions*. World Journal of Mechanics, Volume 7, 35-38. Number 3, March 2017 (Special Issue on Rotational Dynamics: Theory of Dynamic Interactions). doi: 10.4236/wjm.2017.73004. http://www.scirp.org/Journal/PaperInformation.aspx?PaperID=74661

[18] Martín Gutiérrez, Almudena: *The flight of the boomerang: comments.* World Journal of Mechanics, Volume
7. Number 3, March 2017 (Special Issue on Rotational Dynamics: Theory of Dynamic Interactions). http://www.scirp.org/Journal/Home.aspx?IssueID=9235#74661

[19] Bauluz, E.: *New Dynamic Hypotheses.* Madrid, 2011. This video showed the experiments carried out by Advanced Dynamics S. A. to prove and justify the http://www.youtube.com/watch?v=vSUkd4slHGQ&feature=c4-overview&list=UUgDHgaGi2I2rmZNoanNbVWQ http://www.youtube.com/watch?v=vSUkd4slHGQ

[20] Barceló, G.: *Imago Universi: A Story of the Human Conception of the Cosmos*. Ed. Arpegio: Barcelona, 2013. http://www.editorialarpegio.com/ http://imagouniversi.com/

[21] Barceló, G.: *A Rotating World* (Un mundo en rotación). 2008, Editorial Marcombo: Barcelona. http://www.dinamicafundacion.com/

[22] Barceló, G.: *A new rotational dynamics of interactions for the planet Saturn.* (*Una nueva Dinámica Rotacional de Interacciones para el planeta Saturno),* 2006. http://dinamicafundacion.com/wp-content/uploads/2014/02/UNA-NUEVA-DINAMICA-ROTACIONAL-DEINTERACCIONES-PARA-EL-PLANETA-SATURNO.pdf

[23] ESA: https://es.pinterest.com/pin/461407924299218908/

[24] Cano, Julio: *Rotational dynamics: A challenge.* World Journal of Mechanics, Volume 7. Number 3, March 2017 (Special Issue on Rotational Dynamics: Theory of Dynamic Interactions). http://www.scirp.org/Journal/Home.aspx?IssueID=9235#74661

[25] Barceló, G.: *On the equivalence principle.* 61st International Astronautical Congress, American Institute of Aeronautics and Astronautics, Prague, CZ. 2010.

[26] Barceló, G.: *Technological Applications of the New Theory of Dynamic Interactions*. Global Journal of Researches in Engineering: Mechanical and Mechanics Engineering-G, Volume 13, Issue 5, 2013. https://globaljournals.org/GJRE_Volume13/E-Journal_GJRE_(G)_Vol_13_Issue_5.pdf

[27] Martín Gutiérrez, Almudena. *Flight simulator, trip to Saturn.* E.T.S.I. Aeronáuticos (U.P. Madrid). Degree project. May, 2015.

[28] Barceló, G.: *Dynamic Interactions in the Atmosphere.* Atmospheric and Climate Sciences. Vol.4 No.5, November 20, 2014. DOI: 10.4236/acs.2014.45073. http://www.scirp.org/Journal/PaperInformation.aspx?PaperID=51584#.VHB4YTSG_To http://dx.doi.org/10.4236/acs.2014.45073

[29] Barceló, G.: *Dynamic Interaction Confinement.* World Journal of Nuclear Science and Technology Vol.4 No.4, October 29, 2014DOI: 10.4236/wjnst.2014.44031 http://www.scirp.org/journal/PaperInformation.aspx?paperID=51026& http://dx.doi.org/10.4236/wjnst.2014.44031

[30] Barceló, Gabriel: *Dynamic Interaction: A New Concept of Confinement*. Global Journal of Science frontier Research: A physics & space science. GJSFR A Volume 16 Issue 3 https://globaljournals.org/GJSFR_Volume16/E-Journal_GJSFR_(A)_Vol_16_Issue_3.pdf

[31] Barceló, G.: *A Rotating World* (Un mundo en rotación). *Figure 13.33.* Editorial Marcombo, 2008, Barcelona. http://www.dinamicafundacion.com/

[32] Advanced Dynamics, "Theory of Dynamic Interactions. Experimental Tests." http://www.youtube.com/watch?v=P9hGgoL5ZGk&feature=c4overviewvl&list=PL3E50CF6AEBEED47B http://www.youtube.com/watch?v=P9hGgoL5ZGk&list=PL3E50CF6AEBEED47B http://www.youtube.com/watch?v=XzTrGEtJGXU&list=PL3E50CF6AEBEED47B http://www.youtube.com/watch?v=dtMqGSU9gV4&list=PL3E50CF6AEBEED47B http://www.youtube.com/watch?v=qK5mW2j2nzU&list=PL3E50CF6AEBEED47B http://www.advanceddynamics.net/index.php?option=com_content&task=view&id=26&Itemid=39&lang=en

[33] Barceló, G. *Theory of Dynamic Interactions: Laws of Motion*. World Journal of Mechanics, 3, 328-338. (2013) http://dx.doi.org/10.4236/wjm.2013.39036

[34] Research Blog: *Extensive report on the investigations of Gabriel Barceló.* March, 2016. Global Journal of Researches in Engineering The report incorporates the new scientific and technological advances achieved with the Theory of Dynamic Interactions. http://blog.gjre.org/2016/03/behaviour-of-rotational-bodies.html

[35] Barceló. G. Analysis of Dynamics Field Systems Accelerated by Rotation. Dynamics of non-inertial systems. DeMSET-2011 Congress, Miami. USA. http://www.coiim.es/forocientifico/FORO%20CIENTFICO/Documentos/DeMSET_2011_GBarcelo.pdf

[36] Newton, Isaac: *The Mathematical Principles of Natural Philosophy*, Rule I, of Book III 1687

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM
TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE
ISSN 2054-7390

Nonlinear Time Series Prediction Performance Using Constrained Motion Particle Swarm Optimization

[1]Nicholas Sapankevych and [2]Ravi Sankar
[1]Raytheon, St. Petersburg, USA;
[2]Department of Electrical Engineering, University of South Florida, USA;
sankar@usf.edu

ABSTRACT

Constrained Motion Particle Swarm Optimization (CMPSO) is a general framework for optimizing Support Vector Regression (SVR) free parameters for nonlinear time series regression and prediction. CMPSO uses Particle Swarm Optimization (PSO) to determine the SVR free parameters. However, CMPSO attempts to fuse the PSO and SVR algorithms by constraining the SVR Lagrange multipliers such that every PSO epoch yields a candidate solution that meets the SVR constraint criteria. The fusion of these two algorithms provides a numerical efficiency advantage since an SVR Quadratic Program (QP) solver is not necessary for every particle at every epoch. This reduces the search space of the overall optimization problem. It has been demonstrated that CMPSO provides similar (and in some cases superior) performance to other contemporary time series prediction algorithms for nonlinear time series benchmarks such as Mackey-Glass data. This paper details the CMPSO algorithm framework and tests its performance against other SVR time series prediction algorithms and data including the European Network on Intelligent Technologies for Smart Adaptive Systems (EUNITE) competition data and the Competition on Artificial Time Series (CATS) competition data.

Keywords: Support Vector Regression, Constrained Motion Particle Swarm Optimization (CMPSO), Particle Swarm Optimization, Nonlinear Time Series Prediction.

1 Introduction

Nonlinear time series regression and prediction applications range from financial market prediction, electrical load forecasting, dynamic control system design, and a vast array of other real world problems. There are many methods to solve such problems including Auto Regressive Moving Average (ARMA) algorithms (in many different forms), Kalman Filtering (also in many different forms), Artificial Neural Networks (ANNs), Support Vector Machines (SVMs) and Support Vector Regression (SVR), and many others. Any of the above algorithms can be applied to real world problems, some with greater success than others. In many cases, the success of the algorithm for a given application depends heavily on algorithm "tuning" – the process of optimizing the algorithm for the specific problem space. Some examples of "tuning" include model selection (as with Kalman Filtering) and free parameter selection (as with SVR). The employment of some algorithms such as SVR further requires the use of a Quadratic Program (QP) to solve for the given algorithmic parameters, thus increasing the computational complexity of these kinds of approaches.

DOI: 10.14738/tmlai.55.3566
Publication Date: 20th August, 2017
URL: http://dx.doi.org/10.14738/tmlai.55.3566

Although SVMs/SVR algorithms are generally computationally complex, they are effective for time series regression and prediction applications [1]. The challenge remains to optimize free (undefined) parameters associated with SVR. These parameters are essential to the proper operation of the SVR algorithm and are generally user selected in an ad hoc manner. Specifically, three parameters need tuning for proper implementation of SVR: 1. A capacity constant (C) that effects scaling and accuracy (which will ultimately be factored out and eliminated as a free parameter in the Constrained Motion Particle Swarm Optimization (CMPSO) formulation), 2. An error bound (ε that will control the data fit, and 3) a kernel function parameter (σ that is used to help cast data to feature space. The focus of this paper is to detail a unique approach of applying SVR to (almost) any time series regression and prediction problem by using a free parameter optimization framework employing Particle Swarm Optimization (PSO). The CMPSO framework is unique in the sense that it does not require the use of a QP for evaluating a candidate SVR solution associated with each particle of the PSO swarm for every epoch, thus reducing the computational load in solving the problem. CMPSO has been shown to be effective for nonlinear time series regression and prediction problems [2]. As compared to other similar-class algorithmic techniques, CMPSO has met or exceeded performance based on experiments using simulated benchmark time series data referred to as Mackey-Glass time series data. Further, CMPSO has the numerical efficiency advantage. The goal of this paper is to provide a more thorough performance analysis and to evaluate CMPSO against many published benchmark time series data.

The outline of the paper is as follows: Section II gives a brief overview of SVR and PSO theory. Section III details the fusion of SVR and PSO to form CMPSO. Section IV shows how CMPSO improves computationally efficient versus using a separate QP solver and associated PSO optimization scheme. Section V provides CMPSOs performance against published benchmark time series data and Section VI details the advantages of CMPSO and further challenges. This section also provides research areas for future investigations.

2 Methodology Background

2.1 Support Vector Regression

SVR is one of two underlying algorithms used in CMPSO that facilitates the interpolation and extrapolation functions of arbitrary one dimensional time series data. SVR is a supervised learning technique that does not assume any specific, underlying model such as found in Kalman filtering. It has advantages for problems that are non-linear in nature and do not have any defined underlying process. The SVR technique is an extension of SVM classifiers developed by Vapnik *et al.* [3, 4]. This technique has been successfully applied for many time series prediction applications [1].

The fundamental principal behind the SVR approach is to cast the time series data into "feature" space as shown in equation (1). The effect of this process is to transform typically non-linear data (non-linear in the sense of a first order linear fit of the time series) into a first order linear regression in the transformed space. The time series estimate, $\hat{y}$, is the weighted combination of a transformed input variable ϕ(x) (where x can be multi-dimensional) plus a bias term b. This is essentially a linear regression model in feature space. The primal formulation derived by Vapnik and summarized by Smola and Schölkopf [5] attempts to minimize the Euclidian norm of the weights (w) and add a loss function to formulate equation (2). Equation (3), the ε-insensitive loss function in this case, allows for errors within a ε bound around the estimate:

$$\hat{y} = w \cdot \phi(x) + b \tag{1}$$

$$\text{Minimize: } \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} L(y_i, \hat{y}_i) \tag{2}$$

$$L(y_i, \hat{y}_i) = \begin{cases} |y_i - \hat{y}_i| - \varepsilon & \text{if } |y_i - \hat{y}_i| \geq \varepsilon \\ 0 & \text{Otherwise} \end{cases} \tag{3}$$

Where, φ(x): Time series cast in feature space; w, b: Weights and bias term, respectively (to be determined);

L: Loss function; ε: Tube size (free parameter); C: Capacity (free parameter)

With the use of Lagrange multipliers and some mathematical manipulation [6], the dual of the primal quadratic programming (QP) problem for SVR (equation (2)) can be formulated in the Lagrange multiplier equation (4) which is called the dual SVR objective function, and the constraints shown in (5):

$$\text{Maximize } \sum_{i=1}^{N} \alpha_i y_i - \varepsilon \sum_{i=1}^{N} |\alpha_i| - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_i K(x_i, x_j) \tag{4}$$

$$\text{Subject to: } -C \leq \alpha_i \leq C\,,\ \sum_{i=1}^{N} \alpha_i = 0\,, C > 0 \tag{5}$$

Where, α: Lagrange multiplier (to be solved using QP or other technique); $K(A, B)$: Kernel Function (user defined); ε: Tube size (free parameter); C: Capacity (free parameter)

For the data sets analyzed in this paper, the exponential kernel given in equation (6) is used:

$$K(A, B) = exp\left(-\frac{1}{2\sigma^2}\|A - B\|^2\right) \tag{6}$$

Where, σ: Kernel (free parameter); A, B: Input vectors

The resulting time series approximation function is given in equation (7):

$$\hat{y} = \sum_{i=1}^{N} a_i^* K(x, x_i) + b^* \tag{7}$$

Equation (7) essentially replaces equation (1) because in the dual formation, the weights *w* in equation (1) equal the weighted sum of the Lagrange multipliers times the input data [5]. The star (*) notation denotes the data points (and corresponding Lagrange multipliers) that lie on or outside the ε tube (and for the bias term, the notation refers to some optimal bias value). Any data points lying inside the tube will be set to zero per the loss function defined in equation (3). This can lead to a "sparse" solution as data points with associated zero Lagrange multipliers will not be necessary to estimate the function.

As mentioned earlier, solving for the Lagrange multipliers requires a quadratic programming algorithm. There are many well-known techniques to solve these types of problems such as Gradient Ascent and Sequential Minimization Optimization (SMO) [7]. The SMO algorithm described in [6] and [7] is designed for SVM classification problems and has been shown to be an efficient method of solving SVM/SVR quadratic programs. CMPSO uses a modified form of SMO specifically designed to solve for the Lagrange

multipliers for the SVR formulation. From the given SVR formulation, one can quickly see that there are several algorithmic choices required to find a suitable solution to any given time series regression or prediction problem. First, a Kernel function must be selected. As documented in [1], the exponential Kernel function is a primary selection for many SVR applications and has been shown to work well in the CMPSO Mackey-Glass benchmark experiment referenced in [2]. Hence, an exponential Kernel was chosen for our experiments. Secondly, there are three free parameters identified in the SVR formulation as user defined. The CMPSO framework attempts to tune these parameters using PSO.

2.2 Particle Swarm Optimization

The selection of C, ε, and σ parameters associated with SVR has been a topic of much research. One candidate approach is Particle Swarm Optimization (PSO) and is the focus of this research. First developed by Kennedy and Eberhart [8] in 1995, PSO mimics the real life process of a swarm of animals or insects searching for food. The goal for each individual of the swarm, where each individual is termed a "particle", is to scavenge in their search region for the place that has the most food. Each particle will remember the location in their search region where they have found the highest density of food and further the entire swarm will remember collectively where the highest density of food was found in the areas that have been searched. As the particles move in their search region, otherwise known as the "search space", they tend to move based on the force, or pull, in three distinct directions:

1) In the direction they are already traveling (inertial)

2) In the direction where the particle remembers the most food being (cognitive)

3) In the direction where the largest amount of food has been found by the entire swarm (social)

As time progresses (and as in real life), the particles of the swarm tend to gravitate towards the place where they collectively have found the most food, assuming there is only one unique place in the entire region that has the most food. Translated to PSO, one would expect every particle to settle on the spot where a maximum (or minimum) solution is found. It should be noted that PSO technical tutorials are found in [9] and [10] with [11] providing a detailed survey of many different PSO applications used in solving real world problems.

The PSO formulation starts with the particle definition, which includes the particles' position and velocity. Each particle definition includes the free parameters and their associated boundaries in the search space. Equations (8a, 8b) are the particle position and velocity definitions for the SVR example given in the previous section.

Particle i Position: $$p_i(C, \varepsilon, \sigma, \alpha_1 \dots \alpha_N, b) \quad (8a)$$

Particle i Velocity: $$v_i(C, \varepsilon, \sigma, \alpha_1 \dots \alpha_N, b) \quad (8b)$$

For CMPSO, each particle i has a position and velocity in search space that includes the three SVR free parameters C, ε, and σ, the N Lagrange multipliers α_1 thru α_N and the bias term b. The boundaries of each of the free parameters are determined by the user before the PSO algorithm is executed (the boundary values of each of the variables will be discussed further in the next section). The next step in the process is to define a "fitness" function. This function (also sometimes referred to as an objective function – not to be confused with the SVR notation of primal and dual objective functions) is the mechanism by which

each particle will evaluate how much food is at its current location in the physical world analogy. Typically the fitness function will generate a single value representing the "level" of fitness relative to some ideal value. In some cases, the position of the best level of fitness may not be unique and particles tending towards "local maxima/minima" can occur. There are no general heuristics for defining fitness functions as they are designed based on the specific application. In practice, identifying an adequate fitness function can be very challenging and may take several iterations to finalize – this is key to a successful optimization process.

After the particles' search space, search boundaries and fitness function have been defined, the next step in the PSO process is to initialize each particle's position and velocity. For each of the variables, a random location and velocity is selected within the boundary of each variable. The random selection is usually from a uniform random distribution within the boundary limits. After particle initialization, the particle simulates real world animal or insect migration by moving through the search space using the simple kinematic equation (9):

Particle Position at Epoch k+1: $$p_{i,k+1} = p_{i,k} + \Delta_t v_{i,k} \quad (9)$$

The position of each particle is updated every epoch (indexed by k) by adding the velocity term with each updated particle position and setting Δt equal to unity. The velocity of each particle i at epoch k is a weighted sum of the current velocity times the "inertial" weighting factor, the direction in which the particle has found its maximum of the objective function times the "cognitive" weighting factor, and the direction in which the maximum is found by all the particles times the "social" weighting factor as given in equation (10):

$$v_{i,k} = \begin{bmatrix} v_{i,k-1} \\ rand \cdot (p_{i,best} - p_{i,k}) \\ rand \cdot (g_{best} - p_{i,k}) \end{bmatrix}^T \cdot \begin{bmatrix} w_I \\ w_c \\ w_s \end{bmatrix} \quad (10)$$

Where, w_I, w_c, w_s: Inertial, Cognitive, and Social Weights;

$rand$: A random number over the closed interval [0,1]; (note the two *rand* functions in (10) are independent); $p_{i,k}$: Particle i's current position;

$p_{i,best}$: Particle i's position where it has found its best result over all k epochs ("local" best)

g_{best}: The position in search space where the swarm's best result was found over all k epochs ("global" best)

The values of $p_{i,best}$ and g_{best} are determined by the evaluation of the fitness function for every particle at every epoch. As each particle moves, the location where the highest score for each particle is remembered ($p_{i,best}$) as well as the location where highest score found for the entire swarm (g_{best}). There are situations where a particle will travel outside the search space for any or all of the variables. There are several techniques available for handling these cases:

Absorbing walls: The velocity of any variable dimension lying outside that dimensional boundary is set to zero. By zeroing the velocity, the particle tends to be "pulled" back into the search space given the p_{best} and g_{best} positions will always be inside the search space. The term "absorbing" for this technique refers to the effect of absorbing the velocity at the limit, or "wall" of the boundary.

Reflecting walls: The sign of the velocity of any dimension lying outside the boundary is reversed. The velocity vector for that dimension causes the particle to "reflect" back into the search space.

Invisible walls: Particles that move outside the boundaries of the search space are allowed without any changes to their velocities; however, the fitness function will not be evaluated until all dimensions of the particles are within the solution boundaries. P_{best} for a particle will not change. The benefit of using this technique is the savings of computational time. Typically, the fitness function is more complex and takes longer to calculate than the equation of motion.

The PSO algorithm is executed until a stopping criterion is reached. There are several different ways to evaluate the completion of the PSO process:

1. *Fitness Function Limit*: The PSO algorithm is stopped when the g_{best} value reaches a user-defined and/or pre-calculated limit.
2. *Function Time-Out:* The PSO algorithm is stopped after a pre-determined amount of time.
3. *Epoch Limit:* The PSO algorithm is stopped after a pre-determined number of epochs.

As stated earlier, it is possible for the PSO algorithm to "stagnate" such that g_{best} value settles on a non-optimal solution (i.e. a "localized" maximum/minimum). One possible solution can be to re-initialize a subset of the particle population to random locations after a certain amount of time or number of epochs has passed. This technique is not guaranteed to succeed as prior knowledge of the search space would have to be known.

2.3 Combining Particle Swarm Optimization and Support Vector Regression

For the SVR problem as stated in section 2.1, there are essentially two optimization objectives:

1. Find the Lagrange multipliers and bias term (typically accomplished with a QP program)

2. Find optimal values for the SVR free parameters (capacity term C, tube size ε, and kernel free parameter σ)

Multiple Objective optimization techniques must be used for applications that have more than one objective. *Multiple Objective* Particle Swarm Optimization (MOPSO) is a candidate technique that is considered for this application since more than one "fitness" criteria is required. A survey of MOPSOs can be found in [12] along with their implementations and applications. Reference [13] also details the use of MOPSOs and qualitative performance results associated with applications with more than one fitness criteria. Many of the MOPSO techniques reviewed and referenced above were considered for solving this specific SVR problem. However, the solution of the QP program is dependent on $C, \varepsilon, \text{and } \sigma$, making the MOPSO techniques cited difficult to implement for this type of problem. In other words, objective number two stated above cannot be found without evaluating objective number one with a candidate set of $C, \varepsilon, \text{and } \sigma$.

PSO based approaches not based on multiple objectives have been proposed for SVR free parameters ($C, \varepsilon, \text{and } \sigma,$) estimation. Hong [14] proposed the use of Chaotic PSO in use with SVR for electrical load forecasting. The technique uses a parallel approach in applying PSO to find the SVR free parameters, but the technique still requires an SVR solution algorithm for the QP problem. Guo *et al*. [15] uses PSO for finding the free parameters for a Least Squares Support Vector Machine (LS-SVM) application using

medical related data for benchmarking (note that this application was specifically for hyper-parameter selection for SVM classification, not SVR based regression). Other PSO related applications involving SVR and linear constraint problems are being studied. Yuan *et al*. [16] introduced a modified PSO algorithm for SVM training based on linearly-constrained optimization using PSO and techniques proposed by Paquet and Engelbrecht [17, 18]. The method presented by Yuan, the "Modified Linear PSO - MLPSO", initializes the Lagrange multipliers randomly, but relies on re-initialization of the Lagrange multipliers should they go beyond the capacity value boundary C. The focus of this research is the PSO applied process for solving linearly constrained optimization problems. Wang *et al.* [19] uses PSO to solve for the three user-defined SVR parameters (referred to as C, ε, and σ, respectively) for a real world application relating to coal working face gas concentration forecasting. This is a similar approach to the presented research in this paper, although the SVR optimization in this citation still requires a separate QP algorithm. Although Wang presents a feasible approach, there is still the computational overhead associated with using SVR and PSO in a non-integrated methodology.

There are many SVR and PSO combined approaches that have been published recently that use PSO to optimize SVR free parameters [19-28]. The applications in these publications range from power load forecasting, traffic flow optimization, to benchmark data estimation, and many others. There are also other general modifications to the algorithms in the citations, but the core processing algorithms are based on both SVM/SVR and PSO. All of these publications show that a PSO based approach to SVR parameter tuning is viable and effective for many different problems. However, none of the researched publications address the computational overhead involved with using a QP solver to find an appropriate SVR solution. The examples stated above that combine both PSO and SVR approaches illustrate the utility and test validity of this type of technique to solve real world problems. It should be pointed out that the PSO/SVR is relatively new area of research and the work outlined in this paper attempts to advance these concepts by using PSO to simultaneously solve the QP problem and optimize the SVR free parameters (C, ε, and σ).

3 Constrained Motion Particle Swarm Optimization

Current research has shown examples of how PSO can be applied to SVR applications; however, there are many complex design and implementation issues. One of the biggest challenges associated with implementing PSO for optimizing SVR, specifically optimizing C, ε, and σ, is the requirement to re-compute the Lagrange multipliers and bias term via a QP solver for every particle position update. The requirement to execute a QP solver for every particle and for every epoch leads to computational inefficiencies and longer optimization times. Another challenge is adapting a PSO/SVR scheme to different applications. Every time series has different numerical bounds, in both the time (independent) and measured value (dependent) dimensions. One could also consider a multiple dimension input to the Support Vector, as SVR itself is designed to accommodate such input schemes. Flexibility in the PSO/SVR design is necessary to accommodate a wide variety of applications. CMPSO is a design that uses PSO as both the SVR free parameter optimizer as well as the QP solving algorithm. Additionally, this optimization framework is able to adapt (scale) the data of interest in order to extend its functionality to multiple applications without the need for “hand” tuning.

3.1 The CMPSO Framework

The goal of CMPSO is to estimate/predict an arbitrary one-dimensional time series using SVR estimation given in equation (7) in section 2.1. The data inputs to CMPSO are N real valued discrete inputs, $x_1 \ldots x_N$, which represent the independent time samples and a corresponding real valued discrete sample for each time step, $y_1 \ldots y_N$. Note the difference between each time step does not have to be uniform. The output of CMPSO is the estimate ($\hat{y}$) of an arbitrary time sequence. For a regression application, the minimum and maximum values of an arbitrary time variable would lie in the interval $[x_1, x_N]$ where $x_1 < x_2 < \ldots < x_N$. For a prediction application, the arbitrary time variable would have values greater than x_N.

In order to adapt CMPSO to multiple applications without the need for offline data pre-processing, the framework automatically adjusts the input time series such that [min(x), max(x)] lie in the interval [0.0, 0.9] for time series prediction past the last known time sample. For prediction applications, time values in the interval (0.9, 1.0] are used. For time series estimation (regression) applications, the input time series lies in the interval [0.0, 1.0]. The framework also automatically scales the independent variable samples such that $|y_i|$ is less than or equal to 1.0. After CMPSO has optimized the free parameters and estimated the scaled time series, the estimate will be re-scaled back to the original x and y dimensions at which point statistical analysis can be performed between the input values (or reference values) and the CMPSO generated estimate.

3.2 The PSO Algorithm Parameters

The PSO parameters are determined (partially) on the computational environment used for this research, specifically the number of particles that are used in the optimization process. More powerful or parallel processing systems may allow for more particles, less computational processing time and/or larger time series sample sets to be processed at once. A summary of CMPSO computational efficiency and computing environment is given in Section 4. For this research, a summary of PSO specific parameters are given in Table I. It should be mentioned that the determination of the three velocity weighting factors can be adjusted by the user and these values were chosen empirically for the data researched in this paper.

3.3 The SVR Algorithm Parameters

Given the objective of CMPSO to eliminate the need for a QP solver for each epoch k, the Lagrange multipliers and the bias term need to be added to the three free parameter for optimization (one Lagrange multiplier for each data sample and one bias term as outlined in section 2.1). Each particle's position in search space was defined in equation (8a) based on the given SVR variables above. One problem associated with using PSO to solve the SVR optimization problem is defining the Capacity term C. PSO (for CMPSO) requires constant limits for each of the variables, including the Lagrange multipliers. However, C is a variable that constrains the limit of the Lagrange multiplier as seen in the constraint equation in (5). This capacity term C can be viewed as a Lagrange multiplier scale factor. If C is viewed as such, it would be advantageous to remove this factor from the SVR constraint equation if possible, and as it turns out, the elimination of this variable as a constraint is part of the motivation to establish the CMPSO generic framework that scales the data to the same numerical bounds regardless of application.

Table 1: CMPSO – PSO Algorithm Specific Parameters

Parameter	Value	Notes
Number of Particles	500	Limited by computational hardware and software implementation
Number of Particles to Reset	490	Number of particles periodically reset to avoid stagnation
Inertial Weight w_I	0.75	Particle velocity weighting factor based on current particle velocity
Cognitive Weight w_C	0.75	Particle velocity weighting factor based on direction where particle found optimal result
Social Weight w_S	0.25	Particle velocity weighting factor based on direction where optimal result found for entire swarm
Epoch Limit	2000	One of two limits that when reached, halt CMPSO

The Capacity term C can be factored out of equation (4) and the constraints in (5) as follows:

$$\text{Maximize: } C \cdot \left(\sum_{i=1}^{N} \alpha_i y_i - \varepsilon \sum_{i=1}^{N} |\alpha_i| - \frac{C}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_i K(x_i, x_j) \right) \tag{11}$$

$$\text{Subject to: } -1 \leq \alpha_i \leq 1 \, , \sum_{i=1}^{N} \alpha_i = 0, C > 0 \tag{12}$$

As seen in (11), each Lagrange multiplier is scaled by a factor C (illustrated in the equation for completeness), and in (12), the constraint limits of the α_i variables have been set to constants value of +/- 1.0. This will allow for CMPSO to fix the bounds of the Lagrange multipliers and to treat the Capacity value C as a free parameter to optimize in a way that fits the CMPSO framework. This optimization problem is the same as the referenced SVR optimization problem, just with a change in the scaling of the Lagrange multipliers. Note in equations (11) and (12) that the bias term *b* is not present in solving for the Lagrange multipliers. For CMPSO (and SVR), it is not necessary to have the bias term as part of the free parameters to optimize. Assume from equation (7) that some optimal bias term b^* exists. One mathematical model to optimize this free parameter would be to minimize the error mean between the training values *y* and each resulting particle output as seen in equation (13):

$$\text{Minimize:} \quad \mathrm{E}[\mathrm{y} - \hat{\mathrm{y}}_\mathrm{i}] = \mathrm{b}_\mathrm{i}^* \tag{13}$$

This now allows for a single calculation for each particle at each epoch for the bias term *b*. The equations that define each particle position and velocity given in (8a) and (8b) then become equations (14a) and (14b) respectively:

$$\text{Particle } i \text{ Position:} \quad p_i(C, \varepsilon, \sigma, \alpha_1 \ldots \alpha_N) \tag{14a}$$

$$\text{Particle } i \text{ Velocity:} \quad v_i(C, \varepsilon, \sigma, \alpha_1 \ldots \alpha_N) \tag{14b}$$

The last two variables to consider for the SVR parameters are the gap value ε and the Kernel Function (an exponential kernel function for this research – see equation (6)) free parameter σ. Through empirical analysis, the boundaries for these parameters as well as the other variables defined in this section are detailed in Table II.

3.4 Particle Initialization and Constrained Motion

Each particle position and velocity defined in equations (14a) and (14b) must be initialized to some value. For capacity C, gap ε, and kernel constant σ, the initial values are selected from a uniform distribution within the bounds defined in Table II. However the bounds for the Lagrange multipliers have to be dealt with in a separate way. Although the bounds are set within the [-1.0, 1.0] limits, there is the additional constraint that the Lagrange multipliers must sum to zero (see dual constraint equation (5)). CMPSO will randomly select N-1 Lagrange multipliers to initialize. The values will be randomly selected from a uniform distribution between the [-1.0, 1.0] limits. The N^{th} Lagrange multiplier will be the negative of the sum of the N-1 randomly selected Lagrange multiplier values, satisfying the constraint. The exact same process is repeated for the initial velocity values (times a uniform scale value). For each particle, the candidate time series fit is now a feasible (but not necessarily optimal) candidate based on the SVR dual optimization constraints in equation (5). At this point, a fitness function is evaluated (see section E below) and the p_{best} value for each particle is stored. The maximum p_{best} value is then selected as the g_{best} value for the optimization.

As described in section 2.2, each particle must now move in search space searching for an optimal solution. The motion is based on a weighted sum of a particle's initial velocity, its direction towards its pbest value, and its direction towards the optimal place found by the swarm for each variable (gbest). The constraint equation in (5) must still hold after a particle moves. Since a particle's $\alpha_1 \dots \alpha_N$ velocity is equal to zero, and the pbest and gbest locations for the Lagrange multipliers also sum to zero, any movement of the weighted sum of these values also sums to zero. The particle motion for $\alpha_1 \dots \alpha_N$ is constrained to meet the constraint equation in (5). Therefore, every particle at every epoch will always contain a feasible solution to the SVR optimization problem. To ensure this constraint holds when a particle moves outside the Lagrange multiplier (or other variable) bounds, the Invisible Walls technique is used.

Table 2: CMPSO – SVR Specific Parameters

Parameter	Value Range	Notes
Capacity, C	[0.001, 2.0]	Set based on the scaling of the input data
Gap, ε	[0.02, 0.10]	Notional values – can be user defined
Kernel Constant, σ	[1e-5, 1e-3]	Set based on the scaling of the input data
Lagrange Multipliers , $\alpha_1 \dots \alpha_N$	[-1.0, 1.0]	Bounded by the factoring of Capacity C
Bias Term b	{unbounded}	Can be determined after Lagrange multipliers have been found

3.5 The Fitness Function and Iteration Bounds

The fitness function can be one of the most difficult tasks to complete in defining a PSO framework. For time series prediction and regression, there are several choices in evaluation metrics to select such as error mean or mean squared error, etc. However, the CMPSO framework is set up not only to find the best fit for time series data, but also to be a QP solver. This research has found that using the dual gap measure associated with SVR optimization convergence is sufficient to produce good results as published

in [2]. Equation (15) is the ratio of the difference of the primal and dual objectives defined in equations (2) and (4) respectively to the primal objective output plus one. This equation is referenced in [5] and [6] as finding the optimal Lagrange multipliers for the SVR problem:

$$\text{Fitness: } \frac{Primal\ Objective - Dual\ Objective}{Primal\ Objective + 1} \tag{15}$$

As mentioned earlier, one of the iteration stopping points is the number of epochs completed, which is set to 2000 for CMPSO (but can be changed based on user needs). This bound is essentially a computational time limit in case there are issues with any particular data series converging. For CMPSO, there are two numeric limits that will stop processing and produce a final output – both are based on the fitness function in equation (15). The ideal limit based on published recommendations in [5] and [6] is set to 0.001. If equation (15) is evaluated at this value or less, the entire optimization process is complete.

In the results published in [2], it has been found that a value of 0.25 for the fitness function limit is sufficient to stop the PSO optimization process in finding the values of C, ε, and σ. At this point these values are fixed and a separate QP program is executed once. For this research, Sequential Minimization Optimization (SMO) for SVR (an implementation based on [5] and [6]) is implemented. Since the Lagrange multipliers are feasible at every point in the CMPSO process, the SMO can be initially seeded with these values, although there is no guarantee of increased SMO performance between nonzero seeded SMO execution vs. zero seeding the Lagrange multipliers. Nevertheless, the worst case scenario means CMPSO will execute a QP program only once. It turns out that the computational impact is negligible as explained in the next section. Figure 1 illustrates the block diagram for the entire CMPSO process.

4 Computational Efficiency

One of the CMPSO objectives is to increase computational efficiency by eliminating the need for a QP solver. The CMPSO framework achieves this by constraining the motion of the particles in search space such that the SVR objective function constraints are always met for every particle at all times. This yields a feasible candidate result for every particle at every epoch and essentially reduces the search space. A simple experiment was set up to illustrate CMPSO computational efficiency. A small data set was used to estimate the best regression fit by a) CMPSO and b) a "separated" PSO and SVR implementation where each particle's Lagrange multipliers were computed to reach the duality gap limit of 0.001 for every epoch. The data set is a 41-point sample of the function defined in equation (16):

$$f(x) = \sin(0.2\pi x)\, e^{(0.1x)} + 0.5N(0,1) \tag{16}$$

Where, N(.) is a Normal distribution with zero mean and unit variance.

The same computing environment and functional software components were used for both CMPSO and the PSO/SVR "separated" approach – a Windows 7 (Intel) based stand-alone PC. The CMPSO framework, PSO and SVR functions are written in MATLAB and C ("mex" files called by MATLAB), where the exact same software was used for both approaches. MATLAB has the ability to measure the time to execute any particular block of code and this functionality was used in this experiment. For the "separated" approach, the particles would only optimize the SVR free parameters and SMO (for SVR) was used to solve for the Lagrange multipliers. Also, a range of number of particles, down to a minimum of 25, was used to compare results. A total of 300 runs were executed for both methodologies. Several key metrics were analyzed:

1. Time to execute entire algorithm
2. Number of runs that converged to dual gap value of 0.001
3. Normalized Mean Square Error (NMSE) of results

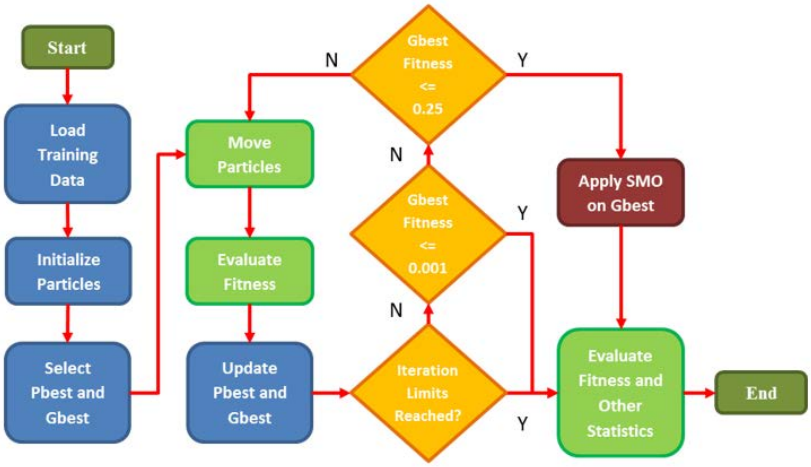


Figure 1 CMPSO Functional Block Diagram

Table III illustrates the results of the experiment. The experimental results show that 99% of the time CMPSO converged to a solution in approximately seven seconds, typically with a very small NMSE of less than 0.005. By contrast, running PSO and SVR separately, the solution never converged with the minimum number of particles (25) and it took more than five times longer for the PSO+SVR method to settle on incorrect results (essentially stagnate). If more particles are used in this method to try and improve the results, the processing time would only increase. This simple experiment shows that not only is CMPSO a viable algorithm for approximating time series functions, it is also more computationally efficient than functionally separated methods.

Table 3: CMPSO – Computational Efficiency

Measured Parameter	CMPSO	PSO+SVR	Notes
Mean Algorithm Execution Time (sec)	7.2749	39.0826	PSO+SVR used minimum of 25 particles
Percentage of Runs that Converged	99.0%	0.0%	PSO+SVR never converged to dual gap limit
NMSE	Typically less than 0.005	{N/A}	Since PSO+SVR never converged, NMSE was not calculated

5 Performance Results

CMPSO algorithm performance was measured against benchmark Mackey-Glass time series data as referenced in [2]. This data is highly nonlinear, but synthesized. In this research, we compared the CMPSO performance against a real world electrical load forecasting application as well as another nonlinear, synthesized data set from the Competition on Artificial Time Series (CATS). CMPSO performance was compared to the original competition contestants as well as more recent SVR and non-SVR based

algorithms that used the data sets post-competition. It is important to note that the CMPSO framework, by design, was not altered for both the presented results.

5.1 EUNITE Competition Data

In 2001, the European Network on Intelligent Technologies for Smart Adaptive Systems (EUNITE – see [29]) initiated a competition [30] to predict maximum electrical loads for the East-Slovakia Power Distribution Company. The goal is to estimate maximum daily power loads for the month of January 1999 based on the daily values of 1997 and 1998 of electrical load and temperature. Accurate predictions are measured by the Maximum Average Percent Error (MAPE) and Maximum Error (MAXE) as shown in equations (17) and (18):

$$MAPE = \frac{100}{31}\sum_{i=1}^{31}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{17}$$

$$MAXE = \max(|y_i - \hat{y}_i|)_{i=1\ldots31} \tag{18}$$

Where,
y_i= Actual Maximum Load for Month of January 1999
$\hat{y}_i$= Predicted Maximum Load

For this data set, the maximum number of data points is 31 as there are 31 days in the month of January. This data set requires some form of strategy for using CMPSO, as there are many data sets available for training (two years of power and temperature data respectively as well as the temperature for January 1999). For the purposes of showing the utility of CMPSO, the CMPSO framework not been altered, but two simple strategies were used on the input data based on distinct factors associated with the EUNITE data set:

1. There appears to be loading trends based on what day of the week it is. It is assumed that weekends tend to have less power demand than the Monday through Friday work week.
2. There is a relationship between the temperature and power consumption for any given day. As temperature decreases, power consumption increases.

Figure 2 shows the January 1997 and January 1998 maximum power loads, time aligned such that the two data sets are aligned by day of the week. There are trends surrounding the weekends vs. the work week, as noted by the relative maximums of five days duration at work week vs. the general minimums of two days duration at weekend that are clearly visible. This figure illustrates the first factor in the prediction strategy. To address the second factor, it is necessary to eliminate the trending in the data as a function of the day of week (assuming there is a relationship between temperature and day of week). This can be achieved by looking at the daily changes in power load and temperature between 1997 and 1998. Figure 3 is an illustration of this relationship. It is a scatter plot of the change in power load vs. the change in temperature for 360 days, time aligned to the day of week. A correlation coefficient of -0.5041 was computed on the 360 data points. This result means there is a moderate inverse correlation between power consumption and temperature. Given the relationship between power consumption and temperature, a simple linear approximation can be generated as shown in equation (19):

$$\Delta_P = -3.4668\Delta_T + 9.9573 \tag{19}$$

Where,

Δ_P = Change in maximum load for same day of the week between years (MW)
Δ_T = Change in temperature for the same day of the week between years (C)

This offset can now be applied to each day of the prediction, if the temperature of Jan 1999 is known (or in reality can be predicted reliably in the short term).

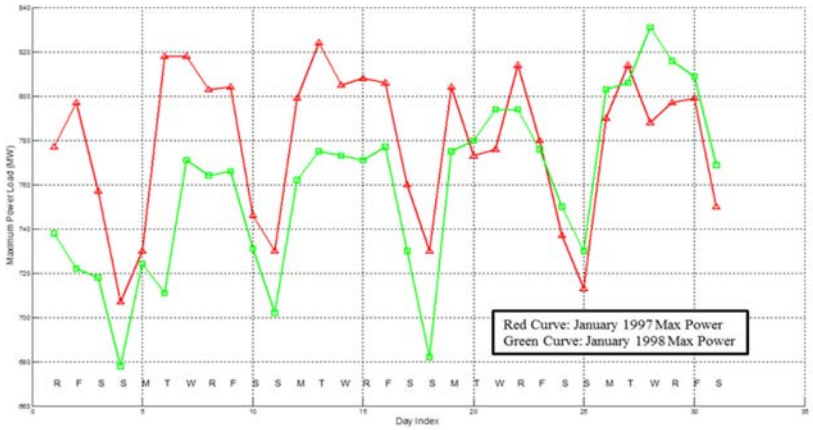


Figure 2: Maximum Power Load for January 1997 and 1998

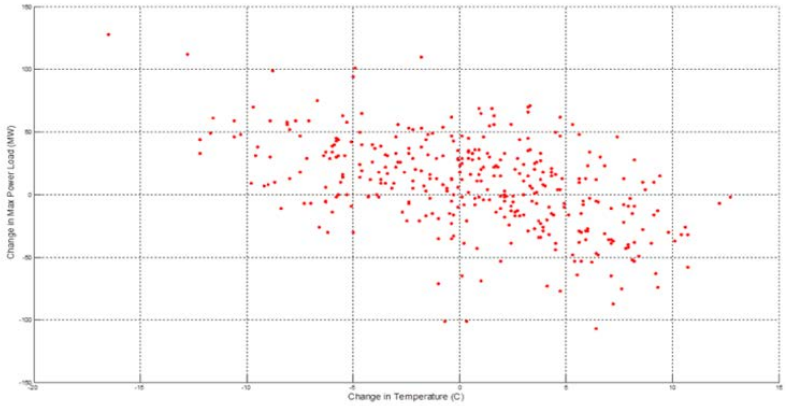


Figure 3: Relationship between changes in Power Load and Temperature for January 1997 and 1998

To predict the trending of the data, CMPSO was used to fit the mean value of the day-of-week aligned maximum power loads of January 1997 and January 1998. In addition, the output of CMPSO was offset by the linear approximation described in equation (19) on a day by day basis based on the difference in temperature between 1998 and 1999. Table IV shows the numerical results of the experiment.

Table 4: CMPSO – EUNITE Competition Results

Measured Parameter	CMPSO	CMPSO + Offset	Notes
MAPE	3.176	2.628	% Error
MAXE	84.931	66.653	Max Error in MW

As shown in Table IV, there is significant improvement by using the temperature data to offset the values. Figure 4 shows the final results of this experiment. As can be seen, CMPSO gives a good approximation

to the actual January 1999 load data (Blue Curve). The simple linear approximation is shown to improve with the offset in power due to temperature (Magenta Curve). In addition, Figure 5 is an approximation of where CMPSO (and the offset addition) would have ranked in the competition based on the values shown in Table V (referenced from [30]).

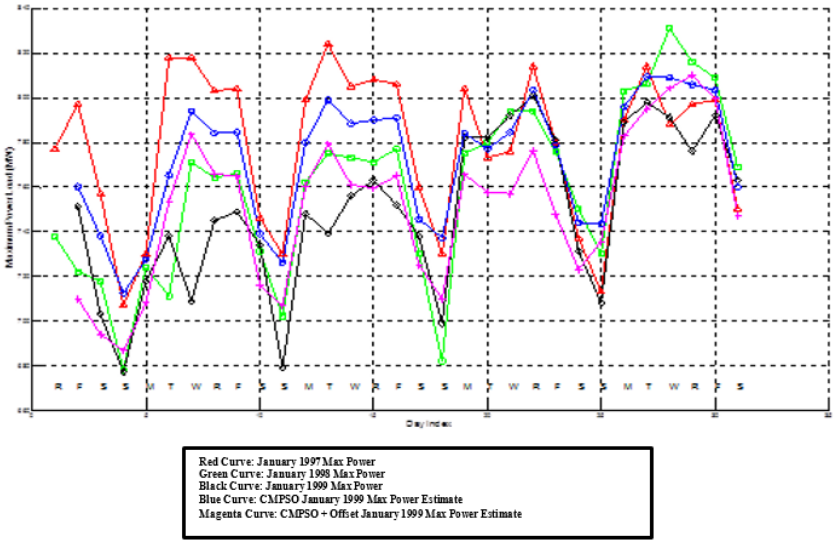


Figure 4: CMPSO EUNITE Results

As seen in Figure 5, an attempt was made to rank CMPSO and CMPSO with the delta offset with the other competitors (in the absence of the actual result values from each competitor). There are two sets of highlighted text boxes and lines for each of the CMPSO results. Based on MAPE, it appears CMPSO alone would have ranked at least sixth. In the case where the January 1999 temperature is included, CMPSO (plus the offset) would have ranked no worse than third based on MAPE. Note that the MAXE values were also close to the other competitors (refer to Table V). The winners of this competition, Chen *et al.* [31] used an SVM/SVR approach with multi-dimensional inputs to predict load. They also took into consideration additional attributes such as if the day being predicted is a holiday. The authors further reported details of their approach in [32], noting that there is no added value in using temperature data to predict the load. Although the weighting of electrical load for a holiday was not taken into consideration for this CMPSO benchmark evaluation, there was clearly an improvement in CMPSO prediction performance by modeling temperature and including the offset in the results.

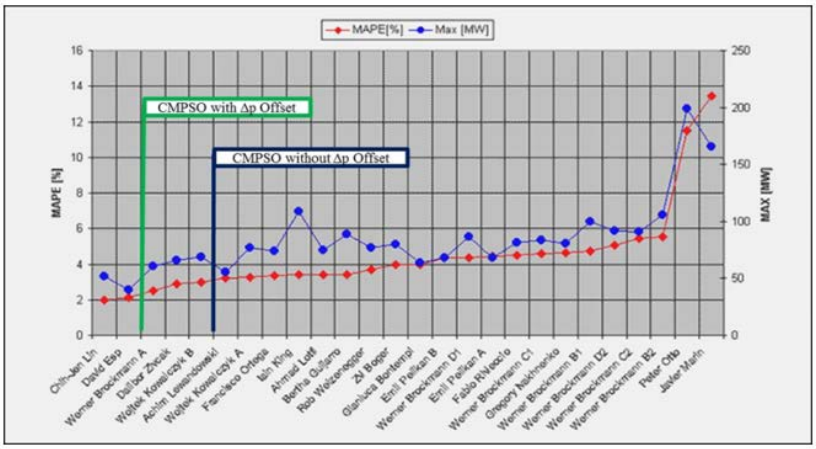


Figure 5: CMPSO EUNITE Results Compared to Other Contestants

In addition, CMPSO is compared to other recent SVM/SVR based approaches ([33] through [37]) that used EUNITE data as a benchmark. Table V shows the algorithm comparison by metric.

Table 5: CMPSO and SVR Based Algorithms – EUNITE Benchmark

Method	Metric		
	MAPE	MAXE	MSE
LS-SVM w/ k-nearest neighbor [37]	1.71	40.99	N/A
LS-SVM [34]	1.97	39.778	N/A
EMD-SVR [33]	1.98	N/A	284.3
CMSPO w/ Linear Offset	2.628	74.663	594.3
CMPSO	3.176	84.931	908.8
SVM Model [36]	3.67	N/A	N/A
AR Model [35]	6.69	N/A	N/A

5.2 CATS Data

In 2004 at the International Joint Conference on Neural Networks, a competition was organized to compare different prediction methods using a proposed benchmark CATS (Competition on Artificial Time Series) time series data of 5000 samples, among which 100 samples are missing [38]. The goal of the competition was to predict five sets of 20 missing data points each from the given data of 5000 samples and compare the methods using two criteria. The first was the Mean Square Error (MSE) - E1 for the four missing data gaps at sample indices 981 to 1000, 1981 to 2000, 2981 to 3000, and 3981 to 4000 in addition to the last omitted 20 data points with sample indices 4981 to 5000. The second was the MSE - E2 for the first four missing data gaps only. E2 was meant to only evaluate the effectiveness of the interpolation of a given algorithm versus the prediction estimate in E1 (the last 20 omitted samples).

Two different winners were selected based on the lowest E1 and E2 MSE scores. Sarkka *et al.* [39] used a Kalman Filter based approach with the lowest E1 MSE for all five missing data sets while Wichard and Ogorzalek [48] used an ensemble Neural Network based approach with the lowest E2 MSE for the first four missing data sets. The strategy for adapting CMPSO was straight forward for this particular application. For each of the first four data gaps, every tenth data sample was used. Half of the samples

were before the gap and the other half after the gap. In addition, ten consecutive samples were used just before and after the data gap for a total of 94 data points per set. For the last data set, a similar strategy of every 10 data samples was selected prior to the end of the data set along with the last 20 sequential samples for a total of 138 samples. Every sample was estimated between the first and last index of the data set and compared to the given results for evaluation.

CMPSO is compared to the top ten entries in the competition ([39] through [48]) as well as several other recent algorithm publications ([49] through [53]) that use CATS data as a benchmark. In all cases, CMPSO outperformed all of the other algorithms for this data set. Table VI shows the comparison of all the algorithms considered. Note the entries in Table VI with an (*) are recently developed approaches that used the benchmark CATS data as a performance indicator. Figures 6 and 7 illustrate CMPSO performance for two of the five data sets for estimation. Figure 6 is the second data set (gap) and Figure 7 is the last data set (prediction). The red curve is the provided competition data, the blue curve is the CMPSO estimation and the red circles are the actual CATS missing data for comparison. As seen from the results, CMPSO provides a very close fit to the missing competition data.

Table 6 CMPSO – CATS Benchmark

E1 MSE	E2 MSE	Model
113	140	CMPSO
143	129	ANN and AdaBoost* [50]; Single Global Model
262	239	ANN and AdaBoost* [50]; Multiple Local Models
287	N/A	Variance Minimization LS-SVM* [52]
390	288	Dynamic Factor Graphs* [53]
408	346	Kalman Smoother [39]
441	402	Recurrent Neural Networks [40]
502	418	Competitive Associative Net [41]
530	370	Weighted Bidirectional Multi-stream Extended Kalman Filter [42]
577	395	SVCA Model [43]
644	542	MultiGrid-Based Fuzzy System [44]
653	351	Double Quantized Forecasting Method [45]
660	442	Time-reversal Symmetry Method [46]
676	677	BYY Harmony Learning Based Mixture of Experts Model [47]
725	222	Ensemble Models [48]
1215	979	Deep Belief Network with Boltzmann Machines* (best value) [51]
2510	2450	ANN Based Approach* [49]

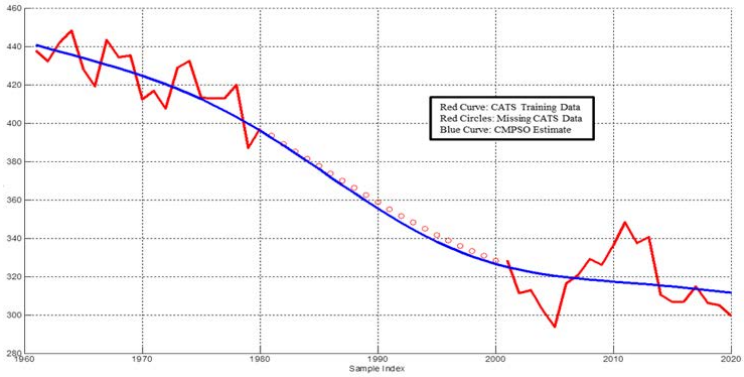


Figure 6: CMPSO Estimate for CATS Data Set 2

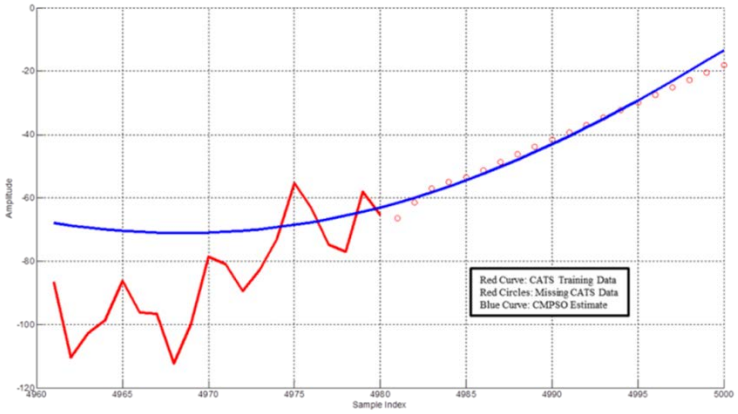


Figure 7: CMPSO Estimate for CATS Data Set 5

6 Conclusion

Support Vector Regression has been shown to be an effective algorithm for nonlinear time series regression and prediction applications across many different domains [1-2]. One of the major challenges associated with SVR is the ability to "tune" SVR to yield adequate results from application to application. Constrained Motion Particle Swarm Optimization (CMPSO) is a unique fusion of Support Vector Regression and Particle Swarm Optimization algorithms that attempts to optimize SVR free parameters while minimizing the need for excess computational resources by ensuring a feasible solution for every particle at every epoch. In addition, CMPSO is a standalone application that does not require user intervention regardless of data type. For the presented data, no algorithmic parameters were ever altered by the user.

CMPSO has performed well for a wide variety of nonlinear time series regression and prediction applications, including benchmark Mackey-Glass time series data [2]. EUNITE and CATS benchmark data results offer further proof of CMPSO performance as shown by comparison to other similar techniques. Based on the results presented in this paper, CMPSO is a viable, generalized approach to time series estimation and regression as compared to both SVM/SVR approaches as well as other Neural Network based algorithms. One unique feature associated with CMPSO is the reduction of the optimization space as the constraints associated with particle motion ensure a feasible solution for every particle at every epoch. Other PSO based SVR optimization techniques will require some form of QP solver for every candidate solution (i.e. every particle) at every epoch. This is an increase in computational efficiency.

Future research should include optimization of the PSO free parameters, choices of SVR Kernel Functions, different loss functions (such as used in LS-SVM), and the adaptation to multi-dimensional input data. It could also be conceivable to extend the CMPSO framework to include multiple models beyond SVR, where each particle could have sub-particles dedicated to any given regression/prediction algorithm.

ACKNOWLEDGEMENT

The authors would like to thank Dr. A. D. Snider, *Professor Emeritus* at the University of South Florida for his help in reviewing the manuscript and providing detailed comments to improve the quality of presentation.

REFERENCES

[1] N. Sapankevych and R. Sankar, "Time Series Prediction Using Support Vector Machines: A Survey", *IEEE Computational Intelligence*, vol. 4, no. 2, pp. 24-38, May 1999.

[2] N. Sapankevych and R. Sankar, "Constrained Motion Particle Swarm Optimization and Support Vector Regression for Non-Linear Time Series Regression and Prediction Applications", *12th International Conference on Machine Learning and Applications (ICMLA 2013)*, December 2013.

[3] V. N. Vapnik, "The Nature of Statistical Learning Theory", Springer, 1995.

[4] V. N. Vapnik, "Statistical Learning Theory", *John Wiley and Sons*, 1998.

[5] A. J. Smola and B. Schölkopf, "A Tutorial on Support Vector Regression", *NeuroCOLT Technical Report*, Royal Holloway College, London, UK, 1998.

[6] N. Cristianini and J. Shawe-Taylor, "An Introduction to Support Vector Machines and Other Kernel-based Learning Methods", *Cambridge University Press*, 2000.

[7] John C. Platt, "Sequential Minimization Optimization: A Fast Algorithm for Training Support Vector Machines", Technical Report MSR-TR-98-14, Microsoft Research, 1998.

[8] J. Kennedy and R. Eberhart, "Particle Swarm Optimization", *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, 1995.

[9] R. Poli, J. Kennedy, and T. Blackwell, "Particle Swarm Optimization: An Overview", *Swarm Intell*, vol. 1, pp. 33-57, 2007.

[10] J. Robinson and Y. Rahmat-Samii, "Particle Swarm Optimization in Electromagnetics", *IEEE Transactions on Antennas and Propagation*, vol. 52, no. 2, February 2004.

[11] R. Poli, "An Analysis of Publications on Particle Swarm Optimization Applications", *University of Essex, UK, Department of Computer Science, Technical Report CSM-469*, May 2007, Revised November 2007.

[12] M. Reyes-Sierra and C. A. C. Coello, "Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art," *International Journal of Computational Intelligence Research*, vol. 2, no. 3, pp. 287-308, 2006

[13] J. Fieldsend, "Multi–Objective Particle Swarm Optimization Methods", *Technical Report No. 419, Department of Computer Science, University of Exeter*, 2004.

[14] W.-C. Hong, "Chaotic Particle Swarm Optimization Algorithm in a Support Vector Regression Electric Load Forecasting Model", Energy Conversion and Management, vol. 50, no. 1, pp. 105-117, 2009.

[15] X. C. Guo, J. H. Yang, C. G. Wu, C. Y. Wang, and Y. C. Liang, "A Novel LS-SVMs Hyper-Parameter Selection Based on Particle Swarm Optimization", Neurocomputing, vol. 71, no. 16-18, pp. 3211-3215, 2008.

[16] H. Yuan, Y. Zhang, D. Zhang, and G. Yang, "A Modified Particle Swarm Optimization Algorithm for Support Vector Machine Training", *Proceedings of the 6th World Congress on Intelligent Control and Automation*, June 21-23, 2006.

[17] U. Paquest and A. P. Engelbrecht, "A New Particle Swarm Optimizer for Linearly Constrained Optimization", *Proceedings IEEE Congress on Evolutionary Computation*, pp. 227-233, 2003.

[18] U. Paquest and A. P. Engelbrecht, "Training Support Vector Machines with Particle Swarms", *Proceedings International Joint Conference on Neural Networks*, pp. 1593-1598, 2003.

[19] X. Wang, J. Lu and J. Liu, "Wavelet Transform and PSO Support Vector Machine Based approach for Time Series Forecasting", *International Conference on Artificial Intelligence and Computational Intelligence*, vol. 1, pp. 46-50, Nov 2009.

[20] J. Hu, P. Gao, Y. Yao, and X. Xie, "Traffic Flow Forecasting with Particle Swarm Optimization and Support Vector Regression", *IEEE 17th International Conference on Intelligent Transportation Systems,* pp. 2267-2268, October 2014.

[21] Y. Wen and Y. Chen, "Modified Parallel Cat Swarm Optimization in SVM Modeling for Short-term Cooling Load Forecasting", *Journal of Software*, vol. 9, no. 8, August 2014.

[22] X. Qiu, L. Zhang, Y. Ren, P. N. Suganthan, and G. Amaratunga, "Ensemble Deep Learning for Regression and Time Series Forecasting", *IEEE Symposium on Computational Intelligence in Ensemble Learning,* pp. 1-6, December 2014.

[23] C. Rajeswari, B. Sathiyabhama, S. Devendiran, and K. Manivannan, "Bearing Fault Diagnosis Using Wavelet Packet Transform, Hybrid PSO, and Support Vector Machine", *12th Global Congress on Manufacturing and Management – Procedia Engineering*, vol. 97, 2014.

[24] P. Shinde, T. Parvat, "Analysis on: Intrusions Detection Based on Support Vector Machine Optimized with Swarm Intelligence", *International Journal of Computer Science and Mobile Computing*, vol. 3, no. 12, pp. 559-566, December 2014.

[25] W. Wenhai, D. Jiandong, "Short-term Wind Power Forecasting Based on Maximum Correntropy Criterion", *International Conference on Power System Technology*, pp. 2800-2805, October 2014.

[26] Z. Hu, Q. Liu, Y. Tian, Y. Liao, "A Short-term Wind Speed Forecasting Model Based on Improved QPSO Optimizing LSSVM", *International Conference on Power System Technology*, pp. 2806-2811, October 2014.

[27] H. Dong, X. Zhu, Y. Liu, F. He, and G. Huo, "Iris Recognition Based on CPSO Algorithm for Optimizing Multichannel Gabor Parameters", *Journal of Computational Information Systems*, vol. 11, pp. 333-340, 2015.

[28] J. F. L. de Oliveira, T. B. Ludermir, "A Distributed PSO-ARIMA-SVR Hybrid System for Time Series Forecasting", *IEEE International Conferences on Systems, Man, and Cybernetics*, pp. 3867-3872, October 2014.

[29] http://www.eunite.org (January 2015)

[30] http://neuron-ai.tuke.sk/competition/ (January 2015)

[31] M.-W. Chang, B.-J. Chen, and C.-J. Lin, "EUNITE Network Competition: Electricity Load Forecasting", November 2001. (see [19]).

[32] B.-J. Chen, M.-W. Chang, and C.-J. Lin, "Load Forecasting Using Support Vector Machines: A Study on EUNITE Competition 2001", *IEEE Transactions on Power Systems*, vol. 19, no. 4, pp. 1821-1830, November 2004.

[33] B. Bican, Y. Yaslan, "A Hybrid Method for Time Series Prediction Using EMD and SVR", *6th International Conference on Communications, Control, and Signal Processing*, pp. 566-569, May 2014.

[34] X. Yang, "Comparison of the LS-SVM Based Load Forecasting Models", *International Conference on Electronic & Mechanical Engineering and Information Technology*, pp. 2942-2945, Aug 2011.

[35] I. Fernandez, C. E. Borges, and Y. K. Penya, "Efficient Building Load Forecasting", *IEEE 16th Conference on Emerging Technologies & Factory Automation*, pp. 1-8, September 2011.

[36] B. E. Turkay and D. Demren, "Electrical Load Forecasting Using Support Vector Machines", *7th International Conference on Electrical and Electronics Engineering*, vol. I, pp. 49-53, December 2011.

[37] T. T. Chen, S. J. Lee, "A Weighted LS-SVM Based Learning System for Time Series Forecasting", *Information Sciences*, pp. 99-116, 2015.

[38] Lendasse, E. Oja, O. Simula, and M. Verleysen, "Time Series Prediction Competition: The CATS Benchmark", *Proc. IEEE International Joint Conference on Neural Networks*, pp. 1615-1620, 25-29 July 2004.

[39] S. Sarkka, A. Vehtari, and J. Lampinen, "Time Series Prediction by Kalman Smoother with Cross Validated Noise Density", *Proc. IEEE International Joint Conference on Neural Networks*, vol. 2, pp. 1653-1657, 25-29 July 2004.

[40] X. Cai, N. Zhang, G. Venayagamoorthy and D. Wunsch, "Time Series Prediction with Recurrent Neural Networks Using a Hybrid PSO-EA Algorithm", *International Joint Conference on Neural Networks,* vol. 2, pp. 1647-1652, 2004.

[41] S. Kurogi, T. Ueno and M. Sawa, "Batch Learning Competitive Associative Net and Its Application to Time Series Prediction", *International Joint Conference on Neural Networks,* vol. 2, pp. 1591-1596, 2004.

[42] X. Hu and D. Wunsch, "IJCNN 2004 Challenge Problem: Time Series Prediction with a Weighted Bidirectional Multi-stream Extended Kalman Filter", *International Joint Conference on Neural Networks*, vol. 2, pp. 1641-1645, 2004.

[43] F. Palacios-Gonzalez, "A SVCA Model for The Competition on Artificial Time Series", *International Joint Conference on Neural Networks,* vol. 4, pp. 2777-2782, 2004.

[44] L. J. Herrera Maldonado, H. Pomares, I. Rojas, J. Gonzalez and M. Awad, "MultiGrid-Based Fuzzy Systems for Time Series Forecasting: CATS Benchmark IJCNN Competition", *International Joint Conference on Neural Networks*, vol. 2, pp. 1603-1608, 2004.

[45] G. Simon, J. A. Lee, M. Verleysen and M. Cottrell, "Double Quantization Forecasting Method for Filling Missing Data in the CATS Time Series", *International Joint Conference on Neural Networks*, vol. 2, pp. 1635-1640, 2004.

[46] P. F. Verdes, P. M. Granitto, M. I. Szeliga, A. Rebola and H. A. Ceccatto, "Prediction of the CATS benchmark exploiting time-reversal symmetry", *International Joint Conference on Neural Networks*, vol. 2, pp. 1631-1634, 2004.

[47] H.-W. Chan, W.-C. Leung, K.-C. Chiu and L. Xu, "BYY Harmony Learning Based Mixture of Experts Model for Non-stationary Time Series Prediction", *International Joint Conference on Neural Networks*, 2004.

[48] J. Wichard and M. Ogorzalek, "Time Series Prediction with Ensemble Models", *International Joint Conference on Neural Networks*, vol. 2, pp. 1625-1630, 2004.

[49] D. Samek and D. Manas, "Comparison of Artificial Neural Networks Using Prediction Benchmarking", *Proceedings of the 13th International Conference on Automatic Control, Modelling & Simulation*, pp. 152-157, 2011.

[50] Y. Dong, J. Zhang, and J. M. Garibaldi, "Neural Networks and AdaBoost Algorithm Based Ensemble Models for Enhance Forecasting of Nonlinear Time Series", *International Joint Conference on Neural Networks*, vol. 1, pp. 149-156, July 2014.

[51] T. Kuremoto, S. Kimura, K. Kobayashi, and M. Obayashi, "Time Series Forecasting Using a Deep Belief Network with Restricted Boltzmann Machines", *Neurocomputing*, pp. 47-56, 2014.

[52] R. Ormandi, "Applications of Support Vector-Based Learning", Research Group on Artificial Intelligence of the University of Szeged and the Hungarian Academy of Sciences, Ph. D. Dissertation, 2013.

[53] P. Mirowski, "Time Series Modeling with Hidden Variables and Gradient-Based Algorithms", Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, Ph. D. Dissertation, 2011.

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM
TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE
ISSN 2054-7390

Application of Artificial Neural Networks ANN and Adaptive Neuro Fuzzy Inference System ANFIS Models in Water Quality Simulation of Tigris River at Baghdad City

[1]Chelang A .Arslan, [2]Waleed M. Sh. Alabdraba, [3]Zainab B. Mohammed
[1]Assist. Prof. Civil Engineering Department Kirkuk University-Iraq
[2]Assist. Prof. Environmental Engineering Department Tikrit University-Iraq
[3]Constructions and Building Department, Technology University – Iraq
dr.chelenkalparslan@gmail.com; walabdraba@tu.edu.iq; zbm_asr@yahoo.com

ABSTRACT

In this paper two different types of artificial neural networks LMNN, SCGNN applied to simulate the total dissolved solids at of Tigris River at El-Wihda station using different water quality parameters data (pH, Temp., Hardness, Turbidity, EC, SO4, CL) at different stations upstream El-Wihda station. Different architecture and different input combinations with trying different numbers of neurons at the hidden layer. In addition, another application, which is an adaptive neuro fuzzy logic inference system ANFIS applied for the same purpose, the results shows that Even though the available data size is relatively small, reasonably a very good results found and a high performance obtained for the water quality prediction. Both ANN and ANFIS models show a very good performance in simulation of the TDS at the required station, and for the two types of ANNs, It can see that LMNN is better than SCGNN.

Keywords: Water quality, ANN, LMNN, SCGNN, ANFIS.

1 Introduction

Over the past two decades, the use of artificial neural networks has become widely used in environmental modeling, such as forecasting the quality and quantity of water in rivers and lakes. The use of this water for different purposes requires determining their suitability for specific use by matching their characteristics with the standards for each use. The aim of this study is to examine the possibility of using this technique in predicting the quality of the Tigris River during its passage in the city of Baghdad and matching the results with the field measurements based on available data for water quality in several sections along the river. Neuro Fuzzy method was used to classify water quality in China's major rivers. Data from 100 monitoring stations were collected over nine weeks for some physical chemical properties, including chemical oxygen demand, dissolved oxygen, and ammonia. Results showed that 89.59% of data could be properly classified using this method, and it is applicable to water quality assessment [1]. The nine-year data from 1990 to 1998 were used to calculate the concentration of dissolved oxygen based on the daily measurements of pH, discharge, temperature, and materials. In addition, the dissolved oxygen concentration in water quality assessment of reservoirs was determined using radial basis neural network (RBNN) and adaptive neuro-fuzzy inference system (ANFIS). The results showed that the RBNN method

DOI: 10.14738/tmlai.55.3511
Publication Date: 20th August, 2017
URL: http://dx.doi.org/10.14738/tmlai.55.3511

was better than the ANFIS method in calculating dissolved oxygen concentration[2]. ANN and ANFIS were used to model the dissolved oxygen concentration of the Khoram-Abad River in Iran for 12 months based on river discharge, flow velocity and dissolved oxygen concentration[3]. For calculating the concentration of dissolved oxygen, biochemical and chemical oxygen demand in the Karun River in Iran through the application of MLP, RBNN, and ANFIS. Nine water quality variables were used in the model. The results showed that RBNN and ANFIS were the best in predicting [4].

The ANN method was used to predict the concentration of total Dissolved solids in the Euphrates River in Iraq. The flow of the river, year, month, and the distance of the measurements station from upstream were adopted as inputs for the model used over 14 years. The results showed that the ANN method can be used to predict dissolved solids, and river flow affect the TDS concentration by 75% followed by 61% for the distance of the station, then 33% for the year of measurements, while the month had no more than 4%[5]. In a study carried out by [6] to evaluate water quality of reservoirs using the ANFIS method and RBNN method where the results proved that ANFIS is better than in the prediction of dissolved oxygen concentrations, total phosphorus, and chlorophyll.

2 Methodology

2.1 Concepts of Artificial Neural Networks

The concept of artificial neurons was produced in 1943 [7], and applications of ANNs in many fields started with the introduction of the back-propagation training (BP) algorithm for feed forward ANNs in 1986 [8]. An artificial Neural networks ANN are computing systems. Each system is composed from highly interconnected neurons which are simple information processing nodes or units .These neurons are gathering the information's (inputs) from both single and multiple sources then provides output which is related with the inputs by nonlinear relation. ANNs explain non-linear relations with different factors that are adjusted (calibrated or trained) in such a way that the ANN's output approximates the observed output on a data set. ANNs need quite enough amount of historical data to be trained upon satisfactory training, an ANN should be able to provide output for previously ''hidden" inputs. The main differences between the various types of ANNs involve network architecture and the method for determining the weights and functions for inputs and neurons this is what could be described as training operation [9].

2.1.1 Levenberg-Marquardt Method for Neural Networks Training

Many researches have been produced many methods to ensure the effeciency of backpropagation BP algorithm. All of these methods lead to low acceptable results. The Levenberg-Marquardt (LM) algorithm [10] ensued from development of BP (Back propogation) algorithm dependent methods. This method gives a good exchange between the speed of the Newton algorithm and the stability of the steepest descent method [11], that those are two basic theorems of LM algorithm. It is impossible to reduce error oscillation. Other effort with variable decay rate has been ensued to reduce error oscillation [10], but offered algorithm had low speed compared standard LM algorithm. LM can be described as a combination of steepest descent and the Gauss-Newton method. When the solution becomes far from the correct one, the algorithm behaves like a steepest descent method: slow, but guaranteed to converge. When the current solution is close to the correct solution, it becomes a Gauss-Newton method [12].

2.1.2 Scaled Conjugate Gradient Type Method With Convergence For Back Propagation Neural Network

Scaled Conjugate Gradient Method SCG is a supervised learning algorithm for feed forward neural networks. This method is used to reduce the error value between the result of the network and the real output or the observed one. It can be represented by following equation that indicates the quadratic approximation to the error E in a neighborhood of a point w by:

$$E_{qw}(y) = E(w) + E'(w)^T y + \frac{1}{2} y^T E''(w) y \quad (1)$$

The critical points for Eqw(y) must be found for calculating the minimum Eqw(y). The critical points are the solution to the linear system defined by Moller, [13] as:

$$E_{qw}(y) = E''(w)y + E'(w)y \quad (2)$$

The step size scaling mechanism, which is used in SCG, could reduce time consuming line search per learning iteration. This can increase the efficiency and performance [14].

2.2 Concepts of ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM (ANFIS)

At 1993, Jang proposed multi-layered neural network. The connections in these layers are not weighted or all weights equal to one [15]. ANFIS can be considered as an alternative manner that consists the advantages of two intelligent approaches neural network and fuzzy logic to ensure rational reasoning in quantity and quality. This new network has a high ability of training by means of neural networks and linguistic interpretation of variables via fuzzy logic. ANFIS implement a first order Sugeno style fuzzy system; it applies the rule of TSK Takagi Sugeno and Kang form in its architecture [16].

2.2.1 Traning of ANFIS (Learning Algorithm)

The membership function parameters of each input, the consequents parameters also number of rules are the most important parameters which should be tuned in an ANFIS model .The training process should consists two important steps which are Structure learning that ensures determining approopriate structure of the network that is, the best partitioning of number of membership functions for each input and number of rules. The second step is parametric learning whch suggest the membership functions and consequents parameters. The most common training algorithm is the hybrid learning. This algorithm is carried out in two steps: forward pass and backward pass, once all the parameters are initialized, in forward pass, functional signals go forward till fourth layer and the consequents parameters are identified by least square errors. After identifying consequents parameters, the functional signals keep going forward until the error measure is calculated. In the backward pass, the error rates propagate backward and the premise parameters are updated by gradient decent.

3 Case study and Nature of data

Tigris River is one of the two main rivers in Iraq. Its length is about 1850 km, while its length at Iraq is about 1418 km. This River is rising in the Taunus Mountains of Eastern Turkey. The river flows for about 400 km through Turkey before entering Iraq. It drains an area of 473 103 km^2.This area is shared with two other countries Turkey and Syria. About 58% of the basin lies in Iraq, and no major tributary joins the Tigris

south of Baghdad [17, 18]. There are more than eight water treatment stations on Tigris River. The well registered data are at stations El-karkh, Eastern of Tigris, El- wethba, El-kerame, El-qadisiye, El-dora, El-rasheed, El-wihda .These stations are at Baghdad city as they are shown in Figure (1). Different water quality parameters measurements are available at the stations .These data are observed as monthly data from 2009-2013. In this paper the observed values of water quality parameters which are selected after a brief correlation test to be Temperature , hardness, PH , Turbidity , Electrical conductivity , SO_4, Cl and total dissolved solids TDS at 9 stations were used in simulation operation . Seven of these stations were considered as upstream stations which are El-karkh, Eastern of Tigris, El- wethba ,El-kerame , El-qadisiye, El-dora, El-rasheed and the last station which is El-wihda st was considered as downstream station.

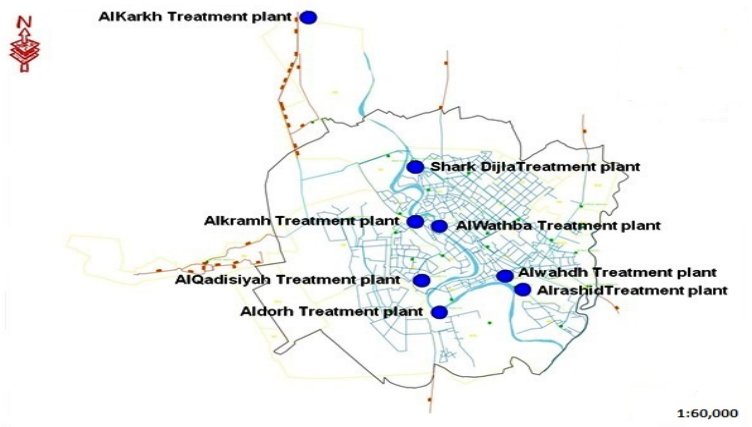


Figure (1) Study Area

3.1 Data Pre-processing

To ensure that all variables receive equal attention during the training process, they should be standardized. There many equations which provide a standardized data. For example, as the outputs of the logistic transfer function are between 0 and 1, the data can be scaled in the range 0.1–0.9 or 0.1–0.8. Before using the data for ANN, the data has been pre-processed (normalized). The same is true for adaptive neuro fuzzy system ANFIS since both models work best if the inputs are scaled to the same range of values [19]. In this study the best equation to standardized the data was found to be

$$Xi = 0.1 + 0.8\frac{(X - Xmin)}{(Xmax - Xmin)} \quad (3)$$

Xmin, Xmax are the minimum and the maximum values of the observed data series.

4 Results and discussion

Two types of artificial neural networks were tried in this study, which are LMNN, SCGNN to simulate the different parameters of water quality of Tigris River. Matlab (2014a, 8.3.0.532) neural network toolbox was used for the implementation of neural networks models. In both applied kinds of ANN, the data set was divided into two sets training and test. The training set was selected to be %60 from the whole data while the test set was chosen as %40 from the data. Different architectures were tried as different input combinations, which mean different water quality parameters at different sites. The output in all tried

models is selected to be Total dissolved solids at downstream of Tigris river which is at the station of Al-wihda. The number of hidden layer was selected to be one hidden layer with different number of neurons. Following Table (1) shows the different applied models of ANN for both types LMNN and SCGNN.

Table 1 Description of the applied ANN models.

Model Name	Inputs	Output
MI	(Tem,Har,pH,Turbidity,EC,SO4,CL,TDS) at 7 upstream stations &(Tem,Har,pH,Turbidity,EC,SO4,CL) at d/s station	TDS at Al-Wihda station
MII	(Tem,Har,PH,Turbidity,EC,SO4,CL) at Al-Wihda station	TDS at Al-Wihda station
MIII	(TDS)at all 7 u/s stations	TDS at Al-Wihda station

Tem: Temperature, Har: Hardness, EC: Electrical Conductivity, CL: chloride, TDS: Total dissolved solids.

4.1 Performance Parameters

Four statistical performance parameters, which are the determination coefficient (R^2), Nash-Sutcliffe efficiency (E_{Nash}), percent bias (R_{Bias}) and mean absolute percent error (MAPE), were used to assess the models' performances. These parameters are defined as:

$$R^2 = \frac{(\sum_{t=1}^{n}(A_t - A_{mean})(S_t - S_{mean}))^2}{\sum_{t=1}^{n}(A_t - A_{mean})^2 \sum_{t=1}^{n}(S_t - S_{mean})^2} \quad (4)$$

$$MAPE = \frac{1}{n}\sum_{t=1}^{n} \left| \frac{A_t - S_t}{A_t} \right|. \quad (5)$$

$$ENash = 1 - \frac{\sum_{t=1}^{n}(A_t - S_t)^2}{\sum_{t=1}^{n}(A_t - F_{mean})^2} \quad (6)$$

$$R\ bias\ = 100 * \frac{\sum_{t=1}^{n}(S_t - A_t)}{\sum_{t=1}^{n} A_t} \quad (7)$$

where A_t is the actual value and S_t is the Simulated value. S_{mean}, A_{mean} are the mean value of the series [20, 21]. The best value of R^2 is 1.0 while, the optimum value of R_{Bias} is 0.0 and a better description of R_{Bias} and E_{Nash} was given also by [22, 23], this description can be summarized as:

$Very\ Good(VG)$	$0.75 < E_{nash} \leq 1$	$R_{bias} < \pm 10$
$Good(G)$	$0.65 < E_{nash} \leq 0.75$	$\pm 10 \leq R_{bias} \leq \pm 15$
$Satisfactory(S\ )$	$0.5 < E_{nash} \leq 0.65$	$\pm 15 \leq R_{bias} \leq \pm 25$
UN$Satisfactory(US\ )$	$E_{nash} \leq 0.5$	$R_{bias} \geq \pm 25$

4.2 ANN application Results

For ANN development, a number of input combinations and input stations were tried, as it is clear from the above Table .This means that the input nodes for the three above models were 71 nodes, 7 nodes, 8 nodes respectively. The output layer had a single node which is the required TDS at d/s station Al-Wihda. During the training process the number of hidden layer nodes was tried to be less than (2* input layer nodes +1) [24]. The comparative performance of different ANN models based on E_{nash}, Coefficient of correlation R^2, R_{bias}, MAPE were calculated, and the selection of the best performance was depended on the best results of these parameters. Table (2) shows the results of performance parameters for the different ANN applied models. The Table concentrates on the best performance among all the applied models since different e number of neurons at the hidden layer were applied until reaching the best result.

Table 2 results of the performance parameters of different ANN applied models for Training Period

Model Name	Type of ANN	Inputs	Output	Best Architectur	E nash	R^2	R bias	MAPE
MI	LMNN	(Tem,Har,PH,Turbidity,EC,SO4,Cl,TDS) at 8 upstream stations &(Tem,Har,PH,Turbidity,EC,SO4,Cl) at d/s station	TDS at d/s station	63-22-1	0.87	0.88	-13.3	28.85
MII	LMNN	(Tem,Har,PH,Turbidity,EC,SO4,CL) at Al-Wihda station	TDS at d/s station	7-10-1	0.98	0.987	+9.1	22.53
MIII	LMNN	(TDS)at all 8 u/s stations	TDS at Al-Wihda station	7-14-1	0.93	0.941	-8.8	24.60
MI	SCGNN	(Tem,Har,PH,Turbidity,EC,SO4,CL,TDS) at 8 upstream stations &(Tem,Har,PH,Turbidity,EC,SO4,CL) at d/s station	TDS at d/s station	63-9-1	0.734	0.734	+16	33.62
MII	SCGNN	(Tem,Har,PH,Turbidity,EC,SO4,CL) at Al-Wihda station	TDS at d/s station	7-7-1	0.75	0.734	-17	38.85
MIII	SCGNN	(TDS)at all 8 u/s stations	TDS at d/s station	7-9-1	0.73	0.73	+16	38.24

Table (3) produces the results of the same process for test period, which is more important in deciding the success of any mathematical model. According to the above results, it is clear that the best result was found using Levenberg-Marquardt Method, since the performance is clearly reduced by using the Scaled conjugated gradient method in training the ANNs. The best input combinations was found to be for model MII with seven input nodes (Temperature ,Hardness, PH, Turbidity, Electrical conductivity , SO4, CL) at downstream station which is El-Wihda station. This was true for both kinds of the artificial neural networks. Table (3) also shows the best number of neurons at the hidden layer for all applied kinds and different input combinations.

Table 3 results of the performance parameters of different ANN applied models for Test Period

Model Name	Type of ANN	Inputs	Output	Best Architecture	E nash	R^2	R bias	MAPE
MI	LMNN	(Tem,Har,PH,Turbidity,EC,SO4,CL,TDS) at 8 upstream stations &(Tem,Har,PH,Turbidity,EC,SO4,CL) at d/s station	TDS at d/s station	63-22-1	0.863	0.861	-12.13	28.77
MII	LMNN	(Tem,Har,PH,Turbidity,EC,SO4,CL) at Al-Wihda station	TDS at d/s station	7-10-1	0.99	0.999	+7.1	22.67
MIII	LMNN	(TDS)at all 8 u/s stations	TDS at Al-Wihda station	7-14-1	0.993	0.993	-7.8	25.98
MI	SCGNN	(Tem,Har,PH,Turbidity,EC,SO4,CL,TDS) at 8 upstream stations &(Tem,Har,PH,Turbidity,EC,SO4,CL) at d/s station	TDS at d/s station	63-9-1	0.74	0.744	+17.1	34.61
MII	SCGNN	(Tem,Har,PH,Turbidity,EC,SO4,CL) at Al-Wihda station	TDS at d/s station	7-7-1	0.76	0.75	-17	38.86
MIII	SCGNN	(TDS)at all 8 u/s stations	TDS at d/s station	7-9-1	0.71	0.71	+15.3	39.21

The best number of neurons at the hidden layer for MII-LMNN type was found to be 10 neurons while for MII-SCGNN was 7 neurons. Values of R_{bias} or first successful model, which is MII–LMNN indicates to an overestimating while the best SCGNN model the R_{bias} value indicates to an underestimating. It is important to mention that all the tested architectures (different input combinations) for LMNN showed better performance if compared with SCGNN networks. Figure (2) represents the best ANN architecture that showed the best performance.

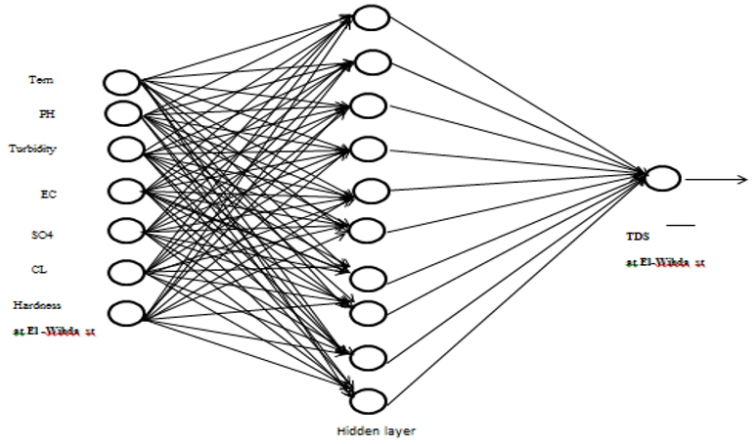


Figure 2 MII-LMNN architecture.

Figure (3) shows the comparison of the best three models with the observed data. The Figure represents the simulated TDS at A-Wihda station for test period by the successive models with the actual data, which means the observed values.

4.3 ANFIS application Results

To apply the adaptive neuro fuzzy interference system ANFIS models, the same architectures were used after applying the same manner in the standardization of data. Fuzzy system is based on the logical rules of premises and conclusions, which cannot be analyzed with the classical probability theories. On the other hand, artificial neural networks are capable of creating relevant relations between input and output variables through their learning capabilities based on various training patterns. A combined system of fuzzy inference and artificial neural network capable of using numerical data for predicting outputs can create a powerful tool that has come to be called the adaptive neural-fuzzy inference system. In this combined system, the fuzzy part establishes relations between input and output variables while the fuzzy membership functions are determined by its neural network part. By using the same input combinations and same training and test periods and by using 112 runs of the adaptive neuro fuzzy inference system with network segregation method, different results were found. This was done through using Matlab (2014a ,8.3.0.532) for network segregation . Tables (4),(5) show the results of application of ANFIS model on the selected of input combinations for the same training and test periods respectively. It is important to mention that the method of segregation was preferred on the partial clustering method due to the choice of type and number of membership functions, which can be determined by the user.

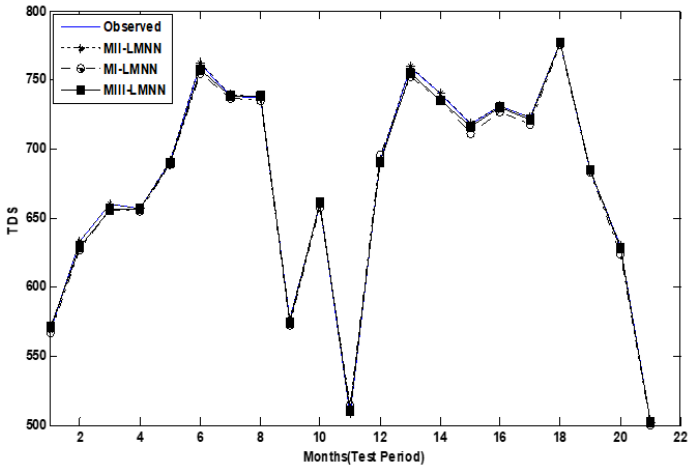


Figure 3 Comparison between the best three ANN models with the Observed data

Table 4 produces the results of the same process for test period

Model Name	Inputs	Output	Description	E nash	R^2	R_{bias}	MAPE
MI	(Tem,Har,PH,Turbidity,EC,SO4,CL,TDS) at 8 upstream stations &(Tem,Har,PH,Turbidity,EC,SO4,CL) at d/s station	TDS at d/s station	Segregation	0.739	0.729	-15.1	32.71
MII	(Tem,Har,PH,Turbidity,EC,SO4,CL) at Al-Wihda station	TDS at d/s station	Segregation	0.961	0.962	**+7.3**	24.007
MIII	(TDS)at all 8 u/s stations	TDS at Al-Wihda station	Segregation	0.851	0.85	-7.9	27.98

Table (5) results of the performance parameters of different ANFIS applied model for Training Period.

The approximate results to the previous ANN results were found since the same input combinations were found to be most successful combinations according to the same evaluation parameters. Figure (4) shows the comparison between the observed values and the simulated values by the best ANFIS model.

It can be seen that al the three models of ANFIS show a good performance in simulating of TDS values This indicates the capability of ANFIS models in making predictions on the basis of input data sets . The second model which inputs are Temperature, Hardness, PH, Turbidity, Electrical conductivity, SO4, CL at the downstream station exhibits a better performance compared to the other two . Comparison of the results obtained from the nine different models (six models ANN and three of ANFIS) revealed that increasing the number of input parameters dose not necessarily enhance model performance or increase the accuracy.

Table 5 results of the performance parameters of different ANFIS model for Test Period

Model Name	Inputs	Output	Description	E_{nash}	R^2	R_{bias}	MAPE
MI	(Tem,Har,PH,Turbidity,EC,SO4,CL,TDS) at 8 upstream stations &(Tem,Har,PH,Turbidity,EC,SO4,CL) at d/s station	TDS at d/s station	Segregation	0.729	0.729	-15.99	34
MII	(Tem,Har,PH,Turbidity,EC,SO4,CL) at Al-Wihda station	TDS at d/s station	Segregation	0.95	0.968	+8.3	**26**
MIII	(TDS)at all 8 u/s stations	TDS at Al-Wihda station	Segregation	0 843	0.849	-9.9	29.6

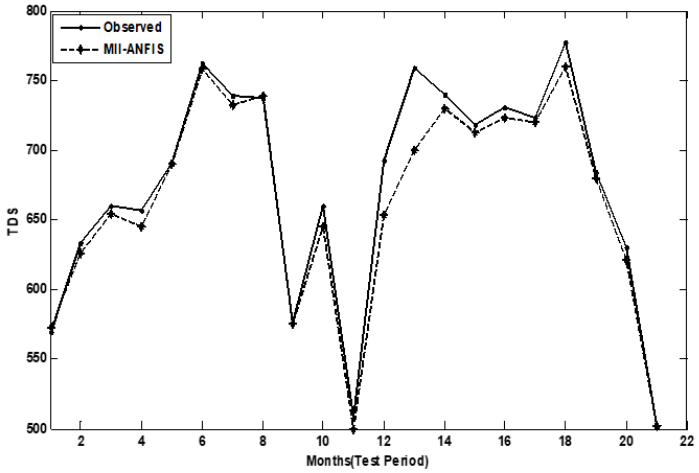


Figure 4 comparison between the best ANFIS (MII-ANFIS) model and the observed data

5 Conclusions

Followings are the most important conclusions, which found;

1. Even though the available data size is relatively small, reasonably a very good results found and a high performance obtained for the water quality prediction. If more data become available, better estimation can be using the successful models.
2. Both ANN and ANFIS models show a very good performance in simulation of the TDS at the required station
3. By comparing the two types of ANNs, It can see that LMNN is better than SCGNN.
4. Increasing the number of input parameters does not necessarily enhance model performance or increase the accuracy; this is true for both ANN models and ANFIS models.

REFERENCES

[1] Zhihong, H., Y. and Wang, H., Adaptive neuro fuzzy inference system for classification of water quality status. Journal of Environmental Sciences, 2010. 22(12): p. 1891-1896.

[2] Ozgur, KISI. And Murat, AY., Comparision of ANN and ANFIS techniques in modeling dissolved oxygen. Sixteenth International Water Technology Conference, IWTC-16 2012. Istanbul, Turkey.

[3] Ahour, M., et al., The study of artificial neural network (ANN} efficiency with neuro-fuzzy inference system (ANFIS) in dissolved oxygen simulation of river water. Bull. Env. Pharmacol. Life Sci., 2013. 2(9): p.30-38.

[4] Emamgholizadeh, S. et al., Prediction of water quality parameters of Karoon river (Iran) by artificial intelligence-based models. Int. J. Environ. Sci., 2014. 11: p.645-656.

[5] Thair, S. K., et al., Prediction of water quality of Euphrates river by using artificial neural network model (spatial and temporal study). International Research Journal of natural Sciences, 2014. 2(3): p.25-38.

[6] Chen, W. and Liu, W., Water quality modeling in reservoirs using multivariate linear regression and two neural network models. Advances in Artificial Neural Systems, 2015. Article ID 521721.

[7] McCulloch, W.S., and Pitts, W., A logical calculus of the ideas imminent in nervous activity. Bulletin and Mathematical Biophysics, 1943. 5: p.115–133.

[8] Rumelhart, D.E., et al, Learning internal representations by error propagation. In: Rumelhart, D.E., McClelland, J.L. (Eds.), Parallel Distributed Processing. 1986. MIT Press, Cambridge

[9] Caudill, M., and Butler, C., Understanding Neural Networks. Basic Networks, vol. 1. 1992. MIT Press, Cambridge, MA.

[10] Hagan, M. T. and Menhaj, M. B., Training feed forward network with the Marquardt algorithm. IEEE Trans. on Neural Net., 1994. 5(6): p.989-993.

[11] Battiti, R., First- and second-order methods for learning between steepest descent and Newton's method. Neural Computation, 1992. 4(2): p. 141-166.

[12] Levenberg, K., A Method for the Solution of Certain Non-linear Problems in Least Squares. Quarterly of Applied Mathematics, 1944. 2(2): p.164–168.

[13] Moller, A., Scaled Conjugate Gradient Algorithm for Fast Supervised Learning. Neural Networks. Water Resour. Res., 1993. 6(4): p.525-533.

[14] Orozco J, et al, Detecting Pathologies from Infant Cry Applying Scaled Conjugate Gradient Neural Networks. European Symposium on Artificial Neural Networks 23-25 April 2003: p.349-354

[15] Jang, J. S., ANFIS: Adaptive Network Based Fuzzy Inference Systems. IEEE Trans, Syst. Man Cybernet. 1993. 23(3): p.665-685.

[16] Takagi, K.I. and Sugeno, M., Fuzzy identification of systems and its applications to modeling and control. IEEE Trans. On Systems, Man and Cybern, 1985. 15(1): p.116- 132.

[17] Al-Ansari, N. A., et al, Evaluation of the water quality for the lower reaches of River Tigris using multivariate analysis. J. Water Res., 1986. 5: p.173–187.

[18] Al-Ansari, N. A., et al., Periodicity of selected water quality parameters for the Tigris water at Baghdad. J. Water Res., 1987. 6: p.11–17.

[19] Heydari M., et al., Development of a Neural Network Technique for Prediction of Water Quality Parameters in the Delaware River, Pennsylvania. Middle-East Journal of Scientific Research, 2013. 13(10): p.1367-1376.

[20] Chokmani, K. et al., Comparison of ice affected streamflow estimates computed using artificial neural networks and multiple regression techniques. Journal of Hydrology, 2008. (349): p.383 – 396.

[21] Teryaki S. et al., Comparison of artificial neural network and multiple linear regression models to predict optimum bonding strength of heat treated woods. International Journal of Adhesion & Adhesives, 2014. (55): p.29–36.

[22] Moriasi, D. N., et ai., Model evaluation guidelines for systematic quantifi cation of accuracy in watershed simulations. Trans. American Society of Agricultural and Biological Engineers ASABE, 2007. 50(3): p.885-900.

[23] Buyukyildis, M. and Arslan, Cheleng A., Comparison of Monthly Streamflow Forecasting Techniques. 9th word congress Water resources Management in Changing world :Challenges and opportunities, 2015. Istanbul Turkey.

[24] Swingler, K. Applying neural networks - A Practical guide. 1996. Academic Press Ltd., London.

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM
TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE
ISSN 2054-7390

A Computational Algorithm for Simultaneously Creating Alternatives to Optimal Solutions

[1] Julian Scott Yeomans
[1] OMIS Area, Schulich School of Business, York University, Toronto, ON, M3J 1P3 Canada;
syeomans@schulich.yorku.ca

ABSTRACT

In solving many practical mathematical programming applications, it is generally preferable to formulate several quantifiably good alternatives that provide distinct perspectives to the particular problem. This is because decision-making typically involves complex problems that are riddled with incompatible performance objectives and contain competing design requirements which are very difficult – if not impossible – to capture and quantify at the time that the supporting decision models are actually constructed. There are invariably unmodelled design issues, not apparent at the time of model construction, which can greatly impact the acceptability of the model's solutions. Consequently, it is preferable to generate several, distinct alternatives that provide multiple, disparate perspectives to the problem. These alternatives should possess near-optimal objective measures with respect to all known modelled objective(s), but be fundamentally different from each other in terms of their decision variables. This solution approach is referred to as modelling to generate-alternatives (MGA). This paper provides an efficient computational procedure for simultaneously generating multiple different alternatives to optimal solutions that employs the Firefly Algorithm. The efficacy of this approach will be illustrated using a well-known engineering optimization benchmark problem..

Keywords: Biologically-inspired Metaheuristics, Firefly Algorithm, Modelling to generate alternatives.

1 Introduction

Typical "real world" decision-making involves complex problems that possess design requirements which are frequently very difficult to incorporate into their supporting mathematical programming formulations and tend to be plagued by numerous unquantifiable components [1][2][3]. While mathematically optimal solutions provide the best answers to these modelled formulations, they are generally not the best solutions to the underlying real problems as there are invariably unmodelled aspects not apparent during the model construction phase [1][2]. Hence, it is generally considered desirable to generate a reasonable number of very different alternatives that provide multiple, contrasting perspectives to the specified problem [4]. These alternatives should preferably all possess near-optimal measures with respect to all of the modelled objective(s), but be as different as possible from each other in terms of the system structures characterized by their decision variables. Several approaches collectively referred to as modelling-to-generate-alternatives (MGA) have been developed in response to this multi-solution creation requirement [4]-[9].

DOI: 10.14738/tmlai.55.3580
Publication Date: 20th August, 2017
URL: http://dx.doi.org/10.14738/tmlai.55.3580

The primary motivation behind MGA is to construct a manageably small set of alternatives that are good with respect to all measured objective(s) yet are fundamentally dissimilar within the prescribed decision space. The resulting set of alternatives should provide diverse approaches that all perform similarly with respect to the known modelled objectives, yet very differently with respect to any unmodelled issues [3][10]. Clearly the decision-makers must then conduct a sub-sequent comprehensive comparison of these alternatives to determine which options would most closely satisfy their very specific circumstances. Therefore, MGA methods must necessarily be classified as decision support processes in contrast to the explicit solution determination methods assumed, in general, for optimization.

In this paper, it is shown how to simultaneously generate sets of maximally different solution alternatives by implementing a modified version of the nature-inspired Firefly Algorithm (FA) [10][11] by extending previous concurrent MGA approaches [6]-[9][12]-[14]. For calculation and optimization, it has been demonstrated that the FA is more computationally efficient than such commonly-employed metaheuristics as enhanced particle swarm optimization, simulated annealing, and genetic algorithms [11][15]. The MGA procedure extends the earlier approaches of Imanirad *et al*. [6]-[9][12][13] to now permit the simultaneous generation of the desired number of alternatives in a single computational run. This new simultaneous FA-based MGA procedure is extremely computationally efficient. This paper illustrates the efficacy of the new FA approach for simultaneously constructing multiple, good-but-very-different solution alternatives on a commonly evaluated benchmark engineering optimization test problem [16].

2 Firefly Algorithm For Optimization

While this section provides only a relatively brief synopsis of the FA procedure [5][12][13], more comprehensive explanations appear in [10][11]. The FA is a biologically-inspired, population-based metaheuristic. Each firefly in the population represents one potential solution to a problem and the population of fireflies should initially be distributed uniformly and randomly throughout the solution space. The solution approach employs three idealized rules. (i) The brightness of a firefly is determined by the overall landscape of the objective function. Namely, for a maximization problem, the brightness is simply considered to be proportional to the value of the objective function. (ii) The relative attractiveness between any two fireflies is directly proportional to their respective brightness. This implies that for any two flashing fireflies, the less bright firefly will always be inclined to move towards the brighter one. However, attractiveness and brightness both decrease as the relative distance between the fireflies increases. If there is no brighter firefly within its visible neighborhood, then the particular firefly will move about randomly. (iii) All fireflies within the population are considered unisex, so that any one firefly could potentially be attracted to any other firefly irrespective of their sex. Based upon these three rules, the basic operational steps of the FA can be summarized within the following pseudo-code [11].

Objective Function $F(\boldsymbol{X})$, $\boldsymbol{X} = (x_1, x_2, \ldots x_d)$

Generate the initial population of n fireflies, $\boldsymbol{X}_i$, $i = 1, 2, \ldots, n$

Light intensity I_i at $\boldsymbol{X}_i$ is determined by $F(\boldsymbol{X}_i)$

Define the light absorption coefficient γ

while (t < MaxGeneration)

for $i = 1: n$, all n fireflies

for $j = 1: n$,all n fireflies (inner loop)

if ($I_i < I_j$), Move firefly i towards j; **end if**

Vary attractiveness with distance r via $e^{-\gamma r}$

end for j

end for i

Rank the fireflies and find the current global best solution $\boldsymbol{G}^*$

end while

Postprocess the results

In the FA, there are two important issues to resolve: the formulation of attractiveness and the variation of light intensity. For simplicity, it can always be assumed that the attractiveness of a firefly is determined by its brightness which in turn is associated with its encoded objective function value. In the simplest case, the brightness of a firefly at a particular location $\boldsymbol{X}$ would be its calculated objective value $F(\boldsymbol{X})$. However, the attractiveness, β, between fireflies is relative and will vary with the distance r_{ij} between firefly i and firefly j. In addition, light intensity decreases with the distance from its source, and light is also absorbed in the media, so the attractiveness needs to vary with the degree of absorption. Consequently, the overall attractiveness of a firefly can be defined as

$$\beta = \beta_0 \exp(-\gamma r^2)$$

where β_0 is the attractiveness at distance $r = 0$ and γ is the fixed light absorption coefficient for the specific medium. If the distance r_{ij} between any two fireflies i and j located at $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$, respectively, is calculated using the Euclidean norm, then the movement of a firefly i that is attracted to another more attractive (i.e. brighter) firefly j is determined by

$$\boldsymbol{X}_i = \boldsymbol{X}_i + \beta_0 \exp(-\gamma (r_{ij})^2)(\boldsymbol{X}_i - \boldsymbol{X}_j) + \alpha \boldsymbol{\varepsilon}_i.$$

In this expression of movement, the second term is due to the relative attraction and the third term is a randomization component. Yang [11] indicates that α is a randomization parameter normally selected within the range [0,1] and $\boldsymbol{\varepsilon}_i$ is a vector of random numbers drawn from either a Gaussian or uniform (generally [-0.5,0.5]) distribution. It should be explicitly noted that this expression represents a random walk biased toward brighter fireflies and if $\beta_0 = 0$, it becomes a simple random walk. The parameter γ characterizes the variation of the attractiveness and its value determines the speed of the algorithm's convergence. For most applications, γ is typically set between 0.1 to 10 [11][15]. In any given optimization problem, for a very large number of fireflies $n >> k$, where k is the number of local optima, the initial locations of the n fireflies should be distributed relatively uniformly throughout the entire search space. As the FA proceeds, the fireflies begin to converge into all of the local optima (including the global ones).

Hence, by comparing the best solutions among all these optima, the global optima can easily be determined. Yang [11] proves that the FA will approach the global optima when $n \rightarrow \infty$ and the number of iterations t, is set so that $t >> 1$. In reality, the FA has been found to converge extremely quickly with n set in the range 20 to 50 [10][15].

3 Modelling To Generate Alternatives

Most optimization methods appearing in the mathematical programming literature have concentrated almost exclusively upon producing single optimal solutions to single-objective problem instances or, equivalently, generating noninferior solution sets to multi-objective formulations [2][3][5][12][13]. While such algorithms may efficiently generate solutions to the derived complex mathematical models, whether these outputs actually establish "best" approaches to the underlying real problems is questionable [1][2][5]. In most "real world" decision environments, there are innumerable system requirements and objectives that are never included or apparent in the decision formulation stage [1][3]. Furthermore, it may never be possible to explicitly incorporate all of the subjective components because there are frequently many incompatible, competing, design interpretations and, perhaps, adversarial stakeholders involved. Therefore most of the subjective aspects of a problem necessarily remain unquantified and unmodelled in the construction of the resultant decision models. This is a common occurrence in situations where final decisions are constructed based not only upon clearly stated and modelled objectives, but also upon more fundamentally subjective socio-political-economic goals and stakeholder preferences [4]. Numerous "real world" examples describing these types of incongruent modelling dualities are discussed in [5][17][18].

When unquantified issues and unmodelled objectives exist, non-conventional approaches are required that not only search the decision space for noninferior sets of solutions, but must also explore the decision space for discernibly *inferior* alternatives to the modelled problem. In particular, any search for good alternatives to problems known or suspected to contain unmodelled components must focus not only on the non-inferior solution set, but also necessarily on an explicit exploration of the problem's inferior decision space.

To illustrate the implications of an unmodelled objective on a decision search, assume that the optimal solution for a quantified, single-objective, maximization decision problem is $\boldsymbol{X}^*$ with corresponding objective value $Z1^*$. Now suppose that there exists a second, unmodelled, maximization objective $Z2$ that subjectively reflects some unquantifiable "political acceptability" component. Let the solution $\boldsymbol{X}^a$, belonging to the noninferior, 2-objective set, represent a potential best compromise solution if both objectives could somehow have been simultaneously evaluated by the decision-maker. While $\boldsymbol{X}^a$ might be viewed as the best compromise solution to the real problem, it would appear inferior to the solution $\boldsymbol{X}^*$ in the quantified mathematical model, since it must be the case that $Z1^a \leq Z1^*$. Consequently, when unmodelled objectives are factored into the decision making process, mathematically inferior solutions for the modelled problem can prove optimal to the underlying real problem. Therefore, when unmodelled objectives and unquantified issues might exist, different solution approaches are needed in order to not only search the decision space for the noninferior set of solutions, but also to simultaneously explore the decision space for inferior alternative solutions to the modelled problem. Population-based solution methods such as the FA permit concurrent searches throughout a feasible region and thus prove to be particularly adept procedures for searching through a problem's decision space.

The primary motivation behind MGA is to produce a manageably small set of alternatives that are quantifiably good with respect to the known modelled objectives yet are as different as possible from

each other in the decision space. The resulting alternatives are likely to provide truly different choices that all perform somewhat similarly with respect to the modelled objective(s) yet very differently with respect to any unknown unmodelled issues. By generating a set of good-but-different solutions, the decision-makers can explore desirable qualities within the alternatives that may prove to satisfactorily address the various unmodelled objectives to varying degrees of stakeholder acceptability.

In order to properly motivate an MGA search procedure, it is necessary to supply a more mathematically formal definition of the goals of the MGA process [4][6][14]. Suppose the optimal solution to an original mathematical model is $\boldsymbol{X}^*$ with objective value $\boldsymbol{Z}^* = F(\boldsymbol{X}^*)$. The following maximal difference model can then be solved to generate an alternative solution, $\boldsymbol{X}$, that is maximally different from $\boldsymbol{X}^*$:

$$\text{Maximize} \qquad \Delta(\boldsymbol{X}, \boldsymbol{X}^*) = \sum_i \; | X_i - X_i^* |$$

$$\text{Subject to:} \qquad \boldsymbol{X} \in D$$

$$| F(\boldsymbol{X}) - \boldsymbol{Z}^* | \leq T$$

where Δ represents some difference function (for clarity, shown as an absolute difference in this instance) and T is a targeted tolerance value specified relative to the problem's original optimal objective $\boldsymbol{Z}^*$. T is a user-supplied value that determines how much of the inferior region is to be explored in the search for acceptable alternative solutions. This difference function concept can be extended into a measure of difference between any set of alternatives by replacing $\boldsymbol{X}^*$ in the objective of the maximal difference model and calculating the overall sum (or some other function) of the differences of the pairwise comparisons between each pair of alternatives – subject to the condition that each alternative is feasible and falls within the specified tolerance constraint.

4 FA-based Simultaneous MGA Computational Algorithm

The MGA method to be introduced produces a pre-determined number of close-to-optimal, but maximally different alternatives, by modifying the value of the bound T in the maximal difference model and using an FA to solve the corresponding, maximal difference problem. Each solution within the FA's population contains one potential set of p different alternatives. By exploiting the co-evolutionary solution structure within the population of the algorithm, the Fireflies collectively evolve each solution toward sets of different local optima within the solution space. In this process, each desired solution alternative undergoes the common search procedure of the FA. However, the survival of solutions depends both upon how well the solutions perform with respect to the modelled objective(s) and by how far away they are from all of the other alternatives generated in the decision space.

A direct process for generating alternatives with the FA would be to iteratively solve the maximum difference model by incrementally updating the target T whenever a new alternative needs to be produced and then re-running the algorithm. This iterative approach would parallel the original Hop, Skip, and Jump (HSJ) MGA algorithm [5] in which, once an initial problem formulation has been optimized, supplementary alternatives are systematically created one-by-one through an incremental adjustment of the target constraint to force the sequential generation of the suboptimal solutions. While this approach is straightforward, it requires a repeated execution of the optimization algorithm [4][12][13].

To improve upon the stepwise alternative approach of the HSJ algorithm, a concurrent MGA technique was subsequently designed based upon the concept of co-evolution [6][8][12][13]. In the co-evolutionary approach, pre-specified stratified subpopulation ranges within the algorithm's overall population were

established that collectively evolved the search toward the creation of the specified number of maximally different alternatives. Each desired solution alternative was represented by each respective subpopulation and each subpopulation underwent the common processing operations of the FA. The survival of solutions in each subpopulation depended simultaneously upon how well the solutions perform with respect to the modelled objective(s) and by how far away they are from all of the other alternatives. Consequently, the evolution of solutions in each subpopulation toward local optima is directly influenced by those solutions contained in all of the other subpopulations, which forces the concurrent co-evolution of each subpopulation towards good but maximally distant regions within the decision space according to the maximal difference model [4].

By employing this co-evolutionary concept, it becomes possible to implement an FA-based MGA procedure that concurrently produces alternatives which possess objective function bounds that are somewhat analogous to those created by the sequential, iterative HSJ-styled solution generation approach. While each alternative produced by an HSJ procedure is maximally different only from the overall optimal solution (together with its bound on the objective value which is at least x% different from the best objective (i.e. x = 1%, 2%, etc.)), a concurrent procedure is able to generate alternatives that are no more than x% different from the overall optimal solution but with each one of these solutions being as maximally different as possible from every other generated alternative that was produced. Co-evolution is also much more efficient than the sequential HSJ-style approach in that it exploits the inherent population-based searches of FA procedures to concurrently generate the entire set of maximally different solutions using only a single population [6][8].

While a concurrent approach exploits the population-based nature of the FA's solution approach, the co-evolution process occurs within each of the stratified subpopulations. The maximal differences between solutions in different subpopulations is based upon aggregate subpopulation measures. Conversely, in the following simultaneous MGA algorithm, each solution in the population contains exactly one entire set of alternatives and the maximal difference is calculated only for that particular solution (i.e. the specific alternative set contained within that solution in the population). Hence, by the evolutionary nature of the FA search procedure, in the subsequent approach, the maximal difference is simultaneously calculated for the specific set of alternatives considered within each specific solution – and the need for concurrent subpopulation aggregation measures is circumvented.

The steps in the simultaneous co-evolutionary alternative generation algorithm are as follows:

Initialization Step. In this preliminary step, solve the original optimization problem to determine the optimal solution, $\boldsymbol{X^*}$. As with prior solution approaches [6]-[9][12][13] and without loss of generality, it is entirely possible to forego this step and construct the algorithm to find $\boldsymbol{X^*}$ as part of its solution processing. However, such a requirement increases the number of computational iterations of the overall procedure and the initial stages of the processing focus upon finding $\boldsymbol{X^*}$ while the other elements of each population solution remain essentially "computational overhead". Based upon the objective value $F(\boldsymbol{X^*})$, establish P target values. P represents the desired number of maximally different alternatives to be generated within prescribed target deviations from the $\boldsymbol{X^*}$.

Note: The value for P has to have been set *a priori* by the decision-maker.

Step 1. Create the initial population of size K in which each solution is divided into P equally-sized partitions. The size of each partition corresponds to the number of variables for the original optimization problem. $\boldsymbol{A}_p$ represents the p^{th} alternative, $p = 1,\ldots,P$, in each solution.

Step 2. In each of the K solutions, evaluate each $\boldsymbol{A}_p$, p = 1,…,P, with respect to the modelled objective. Alternatives meeting their target constraint and all other problem constraints are designated as *feasible*, while all other alternatives are designated as *infeasible*. A solution can only be designated as feasible if all of the alternatives contained within it are feasible.

Step 3. Apply an appropriate elitism operator to each solution to rank order the best individuals in the population. The best solution is the feasible solution containing the most distant set of alternatives in the decision space (the distance measure is defined in Step 5). Note: Because the best solution to date is always retained in the population throughout each iteration of the FA, at least one solution will always be feasible. A feasible solution for the first step can always consists of P repetitions of $\boldsymbol{X^*}$.

This step simultaneously selects a set of alternatives that respectively satisfy different values of the target T while being as far apart as possible (i.e. maximally different as defined in the maximal difference model) from the other solutions generated. By the co-evolutionary nature of the FA, the alternatives are *simultaneously* generated in one pass of the procedure rather than the P implementations suggested by the necessary increments to T in the maximal difference problem.

Step 4. Stop the algorithm if the termination criteria (such as maximum number of iterations or some measure of solution convergence) are met. Otherwise, proceed to Step 5.

Step 5. For each solution k = 1,…, K, calculate D_k, a distance measure between all of the alternatives contained within solution k.

As an illustrative example for determining a distance measure, calculate

$$D_k = \sum_{i=1toP} \quad \sum_{j=1toP} \quad \Delta\,(A_i, A_j).$$

This represents the total distance between all of the alternatives contained within solution k. Alternatively, the distance measure could be calculated by some other appropriately defined function.

Step 6. Rank the solutions according to the distance measure D_k objective – appropriately adjusted to incorporate any constraint violation penalties for infeasible solutions. The goal of maximal difference is to force alternatives to be as far apart as possible in the decision space from the alternatives of each of the partitions within each solution. This step orders the specific solutions by those solutions which contain the set of alternatives which are most distant from each other.

Step 7. Apply appropriate FA "change operations" to the each of the solutions and return to Step 2.

5 Computational Testing Of Simultaneous MGA Algorithm

As described earlier, "real world" decision-makers generally prefer to be able to select from a set of "near-optimal" alternatives that significantly differ from each other in terms of the system structures characterized by their decision variables. The ability of the FA MGA procedure to simultaneously produce such maximally different alternatives will be demonstrated using a non-linear spring design optimization problem taken from [16]. The design of a tension and compression spring has frequently been employed as a standard benchmark test problem for constrained engineering optimization algorithms [16]. The problem involves three design variables: (i) x_1, the wire diameter, (ii) x_2, the coil diameter, and (iii) x_3, the length of the coil. The aim is to essentially minimize the weight subject to constraints on deflection, stress, surge frequency and geometry. The mathematical formulation for this multimodal problem is:

Minimize F(***X***) = $x_1^2 x_2 \left(2 + x_3\right)$

Subject to:

$$g_1(\boldsymbol{X}) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leq 0$$

$$g_2(\boldsymbol{X}) = \frac{4x_2^2 - x_1 x_2}{12566\left(x_1^3 x_2 - x_1^4\right)} + \frac{1}{5108 x_1^2} - 1 \leq 0$$

$$g_3(\boldsymbol{X}) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \leq 0$$

$$g_4(\boldsymbol{X}) = \frac{x_1 + x_2}{1.5} - 1 \leq 0$$

$0.05 \leq x_1 \leq 2.0$ $\quad$ $0.25 \leq x_2 \leq 1.3$ $\quad$ $2.0 \leq x_3 \leq 15.0$

The optimal solution for the specific design parameters employed within this formulation is F(***X****) = 0.0127 with decision variable values of ***X**** = (0.051690, 0.356750, 11.287126) [16].

In order to create the set of different alternatives, extra target constraints that varied the value of *T* by up to 1.5% between successive alternatives were placed into the original formulation in order to force the generation of solutions maximally different from the initial optimal solution (i.e. the values of the bound were set at 1.5%, 3%, 4.5%, etc. for the respective alternatives). The MGA maximal difference algorithm described in the previous section was run to produce the optimal solution and the 10 maximally different solutions shown in Table 1.

Table 1. Objective Values and Solutions for the 11 Maximally Different Alternatives

Increment	1.5% Increment Between Alternatives			
	F(X)	x_1	x_2	x_3
Optimal	0.0127	0.05	0.3174	14.0324
Alternative 1	0.0128	0.05	0.3165	14.1598
Alternative 2	0.0128	0.0514	0.3472	12.0089
Alternative 3	0.0129	0.0529	0.3862	9.9684
Alternative 4	0.013	0.0521	0.3656	11.0667
Alternative 5	0.0131	0.0527	0.3766	10.5179
Alternative 6	0.0134	0.05	0.3157	14.978
Alternative 7	0.0137	0.052	0.3629	12.1615
Alternative 8	0.0140	0.0557	0.4307	9.5783
Alternative 9	0.0143	0.0542	0.4014	11.6481
Alternative 10	0.0146	0.0546	0.4247	10.7556

As described earlier, most "real world" optimization applications tend to be riddled with incongruent performance requirements that are exceedingly difficult to quantify. Consequently, it is preferable to create a set of quantifiably good alternatives that provide very different perspectives to the potentially unmodelled performance design issues during the policy formulation stage. The unique performance features captured within these dissimilar alternatives can result in very different system performance with respect to the unmodelled issues, hopefully thereby addressing some of the unmodelled issues into the actual solution process.

The example in this section underscores how a co-evolutionary MGA modelling perspective can be used to simultaneously generate multiple alternatives that satisfy known system performance criteria according to the prespecified bounds and yet remain as maximally different from each other as possible in the decision space. In addition to its alternative generating capabilities, the FA component of the MGA approach simultaneously performs extremely well with respect to its role in function optimization. It should be explicitly noted that the cost of the overall best solution produced by the MGA procedure is indistinguishable from the one determined in [16].

6 Conclusion

"Real world" decision-making problems generally possess multidimensional performance specifications that are compounded by incompatible performance objectives and unquantifiable modelling features. These problems usually contain incongruent design requirements which are very difficult – if not impossible – to capture at the time that supporting decision models are formulated. Consequently, there are invariably unmodelled problem facets, not apparent during the model construction, that can greatly impact the acceptability of the model's solutions to those end users that must actually implement the solution. These uncertain and competing dimensions force decision-makers to integrate many conflicting sources into their decision process prior to final solution construction. Faced with such incongruencies, it is unlikely that any single solution could ever be constructed that simultaneously satisfies all of the ambiguous system requirements without some significant counterbalancing involving numerous tradeoffs. Therefore, any ancillary modelling techniques used to support decision formulation have to somehow simultaneously account for all of these features while being flexible enough to encapsulate the impacts from the inherent planning uncertainties.

In this paper, an MGA procedure was presented that demonstrated how the population structures of a computationally efficient FA could be exploited to simultaneously generate multiple, maximally different, near-best alternatives. In this MGA capacity, the approach produces numerous solutions possessing the requisite structural characteristics, with each generated alternative guaranteeing a very different perspective to the problem. The computational example has demonstrated several important findings with respect to the simultaneous FA-based MGA method: (i) The co-evolutionary capabilities within the FA can be exploited to generate more good alternatives than planners would be able to create using other MGA approaches because of the evolving nature of its population-based solution searches; (ii) By the design of the MGA algorithm, the alternatives generated are good for planning purposes since all of their structures will be maximally different from each other (i.e. these differences are not just simply different from the overall optimal solution as in an HSJ-style approach to MGA); and, (iv) The approach is computationally efficient since it need only be run a single time in order to generate its entire set of multiple, good solution alternatives (i.e. to generate *n* solution alternatives, the MGA algorithm needs to run exactly once irrespective of the value of *n*). Since FA techniques can be modified to solve a wide variety

of problem types, the practicality of this MGA approach can clearly be extended into numerous disparate planning applications. These extensions will be studied in future research.

REFERENCES

[1] Brugnach, M., A. Tagg,, F. Keil, and W.J. De Lange, *Uncertainty matters: computer models at the science-policy interface*. Water Resources Management, 2007. 21: p. 1075-1090.

[2] Janssen, J.A.E.B., M.S. Krol, R.M.J. Schielen, and A.Y Hoekstra, *The effect of modelling quantified expert knowledge and uncertainty information on model based decision making*. Environmental Science and Policy, 2010. 13(3): p. 229-238.

[3] Walker, W.E., P. Harremoes, J. Rotmans, J.P. Van der Sluis, M.B.A. Van Asselt, P. Janssen, and M.P. Krayer von Krauss, *Defining uncertainty – a conceptual basis for uncertainty management in model-based decision support*. Integrated Assessment, 2003. 4(1): p. 5-17.

[4] Yeomans, J.S., and Y Gunalay, *Simulation-Optimization Techniques for Modelling to Generate Alternatives in Waste Management Planning*. Journal of Applied Operational Research, 2011. 3(1): p. 23-35.

[5] Brill, E.D., S.Y. Chang, and L.D Hopkins, *Modelling to generate alternatives: the HSJ approach and an illustration using a problem in land use planning*. Management Science. 1982. 28(3): p. 221-235.

[6] Imanirad, R., and J.S. Yeomans, *Modelling to Generate Alternatives Using Biologically Inspired Algorithms*. in *Swarm Intelligence and Bio-Inspired Computation: Theory and Applications*, X.S. Yang, Editor 2013. Amsterdam: Elsevier (Netherlands). p. 313-333.

[7] Imanirad, R., X.S. Yang, and J.S. Yeomans, *Modelling-to-Generate-Alternatives Via the Firefly Algorithm*. Journal of Applied Operational Research. 2013. 5(1): p. 14-21.

[8] Imanirad, R., X.S. Yang, and J.S. Yeomans, *A Concurrent Modelling to Generate Alternatives Approach Using the Firefly Algorithm*. International Journal of Decision Support System Technology. 2013, 5(2): p. 33-45.

[9] Imanirad, R., X.S. Yang, and J.S. Yeomans, *A Biologically-Inspired Metaheuristic Procedure for Modelling-to-Generate-Alternatives*. International Journal of Engineering Research and Applications. 2013, 3(2): p. 1677-1686.

[10] Yang, X.S., *Firefly Algorithms for Multimodal Optimization.* Lecture Notes in Computer Science. 2009, 5792: p. 169-178.

[11] Yang, X.S., Nature-Inspired Metaheuristic Algorithms 2nd Edition 2010, Frome: Luniver Press (United Kingdom).

[12] Imanirad, R., X.S. Yang, and J.S. Yeomans, *A Computationally Efficient, Biologically-Inspired Modelling-to-Generate-Alternatives Method*. Journal on Computing. 2012, 2(2): p. 43-47.

[13] Imanirad, R., X.S. Yang, and J.S. Yeomans, *A Co-evolutionary, Nature-Inspired Algorithm for the Concurrent Generation of Alternatives*. Journal on Computing. 2012, 2(3): p. 101-106.

[14] Yeomans, J.S., *An Efficient Computational Procedure for Simultaneously Generating Alternatives to an Optimal Solution Using the Firefly Algorithm*, in *Nature-Inspired Algorithms and Applied Optimization*, Yang, X.S. Editor 2018. Heidelberg (Springer), Germany. Forthcoming.

[15] Gandomi, A.H., X.S. Yang, and A.H Alavi, *Mixed Variable Structural Optimization Using Firefly Algorithm*. Computers and Structures. 2011, 89(23-24): p. 2325-2336.

[16] Cagnina, L.C., C.A. Esquivel, and C.A Coello, *Solving Engineering Optimization Problems with the Simple Constrained Particle Swarm Optimize*r. Informatica. 2008, 32: p. 319-326.

[17] Baugh, J.W., S.C. Caldwell, and E.D Brill, *A Mathematical Programming Approach for Generating Alternatives in Discrete Structural Optimization*. Engineering Optimization. 1997, 28(1): p. 1-31.

[18] Zechman, E.M., and S.R. Ranjithan, *An Evolutionary Algorithm to Generate Alternatives (EAGA) for Engineering Optimization Problems*. Engineering Optimization. 2004,. 36(5): p. 539-553.

SOCIETY FOR SCIENCE AND EDUCATION UNITED KINGDOM
TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE
ISSN 2054-7390

Urban Flood Forecast using Machine Learning on Real Time Sensor Data

[1] Likith Ponnanna P B, [2] R Bhakthavathsalam, [3]K Vishruth
[1,2]Supercomputer Education and Research Centre, Indian Institute of Science, Bangalore, India;
[3]College of Computer and Information Science, Northeastern University, Boston, USA
likith.april13@gmail.com; bhaktha@serc.iisc.in; krishnaprasad.v@husky.neu.edu

ABSTRACT

All the underpasses, flyovers and drainage networks in the urban areas are designed to manage a maximum rainfall. This situation implies an accepted flood risk for any greater rainfall event. This threat is very often underestimated as components such as climate change is disregarded. But even great structural alterations cannot assure that urban flood control precautions would be able to cope with all future rainfall events. Hence, being readily able to forecast city or urban floods in real time is one of the main tasks of this forecast. The current Urban flood forecasting methods involve the use of Geographical Information Systems techniques. Even though, these systems allow to detect and model the flood patterns in a larger perspective. They cannot pin point precise location behavior. Machine Learning models in conjunction with a sensor network can be essential elements of urban flood forecast systems, as an active part of the system or as study tools. The paper goes into the application of machine learning models to better predict flood pattern based on several external factors in real time.

Keywords: Urban Flooding, Flood Forecasting, Machine Learning, Real time Machine Learning.

1 Introduction

As the cities grow at a rapid pace, the city planning usually tends to fall behind. Extreme amount of concretization of these cities has brought up huge issues in the cities flood drains. These issues will be increasing as the concretization increases. Since a huge amount of time and investment is required to make better drainage systems some of the government bodies tend to ignore the issue until it becomes too big to handle. This can cripple most cities. As a result of this, a lot of the poorly planned cities all over the world have huge urban flooding problems. Where even a small rainfall brings the city traffic to a halt. And a larger rainfall causes the cities to come to a complete standstill. Which can be seen in the cases of 25 July 2005 in the city of Mumbai. The government surely would look into the issue had there been an economical alternative to detect, forecast the urban floods [1]. This, in turn, helps the government body to plan strategically and avoid unnecessary spending for projects with lesser overall effect on the flooding patterns, which drives down the cost of planning and implementation by placing the sensors in strategic locations.

The majority of the current solution is based on satellite data [2] using 3D Modelling techniques. Although these solutions have a considerable ability to detect flooding at a bigger area such as a completely

DOI: 10.14738/tmlai.55.3552
Publication Date: 20th August, 2017
URL: http://dx.doi.org/10.14738/tmlai.55.3552

submerged area, they cannot be used for mapping highly localized flooding such as a flyover or an underpass. Also, having these data collections at a highly localized level can help in detecting and fixing fault areas and could help in times where major disaster could be avoided by proper planning, as seen in the case of Chennai [3], India 2015 floods.

2 Setup

The hardware setup comprises of multiple sensors whose readings are handled by a micro-controller in real-time and pushes it to a hosted database. The different sensors [4] are as follows:

2.1 Ultrasonic Sensor

The ultrasonic sensor takes real-time readings of water level over time. The ping generations of the sensor are increased when the water level is found to be increased so that more granular data is available for the modelling. The ultrasonic sensor calculates the distance between itself and the ground. The initial distance and measurements are Pre-calibrated. The trigger part of the sensor bounces ultrasonic waves to the ground surface and waits to receive the echo back. Based on the time taken for the ultrasonic waves to travel back and forth the distance is measured which gets altered in case of a water level rise.

2.2 Thermometer

The thermometer is used to consider the temperature variations that are caused during different scenarios such as rain, heat, also helps to indicate if the flood was formed locally or was it carried from a different source. This helps in checking the slight temperature variations in conjunction with other variables in case of rain, snow, mist, etc., which might affect the precipitation values.

2.3 Barometer

A barometer is used to detect the atmospheric pressure. There is a large amount of correlation between atmospheric pressure and rainfall in a region [5]. Especially in times of cyclones or hurricanes if a very low-pressure area is developed the imminent rain can be sensed beforehand. It helps detect the possibilities of flooding in response to the rainfall.

2.4 Hygrometer

Hygrometer will work in conjunction with the other sensors to effectively monitor rainfall and other correlations driven from the humidity readings. Usually, there is a steep increase in the humidity at a location when there is rainfall.

2.5 Precipitation Sensor

The precipitation sensor is used to measure the rain intensity at any given time. The rain intensity gives an additional dimension when it comes to forecasting the water settlement in areas. In places where other precipitation parameters such as snow and mist are lower, a rain gauge can be used in its place to just monitor the rainfall intensity over time.

2.6 Ammeter paired to Solar Panel

Since cloud cover and day-night cycles affect the production of electricity by a solar panel. An ammeter is placed in conjunction with the solar panel to measure the intensity of the current, which helps in mapping the cloud cover and night cycles.

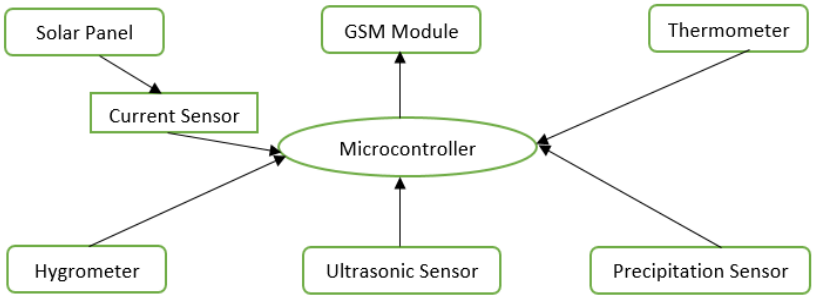


Figure 1. Hardware Setup

3 Methodology

3.1 Dimensionality Reduction

After the features required for the forecast is carefully chosen based on weather data relations. It is observed that the data is in a high dimensional space. To reduce this higher dimensional space to tangible few dimensions, feature extraction techniques are used. As we see the data is taken in real time the data scale is ever growing. For larger dataset, Linear Discriminant Analysis (LDA) is a much more suitable dimensionality reduction technique when compared to Principal Component Analysis (PCA) [6]. From *n* independent variables in the dataset, LDA extracts a more manageable number of new independent variables that separate the most, the classes of dependent variables.

Since there are more than two classes and the LDA is generalized from the Fisher discriminant. The Fisher discriminant could be extended to detect a subspace which encompasses all the class divergences. If each of the C classes has a mean μ_i and the same covariance Σ. The dispersion in between class variability may be outlined by the sample co-variance of the class means [7]

$$\Sigma_b = \frac{1}{c}\sum_{i=1}^{c}[\mu_i - \mu][\mu_i - \mu]^T$$

Where μ - average of class means. The class divergence in a particular direction $\vec{\omega}$ is provided by

$$s = \frac{\vec{\omega}\sum_b^T \vec{\omega}}{\vec{\omega}^T \Sigma \vec{\omega}}$$

$\vec{\omega}$ is an eigenvector of $\Sigma^{-1}\Sigma_b$ separation.

3.2 Classification Techniques

The dataset is run with multiple types of classification techniques which help in detecting the accuracy of different models. Using the accuracy check we can finalize a particular technique to maximize correct predictions.

3.2.1 K-nearest Neighbors(k-NN)

The k-NN is a classification technique, that is a nonparametric method and it makes its predictions based on the target outputs of its k nearest neighbors of any given point to be queried. Euclidean distance is calculated between every point and the training set points. The closest k training points are selected and prediction is taken as the average of the target output values of those k-points [8].

3.2.2 Kernel SVM

Kernel methods are used for pattern analysis. One of their members is support vector machines (SVM). The kernel methods utilize kernel functions to ply in higher dimensional and implicit area and without ever evaluating coordinates of the different data points. It in place uses internal products in between the images of all the data pairs in the feature or characteristic space [9].

3.2.3 Decision Tree Classification

This classification allows to build classification model based on the tree structure. It initially breaks the dataset into smaller subsets and in conjunction an associated decision tree inbuilt step by step. The result obtained is a tree with decision and leaf nodes [10].

3.2.4 Random Forest Classification

The Random Forest Classification develops lots of decision tree based on random selection of data and variables. The class of the dependent variable is based on multiple trees [11].

4 Implementation

4.1 Dataset

Using the hardware setup given above data collection was carried out by multiple sensors in different parts of the city of Bengaluru. This Data stream was collected for a time frame of twelve months. The data was then taken and values with multiple duplicates were removed to avoid redundancy of multiple sensor data. The final dataset was taken out which comprised of 5800 data points. The independent variables in the data set include parameters such as temperature, humidity, atmospheric pressure, current value from the solar panel which is set to three different levels, Precipitation levels, Water level, the dependent variable which is if there is a possibility of a flood is divided into 3 states, namely no flood, light flooding, heavy flooding which are assigned particular values.

Table 1. Database Schema

Sensor ID	Current(mA)	Temperature (℃)	Humidity	Atm Pressure (mb)	Precipitation / Rain Intensity	Water Level (cm)	Flood

4.2 Classifier Process

The dataset obtained is taken and split into training and test dataset. Since the amount of relationship can only be obtained by a significant training set and 20-80 split of training and test data is made. LDA is applied to reduce the dimensionality of the dataset independent variables. Different model classifiers are modelled onto the same dataset and the best model is chosen. The constant real-time stream is taken by the model and the new prediction values are computed constantly based on the optimal model.

Algorithm 1. General algorithm for multiple classifiers

Given a dataset D consisting multiple feature variables making the model of higher dimension. n is the number of feature variables.
for i ← 1 **to** N **do**
Read the dataset and construct dataset matrix
end for
for I ← 1 **to** N **do**
Extract the dependent and independent variable vectors.
Split the dataset into test and training set
Apply feature scaling to standardize the independent variables range
end for
for j ← 1 *to* M **do**
while p>=n
Apply LDA to the test and train independent variables and fit it to the model to reduce the dimensionality to p.
end while
Apply the appropriate classifier for both training and test set.
end for
Compute the confusion matrix to test accuracy for a different model where Ω is accuracy.
if Ω model-x > Ω model-y **then**
Model-x
else
Model-y
end if
Construct the flood forecast.

---

5 Results And Discussion

Since we have performed LDA and decomposed the independent data we can visualize the data clusters in two-dimensions. Where LD1 and LD2 are the dimensions derived from performing the LDA. We see that the plane is clustered for different sets of the algorithm. The dependent variable which has three states are mapped onto these clusters and the prediction value is compared with the cluster boundaries. The training was set to consist of 25 percent of the dataset.

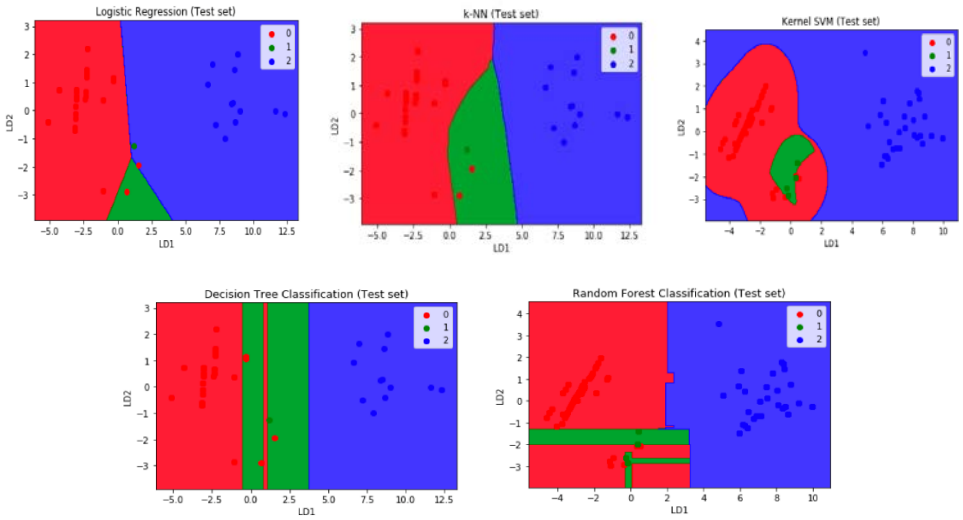


Figure 2. Model comparison on test dataset

Using the confusion matrix True Positives are computed for all the models. Here we observe that Logistic Regression fares the worst among the models for this particular use case. Both k-NN and Random forest classification have very close accuracy scored. Which means they have lower false positive rates. But looking at the visualization of a two-dimensional space we can see that the false positive deviations of the k-NN classifier seem to deviate slightly farther than Random forest classifier. Also, the accuracy obtained with the Random Forest classifier has a slighter better accuracy rate.

Table 2. Accuracy comparison of flood prediction with various classifiers

Methods	Accuracy of classifications
Logistic Regression	0.9562
k-NN	0.9974
Kernel SVM	0.9860
Decision Tree Classification	0.9873
Random Forest Classification	**0.9987**

6 Conclusion

This paper involves the application of multiple machine learning models and uses the model ranking to compute the better model among the several models. This involves a multi-model approach although for the scenario we see that the Random Forest Classification has the higher accuracy. The evaluation of the performance of different models like k-NN, SVR, Decision Tree Classification, Logistic Regression and Random Forest Classifications is very important to choose the right model for the appropriate scenario. The dataset used for this experiment involved weather data for a comparatively shorter time frame.

As the weather is a very complex phenomenon which constitutes of multiple variables and can have other external factors that could affect it. Also, in this particular case, the small shortfall in the accuracy is because of the limited night time sensor data where the current reading from the solar panel shows zero reading. Which the models might interpret as heavy cloud instead of night time. To overcome this UV sensor might be placed which allows determining the day or night cycle, by determining the intensity of sunlight or the lack thereof. During the night the UV levels fall down to negligible levels.

As the data is collected from very few sensor nodes from one city, the diversity in the data is reduced. If the sensor nodes are more diversely scattered and more data is extracted this model can help in better forecasting the Urban flooding.

Thus, by using a hybrid multi model comparison approach when it comes to predicting the data nodes the accuracy of the final model is increased.

ACKNOWLEDGEMENT

The authors would like to place on record their gratitude to authorities of Supercomputer Education and Research Centre, Indian Institute of Science, Bangalore, India for the encouragement and support during the entire course of this work. We also thank the Government of Karnataka for providing us with the necessary permissions to set up the test sensor modules across the city of Bangalore.

REFERNCES

[1] Real-time urban flood forecasting and modelling – a state of the art Justine Henonin, Beniamino Russo, Ole Mark, Philippe Gourbesville Journal of Hydroinformatics Jul 2013, 15 (3) 717-736;

[2] Peter, Lamovec, Mikoš Matjaž, and Oštir Krištof. "Detection of flooded areas using machine learning techniques: Case study of the Ljubljana moor floods in 2010." Disaster Advances 6.7 (2013): 4-11.

[3] Gupta, Anil K., and Sreeja S. Nair. "Urban floods in Bangalore and Chennai: risk management challenges and lessons for sustainable urban ecology." Current Science (2011): 1638-1645.

[4] Ramaswamy, Bhakthavathsalam. "An Intelligent Wireless Modular System for Effective Disaster Management." Transactions on Networks and Communications 4.3 (2016): 22.

[5] Nicholls, N. "A possible method for predicting seasonal tropical cyclone activity in the Australian region." Monthly Weather Review 107.9 (1979): 1221-1224.

[6] Martínez, Aleix M., and Avinash C. Kak. "Pca versus lda." IEEE transactions on pattern analysis and machine intelligence 23.2 (2001): 228-233.

[7] Rao, R. C. (1948). "The utilization of multiple measurements in problems of biological classification". Journal of the Royal Statistical Society, Series B. 10 (2): 159–203. JSTOR 2983775.

[8] Hofmann, Thomas; Scholkopf, Bernhard; Smola, Alexander J. (2008). "Kernel Methods in Machine Learning".

[9] *Altman, N. S. (1992). "An introduction to kernel and nearest-neighbor nonparametric regression". The American Statistician.* ***46*** *(3): 175–185.*

[10] Rokach, Lior; Maimon, O. (2008). Data mining with decision trees: theory and applications. World Scientific Pub Co Inc. ISBN 978-9812771711.

[11] Ho, Tin Kam (1995). *Random Decision Forests* (PDF). Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995. pp. 278–282.

SOCIETY FOR SCIENCE AND EDUCATION UNITED KINGDOM
TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE
ISSN 2054-7390

Implementation and Comparison of Machine Learning Algorithms for Recognition of Fingerspelling in Indian Sign Language

[1]Nikhil Aatrei M, [2]Shreyas H N, [3]Sumesh S. Iyer, [4]Gowranga K H, [5]R Bhakthavathsalam
[1,2,3]Dept. of Computer Science and Engineering, M. S. Ramaiah University of Applied Sciences, India;
[4,5]Supercomputer Education and Research Centre, Indian Institute of Science, Bangalore, India
nikhilaatrei96@gmail.com; hn.shreyas70@gmail.com; sumesh1809@gmail.com;
gowrangakh@gmail.com; bhaktha@serc.iisc.in

ABSTRACT

Communication is the biggest hurdle faced by the hearing and speech impaired in leading a normal life. In this context, Sign Language is the most prominent means of communication. Machine learning and Computer Vision is an integral part of today's computing world. This research paper proposes a Machine Learning based system to recognize fingerspelling gestures present in Indian Sign Language. Edge Frequency technique is chosen for Feature Extraction. The system was implemented using Aforge.NET framework. A comparative analysis of the Machine Learning Algorithms consisting of Support Vector Machine (SVM), K- Nearest Neighbor (KNN), Adaptive Naïve Bayes Classifier (ANBC), Decision Tree (DT) and Random Forests (RF) is performed to find out which algorithm is the most suitable to recognize ISL. Comparison is done based on validation accuracies and confusion matrices obtained. The accuracy for KNN was found to be 97.44% while SVM and ANBC have an accuracy of 96.15% and 82.05% respectively.

Keywords: Machine Learning, Computer Vision, Gesture Recognition, SVM, KNN, ANBC, Edge Frequency.

1 Introduction

Sign Language is the most prominent means of communication for the hearing and the speech impaired. It is estimated that across the world, 70 million deaf people use sign language as their first language [1]. 18% of the 70 million disabled people in India have hearing and speech impairment, as per Census 2011. Based on this data, India has the largest deaf population in the world.

Since the establishment of Indian Sign Language Research and Training Center (ISLRTC) in 2015, developments in the domain of Indian Sign Language (ISL) have been gaining momentum. A team of researchers at ISLRTC is working on Indian Sign Language dictionary. So far the team has compiled 6000 English and Hindi words in ISL. ISLRTC plans to circulate this dictionary in all schools. Their goal is to ensure that each school has at least one teacher who knows ISL [2].

However, according to a recent survey by the Ministry of Social Justice and Empowerment there are merely 300 ISL interpreters in the entire country. It is evident the requirement for the interpreters of ISL is significant. This research paper proposes a Machine Learning based system which recognizes the letters in ISL. A comparative analysis of the different Machine Learning algorithms which can be applied to this

DOI: 10.14738/tmlai.55.3557
Publication Date: 20th August, 2017
URL: http://dx.doi.org/10.14738/tmlai.55.3557

problem is performed. This system will be of great use if implemented in public places like banks, bus stations and malls.

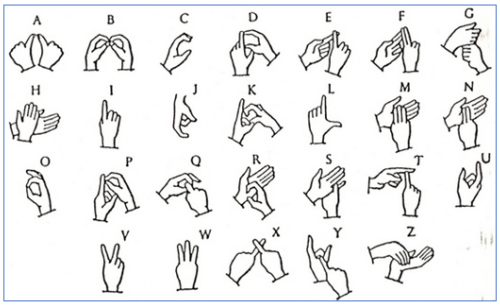


Figure 1: Indian Sign Language Gestures [3]

The rest of this paper is organized as follows: Section II presents the related work. System overview is presented in Section III. Results obtained and its analysis is done in the Section IV. Finally, Section V presents the conclusion and future scope of this work. In the following, the implementation is carried out for 5 algorithms with 25 features and the comparison is done on the top 3 performing algorithms with 36 features.

2 Related Work

Machine learning algorithms have been applied in diverse fields such as detection of phishing websites [4], gesture recognition, converting grayscale image to color [5], email classification [6], and music genre classification [7]. A decent amount of research has been done in the domain of gesture recognition. K-Nearest neighbor was used to recognize gestures of American Sign Language. In this paper, Euclidean distance was computed to find out K nearest neighbors and classify the test sample accordingly. Support vector Machine (SVM) and K-Nearest Neighbor (KNN) were used in [8] to recognize gestures of American Sign Language. This paper used Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) algorithms to compress and analyze the images of training set. Kethki P proposes to take a dynamic approach by using Hidden Markov Model in [9] to recognize One Handed American Sign Language and Two Handed British Sign Language. Another dynamic approach proposed in [10] extracts gestures from each frame of video. The probabilities of each gesture type are computed using combination of both KNN and Naïve Bayes.

3 System Overview and Implementation

A. Image Preprocessing

1) Skin detection

2) Closing

3) Blob

B. Dimensionality Reduction through Feature extraction

C. Classification

D. Implementation

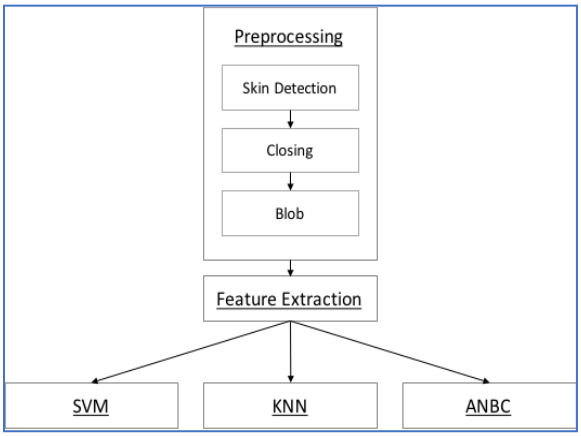


Figure 2. System Framework

3.1 Image Preprocessing

3.1.1 Skin Detection

Skin detection is the extraction of portion of the image that has skin [11]. Pixels that correspond to color range of human skin are extracted. This is done using the pixel intenisities of the image. This is the first step in preprocessing of any image or video. A skin detector function which transforms pixels into corresponding color space is used. This is then given to skin classifier which is responsible for determining if the pixel in question is skin or not. Skin classifier generates a decision boundary of the skin color space with respect to a skin color database [12]. Based on the range of color, skin portion is detected and converted to white while the rest of the pixels are converted to black. This can be seen in Figure 4(b).

3.1.2 Closing

It is responsible for removing small holes in an image. It ensures removal/reduction of noise in an image, due to holes in the image [13]. For example, after skin detection, if some of the pixels which are actually skin pixels are not detected as skin pixels, these pixels are converted to skin pixels by applying closing technique. This can be depicted using an example in Figure 3. There was a part of the main object which was not detected. After closing was applied, the missing part was detected correctly.

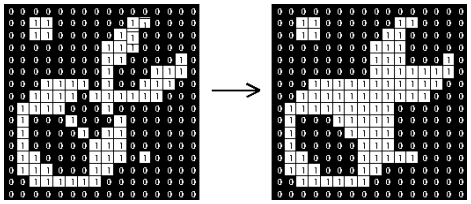


Figure 1. Example for Closing Error! Reference source not found.

3.1.3 Blob

Blob detection is a method which is used to detect regions which differ based on some factors. Blob is a region of an image where pixels of that region have same or constant properties. All points in a blob are similar to each other. In gesture recogntion, blob detection is applied to detect biggest blob or skin region which represents the getsure i.e. sign. It is used to distinguish the gesture region (palm) from other regions (region below wrist). The biggest blob is identified which represents one of the 26 signs of Indian sign

language and is further used as input for feature extraction. This can be better understood by looking at Figure 4(c).

3.2 Dimensionality Reduction through Feature Extraction

Minimal features of an image is extracted which can be later used to describe the image accurately with reduced number of resources. Representing an image with reduced number of features (variables) saves memory and processing time. The Feature extraction technique used in this paper is Edge Frequency (HOEF) [14].

3.2.1 Histogram of Edge Frequency (HOEF)

It is one of the most reliable feature extraction techniques. It divides the gesture image into N*N blocks, where N is the size of row and column. Each ith Block of the gesture image where i=1,2,...N is scanned for edge pixel. Separate edge pixel count is maintained for each block. Whenever an edge pixel is encountered, edge pixel count is incremented for that corresponding block. A histogram is plotted where each bin (column) represents a block and its value represents edge pixel count of that corresponding block. The histogram is further normalized (values are modified for uniformity) and final feature vector is obtained. This feature vector is sent to ML classifiers for recognition of gesture. Advantage of this method is that, for a gesture, there may be blocks which are free of edges i.e. edge pixel count of these blocks is zero. These blocks can be helpful in uniquely distinguishing one gesture from other as the other gesture may have edges in those blocks. Feature extraction is done on Figure 4(d).

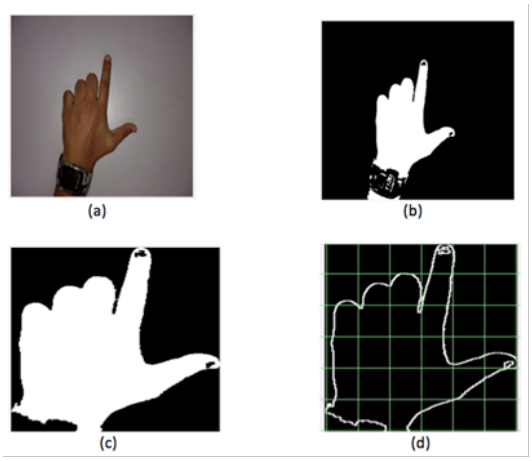


Figure 4. (a) Original Image. (b) After Skin Detection and Closing. (c) After Blob Detection. (d) Feature Extraction using HOEF

3.3 Classifier

3.3.1 Support Vector Machine (SVM)

SVM is basically a classification algorithm that is used to classify data. It works using hyperplanes, meaning these hyperplanes divide the dataset (training set) based on the classes they belong to. SVM is a Supervised Learning Algorithm meaning the categorization is done based on the labels that are given to the training set **Error! Reference source not found.**. The aim of this algorithm is to separate the classes using a hyperplane so that all the training sets that belong to Class 1 are on one side of the hyperplane while the training sets that belong to Class 2 are on the flip side of that hyperplane (Assuming there are only 2 potential classes).

Consider an example in which there are 2 classes x_1 and x_2 which are Linearly Separable.

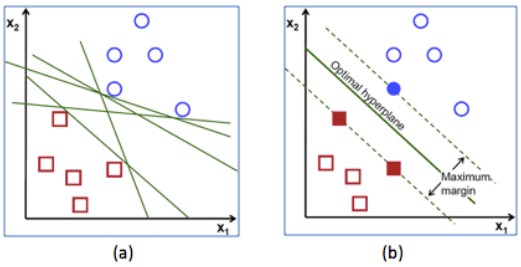


Figure 2. (a) Classification of data without using SVM. (b) Classification of data using SVM Error! Reference source not found.

In Figure 2(a), it can be seen that there are multiple lines that divide the two Classes. It is SVM's job to find the best hyperplane that has the largest minimum distance between the hyperplane and the data points as depicted in Figure 2(b). This is same as maximizing the margin of the training data **Error! Reference source not found.**.

If the classes are not linearly separable, then SVM works in coordination with a kernel function which avoids mapping of a linearly separable dataset to a non-linearly separable dataset, explicitly. Meaning, there will be no need for an explicit mapping between the two, as shown in Figure 3.

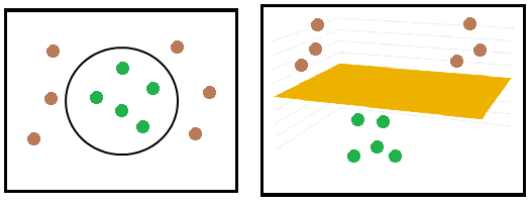


Figure 3. SVM in coordination with the Kernel Function

3.3.2 K-Nearest Neighbor (KNN)

KNN is a classification and regression algorithm used mostly in pattern recognition. KNN is one of the simplest and easily implementable machine learning algorithms. While classifying a sample using KNN, Euclidean distance between the test sample (test feature vector) and rest of the samples (training feature vectors) is computed. 'K' samples whose distance is minimum from test sample i.e. 'K' nearest neighbors are selected.

Test sample is assigned the class label which is majority among the 'K' nearest neighbors **Error! Reference source not found.**.

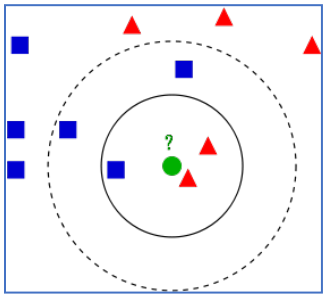


Figure 4. Example for classification of data using KNN Error! Reference source not found.

For example, in Figure 4, there are 2 classes; squares (Blue) and triangles (red). The aim is to classify the green circle into one of the two available classes. The radius of the enclosing circle is determined based

the chosen value of k. When k=3, the enclosing circle is the solid line circle and the neighbors would be 2 triangles and 1 square. So, the green circle will be classified as a triangle according to this algorithm (as there are more instances of triangles in that enclosing circle).

If k=5, the enclosing circle is the dotted line circle and the neighbors would be 2 triangles and 3 squares. So, the green circle will be classified as a square according to this algorithm (as there are more instances of squares in that enclosing circle).

3.3.3 Adaptive Naïve Bayes Classifier (ANBC)

This classification algorithm works based on Bayes theorem of probability to predict class C(i) of test sample x given its observed feature values. An assumption is made that the features are independent of each other. It assigns the most likely class for the given sample given its attribute set or observation variables (feature vector). It computes posterior probability and is based on the prior probabilities of the class, observation variables and observation variables given the class **Error! Reference source not found.**.

For example: If there is a need to classify whether a car is stolen or not given its attributes (Color, Type, Origin, etc.). Initially a given data set is used for training purpose. Based on the data set which is used for training purpose, one can classify test samples. Suppose one has to determine whether Red (Color) Sedan(Type) Domestic(Origin) is stolen or not. By referring the data set, one has to find out the probability of how many Red cars have been stolen or not, how many cars are Sedan and non-Sedan, how many of them are Domestic or Imported. After computing these probabilities, posterior probability is computed which finally determines whether the car (test case) is stolen or not.

3.4 Implementation

The system was implemented using Aforge.NET framework. Aforge is an open source Computer Vision and Artificial Intelligence framework which provides tools for Image Processing, Robotics and Machine Learning application. The User Interface which performs feature extraction is presented in Figure 5.

Figure 5. Feature Extraction User Interface

Skin Detection is implemented using the technique mentioned in **Error! Reference source not found.**. Hue and saturation values are computed for each pixel based on its RGB values. Based on hue and saturation obtained, each pixel is classified as skin and non-skin. Aforge framework provides default closing function as part of its Computer Vision tools. This function is applied on output of Skin Detection process. Futher the processed image undergoes blob detection. “ExtractBiggestBlob” class present in the Aforege library is used to perform this task. Features Extraction is achieved by storing the processed image as a bitmap. This helps dividing the image into a 5X5 grid and detect edges. Finally, EdgeDetecor class is used to detect the Edge Frequency for each block. The extracted features are given as input to the Machine Learning algorithms which are implemented using GRT (Gesture Recognition Toolkit), an open source library. Confusion matrix, precision and recall are computed based on the classification results. This implementation scheme is used to compare the 5 mentioned machine learning algorithms.

4 Results and Discussion

For a dataset consisting of 260 images (10 images for each alphabet gesture of ISL), features are being extracted using Edge Frequency technique. Each image is divided into a 5*5 grid. Edge frequency is computed for each block. This gives a total of 25 features. An open source Computer Vision library Aforge.NET was used to implement the feature extraction module. The 260 images are divided into 70-30 ratio (70% for training and 30% for testing). The program was run on a machine with 1.4 GHz Intel Core i5 processor with 4GB DDR3 Memory and Intel HD Graphics 5000. Feature extraction and training of the model took close to 14 minutes. Initially SVM, KNN, Random Forests, Decision Trees and ANBC Machine Learning algorithms were applied for classifying the dataset. Based on the accuracy obtained as shown in Table 1, the top three algorithms namely SVM, KNN and ANBC were chosen for further testing and analysis.

Table 1: Validation Accuracy obtained for 25 features

Algorithm	Accuracy
Support Vector Machine	94.23%
K-Nearest Neighbor	92.31%
Adaptive Naïve Bayes Classifier	82.05%
Decision Trees	76.92%
Random Forests	71.79%

Extracting more number of features is one of the possible options to improve accuracy. This means the algorithm has more input variables and thus can differentiate between data points effectively. For the feature extraction technique used i.e. HOEF, if an image is further divided into 6X6 blocks instead of 5X5, each image is uniquely identified using 36 feature vectors (each block). This is likely to improve the accuracy of our algorithm. To support this statement, consider the example presented in Figure 6 and Figure 7. The ISL gestures of M and N are similar as shown in **Error! Reference source not found.**. Figure 6 and Figure 7 shows the histograms plots of M and N gestures for 25 and 36 features respectively. It can be observed for 25 features the generated histogram plots are similar. Hence the probability of M/N misclassification is more. But in case of 36 features the generated histogram plots are relatively different. Therefore, the probability of M/N misclassification is less, thus accuracy is likely to improve. The results obtained for 36 features are presented in Table 2. The average of Precision and Recall values of all 26 classes for each algorithm is presented.

Precision and Recall are two parameters used to compare the performance of ML algorithms. Using the most common example, consider an application which is supposed to detect dogs in a picture. A picture containing 12 dogs and some cats are given as input. Assume that the application detects 8 entities as dogs but in reality, 5 out of the 8 detected entities are dogs and the rest are cats. Precision for this case would be 5/8 because among the number of predictions made by the application 5 were correct. Recall would be 5/12 because among the 12 dogs in the picture the application detected 5 correctly.

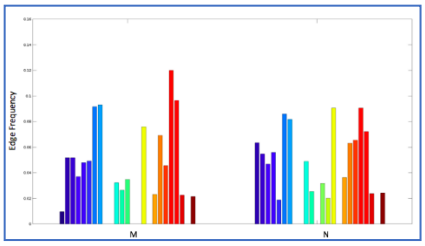


Figure 6. Edge Frequencies for letters M and N with 25 features

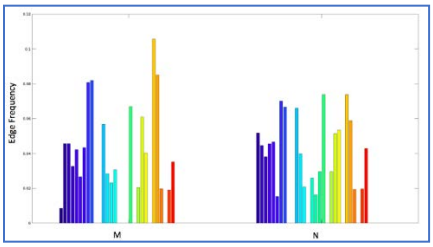


Figure 7. Edge Frequencies for letters M and N with 36 features

Table 2. Results obtained for 36 Features

Algorithm	Accuracy	Precision	Recall
Support Vector Machine	96.15%	0.96794	0.96150
K-Nearest Neighbor	97.44%	0.97520	0.99030
Adaptive Naïve Bayes Classifier	84.61%	0.83663	0.81154

Looking at the confusion matrices of each of these algorithms as shown in Table 3,

Table 4 and Table 5, it can be said that in the case of SVM, the gesture for 'C' is recognized as 'C' 66.7% of the times and is misinterpreted as 'R' 33.3% of the times. Similarly, gesture for Q is misinterpreted 33.3% of the times as G and 'R' is misinterpreted as 'N' 33.3% of the times.

In case of KNN, J is correctly interpreted 75% of the times and is misinterpreted as 'O' 25% of the times.

ANBC has a few more discrepancies like A-E, B-X, H-Q, I-Y, J-Y, L-G, O-R-U, T-K, U-E, V-Y, Y-Z.
Referring to Table 1, on observing the poor performance of Decision Trees, Random Forests algorithm was applied to check if the Decision Tree model was overfitting the data. Random Forests algorithm is an extension of Decision Trees which attempts to prevent overfitting by forming random subsets of input features and building multiple smaller trees. However, a poorer validation accuracy for Random Forests was obtained indicating that overfitting was not a problem in the Decision Tree model.

From the initial results presented in Table 1, it can be inferred that the linear classification property of SVM suits this application. KNN surpassed SVM with a small difference of 0.9% in case of 36 features. However, KNN is known to perform well on small datasets. KNN is not scalable due to the fact it stores all the training data. KNN needs to use all the training data to predict each test sample. However, SVM discards all unwanted vectors during training phase. This makes SVM highly scalable. Keeping this in mind, it can be said that both SVM and KNN show promising results for this application.

Table 3. Confusion Matrix of SVM (Note that the letters with 100% accuracy are not mentioned in the rows below for sake of simplicity)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
C	0	0	66.7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	33.3	0	0	0	0	0	0	0	0
Q	0	0	0	0	0	0	33.3	0	0	0	0	0	0	0	0	0	66.7	0	0	0	0	0	0	0	0	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	33.3	0	0	0	66.7	0	0	0	0	0	0	0	0

Table 4. Confusion Matrix for KNN (Note that the letters with 100% accuracy are not mentioned in the rows below for sake of simplicity)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
J	0	0	0	0	0	0	0	0	0	75	0	0	0	0	25	0	0	0	0	0	0	0	0	0	0	0

Table 5. Confusion Matrix for ANBC (Note that the letters with 100% accuracy are not mentioned in the rows below for sake of simplicity)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	50	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0
H	0	0	0	0	0	0	0	70	0	0	0	0	0	0	0	0	30	0	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0
J	0	0	0	0	0	0	0	0	0	75	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0
L	0	0	0	0	0	0	25	0	0	0	0	75	0	0	0	0	0	0	0	0	0	0	0	0	0	0
O	0	0	0	0	0	0	0	0	0	0	0	0	0	0	70	0	0	15	0	0	15	0	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0
U	0	0	0	0	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	75	0	0	0	0	0
V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	75	0	0	25	0
Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	50

5 Conclusion

This paper deals with the comparison of 3 pattern recognizing Machine Learning Algorithms for ISL recognition. After extracting features using Histogram of Edge Frequency technique, the features are given as input to the 3 algorithms; SVM, KNN and Adaptive Naive Bayes Classifier. Referring to Table 1, SVM shows a promising result of 94.43% accuracy for 25 features. When the number of features is increased to 36, KNN surpasses SVM with an accuracy of 97.44%. While KNN (with k=3) is the most accurate, SVM has an accuracy of 96.15% and ANBC has an accuracy of 82.05%. However, it is to be noted that the test scenario is limited. The images in the dataset are not very diverse in nature. For example, images of different skin tones or of varying lighting conditions are not present in the dataset. So, the results must be analyzed keeping this in mind. Based on our results and analysis, SVM and KNN show favorable results for this application. Referring to Table 2, Precision and Recall values also support this statement.

Regarding future work, an attempt to further improve the accuracy of KNN can be done by including more images in the training set, dividing the image into more number of blocks before extracting features and/or by attempting to use a combination of classification algorithms. Once a sufficiently high accuracy is achieved, this system can be integrated into a mobile application that can recognize gestures in real time. There is scope for modifying this application to facilitate two-way communication i.e. interpret gestures and convert letters/words into gestures. Thus, the results of this paper will be beneficial in eliminating the communication barrier between hearing-speech impaired and normal people, without the need of a human interpreter.

ACKNOWLEDGEMENT

The authors would like to place on record their gratitude to the authorities of Supercomputer Education and Research Centre, Indian Institute of Science, Bangalore, India for the encouragement and support during the entire course of this work.

REFERENCES

[1] M Miller, "Sound of Silence". Presidential Leadership Academy. April 2016.

[2] S Nair, "Coming soon: First-of-its-kind Indian sign language dictionary" The Indian Express, January 23, 2017. [Online]. Available: http://www.indianexpress.com [Accessed June 18, 2017]

[3] Adithya, P. R. Vinod and U. Gopalakrishnan, "Artificial neural network based method for Indian sign language recognition," Information & Communication Technologies (ICT), 2013 IEEE Conference on, JeJu Island, 2013, pp. 1080-1085

[4] Ningxia Zhang, Yongqing Yuan, “Phishing Detection Using Neural Network”, May 2012.

[5] Austin Sousa, Rasoul Kabirzadeh, Patrick Blaes, “Automatic Colorization of Grayscale Images”, June 2013.

[6] David Harwath, Nikhil Johri, Edouard Yin, “An Unsupervised Approach To Email Label Suggestions”, November 2010.

[7] Charles Burlin, Matthew Crème, Raphael Lenain, “Music Genre Classification”, December 2016.

[8] Satish Kumar Kotha, Jahnavi Pinjala, Kavya Kasoju, Manvitha Pothineni, “GESTURE RECOGNITION SYSTEM”, International Journal of Research in Engineering and Technology, May 2015

[9] Ketki P. Kshirsagar, “Key Frame Selection for One-Two Hand Gesture Recognition with HMM”, June 2015.

[10] Pujan Ziaie, Thomas Muller, Mary Ellen Foster, and Alois Knoll, “A Naïve Bayes Classifier with Distance Weighting for Hand-Gesture Recognition”, Communications in Computer and Information Science Conference, September 2008.

[11] A. Albiol, L. Torress, E. J. Delp, “OPTIMUM COLOR SPACES FOR SKIN DETECTION”. Proc. Of International Conference on Image Processing, Vol. 1, 122-124, 2001.

[12] Ahmed Elgammal, Crystal Muang and Dunxu Hu, “Skin Detection - a Short Tutorial”, Encyclopedia of Biometrics by Springer-Verlag Berlin Heidelberg 2009.

[13] R. Fisher, S. Perkins, A. Walker and E. Wolfart., "Morphology - Closing", Homepages.inf.ed.ac.uk, 2017. [Online]. Available: https://homepages.inf.ed.ac.uk/rbf/HIPR2/close.htm.

[14] M Sharma, S Singh, “Evaluation of Texture Methods for Image Analysis”, Seventh Australian and New Zealand Intelligent Information System Conference, 18th November 2001.

[15] Anita Jadhav, Rohit Asnani, Rolan Crasto, Omprasad Nilange, Anamol Ponkshe, “Gesture Recognition Using Support Vector Machine”. International Journal of Electrical, Electronics and Data Communication. May 2015

[16] Docs.opencv.org, (2017). Introduction to Support Vector Machines — OpenCV 2.4.13.2 documentation [Online]. Available: http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html.

[17] S. Nagarajan, T. S. Subahini, "Image Based Hand Gesture Recognition using Statistical Features and Soft Computing Techniques". International Journal of Emerging Technology and Advanced Engineering. July 2015.

[18] Hector Hugo Avilts-Aniaga, Luis Enrique Sucart, Carlos Eduardo Mendozaz, "Visual Recognition of Gestures using Dynamic Naïve Bayesian Classifiers", November 2003.

SOCIETY FOR SCIENCE AND EDUCATION UNITED KINGDOM
TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE
ISSN 2054-7390

Effects of Strengths of Steel and Concrete, Eccentricity and Bar Size on the Optimization of Eccentrically Loaded Footings

[1]Jiin-Po Yeh and [2]Kuan-Hao Huang
Department of Civil and Ecological Engineering, I-Shou University, Kaohsiung, Taiwan
[1]jpyeh@isu.edu.tw, [2]isu10306006m@cloud.isu.edu.tw

ABSTRACT

This paper aims to explore effects of the yield strength of steel, compressive strength of concrete, eccentricity of the axial load and steel bar size on the optimization of reinforced concrete isolated footings. The optimization tool adopted in this paper is genetic algorithms. Based on the ACI Building Code, constraints are built by considering the wide-beam and punching shears, bending moment, upper and lower limits of reinforcement, allowable soil pressure, development length for deformed bars and clear distance between parallel deformed bars. Design variables consist of the width, length and thickness of the footing and the number of bars in the long and short directions, all of which are integers. The objective is to minimize the cost of steel and concrete used in the footing. By changing one of the four factors: the yield strength of steel, compressive strength of concrete, eccentricity and bar size while fixing the other three, this paper finds that the highest yield strength of steel, the lowest compressive strength of concrete, the smallest eccentricity and No. 6 bar, respectively, will lead to the optimal results. In addition, when the size of the reinforcement gets larger, the optimal footing have a tendency to become square and thicker.

Keywords: Genetic Algorithms; Optimal Design; Reinforced Concrete Footings; Eccentricity

1 Introduction

A footing that serves the purpose of transmitting the load from the superstructure to the supporting soil is a very important element in an architectural structure. A conventional way to design a footing is to find its suitable dimensions and amount of reinforcement according to the provisions of a building code. The design results are usually not the most economical. In order to achieve this goal, the optimization techniques can be applied. There have been a number of optimization studies of reinforced concrete footings published over the past few years, such as optimization of combined footings using modified complex method of box [1], optimization of concentrically loaded reinforced concrete footing using an analytical model [2], and optimization of concentrically loaded reinforced concrete footings using genetic algorithms [3].

The fundamental techniques of genetic algorithms are designed to imitate processes in natural evolution. Genetic algorithms are the most effective methods in a search space for which little is known and which is uneven and has many hills and valleys with potential candidate solutions. The idea of genetic algorithms was inspired by the evolution theory of "survival of the fittest," and formally introduced in 1970s by Professor John Holland at the University of Michigan, who in 1975 published the ground-breaking book "Adaptation in Natural and Artificial System" [4] that led to many important discoveries. In 1989, Goldberg

DOI: 10.14738/tmlai.55.3592
Publication Date: 27th August, 2017
URL: http://dx.doi.org/10.14738/tmlai.55.3592

described in more detail the theory of genetic algorithms and its applications [5]. From then on, the continuing improvements of computational techniques have made genetic algorithms more attractive and popular. Genetic algorithms have successfully been applied to many fields, for example, engineering, economics, chemistry, manufacturing, mathematics and physics. Especially in the aspect of engineering structures, there are a lot of applications, such as reliability analysis of structures [6], global optimization of grillages [7], global optimization of trusses with a modified genetic algorithm [8], optimization of pile groups using hybrid genetic algorithms [9], prediction of concrete faced rock fill dams settlements [10], optimization of grid shell topology and nodal positions [11], optimizations of constrained layered damped (CLD) laminated structures [12], calibration of a hydrological model to predict stream flows [13] and optimal design of short columns [14].The fact that they are successfully applied to many problems which are difficult to solve by using conventional optimization techniques prove that genetic algorithms are a powerful, robust optimization technique.

Most optimization approaches have been focused on and developed for continuous variables, while the design variables are usually integers for problems in architectural structures. Due to their abilities to solve discrete optimization problems, genetic algorithms provided by the MATLAB Global Optimization Toolbox [15] are used in this paper to carry out the optimization of eccentrically loaded reinforced concrete isolated footings and explore effects of the yield strength of steel, compressive strength of concrete, bar size and eccentricity on the optimal results. Based on the provisions of the ACI Building Code Requirements for Structural Concrete and Commentary [16], the constraints of genetic algorithms are constructed, considering the wide-beam and punching shears, bending moment, upper and lower limits of reinforcement, allowable soil pressure, development length for deformed bars, clear distance between deformed bars. The design variables are the depth, width and length of the footing, the number of bending reinforcement in each direction of the footing; the objective is to find the minimum cost of concrete and steel.

2 Genetic Algorithms

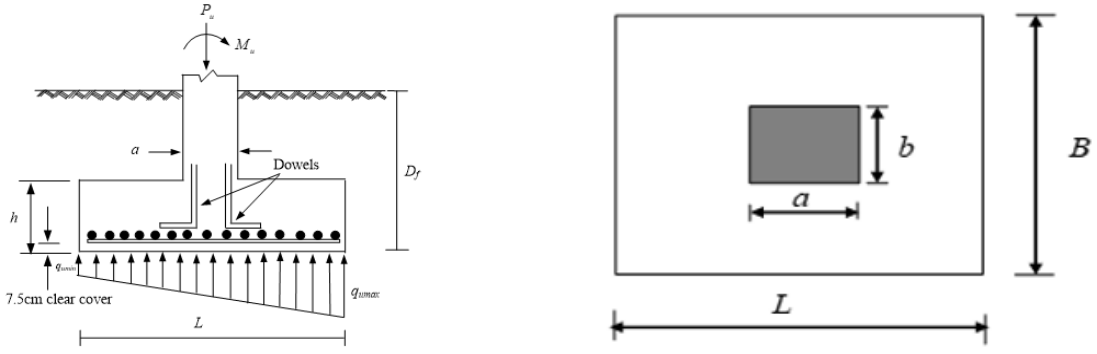
Genetic algorithms are basically a heuristic process for mimicking the survival of the fittest among individuals over a sequence of generations for solving an optimization problem. There is a population of individuals in each generation. Each individual made up of design variables represents a candidate solution to a given problem. The individuals are similar to chromosomes and the design variables to genes. A fitness value is assigned to each solution to measure its competitiveness. The individuals with higher fitness values are more likely to be selected to form the next generation. If the current population can no more produce individuals significantly better than those in the previous few generations, the algorithm is said to converge and the optimal solution are found.

The most common type of genetic algorithms works through the following process of natural selection: (1) Randomly create an initial population of individuals; (2) Score each individual of the current population by computing the value of the fitness function; (3) Scale the raw fitness scores to convert them into a range that is suitable for the selection function; (4) Select a specified number of individuals with lower fitness values, called parents, by using the selection function; (5) Choose a few elite individuals with the lowest fitness values from the current population. These elite individuals are then just passed to the next population; (6) Produce children from the parents. Children are produced either by combining portions of good individuals (i.e., crossover), which aims to create even better individuals or making random changes to a single individual (i.e., mutation), whose purpose is to maintain diversity within the population and

inhibit premature convergence; (7) Replace the current population with the crossover and mutation children and elites to form the next generation and the process repeats form steps (2) to (7). The algorithm stops when one of the stopping criteria is met, such as the number of generation, the weighted average change in the fitness function value over some generations less than a specified tolerance, no improvement in the best fitness value for an interval of time, etc. Genetic algorithms can solve both constrained and unconstrained optimization problems. The constraints built for genetic algorithms can be linear or nonlinear in the form of equality or inequality with bounds on the variables. Each individual made up of the design variables can be real-coded or binary-coded. In this paper, the constraints consist of nonlinear and linear inequalities and all the design variables are integers.

3 Design Considerations in Eccentrically Loaded Footings

Both the concentric compressive force P_u and bending moment M_u, are considered to act on the reinforced concrete isolated footing whose layout is shown in Fig. 1. The rectangular footing has width B, length L and thickness h, and the column size is $a \times b$. The soil bearing pressure distribution on the footing is trapezoidal due to the combined effects of axial load and bending, as shown in Fig. 1(a). All the constraints required to design the isolated footing comply with the ultimate-strength design of ACI 318-11 Code, considering wide-beam and punching shears, bending moment, the development length for deformed bars, clear distance between parallel deformed bars and the upper and lower limits of reinforcement. The units of force and length in the following formulas are kgf (=9.81N) and cm, respectively. The factored load $P_u = 1.2P_D + 1.6P_L$,



(a) (b)

Figure 1 The reinforced concrete footing subjected to the concentric factored load Pu and bending moment Mu: (a) elevation and (b) plan

Figure 1 The reinforced concrete footing subjected to the concentric factored load P_u and bending moment M_u: (a) elevation and (b) plan.

where P_D and P_L are the dead and live loads, respectively. The eccentricity e is defined as M_u / P_u.

3.1 Factored Shears

To have enough shear capacity, there are two kinds of actions that need to be considered: wide-beam action and two-way action.

3.1.1 Wide-beam Action

The maximum and minimum soil pressures on the footing, as shown in Fig. 1, are

$$q_{u\max} = \frac{P_u}{LB}(1+\frac{6e}{L}) \tag{1}$$

and

$$q_{u\min} = \frac{P_u}{LB}(1-\frac{6e}{L}) \tag{2}$$

respectively, where $e \le \frac{L}{6}$ is the eccentricity. The plane of the critical section is assumed to extend in a plane across the entire width and lies at a distance *d* from the face of the column, as shown in Fig. 2(a). The nominal shear strength of this section is

$$V_{c1} = 0.53\sqrt{f_c'}Bd \tag{3}$$

and

$$V_{c2} = 0.53\sqrt{f_c'}Ld \tag{4}$$

respectively in the long and short directions, where *d* is the average effective depth of the footing. Let q_{udL} denote the soil pressure in the long direction of the footing at a distance *d* from the right face of the column and ϕ =0.75 be the strength reduction factor for shear. The constraints for the wide-beam shear are

$$V_{uL} = q_{u\,avgL}(\frac{L-a}{2}-d)B \le \phi V_{c1} \tag{5}$$

and

$$V_{uB} = q_{u\,avgB}(\frac{B-b}{2}-d)L \le \phi V_{c2} \tag{6}$$

respectively for the long and short directions of the footing, where

$$q_{uavgL} = \frac{q_{u\max}+q_{udL}}{2} \tag{7}$$

and

$$q_{uavgB} = \frac{q_{u\max}+q_{u\min}}{2} \tag{8}$$

3.1.2 Two-way Action

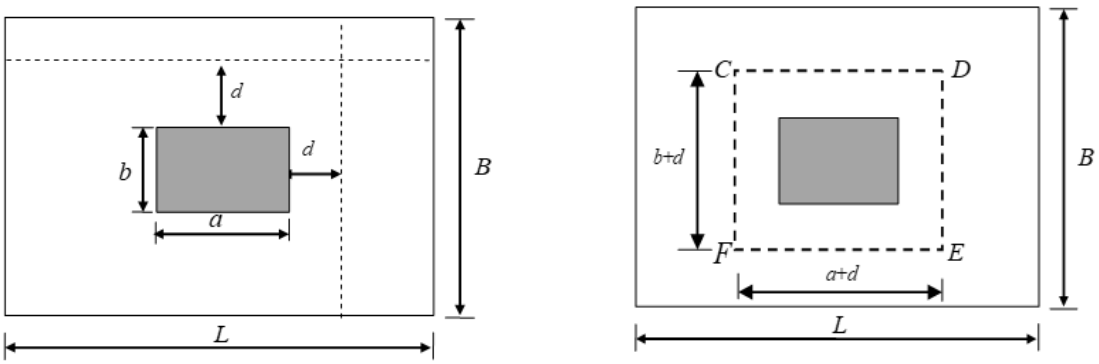
The critical section occurs at a distance *d*/2 from the face of the column, as shown in Fig. 2 (b). The maximum allowable nominal shear strength is the smallest of the following three equations

$$V_c = (0.53 + \frac{1.06}{\beta_c})\sqrt{f_c'}b_0 d \ ,$$
$$V_c = (0.53 + \frac{0.265\alpha_s d}{b_0})\sqrt{f_c'}b_0 d \quad (9)$$
$$V_c = 1.06\sqrt{f_c'}b_0 d$$

where β_c = long side *a*/short side *b* of the concentrated load or reaction area, b_0 = perimeter of the critical section *CDEF* and α_s = 40, 30 and 20 for interior, edge and corner columns, respectively. In this paper, interior columns are considered; therefore, α_s = 40. Let q_{uL1} and q_{uL2} denote the soil pressures at a distance *d*/2 from the left and right faces of the column, respectively. The constraint for the punching shear is

$$V_u = P_u - \frac{q_{uL1} + q_{uL2}}{2}(a+d)(b+d) \le \phi V_{c,\min} \quad (10)$$

where $V_{c,min}$ is the smallest of Eqs. (9).



(a) (b)

Figure 2 Critical sections: (a) wide-beam action and (b) two-way action.

3.2 Factored Moments

Suppose that N_L and N_B are the number of steel bars required in the long and short directions of the footing, respectively, and A_b is the cross-sectional area of the flexural reinforcement. The critical section for moment is taken at the face of the column. Let q_{ul3} denote the soil pressure at the right face of the column. The constraints for the factored moments are

$$M_{uL} = \frac{(q_{uL3} + q_{u\max})B}{2}(\frac{L-a}{2})k \le \phi_{mL} N_L A_b f_y (d - \frac{N_L A_b f_y}{2(0.85) f_c' B}) \quad (11)$$

and

$$M_{uB} = \frac{(q_{u\max} + q_{u\min})L}{2}\frac{(B-b)^2}{8} \le \phi_{mB} N_B A_b f_y (d - \frac{N_B A_b f_y}{2(0.85) f_c' L}) \quad (12)$$

where ϕ_{mL} and ϕ_{mB} are the strength reduction factors for moment and *k* is distance from the face of the column to the centroid of the trapezoid. Let ε_t be the tensile strain of the reinforcement, then

$$0.65 \le \phi_{mL} \text{ or } \phi_{mB} = 0.65 + 0.25\frac{\varepsilon_t - \varepsilon_y}{0.005 - \varepsilon_y} \le 0.9 \tag{13}$$

3.3 Upper and Lower Limits of Reinforcement

To prevent sudden failure with little or no warning when the beam cracks or fails in a brittle manner, the ACI code limits the minimum and maximum amount of steel to be

$$A_{sL,\min} \le N_L A_b \le A_{sL,\max} = \frac{0.85 f_c' \beta B d}{f_y}(\frac{3}{7}) \tag{14}$$

and

$$A_{sB,\min} \le N_B A_b \le A_{sB,\max} = \frac{0.85 f_c' \beta L d}{f_y}(\frac{3}{7}) \tag{15}$$

respectively in the long and short directions, where β is the stress block depth factor,

$$A_{sL,\min} \ge \max\ (\frac{0.8\sqrt{f_c'}}{f_y} Bd,\ \frac{14Bd}{f_y}) \tag{16}$$

and

$$A_{sB,\min} \ge \max\ (\frac{0.8\sqrt{f_c'}}{f_y} Ld,\ \frac{14Ld}{f_y}) \tag{17}$$

The formula for $A_{sL,max}$ in Eq. (14) and $A_{sB,max}$ in Eq. (15) is derived based on the requirement that the tensile strain must be greater than or equal to 0.004. In addition, both the steel ratios $N_LA_b/(Bh)$ and $N_BA_b/(Lh)$ must exceed the minimum value required for temperature and shrinkage: 0.0018 for grade 60 deformed bars and 0.002 for grade 40 or 50 deformed bars.

3.4 Allowable Bearing Capacity of Soil

Bearing capacity is the capacity of soil to support the loads applied to the ground. Usually, only the service loads need to be considered, i.e., P_D and P_L without load factors. Assume that the allowable soil pressure under the base of the footing is q_a. The gross soil pressure must not exceed the allowable soil pressure, that is,

$$\frac{P_D + P_L}{BL} + hw_c + \gamma_s(D_f - h) \le q_a \tag{18}$$

where D_f is the distance from the base of the footing to the ground surface, as shown in Fig. 1, w_c is the weight of concrete and γ_s is the unit weight of soil over the footing.

3.5 Development Length for Deformed Bars

The ACI Code specifies that the equation for the development of deformed bars in tension be expressed by

$$L_d = \frac{0.15 d_b f_y \psi_t \psi_e \lambda}{\sqrt{f_c'}} \geq 30 \ cm \tag{19}$$

$$\text{or} \quad L_d = \frac{0.19 d_b f_y \psi_t \psi_e \lambda}{\sqrt{f_c'}} \geq 30 \ cm \tag{20}$$

for No. 6 and smaller bars or No. 7 and larger bars, respectively, with clear spacing not less than $2d_b$ and clear cover not less than d_b, where d_b is the bar diameter, and ψ_t and ψ_e are the bar location and coating factors, respectively. In this paper ψ_t and ψ_e are assumed to be 1.0 and λ=1 for normal weight concrete. The critical section for development length of the bars in tension is the same as the critical section in flexure, that is, at the face of the column. Hence,

$$0.5(L\text{-}a) \text{ - concrete cover} \geq L_d \tag{21}$$

and

$$0.5(B\text{-}b) \text{ - concrete cover} \geq L_d \tag{22}$$

respectively in the long and short directions. The equation for the development length of bars in compression is

$$L_{dc} = \max \left(\frac{0.075 d_b f_y}{\sqrt{f_c'}}, \ 0.0043 d_b f_y \right) \tag{23}$$

The dowel bars stressed to f_y are required to transfer the axial compression force in the column into the footing, as shown in Fig. 1; hence, there should be minimum extension of the dowels into the footing. Therefore, the thickness h of the footing must satisfy the following constraints:

$$h - \text{concrete cover} - 2d_b \text{(footing bars)} - d_b \text{(dowels)} \geq L_{dc} \tag{24}$$

In addition, depth of footing above bottom reinforcement shall not be less than 15 cm for footings on soil and a practical minimum thickness h should not be less than 25 cm.

3.6 Distribution and Minimum Clear Distance of Steel Bars

The total steel area $N_B A_b$ in the short direction determined from Eqs. (12) and (15) should be uniformly distributed over the central band of the footing, whose width is B, as shown in Fig. 3. The ratio of the reinforcement in the central band to the total reinforcement is equal to $2/(L/B+1)$. The reinforcement that is not placed in the central band is uniformly spaced at each side of the central band. In the long direction, the total steel area $N_L A_b$ determined from Eqs. (11) and (14) is uniformly distributed across the entire width of the footing. The clear distance s between parallel steel bars in both the long and short directions must satisfy

$$Min \ (3h, \ 45 \ cm) \geq s \geq Max \ (2d_b, \ 2.5 \ cm) \tag{25}$$

Instead of d_b, the minimum clear distance $2d_b$ used in Eq. (25) is due to the requirement of Eqs. 19 and 20.

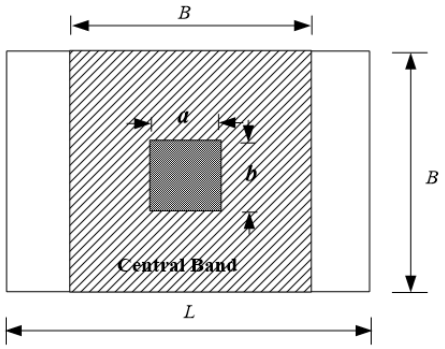


Figure 3 The central band of the footing.

4 Numerical Results

The given design conditions for the eccentrically loaded footings are as follows: the dead load P_D=100 ton, the live load P_L=80 ton, the distance from the footing bottom to the ground surface D_f =1.5m, the unit weight of concrete γ_c=2.4 ton/m^3, the unit weight of soil over the footing γ_s= 2 ton/ m^3 and the allowable soil pressure at the base of the footing q_a=25 ton/m^2. The size of the column transferring the axial load and eccentric moment to the footing is assumed to be 0.40 m×0.40 m. The concrete cover for the reinforcement of the footing is assumed to be 7.5 cm. In order to explore their effects on the optimization of eccentrically loaded footings, the yield strength of steel f_y, compressive strength of concrete f_c', size of the flexural reinforcement or eccentricity e is varied, with the other three fixed. In Taiwan, the unit price of concrete is 1950 NT$/m, 2150 NT$/m^3, 2350 NT$/m^{3}and 2450 NT$/m^3 for f_c' =210 kgf/cm^2 (3000 psi), 280 kgf/cm^2 (4000 psi), 350 kgf/cm^2 (5000 psi) and 420 kgf/cm^2 (6000 psi), respectively; the unit price of steel is 14400 NT$/ton. Design variables are the thickness h, width B and length L of the footing, and the number of steel bars in the long direction N_L and short direction N_B. In this paper, there are six kinds of bar sizes: Nos. 4 to 9; four kinds of f_c': 210 kgf/cm^2, 280 kgf/cm^2, 350 kgf/cm^2 and 420 kgf/cm^2; three kinds of f_y: 2800 kgf/cm^2 (40 ksi), 3500 kgf/cm^2 (50 ksi) and 4200 kgf/cm^2 (60 ksi); and seven kinds of eccentricity: 0 cm, 10 cm, 20 cm, 30 cm, 40cm, 50 cm and 60 cm. The fitness function is the total cost in New Taiwan Dollars of the footing reinforcement and concrete. All the constraints are built according to the formulas discussed in Sec. 3. The population size is set to be 100, crossover rate 0.8, and elite number 5. Furthermore, all the individuals are encoded as integers; "Rank" is used as the scaling function that scales the fitness values based on the rank of each individual; "Roulette" is the selection function to choose parents for the next generation; "Two-point crossover" is used as the crossover method to form a new child for the next generation; The "Adaptive Feasible Function" is selected as the mutation function. The results are discussed as follows.

4.1 Optimal Results by Fixing f_y = 4200 kgf/cm^2, f_c' =210 kgf/cm^2 and e = 10 cm

The size of reinforcement is varied, ranging from No. 4 to No. 9. The optimal results are listed in Table 1, where No. 6 reinforcement has the minimum cost. The lowest cost can be seen clearly in Fig. 4. In addition, when the size of the reinforcement becomes larger, the optimal footing grows square and thicker, as shown in Fig. 5.

4.2 Optimal Results by Fixing f_c' =210 kgf/cm², No. 6 bar and e = 10 cm

There are three kinds of yield strengths of steel: f_y = 4200 kgf/cm², 3500 kgf/cm² and 2800 kgf/cm². The optimal results are listed in Table 2, where f_y = 4200 kgf/cm² has the minimum cost. Besides, when f_y changes, the optimal thickness remains the same.

4.3 Optimal Results by Fixing f_y = 4200 kgf/cm², No. 6 bar and e = 10 cm

There are four kinds of compressive strengths of concrete: f_c' =210 kgf/cm², 280 kgf/cm², 350 kgf/cm² and 420 kgf/cm². The results are listed in Table 3, where f_c' =210 kgf/cm² has the minimum cost. Besides, when f_c' becomes larger, the thickness of the footing turns to be smaller.

4.4 Optimal Results by Fixing f_y = 4200 kgf/cm², f_c' =210 kgf/cm² and No. 6 bar

There are seven kinds of eccentricity explored: e= 0 cm, 10 cm, 20 cm, 30 cm, 40 cm, 50 cm and 60 cm. The results are listed in Table 4, which shows the smaller the eccentricity is, the less cost the footing becomes. Aside from that, when the eccentricity becomes bigger, the optimal footing turns to be thicker.

Table 1 Optimal results by fixing f_y = 4200 kgf/cm², f_c' =210 kgf/cm² and e = 10 cm.

Bar size	h (cm)	B (cm)	L (cm)	N_B	N_L	Cost (NT$)
No. 4	62	270	307	43	38	13,188
No. 5	62	266	311	28	24	13,180
No. 6	62	261	317	20	16	13,166
No. 7	63	300	300	14	14	14,547
No. 8	71	335	335	14	14	20,672
No. 9	79	372	372	14	14	28,628

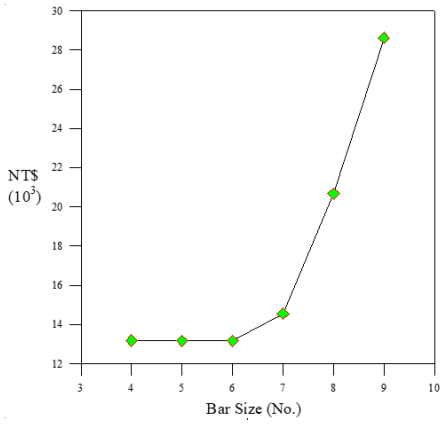


Figure 4 The optimal prices for different bar sizes

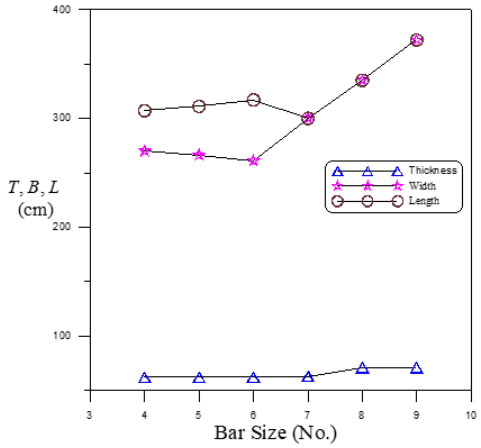


Figure 5 The Optimal Results of thickness, width and length for different bar sizes

Table 2 Optimal results by fixing f_c' =210 kgf/cm^2, No. 6 bar and e = 10 cm.

f_y (kgf/cm^2)	h (cm)	B (cm)	L (cm)	N_B	N_L	Cost (NT$)
2800	62	282	294	27	26	14,715
3500	62	257	322	24	19	13,781
4200	62	261	317	20	16	13,166

Table 3 Optimal results by fixing f_y = 4200 kgf/cm^2, No. 6 bar and e = 10 cm.

f_c' (kgf/cm^2)	h (cm)	B (cm)	L (cm)	N_B	N_L	Cost (NT$)
210	62	261	317	20	16	13,166
280	58	275	301	17	16	13,240
350	55	281	294	17	17	13,683
420	52	285	290	17	17	13,535

Table 4 Optimal results by fixing fy = 4200 kgf/cm2, fc' =210 kgf/cm2 and No. 6 bar.

e (cm)	h (cm)	B (cm)	L (cm)	N_B	N_L	Cost (NT$)
0	62	261	317	20	16	13,166
10	62	261	317	20	16	13,166
20	62	275	301	19	17	13,187
30	64	284	292	19	19	13,715
40	67	286	290	20	20	14,378
50	72	277	300	22	21	15,478
60	86	231	361	33	21	18,724

5 Conclusions

This paper explores effects of the yield strength of steel, compressive strength of concrete, bar size and eccentricity of the axial load transmitted to the footing on the optimization of reinforced concrete isolated footings. Genetic algorithms are used to optimally design the eccentrically loaded reinforced concrete footings. From the numerical results, the principal conclusions may be summarized as follows:

(1) The steel yield strength of 4200 kgf/cm^2, the concrete compressive strength of 210 kgf/cm^2, the smallest eccentricity and No. 6 bar, respectively, will have the optimal results if one of them is varied and the other three are fixed.
(2) When the size of the reinforcement is getting larger, the optimal footing have a tendency to become square and thicker.
(3) When f_y changes, the optimal thickness of the footing remains the same.
(4) When f_c' becomes larger, the optimal footing is getting thinner.
(5) When the eccentricity becomes bigger, the optimal footing grows thicker.

REFERENCES

[1] Rizwan, M., Alam, B., Rehman, F. U., Masud, N., Shahzada, K. and Masud, T., " Cost Optimization of

Combined Footings Using Modified Complex Method of Box," *International Journal of Advanced Structures and Geotechnical Engineering,* Vol. 1, No. 1, pp. 24-28, 2012.

[2] Al-Ansari, M. S., "Structural Cost of Optimized Reinforced Concrete Isolated Footing," *International Journal of Civil, Environmental, Structural, Construction and Architectural Engineering*, Vol. 7, No. 4, pp. 290-297, 2013.

[3] Yeh, J-P and Yeh, S-Y, "Application of Genetic Algorithms Coupled with Neural Networks to Optimization of Reinforced Concrete Footings," *Transactions on Machine Learning and Artificial Intelligence*, Vol. 4, No 4, pp.18-35, 2016.

[4] Holland, J. H., *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, MI, USA, 1975.

[5] Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, Reading, MA, USA, 1989.

[6] Cheng, J and Li, Q. S., "Reliability Analysis of Structures Using Artificial Neural Network Based Genetic Algorithms," *Computer Methods in Applied Mechanics and Engineering*, Vol. 197, No. 45, pp. 3742-3750, 2008.

[7] Belevičius, R. and Šešok, D., "Global Optimization of Grillages Using Genetic Algorithms," *Mechanika*, Nr. 6(74), pp. 38-44., 2008.

[8] Šešok, D. and Belevičius, R., "Global Optimization of Trusses with a Modified Genetic Algorithm," *Journal of Civil Engineering Management*, Vol. 14, No. 3, pp. 147-154, 2008.

[9] Chan, C. M., Zhang, L. M. and Jenny, T. N., "Optimization of Pile Groups Using Hybrid Genetic Algorithms, "*Journal of Geotechnical and Geoenvironmental Engineering*, Vol. 135, Issue 4, pp. 497-505, 2009.

[10] Marandi, S. M., VaeziNejad, S. M. and Khavari, E., "Prediction of Concrete Faced Rock Fill Dams Settlements Using Genetic Programming Algorithm," *International Journal of Geosciences*, Vol. 3, pp.601-609, 2012.

[11] Richardson, J. N., Adriaenssens, S., Coelho, R. F. and Bouillard, P., "Coupled Form-Finding and Grid Optimization Approach for Single Layer Grid Shells," *Engineering Structures*, Vol. 52, pp. 230-239, 2013

[12] Luo, G-M and Hsieh, T-Y, "Optimum Design for CLD Laminate Plates Using Genetic Algorithms," *Open Journal of Composite Materials*, Vol. 4, pp. 106-116, 2014.

[13] Boisvert, J, El-Jabi, N, St-Hilaire, A and El Adlouni, S-E, "Parameter Estimation of a Distributed Hydrological Model Using a Genetic Algorithm," *Open Journal of Modern Hydrology*, Vol. 6, pp. 151-167, 2016.

[14] Yeh, J-P and Hsia, H-M, "Discrete Optimal Design of Reinforced Concrete Short Columns Using Genetic Algorithms," *International Journal of Research Studies in Science, Engineering and Technology*," Vol. 3, Issue 9, pp. 1-10, 2016.

[15] The MathWorks, *Global Optimization Toolbox: User's Guide*, The MathWorks, Inc., Natick, MA, USA, 2015.

[16] ACI 318 Committee, *Building Code Requirements for Structural Concrete* (*ACI 318-11*) *and Commentary*, American Concrete Institute, Farmington Hills, MI, USA, 2011.

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM
TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE
ISSN 2054-7390

Using Artificial Intelligence to Predict Animal Behaviour in Food Webs

[1]Tahir Aduragba, [2]Abdulkadir Ahmed, [3]Bilikisu Ojuolape, [4]Ayodeji Ajani and [5]Yinka Adedoyin
[1234]College of Engineering and Technology, Kwara State University, Ilorin, Kwara State, Nigeria.
[5]Faculty of Communication and Information Science, University of Ilorin, Ilorin, Kwara State, Nigeria.
tahirlanre@hotmail.com; abdulkadiradekunle@live.com; bilkisujimada@yahoo.co.uk;
ajaniaa@gmail.com; yinkaadedoyin22@yahoo.com

ABSTRACT

Overfishing of species in the marine life has caused oceans to become deserts at a fast pace. The population of specific species such as Cod and Haddock has reduced over the years. This has affected countries that hugely depend on them as a source of food. This study used Dynamic Bayesian Network (DBN) to predict animal behaviour in a food web. Two independent biomass surveys from the North Sea were used to learn predictive models and test them on the Northern Gulf Ocean. The resulting predictive model is expected to unveil useful information about what affects the population of fishes in the Northern Gulf Ocean. In addition, the predictive model was used to make predictions into the future about the effects of tampering with the population of specific species of fish in the same region. The focus was on the Cod species in the George's Bank in relationship to species network in their food web. Looking at their biomass states and the effects it has on the hidden dependence when there is a change in their biomass states. Also, the different predictive models were used to evaluate species in the George's Bank based on their performance. The result from the experiment shows that there is a hidden dependence, which is responsible for the collapse of species (Cod); due to the temperature or salinity of the ocean.

Keywords: Fisheries management, data mining, Bayesian network and hidden Markov model

1 Introduction

"About 20% of the world's population derives at least one-fifth of its animal protein intake from fish, and some small island states depend almost exclusively on fish" [1]. Overfishing of species in the marine life has made oceans to become deserts quickly and it has also caused loss of some species as well as entire ecosystem. Due to this, the overall ecological unity of the oceans is under stress and at risk of collapse. The ocean is at risk of losing a valuable food source many depend upon for social, economical or dietary reasons [2]. The over-exploitation and mismanagement of fisheries has already led to some spectacular fisheries collapsing. For example, the Cod fishery of Newfoundland, Canada collapsed in 1992, leading to the loss of some 40,000 jobs in the industry. Further, the Cod stocks in the North Sea and Baltic Sea are now heading the same way and are close to complete collapse [3]. Per Darwin's description of tangled bank, he emphasized that tempering with the population of one species of fish can cause surprising and dramatic changes in the population of others [4]. Overfishing of a specific fish has caused increase in the population of their competitors because there is less consumer-resource competition between the

DOI: 10.14738/tmlai.55.3568
Publication Date: 27th August, 2017
URL: http://dx.doi.org/10.14738/tmlai.55.3568

remaining fishes in the ecosystem. Per the Sequential Population Analysis (SPA), total population of Cod in the Northern Gulf declined from 539 million in 1980 to 31 million in 1994, then slowly increasing to attain 58 million individuals in 2010 [5]. Cods population in this region remains depleted and the reasons behind this remain disputed [6]. Although in the North Sea such collapse in Cod population occurred, the population tends to be recovering over the years [7]. The difference between the recoveries of Cod population in these regions remains unrevealed. Consequently, ban on fishing Cod species from the Northern Gulf Ocean has been introduced but this has not helped in the recovery of these species. However, in this changing world, understanding ecosystem stability and fragility is of growing importance—yet to do so there must be an understanding of the networks that forms the systems [8].

A large amount of data is collected and stored in storage devices every day from business, science and engineering and almost every facet of our daily life. This explosive growth of available data volume is a result of the automation of our society and the progress achieved in developing powerful data collection and storage tools. Therefore, there is desperate need for powerful tools to automatically reveal valuable information from the huge amount of data and to transform such data into organized knowledge. This necessity has led to the birth of data mining [9]. Data mining can be described as the process of discovering patterns in data that are meaningful and can be used advantageously [10]. Although, data collected from ecological sources has been less explorative compared to data collected from other sources mentioned above [11]. In this paper, we apply data mining tools including dynamic Bayesian network and Hidden Markov models to fisheries data to identify species that perform similar functional roles both in the George's Bank and North Sea and these species are used to predict functional collapse in their respective fishing communities.

Bayesian Network (BN) is an exceptional case of a broader class called graphical models, where nodes represent random variables, and the conditional independence assumption is represented by the lack of arcs. BNs do not deal with time, however the Dynamic Bayesian Network (DBN) does by representing how these random variables evolve over time [12]. DBNs are devised to model probability distributions over a sequence of random variables to manage sequenced observations that are propagated by some fundamental hidden states that evolve in time [13]. DBNs consist of two networks. One of them represents the prior probabilities of all variables in the network in the initial time slice $(i.e\ t = 0)$. This is known as the prior network. While the other one represents the probabilities of all the variables in all other time slices $(i.e\ t = 1,2, \dots n)$. This is known as transition network [14].

From the graph below (figure I), X_3 is independent of X_1 given X_2. This explains the first order Markov property that the future is independent of the past given the present. Hence, this graph entails the basic concept of HMM. HMMs, Kalman Filters, Vector Quantization, etc. are all variants of DBNs [15]. This project will implement Hidden Markov Models (HMM) as DBNs. The goal of implementing an HMM as a DBN is to infer the hidden state given the observation sequence, which can be represented as $P(X_t = i|O_{1:t})$. Implementing HMM based models as Dynamic Bayesian Networks (DBN) facilitates compact representation as well as additional flexibility regarding the model structure [14].

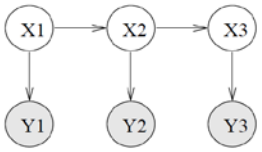


Figure I - Representation of Dynamic Bayesian Network. Observed nodes are shaded, whereas hidden nodes are not shaded

Tucker and Duplisea (2013) used bioinformatics techniques to exploit functional equivalence between different fisheries datasets and used the identified species in conjunction with a dynamic model that uses latent variables to predict functional collapse (and future biomass). The latent variable was used partially to represent something's external to the fish community such as oceanographic conditions. They explored this further by using data of likely factors such as temperature, nutrients and fishing mortality. The result suggests that changes in conditions external to the fish community may be responsible for collapse in GB and ESS.

2 Material and Methods

2.1 Data Source

To conduct the research the data about species biomass and food webs in the studied oceans were used. The information about the species and their food webs in the George's Bank Ocean were provided by the Department for Fisheries and Ocean in Canada while the data sets for the North Sea were provided by the International Council for the Exploitation of Sea. In the George's Bank, the data set contains biomass surveys of thirty-nine species from 1963 - 2008 and for the North Sea biomass surveys of forty-four species from 1967 - 2009 were provided in the data sets. This data is all recorded as typical continuous values. Information about the food web was collected by examination of the species stomach content and recording the fishes that were found in their stomach. This was done for the species in both oceans. Although there are many species, only an overlap of species will be used to carry out the various experiments. These species include: Cod, Herring, Cusk, Mackerel, Spiny dogfish and Red Hake. The choice of these species was based on their trophic relationships to one another. For example, Herring was chosen because it is a prey of Cod. However, the data sets contain missing values, so it was decided to interpolate the data to fit in missing values in the data sets. Although, DBNs are capable of handling missing values from the data sets. Furthermore, the data will also be discretized into simple discrete values.

2.2 Experiments

The experiments undertaken in this paper involve discretization, inference and prediction. To represent the continuous values into a small number of finite values, the data can be discretized. Discretization of real data into a typically small number of finite values is often required by Bayesian Net apps [16]. The biomass surveys are discretized into two qualitative states as low and high. These states are interpreted as the relative weight of species at time slice. For example, if the relative weight of Cod species is 1 at a time, then it is in a low state. However, sometimes they are discretized into three qualitative states low, medium and high to test how they affect the predictions. Number 1 represents low, number 2 medium and number 3 high. Table 1 shows the hypothetical relative weight of species.

Table 1: Hypothetical relative weight for species

States	Low	High
Relative weight	1	2

An important aspect of this paper is trying to predict the behaviour of species over time. To do this, $P(X_{t+h}|y_{1:t})$ needs to be computed where h is how far to predict. Once the future hidden state is known,

the information about the future observation state can be retrieved by marginalizing out X_{t+h} as shown in equation 2.1.

$$P(Y_{t+h} = y|y_{1:t}) = \sum_x P(Y_{t+h} = y|X_{t+h} = x)P(X_{t+h} = x|y_{1:t}) \quad (1)$$

The goal of inference can be achieved by using the Junction Tree algorithm. The simplest approach to applying junction tree algorithm to DBNs for inference is by unrolling the DBN for "T" slices and applying the algorithm to each static Bayes net. The disadvantage of this approach is that it takes too long for algorithm to run out of memory if the unrolled DBN becomes huge, i.e. there are many time steps. This is the approach used for the all experiments because it allows relative weight of species over a period to be entered in form of evidence. It was used because the number of time steps involved is not much. However, a better approach to applying Junction Tree algorithm to a DBN is by representing the DBN using only the first two time slices of a process. This approach is based on the first order Markov property that the current time slice is only dependent on the preceding time slice and not any previous time slices. It is much faster than unrolling the DBN into several time slices. The Junction Tree algorithm applied to DBNs for inference can be outlined as follows [17]:

1. Initialization
   A. On initialization, Junction Trees, J_1 and J_t are created
      I. J_1 is the junction tree for the initial time slice and is created from time slice 1 of the 2TBN
      II. J_t is the junction tree for each subsequent time slice and is created from time slice 2 of the 2TBN and the outgoing interface of time slice 1
   B. Time is initialized to 0
2. Queries
   A. Marginals of nodes at the current time slice can be queried
      I. If current time = 0, queries are performed on "_1" nodes in J_1
      II. If current time > 0, queries are performed on "_2" nodes in J_t
3. Evidence Application
   A. Evidence can be applied to any node in the current time slice
      I. If current time = 0, evidence is applied to "_1" nodes in J_1
      II. If current time > 0, evidence is applied to "_2" nodes in J_t
4. Advance
   A. Increment time counter
   B. Use outgoing interface from active time slice to do inference in next time slice
      I. Since the outgoing interface d-separates the past from the future, this ensures that when inference is done in the next time slice it takes everything that has occurred "so far" into account.

The detailed biomass state prediction procedure is as follows:

Step 1: Collect data about biomass surveys of species to use in building the model.

Step 2: Discretize data sets into qualitative states. For example, into two states: low and high.

Step 3: Build the DBN from the collected data and train it.

Step 4: After training the model, the inference model can be used to compute the future states, $P(X_{(t+h)}|y(1:t))$ using equation (2.1)

Step 5: Find the value of $maxP(X_{(t+h)}|y(1:t))$ using equation 2.1 and return as the most probable state.

The first experiment involves simulating population of species over a certain period. Essentially, specific states of biomass of species are entered for certain annual periods and predictions over certain annual periods are made based on the built model as to how these populations behave over certain annual periods. These states are entered in the form of evidence. For example, the biomass state of Cod is set to be low and the remaining species used to build the model are set to be high over a five-year period. So, the behaviour given overtime can monitor this information. Experiment 1 tries to predict if Cod species can recover in the George's Bank and identifies the conditions that are responsible for this behaviour. Throughout this experiment, four species are used to build the model which includes Cod and Haddock. However, the other two species used have a kind of trophic impact be it negative or positive on them.

While conducting the first experiment, predictions were made. To determine how accurate these predictions are, cross validation was used to evaluate the built predictive model. The second experiment involves performing cross-validation to ascertain how accurate the predictions are. Cross-validation is a statistical method of evaluating and comparing learning algorithm by dividing data into two segments: one use to train a model and the other used to validate the model. There are several methods that can be used for cross-validation but in this study, 10-fold cross-validation was used because it is the most common method in data mining and machine learning [18]. While performing cross-validation, certain performance measures can be used. This experiment used sensitivity analysis in the form of confusion matrix to effectively compare the performance of predictive models. A confusion matrix is a visualization tool used to determine the predictive capability of a model. Each row of the matrix represents the actual value while the column of the matrix represents the predicted value. From this, the accuracy of prediction can be determined by estimating the proportion of the total number of correct predictions. The equation is given below:

$$\text{Accuracy, A = correct / (correct + incorrect)} \tag{2}$$

The cross-validation estimation of the overall accuracy can be defined as,

$$CVA = \frac{1}{k}\sum_{i=1}^{k} A_i \tag{3}$$

Where, CVA is the average accuracy at each fold, k is the number of fold and A is the accuracy at each fold.

The third experiment involves testing the learnt model on an unseen data. This is similar to the second experiment however; the difference is that the model is tested on an independent data. In this case, several species from the North Sea were used to build a predictive model and this model was tested on the same species from the George's Bank. The main aim of this experiment is to see if the conditions that helped Cod species recover in the North Sea can be applied to the Cod species in the George's Bank.

The final experiment uses information from the food web to build a model using different species and trying to predict the behaviour over their population over time by simulating their populations over

certain periods. Different species were used in building a model based on the information about their trophic impact on each from the food web. For example, predators or prey of Cod are switched to see how Cod relates to them in different conditions.

3 Results and Discussion

In the first experiment, an initial model was built using Cod, Haddock, Herring and Cusk from the George's Bank. These species were chosen based on the information that Cusk is the predator of Cod and Haddock, Cod is the predator of Haddock and Haddock is the predator of Herring. The entire species nodes were represented in the two usual states (low and high). After building the model it was decided to enter evidence that for the first-time slices, Cod is set to be low. This is because the major interest is trying to simulate recovery of Cod. However, the biomass states of other species are randomly entered. In this case, the states of other species are set to be high.

Figure 2: The prediction results of the biomass state of four species

Given these evidences, the model produced the result in figure II. It is expected for the biomass state of Cod to be high because the biomass state of Haddock (prey of Cod) is also high. The biomass state of Cod increases almost immediately. After certain time slices, the biomass state of Cod dropped to a low state. Notes were taken on the hidden states at each of the time slices. From figure III, it was discovered that there was change of state in the hidden dependence immediately the biomass state of Cod dropped. This result suggests that there is a hidden dependence, which is involved with the collapse of Cod biomass state in the George's bank.

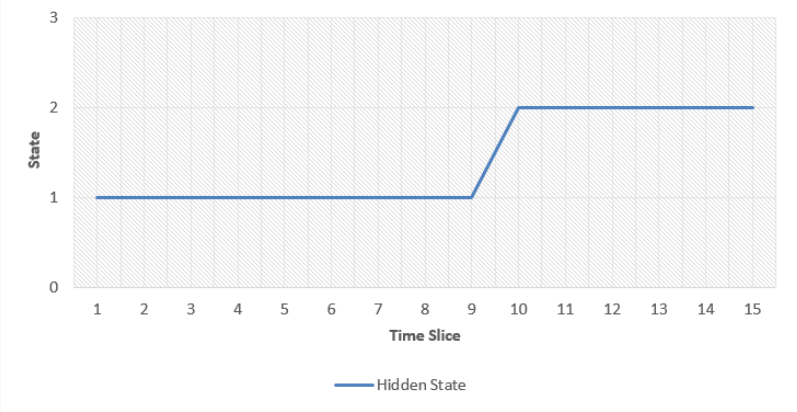


Figure 3: The plot of the hidden dependence

Based on this network, evidences were given that the biomass state of Cod is high, the biomass state of Haddock is high and the biomass state of Cusk (the predator of Cod) is low. It is expected that Cod should remain high due to high biomass of its prey (Haddock) and low biomass of its predator (Cusk). Given this condition, the Cod species remained high for a long number of time slices but later dropped after twelve time slices. (See figure IV)

Figure 4: The prediction results of the biomass state of four species

A new model was constructed but this time the network excluded the predator of Cod. It was replaced with a prey of Cod. The prey of Cod used is Spiny dogfish. Evidences were entered that Cod species are in a high state as well as other species in the network for five time slices. This favours the Cod species, as they remained in a high state most of the time; also, there is no change of state in the hidden dependence. (See figure V)

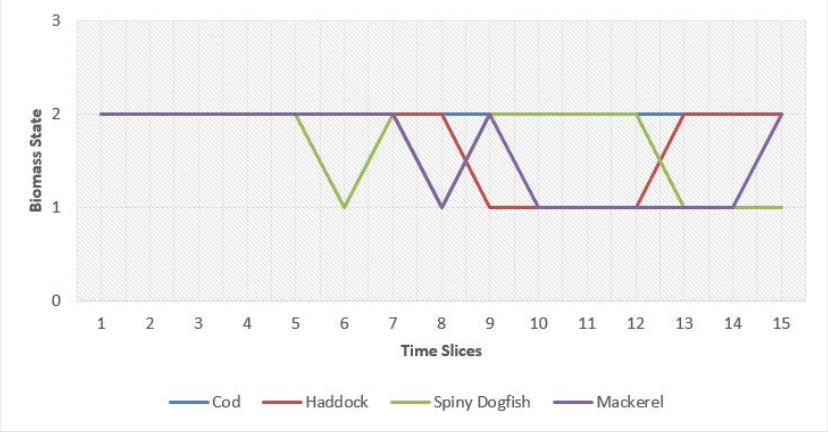


Figure 5: The prediction results of the biomass state of four species

Following the initial model, where the network comprised of the prey of Cod, a new model was introduced to include a predator of Cod, Red Hake. The evidence entered is that the biomass of Cod is low, the biomass of Haddock is high, as well as Spiny dogfish and the biomass of Red Hake is low over five time slices. Interestingly, Cod rises immediately, however after two time slices it drops into a low state. Moreover, all species in the model, dropped immediately, while the hidden dependencies were changing. This suggests a possible correlation between the biomass of Cod and the hidden dependencies. The result is shown in the figures VI & VII.

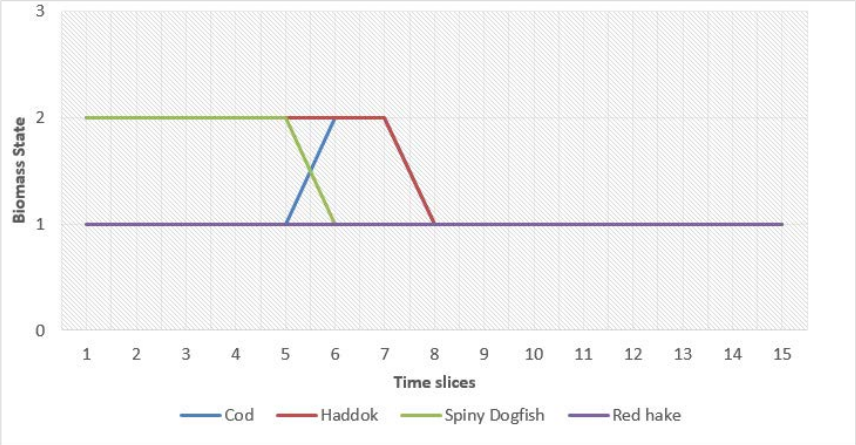


Figure 6: The prediction results of the biomass state of four species

Figure 7: The plot of the hidden dependence

In a completely new model, evidence was entered that the biomass state of Cod and Herring were both high, with Haddock and Cusk to being low, for the first five time slices. There was no recovery of Cod in this network, where changes to the hidden states were again found.

The models used for prediction were evaluated based on the performance measures described earlier using equations 2.2 and 2.3. The results were achieved by using ten-fold cross-validation for each model. Each model was built using the same species. For the second experiment, the models were built using Cod, Haddock, Herring and Cusk species from the North Sea and the observation sequence is represented using two states (low and high). These built models are used to predict the biomass states of Cod species over time and the results are validated using the known biomass of Cod species from the George's Bank Ocean. Table 2 shows the complete set of results for each fold with the overall accuracy calculated based on the average results obtained for each fold. The correctly classified states in the datasets are in yellow and the incorrect are in white. On average, it was discovered that the Auto-Regressive HMM model achieved an accuracy of 0.8150 while the DBN achieved an accuracy of 0.7900. In addition, the standard deviation of the accuracy for the Auto-Regressive HMM model is lower than that of the DBNs. This implies that the Auto-Regressive HMM model show more consistency while predicting. Overall, the Auto-Regressive HMM model performed best of both the models used for this experiment. Confusion matrices were also created based on the total number of correct and incorrect classifications (after the 10^{th} fold) to illustrate the performance of these models.

Table 2: Details of accuracy and confusion matrix at each fold of 10-fold Cross-Validation

Fold No.	Auto regressive HMM			Dynamic Bayesian Network		
	Confusion Matrix		Accuracy	Confusion Matrix		Accuracy
1	0	0		0	0	
	0	4	1.0000	0	4	1.0000
2	0	1		0	1	
	1	3	0.6000	0	4	0.8000
3	2	1		0	3	
	1	1	0.6000	0	2	0.4000
4	0	0		0	0	
	0	5	1.0000	0	5	1.0000
5	0	1		0	1	
	0	4	0.8000	0	4	0.8000
6	5	0		5	0	
	0	0	1.0000	0	0	1.0000
7	1	1		1	1	
	2	1	0.4000	2	1	0.4000
8	2	1		3	0	
	0	1	0.7500	1	0	0.7500
9	4	0		4	0	
	0	0	1.0000	0	0	1.0000
10	4	0		3	1	
	0	0	1.0000	0	0	0.7500
Mean			0.8150			0.7900
St Dev.			0.2212			0.2319

While defining DBNs, assumptions are made as to the number of states used to represent the observation sequence of the model. These assumptions are made based on the observed data sets. For this project, all models were built using two states based on the biomass for all species (Low and High). However, it was decided to increase the number of states to three (low, medium and high) and see what effects it has on the accuracies of the models. Same information except for the number of states, which was increased to three, was used to perform 10-fold cross-validation.

As shown in table 3, the accuracies for both the Auto-Regressive HMM model and the DBN dropped to 0.6250 and 0.5870 respectively. Increasing the number of states does not increase performance in terms of accuracy rather it reduces it. Hence, it is concluded that the models should be built using small groups of data sets to achieve better performance.

Table 3: Accuracies of three state representations

Auto-Regressive HMM			Dynamic Bayesian Network		
15	0	0	12	3	0
2	3	10	1	6	8
1	5	10	0	7	9
Accuracy = 0.6250			Accuracy = 0.5870		

As earlier described the Cod species suffered overfishing in both the North Sea and the George's bank. However, the Cod species in the North Sea tends to be recovering but it's a different case in the George's bank. One of the experiments conducted in this project is to see how a predictive model learnt from an ocean performs on another independent ocean. The aim of this experiment is to see whether the condition that favored cod species in the North Sea can be applied to Cod species in the George's bank. The network was built using species present in both oceans. This includes Cod Haddock, Spiny Dogfish and Cusk. The two usual biomass states (low and high) were used based on this network.

Table 4 shows the result of the experiment. Analysing the confusion matrix, the accuracy of prediction is 0.50. Also, it tends to predict the high state of biomass for Cod species most of the time. This is expected because this is the situation of Cod in the North Sea. This suggests that the model doesn't fit in and the conditions present in both oceans are different. However, to ascertain this, another network was built using different species. The species used are Haddock, Red Hake, and Spiny Dogfish. The choice of choosing these species was based on the fact that apart from Cod, the other species behave in similar ways in both oceans. From this network, an accuracy of 0.53 was achieved. This showed an improvement from the previous network. However, it doesn't still give an encouraging result to draw a conclusion on this experiment.

Table 4: Confusion Matrices for (a) Auto-regressive HMM and (b) Dynamic Bayesian Network

		Predicted	
		Low	High
Actual	Low	12	9
	High	15	10

(a)

		Predicted	
		Low	High
Actual	Low	13	8
	High	12	12

(b)

The final experiment is similar to the first, where initially there were four species, now it is reduced to two, where the species were switched based upon the food web information. The entire species nodes were represented in the two usual states (low and high). The species Cod and Cusk were used to build the first model. Cusk is a primary predator of Cod. Evidences were entered that the biomass states of Cusk were high for the first time slices, while the biomass state of Cod was set to be low, with a fluctuation between the first 5 times slices. Given the evidence, the biomass state of Cod remained high up to the ninth time slice, then dropped immediately to a low state thereafter, while the biomass state of Cusk

remained high, then it dropped after three times slices. However, there was no change in the hidden states until after this occurs. The result of this model is shown in the figures VIII & IX below.

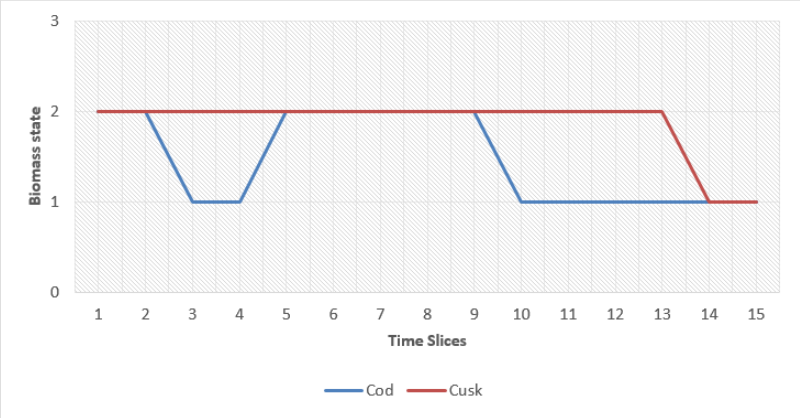


Figure 8: The prediction results of the biomass state of two species

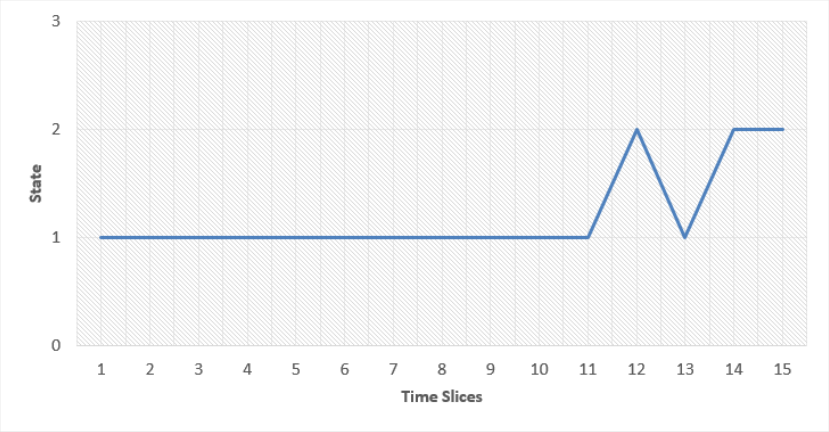


Figure 9: Plot of the hidden dependence

In a second model, using the network of Cod and Haddock (the prey of Cod), the evidence entered that the biomass state of Cod fluctuated between high and low, in respect to the biomass state of Haddock which was relatively high for the first five times slices. Interestingly, after the evidence was entered, the hidden dependences were unchanged for up to ten time slices; even though the biomass state of Cod continued to fluctuate. However, when the state of the hidden dependence changes, the biomass state of Cod, drops and then remains in a low state, which prior to this had consistently been in fluctuation. The result of this model is shown in the figures X & XI below.

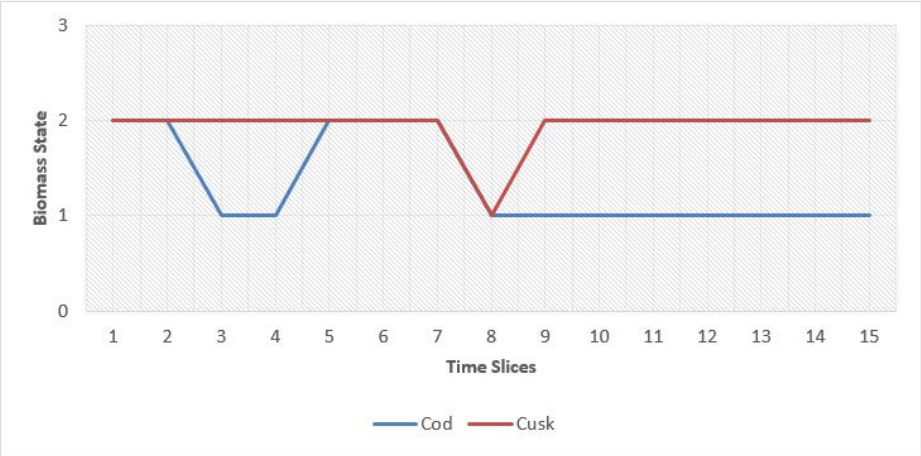


Figure 10: The prediction results of the biomass state of two species

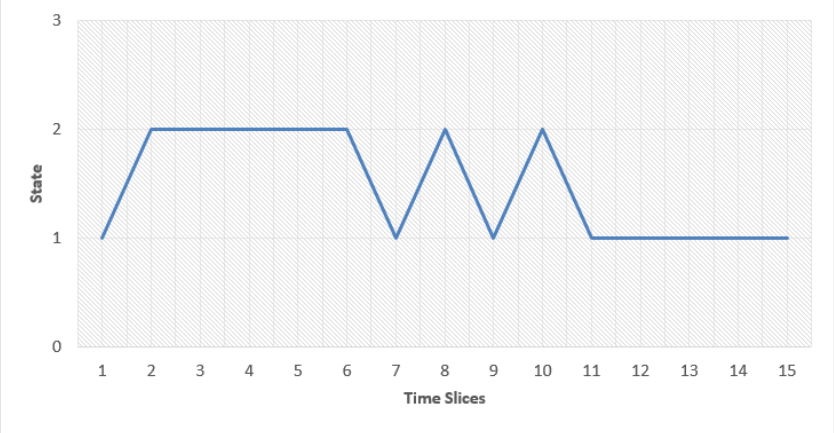


Figure 11: Plot of the hidden dependence

4 Conclusion

This paper looks at the potential correlation between the biomass state of Cod and the changing or unchanging state of the hidden dependencies. Overall, the Cod species did not recover, however it was discovered that there is a change in the state of the hidden dependencies when the biomass state of Cod drops, suggesting that there might be other factors affecting the Cod population. Species such Cod have suffered overfishing, which appear to cause danger to their biomass state, however it is not a conclusive reason for their inability to make a recovery, as they are no longer being fished. New experimentation suggests that the biomass state of the Cod and other network species such as Haddock (the prey of Cod) and Cusk (a predator of Cod), while observing the state of the hidden dependencies, may all be affected by externalities which remain unknown. However, this study investigated this, using data mining techniques to model dependencies of fishes in marine life and it was discovered that there is a hidden dependence, which is responsible for the collapse of species (in this case Cod); possibly the temperature or salinity of the ocean that has affected the species.

REFERENCES

[1] World health Organisation (2011). *Global and regional food consumption patterns and trends.* Available: http://www.who.int/nutrition/topics/3_foodconsumption/en/index.html

[2] Overfishing.org (2011) *Why overfishing is a problem.* Available at: http://overfishing.org/pages/why_is_overfishing_a_problem.php

[3] Greenpeace International (2010). *Overfishing. [Online]* Available at: http://www.greenpeace.org/international/en/campaigns/oceans/overfishing/

[4] Ali, F., Frank, D., Dirk H., Colin M. (2010). Inferring Species Interaction Networks from Species Abundance Data: A comparative evaluation of various statistical and machine learning methods. *Ecological Informatics*, 5 (6), 451-464.

[5] DFO (2010). Assessment of Cod stock in the northern Gulf of St. Lawrence (3Pn, 4RS) in 2009. *DFO Canada Science Advisory Secretariat Science Advisory Report 2010/011*

[6] DFO-SCIENCE (2009) *What's Holding Back the Cod Recovery?* [Online] Available at: http://www.dfo-mpo.gc.ca/science/Publications/article/2006/01-11-2006-eng.htm

[7] The Independent (2007). *North Sea cod population recovering, scientists say*. [Online] Available: http://www.independent.co.uk/environment/nature/north-sea-cod-population-recovering-scientists-say-397350.html

[8] Dunne, J. A., Richard J. W, and Martinez N. D. (2002). Network structure and biodiversity loss in food webs: robustness increases with connectance. *Ecology Letters* Vol. 5:558–567

[9] Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.

[10] Witten I.H. & Frank E. (2005) *Data Mining: Practical Machine Learning Tools and Techniques*. 2nd ed. San Francisco: Morgan Kaufmann.

[11] Tucker, A. and Duplisea, D. (2012). Bioinformatics tools in predictive ecology: applications to fisheries. *Philosophical Transactions of the Royal Society, Biological Science* 367(1586), 279-290

[12] Kevin Murphy (2002). Dynamic *Bayesian Network: Representation, Inference, and Learning*. PhD Dissertation, University of California, Berkeley.

[13] Thomas Dean and Keiji Kanazawa (1989). A model for reasoning about persistence and causation. *Artificial Intelligence*, 93(1-2), 1-27.

[14] Fatih C, Ratna B C (2005). Dynamic Bayesian Networks for Machine Diagnostics: Hierarchical Hidden Markov Models vs. Competitive Learning. *Proceedings of International Joint Conference on Neural Networks.* Vol. 3. pp 1752 – 1757.

[15] Smyth, P. (1997). Belief networks, hidden Markov models, and Markov random fields: a unifying view, *Pattern Recognition Letters* 18(11-13) 1261-1268.

[16] Friedman, N. and Goldszmidt, M. (1996) Discretization of continuous attributes while learning Bayesian networks. *Proceedings Of the 13th International Conference on Machine Learning*, 157-165.

[17] Gimpel K (2005). Junction Tree Algorithm for Inference in DBNs. School of Computer Science, CMU, United States.

[18] Payam R, Lei T, Huan L (2008). *Cross-Validation*. Arizona State University, Arizona, USA.