Analysis of ANFIS and ANN for Building Automated Surrogate Algorithms

Devi Priyanka Pantula

A Dissertation Submitted to

Indian Institute of Technology Hyderabad

In Partial Fulfillment of the Requirements for

The Degree of Master of Technology

भारतीय प्रौद्योगिकी संस्थान हैदराबाद

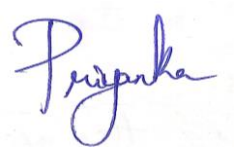
Indian Institute of Technology Hyderabad

Department of Chemical Engineering

June, 2016

Declaration

I declare that this written submission represents my ideas in my own words, and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources that have thus not been properly cited, or from whom proper permission has not been taken when needed.



Devi Priyanka Pantula

CH14M16P000004

Approval Sheet

This thesis entitled "Analysis of ANFIS and ANN for Building Automated Surrogate Algorithms" by Devi Priyanka Pantula is approved for the degree of Master of Technology from IIT Hyderabad.

Dr. Debaprasad Shee
Examiner

Dr. Chandra Shekhar Sharma
Examiner

Dr. Kishalay Mitra
Adviser

Dr. Saptarshi Majumdar
Co-Adviser

Dr. Raja Banerjee
Chairman

Acknowledgements

I would like to express my humble gratitude to my thesis supervisor Dr. Kishalay Mitra. This project would not have been possible without the continuous support and guidance of Dr. Kishalay Mitra who always encouraged me by giving values to my thoughts and motivated me to move in correct path.

I am very much grateful to my co guide Dr. Saptarshi Majumdar for his motivation and encouragement throughout the course of project.

I would like to specially thank my committee members: Dr. Chandra Shekhar Sharma, Dr. Debaprasad Shee, and Dr. Raja Banerjee for their time to time support and guidance throughout my research in my Master's program.

I would like to thank all the esteemed faculty members of Department of Chemical Engineering and the staff of IIT Hyderabad for never letting me down at any moment during my project tenure.

For all my friends, colleagues and family members in and outside IIT Hyderabad who were always there for me in the times of joy and sorrow, thank you very much.

I would finally like to thank my parents without whose support I would have never been able to fulfill any of my aims.

Thank you one and all.

Abstract

While attaining the objective of online optimization of complex chemical processes, the possibility of using the first principle based models is rarely an option, since such models demand large computational time. Surrogate models, which can emulate first principle based models, offer a credible solution to this problem by ensuring faster optimization. Thus, the entire challenge of enabling online optimization of complex models depends on construction of efficient surrogate models. Often, the surrogate building algorithms have certain parameters that are usually fixed based on some heuristic, thereby inviting potential errors in building such surrogate models. This work aims at presenting an elaborate study on the effect of various parameters affecting the predictability of Adaptive Neuro Fuzzy Inference Systems viz. (a) architecture of ANFIS, (b) sample size required by the ANFIS, (c) maximum possible accuracy of prediction, (d) a robust sampling plan. The ANFIS is then utilized as surrogates for a highly nonlinear industrial PVAc process, the optimization of which is then realised nearly 9 times faster than the optimization study using the expensive phenomenological model. A brief study was also conducted on another well-known class of surrogates, Artificial Neural Networks, for modelling of the Electrospinning process.

Index Terms— ANFIS, nonlinear models, Online optimization and control, Parameter in surrogate construction, Surrogate models, PVAc process, ANNs, Electrospinning process.

List of Figures

List of Tables

.

Contents

5 Conclusion

Chapter 1

Introduction

Modern technologies demand for optimization of processes to ensure their practical implementation in the industrial scale. Owing to the usage of robust first principle models, which try to capture the dynamics of reaction networks in a polymer industry or a model handling the wake effects or turbulence in fluid flow, etc., process modelling and optimization of complex industrial problems turn out to be a rigorous task. The first principle models involve solving of several nonlinear coupled ordinary and partial differential equations (ODEs and PDEs) [1] thereby compelling the usage of time consuming simulation packages such as Differential Algebraic Equations solvers (DAE), Computational Fluid Dynamics (CFD) and so on. The inherent complexity of these models considered for optimization forms the genesis for the large computational time consumed by the optimizer, thus compelling the entire process to run over several days or months [2]. The problem increases by multiple folds when the considered system is multi-dimensional in nature (say n dimensions) with optimization formulation involving multiple conflicting objective functions instead of one. The conflicting nature of objective functions results in a set of non-dominating solutions known as Pareto-Optimal (PO) solutions, which are mostly obtained using a multi-objective evolutionary or classical algorithm. A single solution is chosen from the Pareto optimal set based on some higher order information, often provided by the decision maker [3]. The solution obtained in

such a way aims at enabling a decision support system to program and simulate the given process in an optimal fashion. Industrial sectors practically implement this concept of online optimization only when the combined functioning of optimizer and controller is realised in real time of the live process. The tremendous industrial growth and ever expanding demand over the last decade have created strong need for the solutions, which could cater multiple objectives at the same time. This requires solving the underlying multi-objective optimization problem (MOOP). Till date, owing to the advent of fast computing machines, the ability of modern evolutionary methods for solving the MOOP has remained unparalleled [4]. On the other hand, due to the predominant condition, wherein lack or expensive computation of gradient information of the complex models has become a common scenario, the modern evolutionary optimization techniques have gained enormous prominence over their classical counterparts, which provide every future course of movements depending on the current gradient information [3]. The procedure of solving the MOOP by the robust evolutionary techniques, which primarily work with population of candidate solutions, necessitates multiple function evaluations in order to generate those solutions required in the optimization process [5]. These aspects together make the concept of online optimization a far-fetched impractical concept confined to theory, which cannot be realized practically unless the optimization happens in real time.

The key to this problem lies with fast and accurate surrogate models, which essentially are data based models trying to emulate the given complex first principle or physics based models. These surrogates then replace the original physics based models in the optimization algorithm thereby shielding them from the optimizer while generating the candidate solutions. With surrogates in place, the entire optimization algorithm may proceed in a fast manner thus enabling a step towards online optimization. In spite of displaying immense applicability in scientific and engineering domains, the data-based modelling still has large amount of

unexplored potential, when unleashed would enable fast and efficient methods of modelling, eliminating the practical issues associated with the physics based models such as stability, convergence, large computational times, feasibility, etc. Particularly, the thrust lies with enabling the state-of-the-art data based modelling approachess such as Kriging Interpolators (KI) [9], Artificial Neural Networks (ANNs) [16] and Support Vector Machines (SVM) [6], to capture the voluminous information about the system or process by incorporating the easily available knowledge base in the form of human experience [11]. On the other hand, credible amount of research put in Fuzzy Information Systems (FIS) over the past few years [11] has led to the development of robust inferencing engines which can efficiently model the available knowledge base using a set of simple fuzzy rules. However, they remain incapable of learning from crisp data set. Recent works [23,24] reported in the literature are aimed at exploiting the advantage of knowledge based models such as FIS and data-based models such as ANNs to develop the Adaptive Neuro Fuzzy Inference Systems (ANFIS) that can learn from crisp data apart from being able to incorporate the available expert knowledge. ANFIS is one of the prominent candidates of surrogate models which has the ability to combine the benefits of Neural Networks and Fuzzy Inference system [12]. An adaptive neuro fuzzy architecture consists of a feed forward multi layered network where each node performs a particular function on the incoming signals as well as a set of parameters pertaining to this node.

Consider a system having two inputs x and y and one output z. For a first-order Takagi-Sugeno fuzzy model, a common rule set with two fuzzy if-then rules is the following:

- **Rule 1:** If x is A_1 and y is B_1, then $f_1=p_1x+q_1y+r_1$;
- **Rule 2:** If x is A_2 and y is B_2, then $f_2=p_2x+q_2y+r_2$.

The corresponding equivalent ANFIS architecture is as shown in Fig.1. The description of each the layers is given below [12].

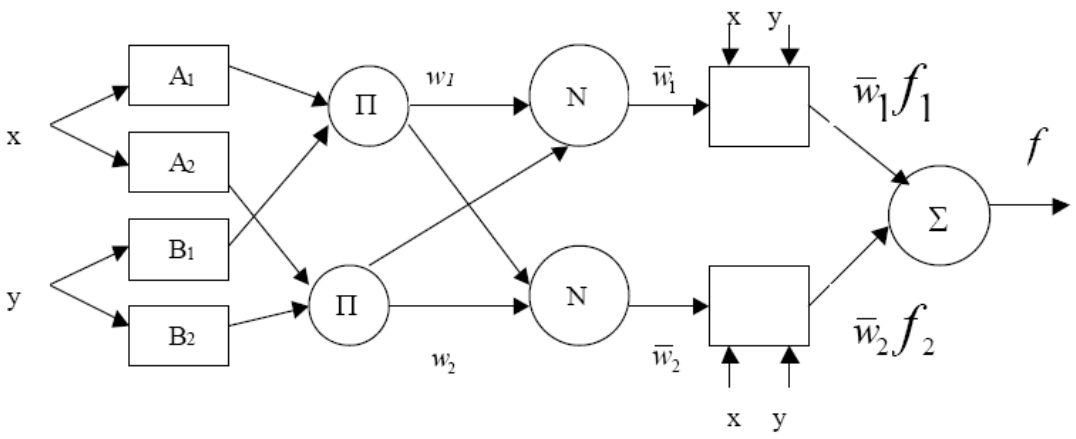


Fig. 1. Structure of ANFIS

Layer 1: Every node *i* in this layer is an adaptive node with a node function as shown below:

$$O_{1,i} = \mu_{A_i}(x), \quad \text{for } i = 1, 2, \text{ or}$$
$$O_{1,i} = \mu_{B_{i-2}}(y), \quad \text{for } i = 3, 4,$$

Here x (or y) is the input to node i and A_i (or B_{i-2}) is a linguistic label (such as "small" or "large") associated with this node. In other words, $O_{1,i}$ is the membership grade of a fuzzy set A ($=A_1, A_2, B_1$ *or* B_2) and it specifies the degree to which the given input x (or y) satisfies the quantifier A. Here the membership function for A can be any appropriate parameterized membership function such as the generalized bell function:

$$\mu(x) = \frac{1}{1 + \left|\frac{x-c}{a}\right|^{2b}}$$

where $\{a, b, c\}$ is the parameter set. As the values of the parameters change, the shape of the bell-shaped function varies. Parameters in this layer are called *premise parameters*.

Layer 2: Every node in this layer multiplies the incoming signals and sends the output signal to the next layer. In general, any other T-norm operator that performs fuzzy AND can be used as the node function in this layer. Each node output represents the firing strength of a rule.

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

Layer 3: In this layer, the i^{th} node calculates the ratio of the i^{th} rule's firing strength to the sum of all rules' firing strengths. The outputs of this layer are called normalized firing strengths.

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

Layer 4: Every node i in this layer is an adaptive node with a node function represented as:

$$O_{4,i} = \overline{w}_i f_i = \overline{w}_i(p_i x + q_i y + r_i),$$

where $\bar{w_i}$ is a normalized firing strength from layer 3 and $\{p_i, q_i, r_i\}$ is the parameter set of this node. Parameters in this layer are referred to as consequent parameters.

Layer 5: The single node in this layer computes the overall output as the summation of all incoming signals:

$$\text{overall output} = O_{5,1} = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}$$

Thus an adaptive network that is functionally equivalent to a Sugeno fuzzy model has been constructed.

Apart from the network structure, the sample size required for training also effects the predictability of the network significantly in accordance with the network architecture [12]. Thus, there is strong obligation to device a logical approach to design the architecture of a given network, simultaneously, along with sample size determination. With this backdrop, we can clearly say that the surrogate building algorithm is governed by several parameters whose values are usually fixed based on some heuristic, thus inviting potential errors and credible variations in the predictability of the surrogates. Also, any extrapolation out of the n-

dimensional input space calls for re-construction of the surrogate model, which would require a significant amount of computational time. Thus, the surrogate building algorithm should be fast enough, apart from being parameter free to make the surrogate models universal and process of optimization online.

In this work, the effect of several parameters governing the ANFIS surrogate building procedure has been studied. The work presents a sound basis and justification for the need of a novel parameter free surrogate building algorithm especially focusing on the automated design of configuration of ANFIS along with the simultaneous determination of the sample size required for maximizing the prediction accuracy, without over-fitting the network. The individual effect of each of the parameters like architecture, sample size, sampling plan, etc. on the aspects of predictability and parsimonious behaviour of the surrogate model has been investigated. An industrially validated model for Poly Vinyl Acetate (PVAc) process is considered for all the sensitivity analysis and optimization studies. A comprehensive comparative study between the results obtained using several ANFIS surrogates obtained by varying the aforementioned parameters, is presented in details. This article contains the following sections: the Introductory section which is followed by the Formulation section comprising a detailed description of the PVAc model and the ANFIS sensitivity analysis. This is then followed by the Results and Discussions section before concluding the work in the Conclusion section. The last chapter of this article deals with the modelling and optimization of electrospinning process using ANNs as surrogates.

Chapter 2

Formulation

2.1 Industrial problem – Modelling and Optimization of PVAc model

2.1.1 PVAc kinetic model

Owing to the wide range of applicability of polymers at industrial scale, the major focus of research in this area has been over process modelling and optimization of long chain branched polymers. The kinetic mechanism for a batch free radical polymerization of vinyl acetate is as shown in Table 1 [1] where "n" represents the total chain length and $P_{b,n}$ and $D_{b,n}$ indicate the corresponding live and dead polymer chains having "b" long chain branches. It is assumed that the dead polymer chains with a double bond have a concentration equal to some known fraction of the total number of dead polymer chains. The net rate of production of live polymer and dead polymer chains are represented in Table 2 [1]. This classification, based on long chain branching (LCB), results in a set of very large number of ODE-IVPs, which must be solved in order to obtain the Molecular Weight Distribution (MWD).

Table 1: Radical chain reaction of PVAc polymerization

1.	Initiator Decomposition	$I \overset{k_d}{\rightarrow} 2PR^{\blacksquare}$
2.	Initiation	$PR^{\blacksquare} + M \overset{k_I}{\rightarrow} P_{0,1}$
3.	Propagation	$P_{b,n} + M \overset{k_p}{\rightarrow} P_{b,n+1}$

4. 5.	Chain transfer • To monomer • To solvent	$P_{b,n} + M \xrightarrow{k_{fm}} D_{b,n} + P_{0,1}$ $P_{b,n} + S \xrightarrow{k_{fs}} D_{b.n} + P_{0,1}$
6.	Reaction with terminal double bond	$P_{b,n} + D_{r,m} \xrightarrow{k_{db}} P_{b+r+1,n+m}$
7.	Chain transfer to polymer	$P_{b,n} + D_{r,m} \xrightarrow{k_{fp}} D_{b.n} + P_{r+1,m}$
8. 9.	Termination by • Combination • Radical disproportionation	$P_{b,n} + P_{r,m} \xrightarrow{k_{tc}} D_{b+r,n+m}$ $P_{b,n} + P_{r,m} \xrightarrow{k_{tb}} D_{b,n} + D_{r,m}$

Table 2: Rate of production of Live and Dead Polymer chains

Live polymer Chain	$\frac{dP_n}{dt} = \{k_I[R^\bullet][M] + (k_{fm}[M] + k_{fs}[S]) \sum_{n=1}^{\infty} P(n)\}\delta(n-1) + k_p[P(n-1) - P(n)][M]$ $-(k_{fm}[M] + k_{fs}[S])P(n) + k_{fp}[nD(n)]\left(\sum_{n=1}^{\infty} P(n)\right) - k_{fp}P(n)\sum_{n=2}^{\infty} nD(n)$ $-k_{tc}P(n)\sum_{n=1}^{\infty} P(n) - k_{td}P(n)\sum_{n=1}^{\infty} P(n)$
Dead Polymer Chain	$\frac{dD_n}{dt} = (k_{fm}[M] + k_{fs}[S])P(n) + k_{td}P(n)\sum_{n=1}^{\infty} P(n) + \frac{1}{2}k_{tc}\sum_{y=1}^{n-1} P(n)P(n-y)$ $+k_{fp}P(n)\sum_{n=2}^{\infty} nD(n) - k_{fp}nD(n)\sum_{n=1}^{\infty} P(n)$

A moment based modeling was taken up for each class of branched polymer where the 0th, 1st and 2nd ordered moments for live and dead polymers of each class were derived as shown in Table 3.This avoids the complication involved in solving such a large number of ODE-

IVPs. The number of classes for constructing the moments was carefully chosen ensuring that the summation of first moments of each class remains equivalent to the overall first moment of polymer. This has ensured the complete construction of overall MWD from MWD of individual moments, but resulted in a large number of classes since the polymer was highly branched. The resulting large number (e.g. 285) of highly nonlinear ODE-IVPs demands relatively larger time for simulating a single run using the differential algebraic equation (DAE) solver LIMEX. This large time of execution greatly reduces the scope of online optimization of PVAc reaction process with LCB in real world applications.

The Eqs. (1)–(4) define the polymer properties such as number average molecular weight (M_n), weight average molecular weight (M_w), polydispersity index (PDI) and branching index (B_n), respectively, in terms of moments obtained by solving the aforementioned set of ODEs. With these equations in place, if the batch monomer concentration (M), amount of initiator (I) and temperature of the isothermal batch polymerization process (T) are given as inputs, the model can then be solved to obtain M_n, M_w, B_n and PDI as outputs after a batch polymerization time of t_{poly} [1].

Table 3: 0^{th}, 1^{st} and 2^{nd} Moments for Live and Dead polymers

Moments for Live Polymer Chains $\mu_j = \sum_{x=1}^{n} x^j P(x)$	$\frac{d\mu_0}{dt} = 2fk_d[I] - (k_{tc} + k_{td})\mu_0^2$ $\frac{d\mu_1}{dt} = 2fk_d[I] + k_p[M]\mu_0 - (k_{fm}[M] + k_{fs}[S])\mu_1 - (k_{tc} + k_{td})\mu_0\mu_1 + k_{fp}(\mu_0\nu_2 - \mu_1\nu_1)$ $\frac{d\mu_2}{dt} = 2fk_d[I] + 2k_p[M]\mu_1 - (k_{fm}[M] + k_{fs}[S])\mu_2 - (k_{tc} + k_{td})\mu_0\mu_2 + k_{fp}(\mu_0\nu_3 - \mu_2\nu_1)$

Moments for Dead polymer Chains $\nu_j = \sum_{x=1}^{n} x^j D(x)$	$\frac{d\nu_0}{dt} = (k_{fm}[M] + k_{fs}[S])\mu_0 + (\frac{1}{2}k_{tc} + k_{td})\mu_0^2$ $\frac{d\nu_1}{dt} = (k_{fm}[M] + k_{fs}[S])\mu_1 + (k_{tc} + k_{td})\mu_0\mu_1 - k_{fp}(\mu_0\nu_2 - \mu_1\nu_1)$ $\frac{d\nu_2}{dt} = (k_{fm}[M] + k_{fs}[S])\mu_2 + (k_{tc} + k_{td})\mu_0\mu_2 + k_{tc}\mu_1^2 - k_{fp}(\mu_0\nu_3 - \mu_2\nu_1)$

$$M_n = \left(\frac{\mu_1}{\mu_0}\right) MW_{VA} \quad \ldots (1)$$

$$M_w = \left(\frac{\mu_2}{\mu_1}\right) MW_{VA} \ldots (2)$$

$$PDI = \left(\frac{M_w}{M_n}\right) \quad \ldots (3)$$

$$B_n = \sum_{n=1}^{\infty} \sum_{b=0}^{N_c} \left(\frac{bN_{n,b}}{\mu_0}\right) \ldots (4)$$

2.1.2 Optimization Problem Formulation

The current industrial scenario demands the following three objectives which are conflicting in nature –

i. Maximize the production of high molecular weight branched /cross linked polymer (M_w).

ii. Minimize the time of operation (t_{poly}).

iii. Maximize the branching index (B_n).

The polymerization needs to be run for a longer time in order to achieve high molecular weight polymers. Therefore maximizing M_w and minimizing t_{poly} simultaneously leads to a conflicting scenario. The third objective of maximizing the B_n is a direct result of the current necessity in polymer industry. Thus a multi - objective optimization problem (MOOP-1) is formulated within this framework where the conflicting objectives are to maximize the M_W, minimize the t_{poly} and maximize the B_n. The optimization problem presented in the article is formulated to fulfill this desire and is shown in Table 4.

Table 4 : Optimization Formulation

Objective functions	Constraints	Decision Variables
Maximize M_w	$M_w \geq (M_w)^{min}$	10 mol/lit $\leq M \leq$ 14 mol/lit;
Maximize B_n	$PDI \leq (PDI)^{max}$	0.00003 mol/lit $\leq I \leq$ 0.00015 mol/lit;
Minimize t_{poly}	$Conv \leq 0.97\{ Conv_{gel}(T) \}$	$333K \leq T \leq 352K$

where gel conversion is given by the correlation, $Conv_{gel}(T) = 1.47\times10^{-3}\times T+0.32$

The relationship between the process parameters (*M*, *I* and *T*) and the objectives pertaining to M_w, B_n and t_{poly} is provided by the first principle model described above. The process parameters form the decision variables for the optimization problem which aims at finding the optimum values of the same which would satisfy the objective functions. The upper and lower bounds on the decision variables *M*, *I* and *T* are obtained from the experimental study so that the search space remains feasible as well as practical [1]. The phenomenological constraints are in place for the molecular weight and polydispersity index. The constraint owing to gel formation is obtained by constructing an empirical relation from the experimental data of monomer con- versions at various temperatures [1]. In order to avoid the gel formation effect, the simulation was allowed to run upto a conversion level (expressed in %), which is 3% less than the gel conversion point (expressed in %). Population based real coded NSGA II framework [3] (number of population = 70, generations = 40, crossover probability = 0.9 and mutation probability = 0.1) has been used as optimization algorithm to solve this multi-objective optimization (MOOP) due to the advantage non-dominated sorting provides in establishing the wide spread PO front as compared to its counter parts in classical optimization. Since the reaction network is highly complicated, it usually consumes an overwhelming amount of time for obtaining a single input–output sample data as pointed out earlier. In order to optimize the

process, the optimizer is in need of such function evaluations to be carried out many times. Thus the target was to replace the original physics based model with a fast and accurate Sobol set based ANFIS model in optimization algorithm and study the advantages and disadvantages that surrogate based optimization gives in terms of computational time and accuracy.

2.2 Surrogate building algorithm: ANFIS

2.2.1 Practical Considerations in ANFIS

The sheer experience of individuals is always susceptible to biased information and other practical errors which might result in sub-optimal solutions. Thus, fine tuning of membership functions is required when the given input-output data set is large enough. However, if the data set is too small then it is not worthwhile to further tune the MFs since we might lose any important information about the system. In such situations the membership functions must be kept fixed throughout the learning process. In a conventional fuzzy inference system, the number of rules is decided by an expert who is familiar with the system to be modelled. Each fuzzy rule can be viewed as a local description of the system under consideration. However, when there is no prior knowledge available which is usually the practical case, the number of rules is determined by examining the desired input-output data such that it can be able to capture the non - linearity present in the system. On a similar basis, the order of Sugeno fuzzy rules is considered as zero, first or any other higher order. The number of membership functions assigned to each input variable is chosen empirically based on the given data set. This scenario is much similar to that of ANNs where the minimum number of hidden layers is not known in advance for achieving a desired performance. The type of MF and the fuzzy T-norm operator as described in the introductory section has always been determined by a trial and error method until some desired accuracy is found. In this current work, the author proposes an elaborate study on the effect of architecture design on the

predictability of the network and provides an appropriate justification for the need of an optimal design of the ANFIS architecture which would enable the network to predict results with maximum accuracy.

2.2.2 Accuracy of Prediction

Prior to modelling, the accuracy of the surrogate model which is a necessary parameter, needs to be specified by the decision maker. It is obvious that any decision maker would like to have a maximum value of accuracy for the surrogate model, which may come at the cost of large computational time and large number of sample points for training. With the dubious nature of this issue, the decision maker without any specific prior experience in the domain of surrogate modelling, would hesitate to provide a particular value of accuracy. This may not allow the algorithm to build a surrogate model capable of maximum predictability. Thus, there is a need to ensure that without providing a specific value of accuracy as an input to the algorithm, it must be able to build a surrogate model having maximum predictability. Accuracy of the predicted output of the network can be estimated using two well-known statistical measures [10]:

i) Root mean square error: RMSE

ii) Correlation coefficient R^2

$$R^2 = \left(\frac{\mathrm{cov}\,(y,\hat{y})}{\sqrt{\mathrm{var}(y)\mathrm{var}(\hat{y})}}\right)^2$$

$$\mathrm{cov}(y,\hat{y}) = n_t \sum_{i=0}^{n_t} y^{(i)}\hat{y}^{(i)} - \sum_{i=0}^{n_t} \hat{y}^{(i)} \sum_{i=0}^{n_t} y^{(i)}$$

$$\mathrm{var}(y) = n_t \sum_{i=0}^{n_t} y^{(i)2} - \left(\sum_{i=0}^{n_t} y^{(i)}\right)^2$$

where y is the original output coming from physics driven model or data and $\hat{y}$ is the predicted output from the surrogate model.

2.2.3 Sampling plan or Design of Experiments (DOE)

The sampling plan is an important aspect of the surrogate building algorithm as it directly influences the number of sample points, accuracy of prediction and architecture of the network. The sampling plan can be easily interpreted as a scheme of placing some arbitrary probes in an m-dimensional space to capture the behaviour of the model (m being the number of inputs). An ideal case would be to divide the entire space into grids and place a probe at every junction which leads to the full factorial sampling plan [17]. This will ensure maximum accuracy based on the precision of the grid size, but the number of probes required will be extremely large making it an impractical proposition. However, the ability to capture the dynamics of the system at every cross joint would certainly make the sampling plan uniform and such a sampling plan is thus said to have the feature of space-filling [10]. The characteristic trait of any efficient sampling plan should be able to probe the dynamics of the entire m-dimensional input space with least possible function evaluations or in other terms least possible sample points. Several sampling plans exist in literature displaying the feature of space filling, but none of them reports of performing the task in least possible number of function evaluations. One such example is Latin Hyper-cube Sampling technique (LHS) [10, 15], which would ensure the space filling nature of the system but when prompted for an additional sample point, would generate a set of points, completely different from previous set constituting the sampling plan. This essentially abandons the previously collected sample points and calls for several new function evaluations. Sobol sampling plan [7], based on highly convergent Sobol sequence, is one sampling plan, which ensures both space filling attribute and maintains the sequence even if prompted for a new sample point. The projection of the distribution of 200 sample points in 3-dimensional space obtained using the Sobol sampling plan is compared with the distribution of those obtained using LHS sampling plan and is presented in Fig 2. One can easily decipher qualitatively the enhanced uniformity and space filling nature of Sobol points

over the LHS points. A metric, called the φ (PHI) metric, proposed in literature [10, 21] of sampling techniques, measures the space filling attribute of any given sampling plan. Lower the value of this φ metric, better the space filling ability of the sampling plan. The space filling nature of both LHS and Sobol sampling plans are measured using this PHI metric for the same distribution of 200 points as given in Fig. 2 and the results are presented in Table 1. It is evident from Table 1 that Sobol sampling plan emerges out to be one of the best alternatives among the existing options. Thus, Sobol sampling plan is selected in this work for implementation in the surrogate building algorithm.

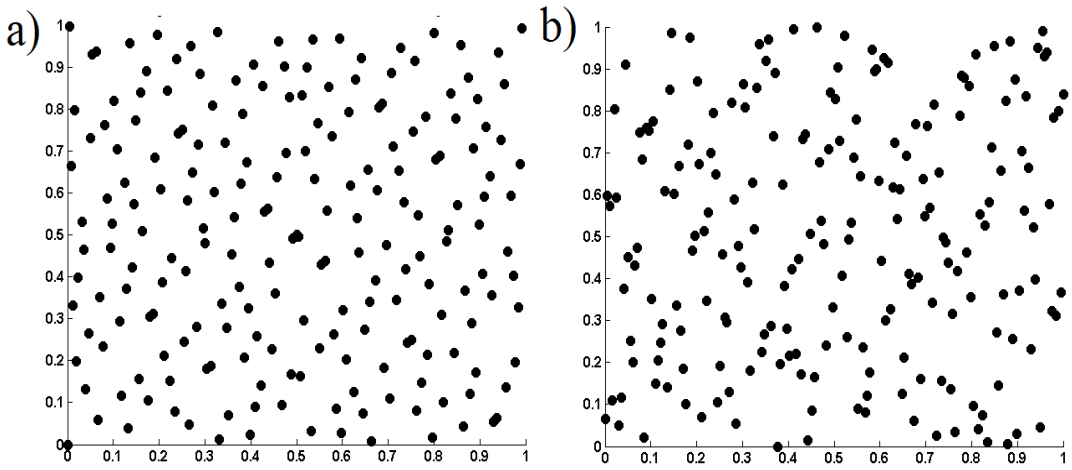


Fig. 2: The distribution of 200 sample points using the a) Sobol sampling plan and b) LHS sampling plan.

Table 5: Comparison of different sampling plans in terms of PHI metric and computational time for 200 sample points.

Sampling Plan	**PHI metric measure**	**Computational time**
LHS-200	205.367	588.69 seconds
Sobol-200	201.939	0.0251 seconds

2.2.4 Sample size

From the study of literature, it can be said that no proper rationale is devised to decide the number of data points required for training [19]. In most of the cases, the general rule of thumb of considering 70% of the available data for training is applied. Such kind of heuristic based assumption may cause the network to be either over-fitted or under trained because of the unavailability of any exact measure of the number of sample points required for training. One significant contribution in literature [22] showcases a novel algorithm for sample size determination of the given network. Their approach is based on the fact that the training error of the network is minimized by increasing the sample size. Although this is true, but the fact that the network might get over-fitted as the sample size is increased cannot be ruled out. Thus, in order to ensure the parsimonious nature of the network, they incorporated the K-fold model evaluation technique [25, 26] (with K = 10) along with a variant of LHS called the incremental-LHS (i-LHS) sampling plan for sample size determination. Their algorithm starts with an initial guess value of the sample size, for a given architecture, which they proposed to consider 10 times the number of dimensions in the model. The sample size is given to the i-LHS sampling algorithm, which then generates the training set and it is then divided equally into K-groups or folds. Out of the K available folds, one group is selected for validation and the remaining groups are used for training the network. A validation error is obtained, which is defined as the maximum of the absolute values of the deviations between original output and fitted quantities. The fold for validation can be considered in K different ways thereby resulting in K number of validation errors. A mean of those errors is thus considered and is denoted as the cross validation error of the current model (models are differentiated by the sample sizes). Then the sample size is incremented by a user defined value (say plus 10) and the entire procedure is repeated for this new model. A quantity is then evaluated for each iteration which is defined as the ratio of the differences of the cross validation errors of two consecutive iterations with the

difference in their corresponding sample size. This ratio is divided by the maximum value of such ratios found till the current iteration to obtain the measure called slope ratio percentage (SRP). If this SRP is less than some tolerance value which is again user specified (say 0.01), the algorithm is terminated and the current sample size is fixed as the final sample size. The essence of their algorithm in brief is to find a minima of cross validation error metric, which is a function of the sample size. One of the major drawbacks of this algorithm is the large computational time of K-fold based validation method. Another disadvantage is the extensive number of function evaluations deliberately called by the i-LHS sampling plan, as described previously.

Hence, the real challenge in this study is to build ANFIS surrogate in place of the phenomenological model for this complex polymerization process along with simultaneous determination of sample size, sampling plan to predict maximum accuracy and carry out the optimization exercise using the surrogate model.

2.3 ANFIS: The Algorithm and its functioning

A Matlab© source code has been developed for successful implementation and functioning of the ANFIS. In order to test the scope and applicability of any type of network, the code developed was a generic code which can practically take the following as inputs

1. Any architecture in the form of a row vector where the entry in each column would correspond to the following:
    a. First column - Number of MFs assigned per input (m); varies from 1 to 4.
    b. Second column – Number of fuzzy rules; varies from m to m^n where n indicates the dimensionality of the system.
    c. Third column – Type of MF; Gaussian, Bell or Triangular MF.
    d. Fourth column – Fuzzy T-norm operator; Product or Minimum.
    e. Fifth column – Order of FIS; first or second.

The code accepts the numerical values of 1, 2 or 3 depending on the choice of architecture to be implemented in the network (Table 6).

Table 6. Choice of variables for ANFIS architecture

Variable name	**Choice**
Fuzzy operator	1 – Product, 2 - Minimum
Order of FIS	1 – First order, 2 - Second order
Type of MF	1 – Triangular MF 2 – Gaussian MF 3 – Bell shaped MF

2. The data set required for training and validation needs to be sent into the code to ensure proper training and validation. The code can accept any number of training and validation sample points.

The outputs from the code are listed below:

1. Original outputs and ANFIS predictions - The predicted values of the outputs corresponding to the inputs in the validation set are sent as outputs of the ANFIS code along with the original outputs of the model which were sent as validation set.
2. Root Mean Square Error (RMSE)
3. R^2
4. Premise and consequent parameters of the trained neuro fuzzy network which will enable it to interpolate any new value.

The working of the code, as per the sequential flow of the steps, is described further in the article.

1. Normalization of the training data: The training data needs to be normalized before it is utilized for training the given network. This ensures scaling of all the input data in the range [0, 1], which is similar to the range of membership function values.
2. Declaration and initialization of premise parameters: The number of premise and consequent parameters keep changing based on the architecture of the network. Since there is no human expertise available, the premise parameters are declared and initialized, to a suitable value, once the architecture is sent to the code as input. These initial values must satisfy ε - completeness with $\varepsilon = 0.3$, which means that given a value of input 'x' in the operating range, there is always a membership value $\mu(x) \geq \varepsilon$. This constraint enables sufficient overlapping and smooth transition from one fuzzy subset/linguistic variable to another.
3. Network Training and Validation: Evolutionary Genetic Algorithm (GA) [4] is adopted as a learning rule for updating the premise and consequent parameters of ANFIS. After training the network using GA, the network predictions are validated using a set of 200 sample points obtained using the LHS sampling plan. The corresponding RMSE and R^2 values are sent as outputs.

Chapter 3

Results and Discussions

The data for surrogate building was procured from a physics based model capable of predicting the overall MWD of Polyvinyl Acetate (PVAc) [1] which was developed using a validated reaction network describing the kinetics of PVAc. The MOOP formulation presented in Table 3 is solved using the real and binary coded NSGA II algorithm whose credentials are given in Table 7. Although the NSGA II algorithm was run for 100 generations, it was observed that the PO front was saturated at generation number 40 with each generation containing 70 populations. Thus, the total function evaluations required to perform the optimization run with original model in place, were nearly 2800. The results of the current work are reported below in the sequence of the simulations conducted. The sampling plan is fixed to Sobol for the reasons mentioned previously.

Table 7: NSGA II parameters for solving the MOOP problem of PVAC system

Parameters	**Values**
Population Size	70

Number of Generations	100
Crossover probability	0.9
Mutation probability	0.01

A. Effect of variation of architecture and exploring multi-layered networks

The architecture was varied along with variations in sample size and the surrogate ANFIS models thus built are reported in Table 8. For each of the architectures considered for investigation, the sample sizes were also varied within a range of 30 to 250. Several possible architectures (nearly 200) were investigated over a long period of time and the potential results which could serve with better accuracy are only reported in Table 8. N indicates the total number of premise and consequent parameters n indicates the sample size. As mentioned earlier, more number of fuzzy rules (second column) and higher order FIS (fourth column) enable to capture the nonlinearity of a system. Clearly, from Table 8, one can observe that this is not always true since higher number of parameters may have led to over fitting of the model thereby reducing its prediction accuracy R^2. When compared with the architecture [3-2-2-2-2] ($R^2 = 0.810$), the architecture [3-1-2-1-1-2] is predicting the results far better ($R^2 = 0.972$) even with less sample size. Therefore, in the absence of expert data, we need to find the optimum number of rules and order of the system from the given dataset, which can predict maximum accurate results without overfitting the network.

On varying the type of membership function, we can see that Gaussian or bell shaped MFs predict better results than linear or triangular MFs, which implies that the system is nonlinear. Regarding operator, no particular fuzzy T-norm operator evolves as a clear winner thereby suggesting that both the minimum and product operator should be explored prior to training the networks. Although this study corresponds to only output 1, a simultaneous study

was also performed for outputs 2, 3 and 4. Thus, from these results, a clear rationale is observed for devising a logical approach for optimal design of the architecture of ANFIS networks.

Table 8: Effect of Architectures on network predictability for output 1

Number of MFs/input	**No. of rules choice**	**Choice of Fuzzy operator**	**Order of FIS**	**Choice of Type of MF**	**R^2**	**N**	**n**
3	2	2	2	2	0.810	288	72
2	2	2	1	1	0.889	50	108
1	2	2	2	1	0.902	19	189
4	1	1	2	2	0.922	64	90
2	1	1	2	2	0.936	32	54
3	1	2	1	3	0.954	39	177
4	1	1	2	3	0.968	76	159
3	1	2	2	3	0.965	57	93
3	1	2	1	2	0.972	30	57
3	1	2	2	2	0.969	48	66
2	1	2	1	3	0.973	26	57
1	1	2	2	2	0.975	16	72
2	1	1	1	3	0.964	26	69
1	2	1	2	3	0.988	19	72
1	1	1	1	3	0.984	13	105
1	1	2	1	2	0.963	10	42

2	1	1	1	2	0.983	20	72
3	1	1	2	2	0.985	48	99
1	2	2	2	3	0.988	19	84
1	2	2	1	2	0.990	10	63
1	2	1	2	2	0.995	16	45
1	2	2	2	2	0.997	16	93
2	1	2	2	2	0.998	32	90

B. Effect of Variation of Sample Size on predictability of an ANFIS.

In order to study the effect of sample size on the prediction accuracies of the networks, one of the architectures with maximum prediction accuracy was selected from Table 8 and its predictability was studied by varying the sample size. The results of this study are reported in Table 9. Here, one can observe that initially, as the sample size increases, the prediction accuracy of the architecture also increases. But after sample size 90, the improvement slows down and reaches saturation, and after sample size 102, it starts decreasing. The overfitting of the network might be a reason for this anomaly. Since the R^2 measured is with respect to the validation set, the validation error decreased as the sample size for training increased till the network got over-fitted. Thereafter, the validation error again increased indicating that the network is over-fitted to the training data. Thus, the sample size for training cannot be arbitrarily given to the network but a quantitative measure should be devised to evaluate systematically the sample size required for allowing a given architecture to predict till maximum accuracy possible without over-fitting the network.

Table 9: Effect of Sample size on network predictability for output 1

Number of MFs/input	**No. of rules choice**	**Choice of Fuzzy operator**	**Order of FIS**	**Choice of type of MF**	**R^2**	**N**	**n**
2	1	2	2	2	0.895	32	36
2	1	2	2	2	0.933	32	42
2	1	2	2	2	0.964	32	48
2	1	2	2	2	0.940	32	54
2	1	2	2	2	0.956	32	60
2	1	2	2	2	0.969	32	66
2	1	2	2	2	0.975	32	72
2	1	2	2	2	0.984	32	78
2	1	2	2	2	0.992	32	84
2	1	2	2	2	0.998	32	90
2	1	2	2	2	0.9978	32	96
2	1	2	2	2	0.9976	32	102
2	1	2	2	2	0.983	32	108
2	1	2	2	2	0.942	32	114

C. Process optimization using ANFIS surrogates

These *results justify the need for a parameter free ANFIS surrogate building algorithm which can intelligently devise the architecture i.e., number of rules, membership function, fuzzy operator and order of FIS, along with simultaneous determination of sample size such that, the*

network predicts with maximum accuracy without being over-fitted. However, with the help of the laborious hit and trial routine, four architectures with appropriate sample size were selected for emulating the outputs of the PVAc model. These surrogates with their credentials are presented in Table 10.

Table 10: ANFIS surrogates for PVAc model

Output	**Architecture** (Input partitions – Rules – Operator - Order - MF)	**N**	**R^2**	**Sample size**	**Total function calls**
M_w	[2-2-2-2-2]	32	0.998	90	120 + 200 (training + validation set) = **320**
B_n	[1-2-2-2-3]	19	0.998	60	
t_{poly}	[3-2-1-2-2]	48	0.995	120	
PDI	[2-2-1-2-2]	32	0.9964	80	

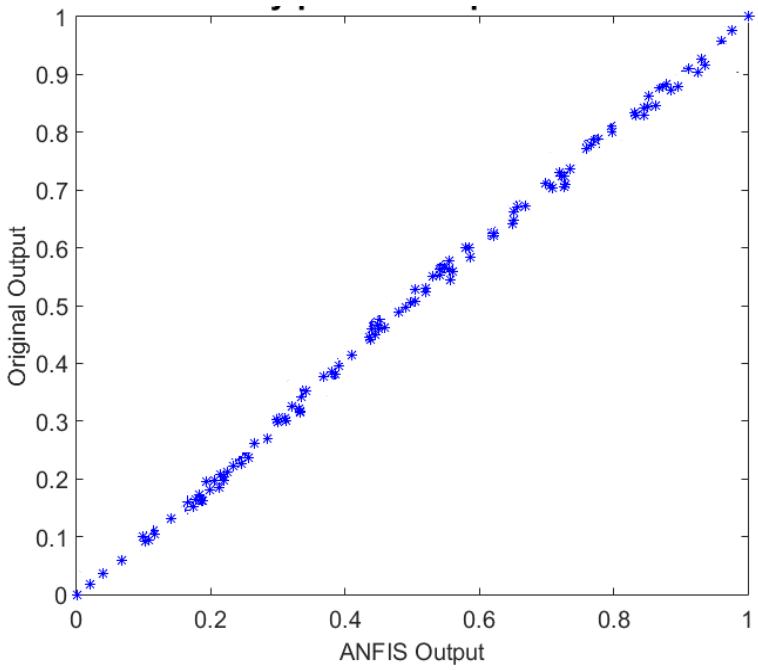


Fig. 3: Parity plot for Output 1 using the architecture = 2-2-2-2-2 with R2 = 0.998 obtained using K-fold sampling technique

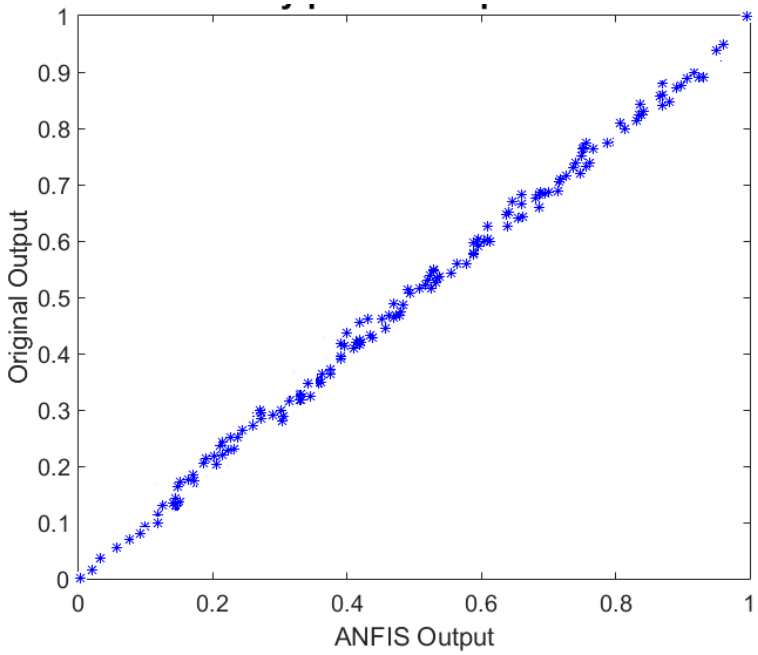


Fig. 4: Parity plot for Output 2 using the architecture = 1-2-2-2-3 with R2 = 0.9977 obtained using K-fold based technique

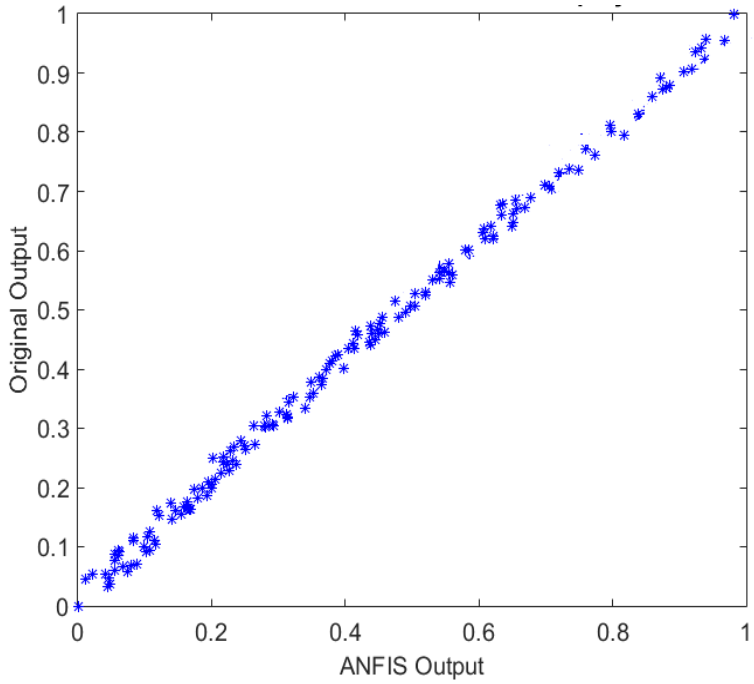


Fig. 5: Parity plot for Output 3 using the architecture = 3-2-1-2-2 with R2 = 0.995 obtained using K-fold based technique

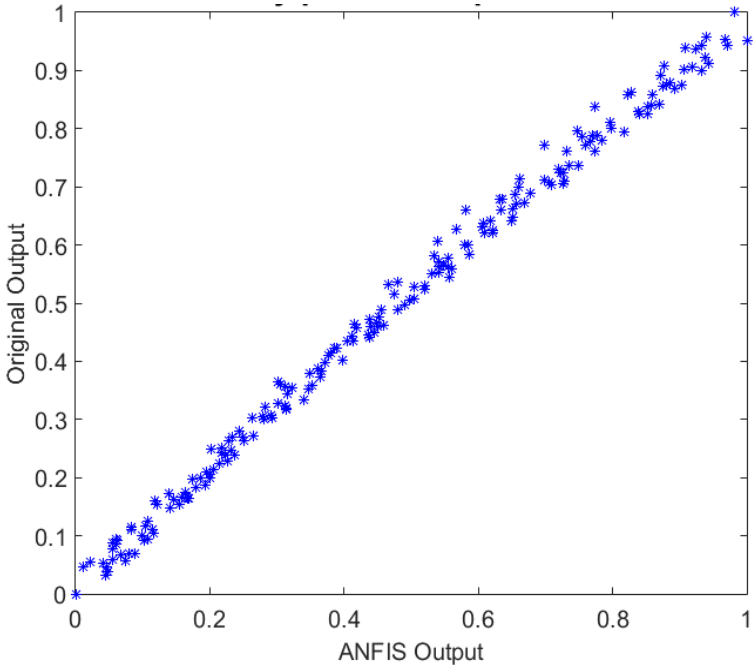


Fig. 6: Parity plot for Output 4 using the architecture = 1-2-2-2-3 with R2 = 0.9964 obtained using K-fold based technique

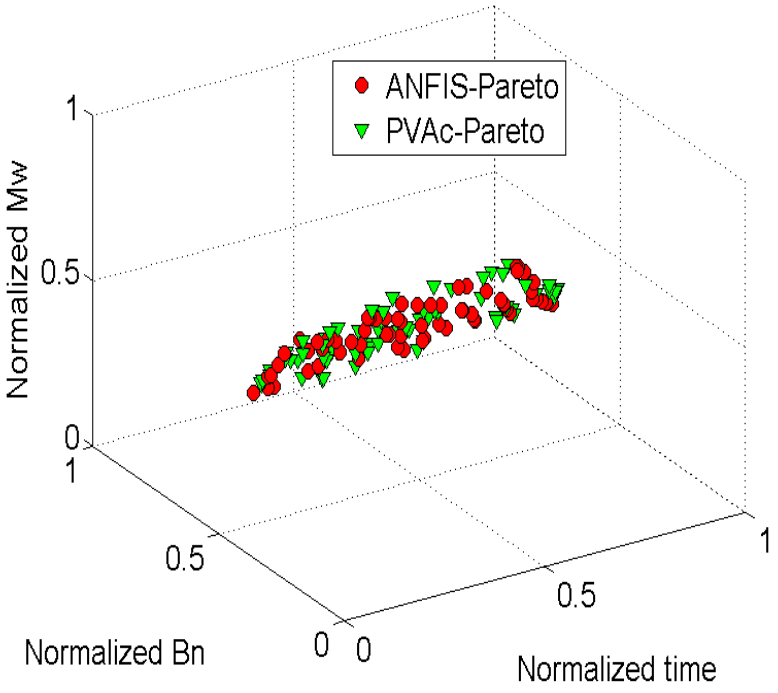


Fig. 7: PO front comparison of optimization using ANFIS surrogate built by K-fold based sampling method and original first principle PVAc model

The corresponding parity plots are presented in Figs. 4, 5, 6 and 7. The ANFIS surrogate based optimization has reduced the number of function evaluations of the first principle model by nearly 9 (2800/320) times when compared with conventional optimization approach. The ANFIS surrogate models obtained for all the outputs are then allowed to replace the original PVAc model in the conventional optimization algorithm. The NSGA II simulation runs were completed in no time and the final Pareto Optimal front comparisons are shown in Fig 8. For the sake of obtaining a clear cut qualitative estimation of the result observed in Fig 8, the inputs of the PO points obtained using the ANFIS surrogate based optimization are sent to the original polymerization model and the corresponding outputs are compared. The ANFIS surrogates were able to predict the global Pareto optimal front with 97% accuracy. This has enabled online optimization of the PVAc-LCB complex model. All the simulations were carried out in Intel(R) Xeon(R) CPU E5-2690 0 @ 2.90GHz (2 processors) 128 GB RAM machine.

D. Knowledge Discovery

In this case study, since there was no availability of expert knowledge, the author has extracted the knowledge (fuzzy rule set) from the given input-output data. This aspect can be justified by considering an ANFIS architecture consisting of 8 rules for the 3-dimensional PVAc system. Gaussian membership functions are considered with product T-norm operator for a first order FIS.

Fuzzy Rule set

If M is LOW, I is LOW and T is LOW then f(1) = p1x + q1y + r1z + s1

If M is LOW, I is LOW and T is HIGH then f(2) = p2x + q2y + r2z + s2

If M is LOW, I is HIGH and T is LOW then f(3) = p3x + q3y + r3z + s3

If M is LOW, I is HIGH and T is HIGH then f(4) = p4x + q4y + r4z + s4

If M is HIGH, I is LOW and T is LOW then f(5) = p5x + q5y + r5z + s5

If M is HIGH, I is LOW and T is HIGH then f(6) = p6x + q6y + r6z + s6

If M is HIGH, I is HIGH and T is LOW then f(7) = p7x + q7y + r7z + s7

If M is HIGH, I is HIGH and T is HIGH then f(8) = p8x + q8y + r8z + s8

The premise parameters of the Gaussian membership functions, which were updated using the genetic algorithm [1], form the two fuzzy subsets - LOW and HIGH for each of the 3 input variables. The outputs of the all the fuzzy rules are represented as a first order function of the inputs x, y and z and the coefficients of the function (p, q, r and s), which are the consequent parameters, are obtained using the learning algorithm. The overall output (f) is the summation of the individual outputs of each fuzzy rule: f = $\sum_{i=1}^{8} f(i)$.

Let us consider that an industry desires to have a polymer of molecular weight $3.2694*10^6$ g/mol (normalized M_w = 0.9815), B_n = 1.2217 (normalized B_n = 0.0922) in time $7.204*10^4$ sec (normalized t_{poly} = 0.3320). In order to obtain the desired output, the decision variables M, I, T that need to be considered are obtained as shown below:

(p1+p2+….p8) M + (q1+q2+….q8) I + (r1+r2+…r8) T + (s1+s2+….s8) = 0.9815

(p1+p2+….p8) M + (q1+q2+….q8) I + (r1+r2+…r8) T + (s1+s2+….s8) = 0.0922

(p1+p2+….p8) M + (q1+q2+….q8) I + (r1+r2+…r8) T + (s1+s2+….s8) = 0.3320

On substituting the values of p, q, r and s for each of the outputs, we obtain the following set of equations:

(0.443) M + (0.557) I + (0.648) T + (0.795) = 0.9815

(0.126) M + (0.243) I + (0.114) T + (0.024) = 0.0922

(0.219) M + (0.321) I + (0.532) T + (0.227) = 0.3320

On solving these three equations, the values of M, I and T are obtained and thus the eight rules can be extracted based on their respective LOW and HIGH membership function values. Depending on the degree of fuzziness/MF values, the decision maker can choose any one rule from the extracted rules such as: *"M high, I low and T low"*. This implies that, even-though the expert rule base was not available initially, the ANFIS surrogate model was able to extract knowledge from the given input-output dataset.

Chapter 4

Electrospinning Process

4.1 Modelling and Optimization formulation

Electrospinning process is a simple, versatile and widely used method for the production of nanofibers at a large scale. A large of polymers have been electro spun into nano fibres. These fibres have been successfully applied in various fields, such as, nano catalysis, tissue engineering scaffolds, protective clothing, filtration, biomedical, pharmaceutical, optical electronics, health-care, biotechnology, defence and security, environmental engineering and so on due to their higher surface area and smaller pore size as compared to the regular fibres [28]. Overall, this is a relatively robust and simple technique to produce nanofibers from a wide variety of polymers.

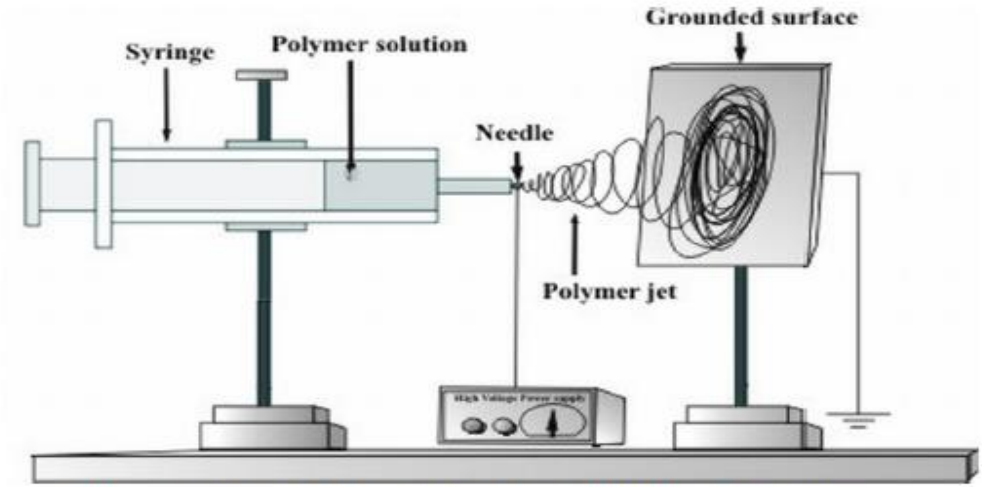


Fig. 8: Electrospinning setup

Unlike PVAc system, electrospinning process does not have a standard first principle based model. It is necessary to build a mathematical model for this process as it not only helps in understanding the effects of process parameters, but also reduces the burden of conducting time consuming experiments to estimate the features of the product under new set of conditions. Additionally, this would help to predict the results under a new combination of parameters and it is also beneficial for the control and optimization of process parameters. The governing parameters of the electrospinning process, which are effective on the macroscopic nanofiber properties are volumetric flow rate, polymer concentration, the applied voltage, syringe diameter and the distance between the collector and tip of the syringe [28]. Here, the modelling could have been performed in two ways: physics based modelling or data based modelling. Since the physical process of electrospinning involves phenomena of multi-physics with phase changes, building a physics based model could be extremely challenging since each of these physics driven sub-processes could have their individual modelling challenges. So, it has been decided to take the root of data based modeling and the real challenge in this study is to build a data based / surrogate model for this process. The author has considered the above mentioned process parameters as inputs to the system and the diameter of the fiber and its corresponding features are obtained as outputs from the system for modelling (Table 11).

Table 11: Inputs and Outputs required for modelling electrospinning process

Inputs	**Range of Inputs (Minimum:gap:Maximum)**	**Outputs**
Syringe diameter(mm)	18:2:26	Diameter of the fiber
Voltage (KV)	8:1:15	Length of fiber - Short / Long
Distance from collector (cm)	5:1:15	
Flowrate of polymer (microliter/min)	2:2:10	

Concentration of polymer (wt %)	3:1:8	Shape of the fiber - Bead / fiber / Beaded fiber

Experimentation was performed over the entire range of input variables in order to obtain the training and validation sets required for modelling. Polyacrylonitrile (PAN) was used as the polymer solution. After building surrogate model with maximum accuracy, the next task is to find the optimum diameter of the fiber. This might be a single objective optimization problem but the main challenge lies in building the equivalent surrogate model for the process.

4.2 Surrogate building algorithm - Artificial Neural Networks

4.2.1. Introduction

ANNs are mathematical models, which try to mimic the functioning of biological neural network of human brain. A biological neuron and its mathematical counterpart, called the node, are described in Fig. 9. They are widely acknowledged for their immense applications in pattern recognition problems, image processing and many other chemical engineering applications.

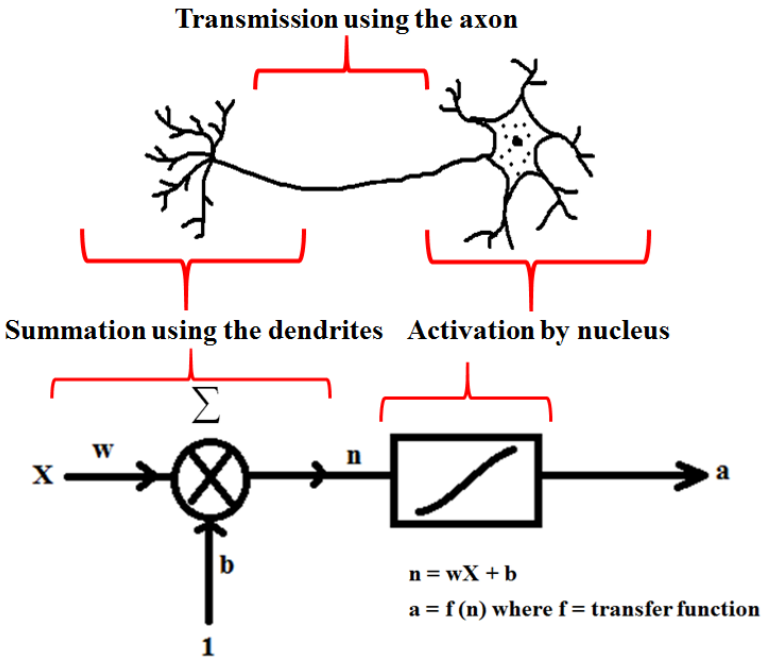


Fig. 9. Basic Structure of a neuron and node

The number of nodes in a single layer and the number of layers in the network together constitute the architecture of the network. One of the flaws with implementation of the ANNs is the inability to optimally design the architecture of the network. The architecture of the network is obtained based on the method of hit and trial, which often leads to an impasse.

One rule of thumb in this heuristic based design, applied widely in order to reduce the complexity of aforementioned hit and trial procedure, is the assumption that for any given data, a single hidden layer with some arbitrary number of nodes would be sufficient to predict any model with reasonable accuracy [14]. The potential of ANNs lies within their ability to segregate the data into exclusive regions. This can be visualised geometrically by considering one layer as an m-dimensional hyper-plane trying to separate out the existing data into two sub spaces, where m is the number of inputs feeding to that layer. A multi-layer perceptron network may, therefore, provide more accuracy for an unseen data, which might be linearly inseparable [8]. This rationale justifies for the fact that the aforementioned assumption may not be true in all cases. As mentioned in the previous chapters, the sample size required for training also effects the predictability of the network significantly in accordance with the network architecture [8]. Some of the prominent contributions in the literature are mixed integer nonlinear programing (MINLP) approach [13], the Akaike Information Criteria (AIC) [15], etc. to come up with the optimal design of the architecture. However, apart from being computationally expensive, none of them addressed the problem of simultaneous design of architecture and sample size determination.

4.2.2 Artificial Neural Network: The Algorithm

A Matlab© source code has been developed for successful implementation and functioning of the ANNs. In order to test the scope and applicability of the multi-layered perceptron networks, the code developed was a generic code which can practically take the following as inputs

1. Any architecture in the form of a row vector where the entry in first column would correspond to number of inputs, the entry in last column would correspond to number of outputs while the number of entries in between first and last column would determine the number of hidden layers. The values in these in between entries will determine the number of nodes in the hidden layer.

2. A numerical value for determining the transfer function. The output layers were all activated by the linear transfer function, but the activation of the hidden layers needs to be specified prior to the design of neural networks. Thus the code accepts the numerical value of 1 for implementing the tan sigmoidal activation function while the numerical value 2 would trigger the implementation of log sigmoidal activation.

3. The data set required for training and validation needs to be sent in to the code to ensure proper training and validation. The code can accept any number of training and validation sample points.

The outputs from the code are listed below:

1. Original outputs and ANN predictions. The predicted values of the outputs corresponding to the inputs in the validation set are sent as outputs of the ANN code along with the original outputs of the model which were sent in as validation set.
2. RMSE
3. R^2
4. Weights of the trained neural network which will enable it to interpolate any new value.

As the number of nodes in the neural network increases, its ability to interpolate with accuracy also increases. Evidently, the increase in number of parameters help in capturing the nonlinear behavior of the system. However, this increase in accuracy comes at the cost of obtaining large

sampling data and time for training the increased number of parameters in the network. But large sample set for training leads to the problem of over fitting. Thus the goal while designing a neural network is to find a simple ANN structure (less number of nodes) that can fit a given set of data with maximum accuracy. This conflicting nature of objectives led to the formulation of a multi-objective optimization problem (MOOP-2), where the aim is to maximize the accuracy of the network while simultaneously minimize the total number of nodes in the architecture. The functioning of this multi-objective optimization algorithm is clearly described in [27].

4.3 Results and Discussions

The sampling plan is fixed to Sobol for the reasons mentioned previously. K-fold sampling, which was described in the first part of this article, is used for determining the sample size required for training the model. The architecture was varied along with variations in sample size and the surrogate ANN models thus obtained using the above algorithm are reported in Table 12. The number of layers were varied up to a maximum value of 3 and number of nodes per each layer were varied up to 8, thus leading to 512 possible architectures. For each of the architectures considered for investigation, the sample sizes were also varied within a range of 30 to 200. Several possible architectures (nearly 200) were investigated over a long period of time and the potential results are reported below. The entries in the columns of the Table are N1 – number of nodes in hidden layer 1, N2- number of nodes in hidden layer 2, N3 – number of nodes in hidden layer 3, N_TF – Numerical indicator for transfer function where 1 indicates tan sigmoidal activation and 2 indicate log sigmoidal activation, N indicates the total number of nodes which is the sum of entries in first three columns and n indicates the sample size. Clearly, one can observe that *none* of the architectures were able to predict the output of the system with good accuracy. Thus from these results, the following points can be deduced:

1. Owing to the fact that there is no existing first principle model for the electrospinning process, it can be said that the process is highly nonlinear and complex. This implies that more number of data points may be required for modelling this complex process.
2. Removing the noise present in the experimental data might help to improve the results upto some extent.

All this study corresponds to only output 1 and a simultaneous study was also performed for outputs 2 and 3. The prediction accuracy was very low for all the outputs.

Table 12: Effect of Architectures on network predictability for output-1

N1	**N2**	**N3**	**N_TF**	**R^2**	**N**	**n**
3	5	6	2	0.364	14	200
4	7	0	2	0.901	11	176
1	1	6	1	0.670	8	125
6	4	4	1	0.645	14	179
8	7	2	1	0.682	17	200
1	1	5	2	0.324	7	170
4	4	3	2	0.354	11	197
2	2	0	2	0.349	4	134
7	5	2	1	0.657	14	164
8	6	2	2	0.395	16	200
5	6	2	1	0.682	13	200
1	1	6	1	0.677	8	137
6	3	7	2	0.343	16	170

6	4	2	1	0.652	12	170
3	5	0	2	0.537	8	101

Chapter 5

Conclusion

5.1 ANFIS Surrogate building algorithm – PVAC system

In this work, the author has presented a comprehensive research over design of ANFIS surrogate models for enabling the optimization of complex industrial process by making use of surrogate based optimization methods. ANFIS are specifically selected due to their ability to incorporate the actual physics of the system along with black box modelling. Since the predictability and efficiency of the surrogate model play a dominant role in success of surrogate based optimization, the effect of various parameters on ANFIS surrogate building process has been studied. It was found that the parameters viz. (a) architecture of ANFIS, (b) sample size required by the ANFIS, (c) maximum possible accuracy of prediction and (d) a robust sampling plan are the major parameters which effect the surrogate building process. Along with that, the author also studied and justified the fact that, in case of no expert data, choosing the maximum possible fuzzy rules is not always the right approach for capturing the dynamics of the system. Therefore, it has been suggested that there is a need for a novel parameter free ANFIS surrogate building algorithm, which can estimate all the parameters automatically, thus eliminating the human intervention ANFIS design. The prime objective of this study is to understand the functioning of the ANFIS and layout a blue print for the intelligent design of Neuro Fuzzy

networks. The ANFIS surrogate models are utilized to emulate a highly complex PVAc-LCB model consisting of 3 inputs and 4 outputs. An optimization problem was solved using the ANFIS surrogates in place, for finding the best process conditions to maximize the branching index, molecular weight in minimum polymerization time. The results of the surrogate based optimization revealed that the surrogate based optimization method could reduce the function evaluations by nearly 9 times as compared to conventional approach, thereby making way for real time optimization of the complex PVAc model.

5.2 ANN Surrogate building algorithm – Electrospinning process

In the previous chapter, the author has presented a brief research over design of ANN surrogate models for enabling the modelling and optimization of electrospinning process by making use of surrogate based optimization methods. ANNs are specifically selected due to their robustness and inherent potential to capture the behavior of any complicated nonlinear system. However, in this case-study, the inability of ANN based surrogate building algorithm for the prediction of maximum accurate results might be due to insufficient number of sample points for training the surrogate model or the experimental input output data may contain noise which needs to be pre-filtered before sending it to the model.

In order to test the robustness of this algorithm, a highly non-linear test function (Ackley function) was considered to perform ANN based interpolation. From the Fig.12, it is clear that the predicted ANN model was able to mimic the complicated function with high accuracy.

Ackley function: $$f(x,y) = \{-20\exp\left(-0.2\sqrt{0.5(x^2+y^2)}\right) - \exp(0.5\,(\cos(2\pi x)+\cos(2\pi x)) + e + 20\}$$

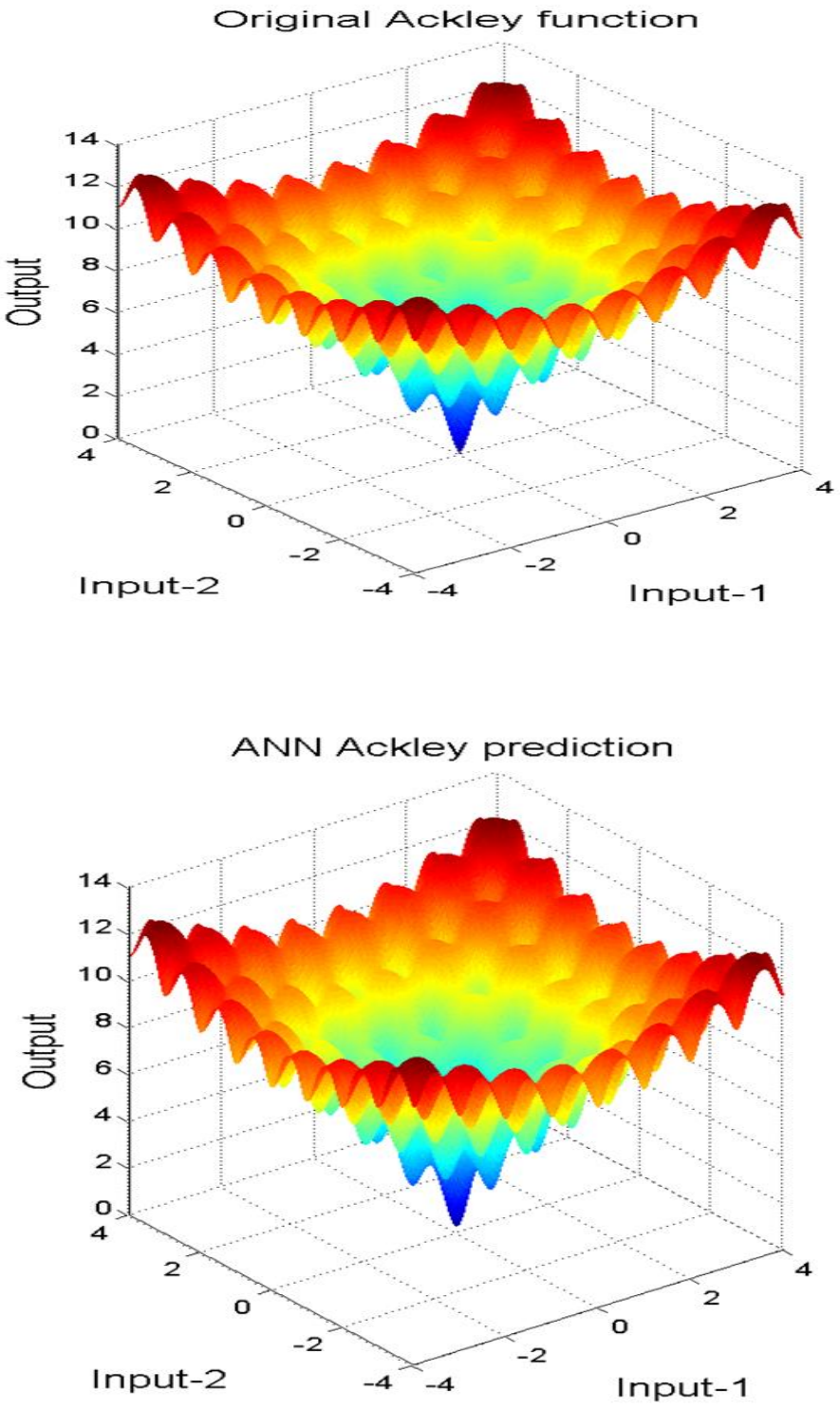


Fig. 10: Comparison of ANN based Interpolation of Ackley function with original function

Future Work

1. To develop a novel parameter free ANFIS surrogate building algorithm for intelligent design of neuro fuzzy networks along with simultaneous estimation of parameters such as sample size, sample plan, membership function and so on to enable it to emulate with maximum accuracy without being over-fitted.

2. To apply the proposed surrogate building algorithms to build ANFIS surrogate models for emulating several industrially validated models and enable the online optimization of such complex models.

3. To apply the proposed algorithm to build ANFIS models for an experimental setup and ensure the successful working of the proposed ANFIS surrogate building algorithm with experimental setups.

4. To successfully eliminate the human intervention and heuristic based inputs in ANFIS design and implementation with the help of the proposed parameter free algorithm.

5. To mimic the highly complex electrospinning process using the ANN based surrogate building algorithm.

References

[1] A. Mogilicharla, T. Chugh, S. Majumdar and K. Mitra “Multi-objective optimization of bulk vinyl acetate polymerization with branching”, Mat. and Man. Proc., vol. 29, pp. 210-217, 2014.

[2] A. Mogilicharla, P. Mittal, S. Majumdar, K. Mitra, “Kriging Surrogate based Multi-objective Optimization of Bulk Vinyl Acetate Polymerization with Branching”, Mater. and Manuf. Process. Genetic Algorithms special issue, vol. 30, pp. 394-402, 2015.

[3] K. Deb, Multi-objective Optimization using Evolutionary Algorithms, Wiley, Chichester, UK, 2001.

[4] K. Deb, “A fast and elitist multi-objective genetic algorithms”, IEEE Trans. Evo. Comp., vol. 6, pp 181–197, 2002.

[5] P. Nain and K. Deb, “A computationally effective multi-objective search and optimization techniques using coarses-to-fine grain modelling”, In Proceedings of the PPSN Workshop on Evolutionary Multi-objective Optimization, 2002.

[6] Hearst, Marti A., et al. "Support vector machines." Intelligent Systems and their Applications, IEEE, vol.13(4), pp 18-28, 1998.

[7] I. M. Sobol, “On the distribution of points in a cube and the approximate evaluation of integrals”, Zh Vychisl. Mat. Mat. Fiz. vol. 7, no. 4, pp. 784-802, 1967

[8] S. Haykin, “Neural Networks: A Comprehensive Foundation”, Macmillan College Publishing Company, New York, 1994.

[9] D. R. Jones, “A taxonomy of global optimization methods based on response surfaces”, J. of Global Optim., vol. 21, pp. 345-383, 2001.

[10] I.J. Forrester, A. Sobester, A.J. Keane, Engineering Design via Surrogate Modelling, A practical guide, Wiley, 2008.

[11] T. Takagi and M. Sugeno, “Fuzzy identification of systems and its application to modeling and control”, IEEE Transaction on Systems, Man and Cybernetics, vol. 15, pp. 116–132, 1985.

[12] Jyh-Shing Roger Jang, “ANFIS – Adaptive Network Based Fuzzy Inference system”, IEEE Transactions on System, Man and Cybernetics, vol. 23, pp. 665-685, 1993.

[13] V. Dua, “A mixed-integer programming approach for optimal conFiguration of artificial neural networks”, Chem. Engg. Res. and Des., vol. 88, pp. 55–60, 2010.

[14] M. Hagen, H. B. Demuth and M. H. Beale, “Neural Network Design”, 2002.

[15] B. K. Giri , F. Pettersson , H. Saxn and N. Chakraborti, “Genetic programming evolved through bi-Objective genetic algorithms applied to a blast furnace”, Mat. and Man. Proc., vol. 28, no. 7, pp. 776-782, 2013.

[16] B Yegnanarayana, “Artificial neural networks for pattern recognition”, Sadhana vol. 19, no.2, pp 189-238, 1994.

[17] M. D. Morris, “Factorial sampling plans for preliminary computational experiments”, Technometrics, vol. 33, no. 2, pp. 161–174, 1991.

[18] B. Tang “Orthogonal array-based latin hypercubes”, Jour. of the Amer. Statis. Asso., vol. 88, pp. 1392–1397, 1993.

[19] N. K. Roy, W. D. Potter, and D. P. Landau, “Polymer property prediction and optimization using neural networks”, IEEE Trans. On Neu. Net., vol. 17, no.4, pp. 1001-1014, 2006.

[20] Rao, R. Bharat, Glenn Fung, and Romer Rosales. "On the Dangers of Cross-Validation. An Experimental Evaluation." SDM., pp .588-596, 2008.

[21] M.D. Morris, T.J. Mitchell, Exploratory designs for computational experiments, J. of Statistical Plan. and Inference, vol. 43, pp. 381–402, 1995

[22] A. Nuchitprasittichai and S. Cremaschi, “An algorithm to determine sample sizes for optimization with artificial neural networks”, AIChE J, vol. 59, no.3, pp. 805 – 812, 2012.

[23] Wong, Shen Yuong, et al. "On Equivalence of FIS and ELM for Interpretable Rule-Based Knowledge Representation." Neural Networks and Learning Systems, IEEE Transactions on, vol. 26(7), pp 1417-1430, 2015.

[24] Amirkhani, S., et al. "ANN and ANFIS models to predict the performance of solar chimney power plants." Renewable Energy, vol. 83, pp. 597-607, 2015.

[25] K. Ron. “A study of cross validation and bootstrap for accuracy estimation and model selection”, Proceedings of 14th International Joint Conference on Artificial Intelligence, vol. 12, no. 2, pp. 1137-1143, 1995.

[26] G. Fung, R. B. Rao and R. Rosales, “On the dangers of cross-validation - an experimental evaluation (SIAM)”, in Proceedings of the SIAM International Conference on Data Mining, pp. 588–596, 2008.

[27] Miriyala, Srinivas Soumitri, Prateek Mittal, Saptarshi Majumdar, and Kishalay Mitra. "Comparative study of surrogate approaches while optimizing computationally expensive

reaction networks." Chemical Engineering Science, vol. 140,pp. 44-61, 2016.

[28] Bhardwaj, Nandana, and Subhas C. Kundu. "Electrospinning: a fascinating fiber fabrication technique." Biotechnology advances 28, no. 3, 325-347, 2010.