

Analysis of Artificial Neural Networks For Building Automated Surrogate Algorithms

M Srinivas Soumitri

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

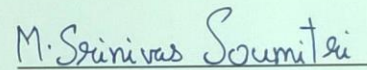
June, 2015

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.



(Signature)



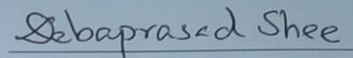
(– Student Name –)



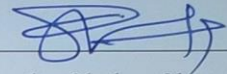
(Roll No)

Approval Sheet

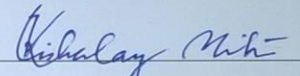
This thesis entitled “Analysis of Artificial Neural Networks For Building Automated Surrogate Algorithms” by Srinivas Soumitri Miriyala is approved for the degree of Master of Technology from IIT Hyderabad.




Dr. Debaprasad Shee
Examiner



Dr. Chandra Shekar 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 Sekhar 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 and sister, without whose support I would have never been able to fulfill any of my aims.

Thank you one and all.

Dedicated to

My Mother T. Annapurna.

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 artificial neural networks viz. (a) architecture of ANN, (b) sample size required by the ANN, (c) maximum possible accuracy of prediction, (d) a robust sampling plan and (e) transfer function choice for node activation. The ANNs are then utilized as surrogates for a highly nonlinear industrial sintering process, the optimization of which is then realised nearly 7 times faster than the optimization study using the expensive phenomenological model.

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

List of Figures

- Figure 1 Basic Structure of a neuron and node [4]
- Figure 2 Schematic showing the generic surrogate building algorithm [6]
- Figure 3 The distribution of 200 sample points using the a) Sobol sampling plan and b) LHS sampling plan. [10]
- Figure 4 Schematic of Industrial Sintering Process [14]
- Figure 5 Contour plots of Output-1 with respect to two inputs considered at a time. [22]
- Figure 6 Contour plots of Output-2 with respect to two inputs considered at a time. [22]
- Figure 7 Evolution of ANN surface for the architecture [3-6-2-0-1] for output -1 [28]
- Figure 8 Parity plot for Output 1 using the architecture = 3-6-2-0-1 with $R^2 = 0.99993$ obtained using HC sampling technique [31]
- Figure 9 Parity plot for Output 2 using the architecture = 3-5-4-1-1 with $R^2 = 0.993$ obtained using HC based technique [31]
- Figure 10 PO front comparison of optimization using ANN surrogate built by HC based sampling method and original first principle Sintering model [33]

List of Tables

Table 1	Comparison of different sampling plans in terms of PHI metric and computational time for 200 sample points.	[10]
Table 2	Kinetic model of the Sintering system	[16]
Table 3	Multi objective optimization formulation of the Sintering model.	[21]
Table 4	NSGA II Parameters for solving the MOOP problem of Sintering system	[23]
Table 5	Effect of Architectures on network predictability for output-1	[25]
Table 6	Effect of Sample size on network predictability for output-1	[27]
Table 7	Effect of Activation function on network predictability for output-1	[30]
Table 8	ANN surrogates for Sintering model	[30]

Contents

Declaration.....	Error! Bookmark not defined.
Approval Sheet	Error! Bookmark not defined.
Acknowledgements	iv
Abstract.....	vi
List of Figures.....	vii
List of Tables	viii
1 Introduction.....	1
2 Formulation.....	7
2.1 Parameters in Surrogate building algorithm.....	7
2.1.1 Accuracy of prediction.....	7
2.1.2 Sampling plan or design of experiments (DoE).....	8
2.1.3 Sample size.....	11
2.1.4 Architecture of the network.....	12
2.1.5 Activation function.....	13
2.2 Industrial Sintering	14
2.2.1 Modeling of Sintering Process.....	14
2.2.2 Optimization of Sintering Process	17
3 Results and Discussions.....	21
4 Conclusion	34
Future work.....	36
References.....	38

Chapter 1

Introduction

Process control and optimization of industrial problems, often involving huge computational rigour owing to the usage of complex and robust first principle based models, demand large times for computation. The first principle models, such as, those trying to capture the dynamics of reaction networks in a polymer industry or a model handling the wake effects or turbulence in fluid flow, etc. usually involve several highly nonlinear coupled ordinary and partial differential equations (ODEs & PDEs) [1]. This necessitates the involvement of time consuming simulation packages, such as, Computational Fluid Dynamics (CFD), or some differential algebraic solvers (DAE) etc., to solve the system of ODEs and PDEs to facilitate their implementation at pilot plant level or at an industrial scale. The intrinsic 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 grows by multiple folds when the considered system is multi-dimensional in nature (say m dimensions) with optimization formulation involving multiple conflicting objective functions instead of one. The conflicting nature of the objective functions results in a set of non-dominating solutions called Pareto Optimal (PO) solutions from which a single solution is obtained using 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. This concept of online optimization is practically imbibed in industry 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 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. Artificial Neural Networks (ANNs) are one of the prominent candidates for surrogate models by virtue of their immense potential to recognize complex patterns [6]. ANNs are mathematical models in form of network of nodes, whose functioning is motivated by the impeccable parallel networking of neurons in the human brain. They are widely acknowledged across all engineering and scientific disciplines for their immense applications in various fields such as computer science and electrical engineering [6], nanoscience [7], geosciences [8], chemical engineering [9, 10] and biological sciences [11] and so on. In this article, an elaborate study is conducted on the perceptron networks, which without any loss of generality can be extended to the specific class of recurrent networks. The

parameters, in terms of weights and biases of the network, enable several degrees of freedom to capture the overall nonlinear behaviour in the given complex system. This unique ability of the ANN to capture the global trend of the complex model with maximum accuracy, not only assures it a status of an efficient surrogate for optimization but also allows for its wide applicability as a highly efficient interpolator which then finds an edge in several numerical techniques [12]. Several other notable surrogate models listed in literature are Kriging Interpolation (KI) [13], Support Vector Machines (SVM) and Response Surface Methodologies (RSM) [14]. The philosophy of surrogate models is to generate accurate functional relationship among inputs and outputs of a given process. RSMs are statistical models, which try to regress lower order (commonly, second order) polynomial models followed by conducting a sequence of designed experiments to guide the optimization search in a direction of optimal response of the objective function. Several instances of failure in capturing the local surface utilizing lower degree polynomials led the RSM research into dealing with higher degree polynomials. KI, which has proved its immense scope of applicability in the areas of system identification, parametric analysis and optimization, geosciences, statistics, design and analysis of computer experiments, is yet another popular function approximation technique. It uses Gaussian distribution functions to fit the training data with a set of parameters which can be tuned based on the estimation of potential error in interpolation. The interpolator predicts the output using the weighted combinations of predictions from simple basis functions.

On the other hand, 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. 1. They are widely acknowledged for their immense applications in pattern recognition problems, image processing and many other chemical engineering applications. 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.

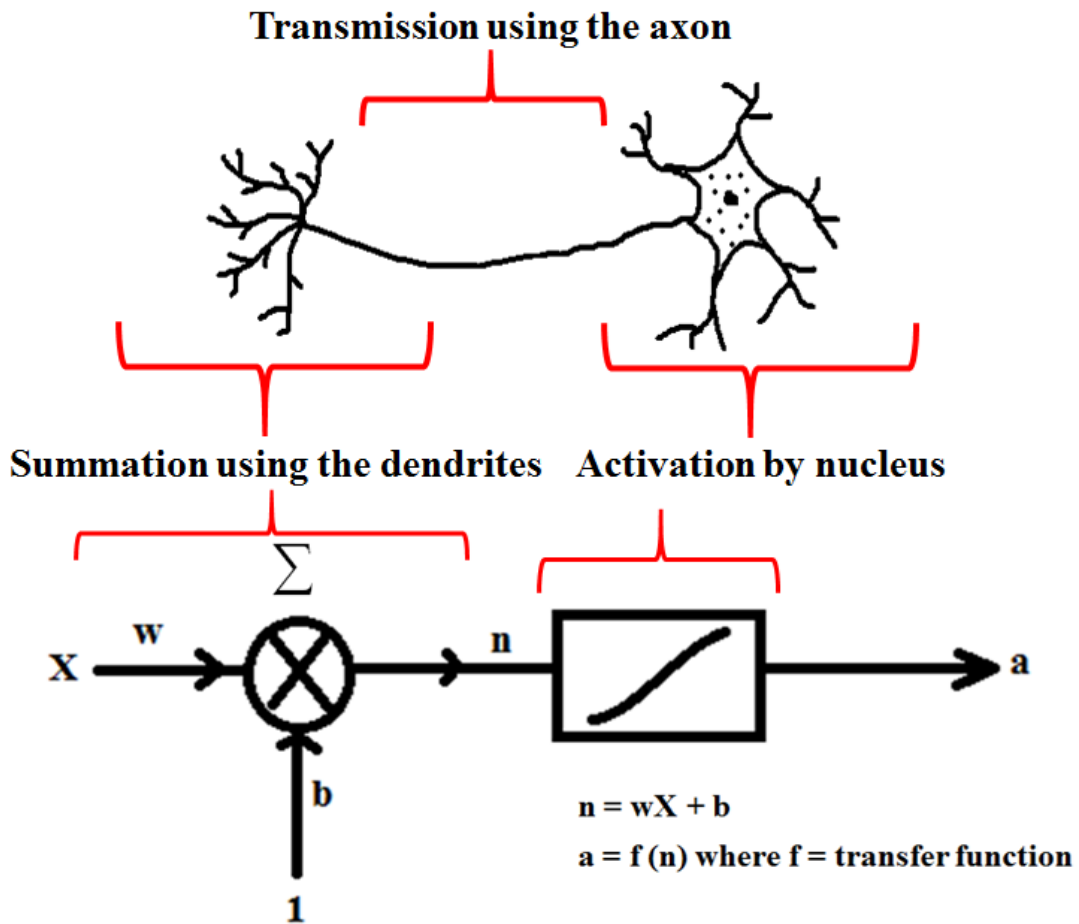


Fig. 1. Basic Structure of a neuron and node

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 [15]. 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 [12]. This rationale justifies for the fact that the aforementioned assumption may not be true in all cases. Apart from this, 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 devise a logical approach to design the architecture of a given network, simultaneously, along with sample size determination. Some of the prominent contributions in the literature are mixed integer nonlinear programming (MINLP) approach [16], the Akaike Information Criteria (AIC) [17], 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. With this backdrop, a schematic for the current scenario of a primal surrogate building algorithm for ANN as surrogate model has been presented in Fig. 2. The simple layout in Fig. 2 clearly shows 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 m -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 ANN 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 ANNs 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, transfer function, etc. on the aspects of predictability and parsimonious behaviour of the surrogate model has been investigated. The potential dangers associated with heuristic based design of ANN with respect to recognizing the capability of the ANNs as surrogate models have also been presented. An industrially validated model for sintering process, used in steel plants, is considered for all the sensitivity analysis and optimization studies. A comprehensive comparative study between the results obtained using several ANN surrogates obtained by varying the aforementioned parameters, is presented in details. The Introductory section in the article is followed by the Formulation section

comprising a detailed description of the sintering model and the ANN sensitivity analysis. This is then followed by the Results and Discussions section before concluding the work in the Conclusion section.

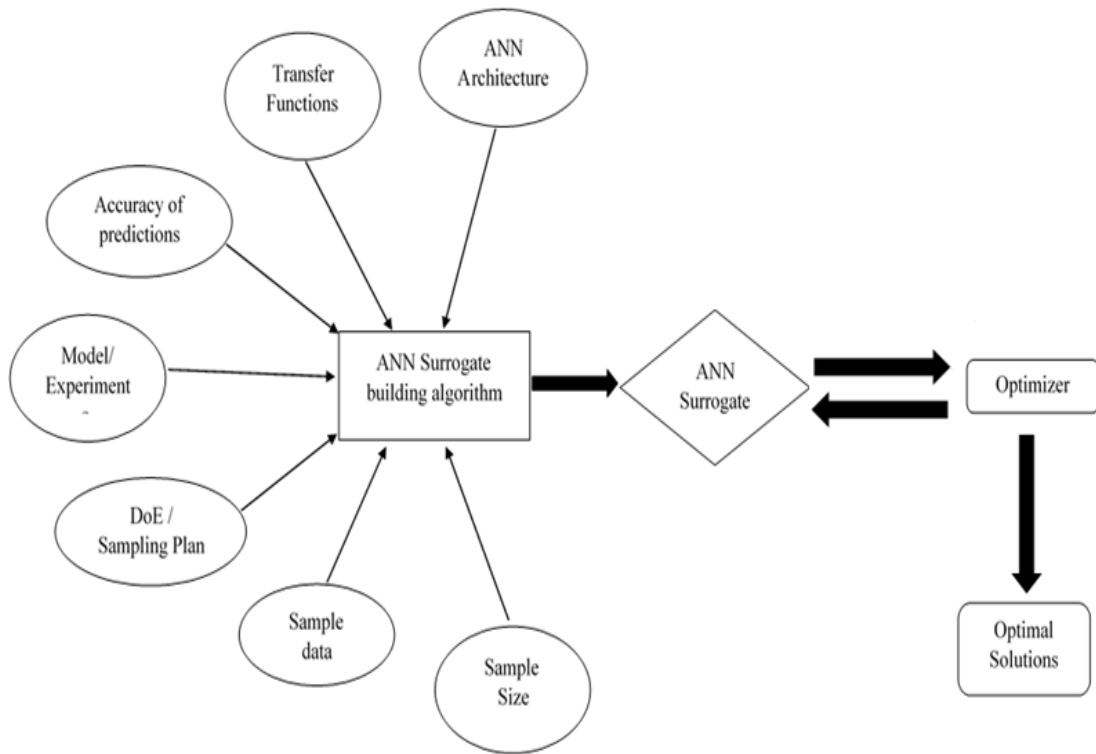


Fig. 2: Schematic showing the generic surrogate building algorithm.

Chapter 2

Formulation

2.1 Parameters in Surrogate building algorithm

With reference to Fig. 2, the parameters involved in surrogate building algorithm are listed down before describing them further in details.

- i) Accuracy of prediction.
- ii) Sampling plan or design of experiments (DoE).
- iii) Sample size.
- iv) Architecture of the network.
- v) Activation function.

2.1.1 Accuracy of prediction.

The accuracy of the surrogate model is a necessary parameter, which needs to be specified prior to the modelling 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. Two well-known statistical measures [18] for estimating the accuracy of the predictions by the network have been considered:

i) *Root mean square error: RMSE*

ii) *Correlation coefficient r^2*

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

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

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

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

2.1.2 Sampling plan or DoE

The sampling plan is at the heart 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 [19]. 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 [18]. 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) [18, 20], 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 [21], 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 3. 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 [18, 22] 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. 3 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.

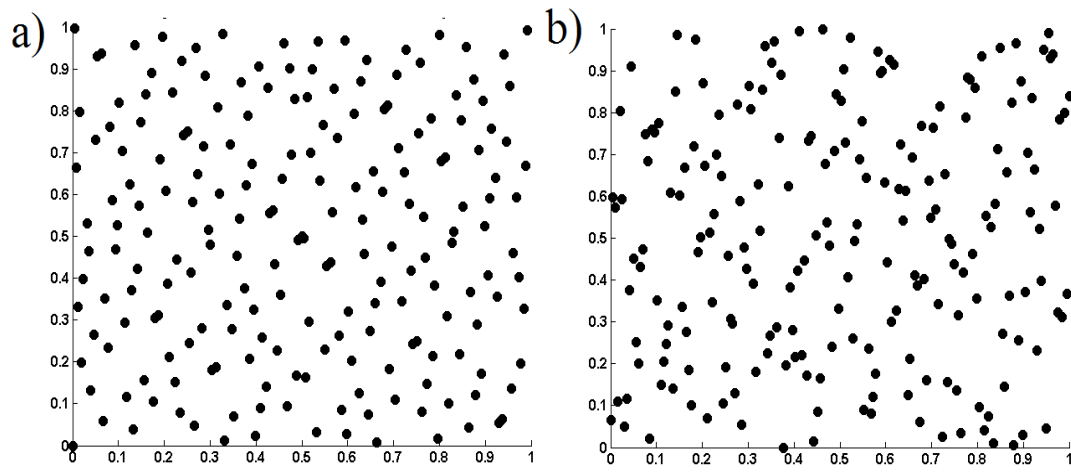


Fig. 3: The distribution of 200 sample points using the a) Sobol sampling plan and b) LHS sampling plan.

Table 1: Comparison of different sampling plans in terms of PHI metric and computational time for 200 sample points.

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

2.1.3 Sample size

ANNs are infamous for their data greedy nature. The scarce amount of literature available on realizing the ANNs as potential surrogates reveals that no proper rationale is devised to decide the number of data points required for training [23]. 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 [24] 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.

2.1.4 Architecture of the network

The architecture of the network is perhaps the most important input, which influences the ANN surrogate building algorithm to maximum extent than any other parameter. The design of architecture as described in the introductory section has always been determined through a heuristic assumption of considering a single hidden layer and varying the nodes in that layer until some desired accuracy is found. Since a hidden layer in a perceptron structure can be visualised geometrically

as an m-dimensional hyper-plane trying to separate out the existing data into two sub spaces, a multi-layer perceptron network may, therefore, provide more accuracy for an unseen data, which might be linearly inseparable [12]. This rationale justifies for the fact that the aforementioned assumption may not be true in all cases. The author in this current work 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 a optimal design the architecture of the network along with simultaneous determination of the sample size and sampling plan, which would enable the network to predict results with maximum accuracy.

2.1.5 Transfer function

This parameter specific to the ANNs describes the necessity of considering various possible alternatives to ensure proper activation of the inputs, which would lead to an efficient training of the ANN. Two prominent activation functions listed in literature [15] have been considered to analyse their effect on the performance of ANNs.

i) The continuous log-sigmoid transfer function:

$$y(x) = \frac{1}{1 + e^{-x}} \text{ and } \frac{dy}{dx} = y(1 - y)$$

and

ii) The continuous tan-sigmoid transfer function:

$$y(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \text{ and } \frac{dy}{dx} = (1 - y^2)$$

These activation functions are enabled into the architecture as the decision variables of the optimization formulation mentioned in the previous section. The author in this work has limited the variability of the activation function to the entire network and

thereby restricting the variation at the level of each node. Although this can be implemented with slight modification, it has been intentionally avoided to honor the computational time constraint on the ANN design. The output layers in any given network are always activated by pure linear activation function [15].

2.2 Industrial Sintering

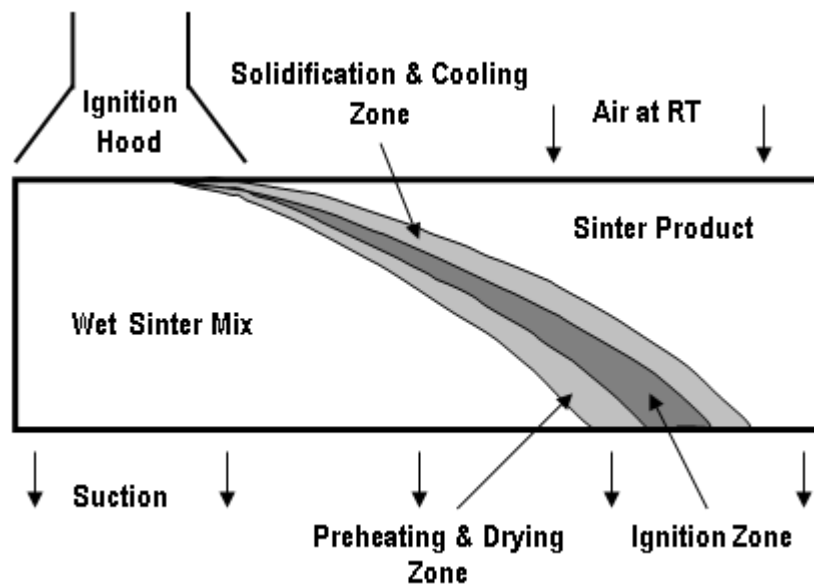


Fig 4. Schematic of Industrial Sintering Process

2.2.1 Modeling of Sintering Process

The industrial process considered (Fig 4) in this article is the iron ore sintering process that produces raw material for the blast furnace operation in steel plants. High quality sinter is essential for better chances of running the blast furnace. Optimum melting achieved during the sintering operation determines the quality of the sinter. Therefore, it is one important parameter supervising the performance of the sintering process. If the melting is less than the optimum, the blast furnace

witnesses the granule breakage, while if the melting is more than optimal point, the reducing ability of the sinter drastically decreases leading to operational problems in the blast furnace. The consumption of coke is one another important metric of sintering operation. In order to ensure better sintering process, consumption of coke should be minimal due to the direct correlation of coke consumption with carbon footprint of the plant. The lesser the coke consumption, higher the efficiency of the plant in terms of energy, which leads to lower carbon footprint value of the operation and thus lower the cost of operation. The conventional sintering process starts with the raw materials being charged on a moving strand (30–60 cm thick) proceeding for sintering. The combustion of the coke, to attain the desired temperature during sintering process, begins in the top where the charge is ignited. This is the region where cold air is forced inside by the vacuum created by suction pressure. The cold air cools the corresponding zone resulting in melting lower than the desired. On the other hand, the preheated air, making its way from the top zone where coke is burnt, creates a broad melting zone in the bottom region, which is way higher than the optimum. Thus, the melting of the charge is not uniform because of different temperatures owing to the different conditions of combustions at both upper and lower regions of the charge. In order to avoid this, the charging process is split into two layers, where the combustion of coke is different but uniform in each of the layers, and this ensures uniform melting of the sinter mix.

A dynamic sintering process model has been developed considering a two-dimensional Cartesian coordinate system. The variations in the lateral direction are assumed to be negligible. The temperatures and compositions of both solids and gas

at any position of the sinter bed can be calculated using this model. For predicting the gas velocity, the well-known Ergun's equation is used:

$$\frac{\Delta P}{L} = \frac{150\mu \cdot V(1 - \varepsilon^2)}{d_p^2 \varepsilon^3} + \frac{1.75\rho_g \cdot V^2(1 - \varepsilon)}{d_p \varepsilon^3}$$

The following equation is used in general for predicting the transport variables, temperature, and concentration of gas and solid state species:

$$\frac{d\phi}{dt} + \left(V_x \frac{d\phi}{dx} + V_y \frac{d\phi}{dy} \right) = \alpha \left(\frac{d^2\phi}{dx^2} + \frac{d^2\phi}{dy^2} \right) + S_c + S_p \phi$$

where ϕ is transport variable, and S_c and S_p are the source terms.

The initial conditions for solving this ODE are provided at the inlet boundary while zero gradient condition is used at the outlet. The convective terms in the rate expression (given below) are used to calculate the velocities of the solids.

$$\frac{dX}{dt} + V_s \frac{dX}{dx} = \sum \frac{R_i}{\rho_{s,i}}$$

All the prominent reactions and phase transformations considered for developing the sintering model are listed in Table 2. The details of the kinetic models and the parameters involved in reaction mechanisms can be obtained from the literature [28, 29].

2.2.2 Optimization of Sintering Process

Extensive simulation studies reveal that the twin objectives of

- a) achieving good quality sinter by maximization of melting with
- b) minimum coke consumption

are conflicting in nature. This kind of optimization problem with conflicting objective functions is ideal for a multi-objective optimization framework where the trade-off between the objectives can be captured. For the current sintering problem

considered, it is observed that 30% melting is the optimal value for the sinter quality for melting (SQM). The first objective is, therefore, defined as to achieve a maximum of 100% when the SQM equals 30%. The combined weighted average of the coke consumed in both the layers (C_w) is considered as the second objective. The height (B) of any one of the two layers in sintering process and the percentages of coke present in each of the two layers (C_A , C_B) can be considered as the decision variables of this optimization problem. The lower and upper bounds for the decision variables are represented by superscripts L and U, respectively.

Table 2: Kinetic model of the Sintering system

	Reaction name	Formula
1	Iron oxides reduction	$\text{Fe}_2\text{O}_3 \xrightarrow{\text{CO}/\text{H}_2} \text{Fe}_2\text{O}_3 \xrightarrow{\text{CO}/\text{H}_2} \text{FeO} \xrightarrow{\text{CO}/\text{H}_2} \text{Fe}$
2	Limestone decomposition	$\text{CaCO}_3 \rightleftharpoons \text{CaO} + \text{CO}_2$
3	Combustion of coke	$\text{C} + \text{O}_2 \rightleftharpoons \text{CO}_2$; $2\text{C} + \text{O}_2 \rightleftharpoons 2\text{CO}$
4	Solution loss reaction	$\text{C} + \text{CO}_2 \rightleftharpoons 2\text{CO}$
5	Water gas reaction	$\text{C} + \text{H}_2\text{O} \rightleftharpoons \text{CO} + \text{H}_2$
6	Water gas shift reaction	$\text{H}_2\text{O} + \text{CO} \rightleftharpoons \text{CO}_2 + \text{H}_2$
7	Water vapor formation	$\text{H}_2 + 0.5\text{O}_2 \rightleftharpoons \text{H}_2\text{O}$
8	Condensation and drying	$\langle \text{H}_2\text{O} \rangle \rightleftharpoons (\text{H}_2\text{O})$
9	Solidification and melting	$[\text{Solid Sinter}] \rightleftharpoons \langle \text{Liquid Sinter} \rangle$

The well-known NSGA II was chosen to solve the aforementioned optimization problem. When the above-mentioned multi-objective optimization is carried out using the complex phenomenological model presented in section 2.2, the

optimization takes a large amount of time creating a deterrent to use the optimization approach as an online one. So, the real challenge in this study is to build ANN surrogates in place of the phenomenological model for this complex sintering process and carry out the optimization exercise using the surrogate model.

2.3 Artificial Neural Network: The Algorithm and its functioning

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. Although as mentioned previously, 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.

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. The normalization needs to be performed in the range according to the transfer function used in the code. This is specifically to capture the sigmoidal shape and to thereby enable the network to perform to its full capability. Thus the data is normalized between -10 to 10 for log sigmoidal activation and between -5 to 5 for tan sigmoidal activation.
2. Declaration and initialization of weights: The number of weights keep changing based on the architecture of the network. Thus, once the

architecture is sent to the code as input, the weights are declared and initialized, to a suitable value, according to the architecture. The initial values of the weights will affect the optimization routine used in determining the weights of the network.

3. Network Training and Validation: The network is trained using the back propagation algorithm and weights are estimated using the Levenberg Marquardt (LM) procedure. The LM algorithm will ensure faster convergence to the optimum values as it can be made to work like both the fast Gauss Newton algorithm and highly converging steepest descent algorithm by changing the damping factor. 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 robust industrially validated model considered in the current study is a 3 input 2 output Sintering model whose validation results can be obtained from literature [28, 29]. The MOOP formulation presented in Table 3 is solved using the real and binary coded NSGA II algorithm whose credentials are given in Table 4. Although the NSGA II algorithm was run for 30 generations, it was observed that the PO front was saturated at generation number 25 with each generation containing 50 populations. Thus, the total function evaluations required to perform the optimization run with original model in place, were nearly 1300 (= 50*26, including the 0th generation). 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 3: Multi objective optimization formulation of the Sintering model.

Objective functions	Decision Variables
$\begin{matrix} \text{Max} \\ C_A, C_B, B \end{matrix} \quad SQM$	$C_A^L \leq C_A \leq C_A^U$
$\begin{matrix} \text{Min} \\ C_A, C_B, B \end{matrix} \quad C_W$	$C_B^L \leq C_B \leq C_B^U$
	$B^L \leq B \leq B^U$

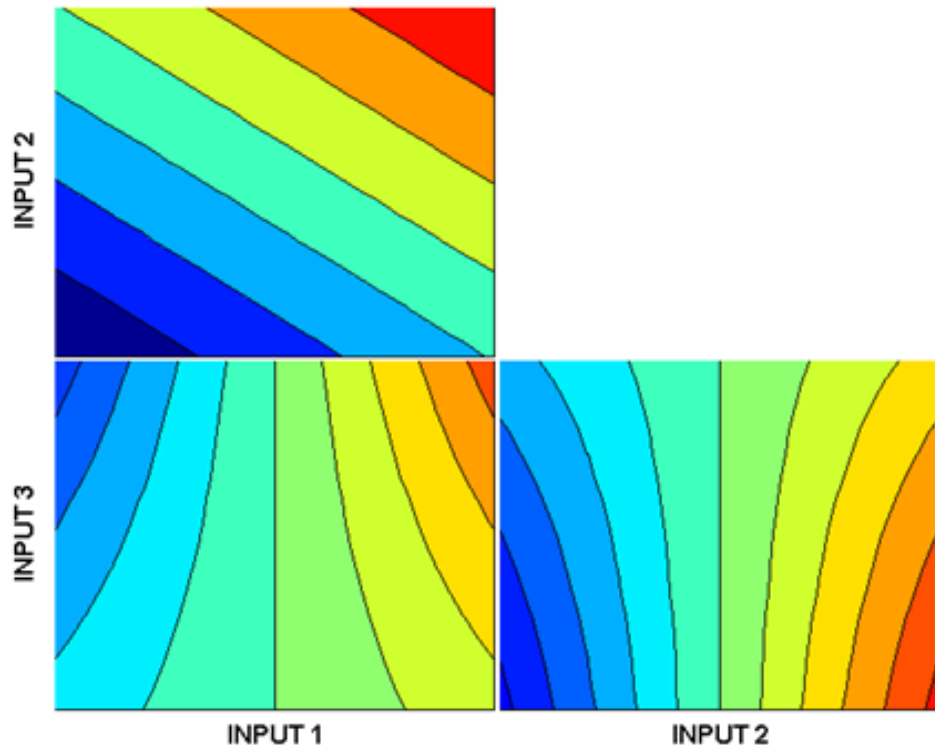


Fig. 5: Contour plots of Output-1 with respect to two inputs considered at a time.

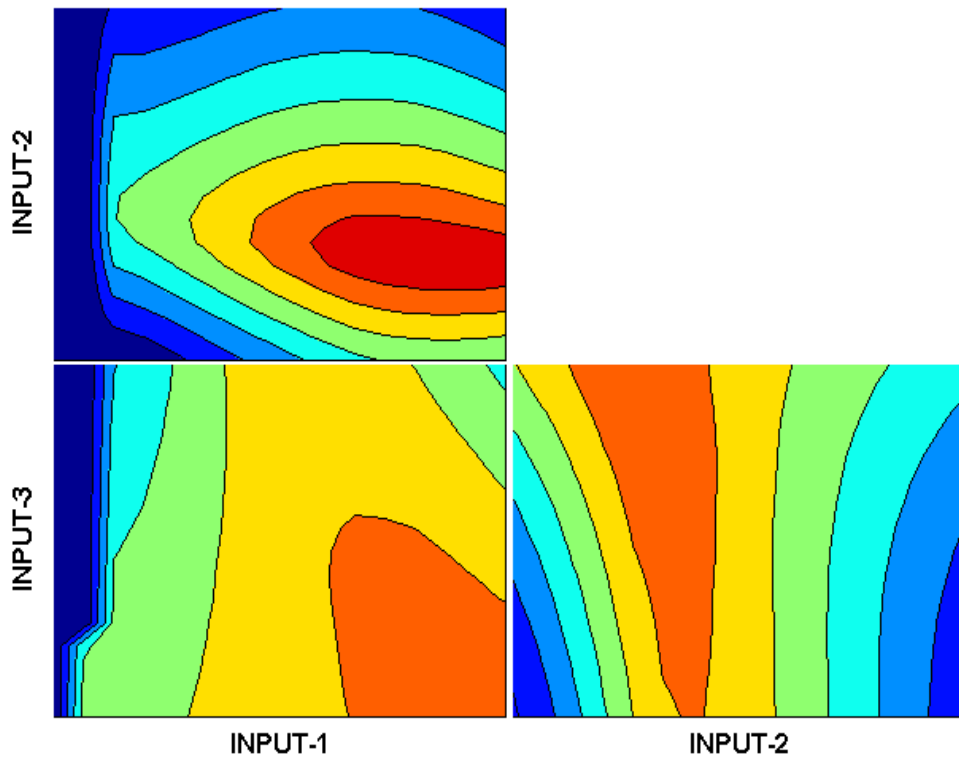


Fig. 6: Contour plots of Output-2 with respect to two inputs considered at a time.

Table 4: NSGA II Parameters for solving the MOOP problem of Sintering system

No.	Parameter name	Value
1	Maximum generation size	30
2	Population size	50
3	Crossover probability	0.9
4	Mutation probability	0.01
5	C_A^L, C_A^U (in %)	3,10
6	C_B^L, C_B^U (in %)	3,10
7	B^L, B^U (in cm)	1,16

I. Non linearity check.

In order to assess the amount of nonlinearity present in the considered sintering model, the contour plots in the form of tile plots for two inputs taken at a time are represented in Fig. 5 and Fig. 6 for output 1 and output 2, respectively. The nonlinear curves and the drastic intensity variations in these tile plots clearly indicate the complicated behavior of the sintering model. Although full factorial experiment was required for generation of these tile plots, none of the sample points were used for either training or validation of the ANN models. These plots were created purely to show the nonlinearity present in the model.

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

Keeping transfer function fixed to tan sigmoidal activation, the architecture was varied along with variations in sample size and the surrogate ANN models thus built

are reported in Table 5. 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 250. Since this would lead to numerous possible case studies, a progressive study has been adopted by first fixing the number of layer and then varying the number of nodes in that layer. 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 5. 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 for single hidden layer, lower the number of nodes larger is the sample size requirement. This study justifies the fact that as the number of parameters of the network increases, the number of sample points required for training decreases. Another interesting observation from this Table reveals that, for a multi-layered network the above mentioned observation does not hold well. As mentioned previously in the article, a large amount of nonlinearity in the sampled data would require more number of layers than more number of nodes in a single layer. Therefore, when compared with the architecture [3-8-0-0-1], the architecture [3-6-2-0-1] is predicting the results far better ($r^2 = 0.999$) even with less sample size. All this study corresponds to only output-1 and a

simultaneous study was also performed for output-2. Thus from these results, a clear rationale is observed for

1. Devising a logical approach for optimal design of the architecture of the neural networks.
2. Exploration of multi-layered architectures for better system identification.

III. Effect of Variation of Sample Size on predictability of an ANN.

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 5 and its predictability was studied with variation in sample size. The results of this study are reported in Table 6. The evolution of the ANN surrogates with increment in sample size for output 1 is shown in Fig. 7. These Figures show the distribution of the sample points in the three dimensional space in the left subfigure, the surface plot of the formed ANN surrogate model in center while the parity plot of the corresponding ANN surrogate is depicted in the right subfigure. The results in Table 5 and 6 clearly indicate that as the sample size increases, the prediction accuracy of the architecture also increases. But, in Table 6, one can observe that the accuracy of predictions (r^2) increases with increase in sample size. But after sample size 50, the improvement slows down and reaches saturation, and after sample size 80, 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.

Table 5: Effect of Architectures on network predictability for output-1

N1	N2	N3	N_TF	r²	N	n
1	0	0	1	0.945	1	210
2	0	0	1	0.987	2	190
3	0	0	1	0.910	3	110
4	0	0	1	0.913	4	110
5	0	0	1	0.900	5	100
8	0	0	1	0.976	8	90
1	1	0	1	0.872	2	120
1	2	0	1	0.854	3	110
1	3	0	1	0.921	4	90
2	1	0	1	0.913	3	80
3	1	0	1	0.956	4	80
2	6	0	1	0.964	8	60
2	7	0	1	0.978	9	50
2	2	1	1	0.988	5	50
2	7	2	1	0.943	11	50
2	1	2	1	0.975	5	50
2	7	4	1	0.970	9	50
5	2	1	1	0.999	8	70
6	2	0	1	0.999	8	50

Thus, sample size plays one critical role in over-fitting the 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 6: Effect of Sample size on network predictability for output-1

N1	N2	N3	N_TF	r²	N	n
6	2	0	1	0.923	8	10
6	2	0	1	0.935	8	15
6	2	0	1	0.910	8	20
6	2	0	1	0.953	8	25
6	2	0	1	0.897	8	30
6	2	0	1	0.986	8	35
6	2	0	1	0.991	8	40
6	2	0	1	0.994	8	45
6	2	0	1	0.999	8	50
6	2	0	1	0.999	8	60
6	2	0	1	0.999	8	80
6	2	0	1	0.998	8	100
6	2	0	1	0.982	8	130
6	2	0	1	0.912	8	180

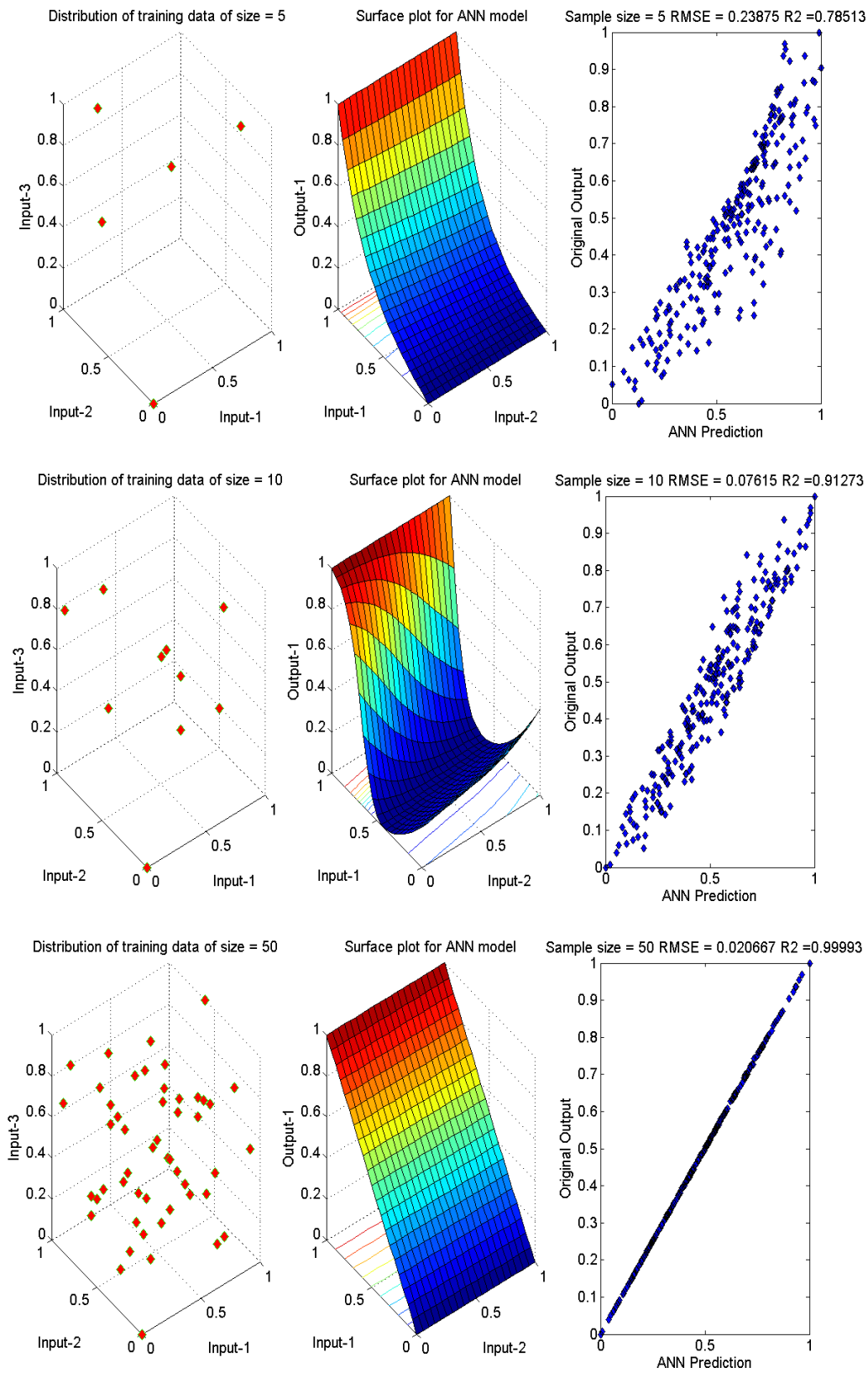


Fig. 7: Evolution of ANN surface for the architecture [3-6-2-0-1] for output -1

IV. Effect of transfer function.

In this study, three different architectures with completely different number of hidden layers have been considered and they are allowed to be trained with both tan sigmoidal and log sigmoidal activation functions. The results of the same are presented in Table 7. From these results no particular transfer function evolves as a clear winner thereby suggesting that both the log and tan sigmoidal activation functions should be explored prior to training the networks.

V. Process optimization using ANN surrogates

Clearly these *results justify the need for a parameter free ANN surrogate building algorithm which can intelligently devise the architecture along with simultaneous determination of sample size and transfer function such that, the network predicts with maximum accuracy without being over-fitted.* However, with the help of the laborious hit and trial routine, two architecture with appropriate sample size and transfer function were selected for emulating the output-1 and output-2 of the sintering model. These surrogates with their credentials are presented in Table 7.

Table 7: Effect of Activation function on network predictability for output-1

N1	N2	N3	N_TF	r ²	N	N
6	2	0	1	0.999	8	50
6	2	0	2	0.878	8	150
5	2	1	1	0.999	8	70
5	2	1	2	0.913	8	100
2	7	0	1	0.978	9	50
2	7	0	2	0.986	9	80

Table 8: ANN surrogates for Sintering model

	Architecture (Inputs-N1- N2-N3- outputs)	N_TF	N	r ²	Sample size	Total function calls
Output 1	3-6-2-0-1	1	8	0.9999	50	190 + 200 (training + validation set) = 390
Output 2	3-5-4-1-1	1	10	0.9928	190	

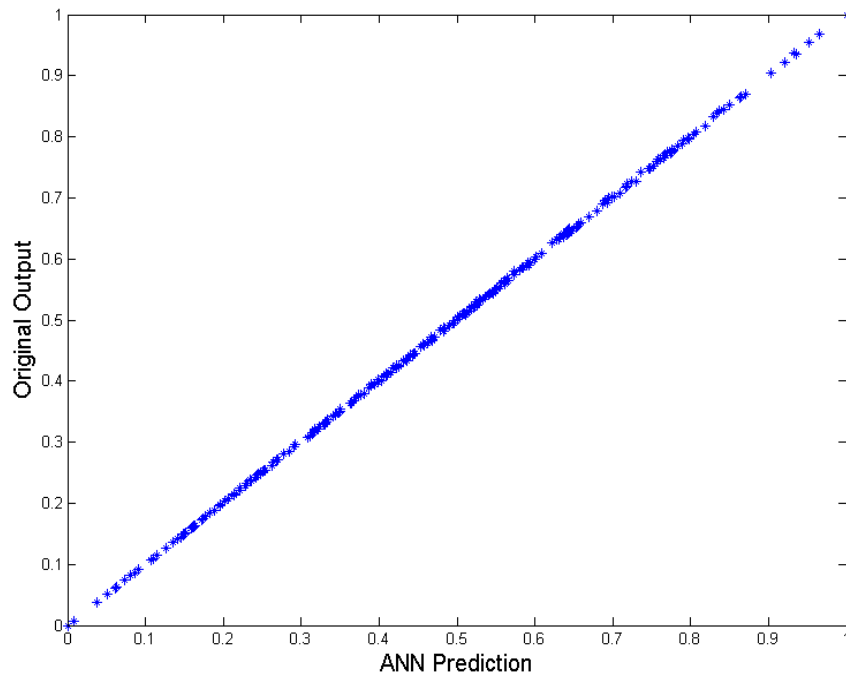


Fig. 8: Parity plot for Output 1 using the architecture = 3-6-2-0-1 with $R^2 = 0.99993$ obtained using HC sampling technique

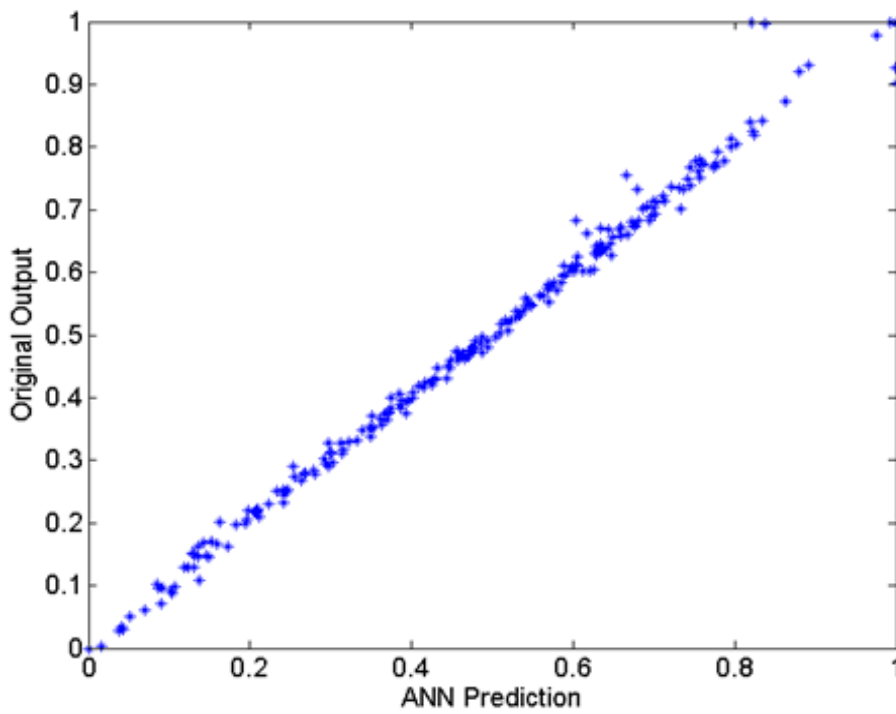


Fig. 9: Parity plot for Output 2 using the architecture = 3-5-4-1-1 with $R^2 = 0.993$ obtained using HC based technique

The corresponding parity plots are presented in Figs 8 and 9. The following results can be drawn out of the observations in Table 8:

1. The complete replacement of the original model with ANN surrogate in the optimization algorithm resulted in saving an enormous 70 % of the function evaluations $\{[(1300 - 390)/1300] * 100\}$ thereby resulting in nearly 4 times $\{1300/390\}$ faster optimization.
2. Although for a safer side, the validation set is considered here to be of 200 points, the optimization run can be made much faster by considering lower number of validation points. With respect to training set alone, the proposed algorithm resulted in multilayered architectures which emulate the original model with an average accuracy of 99 % and performed the optimization run nearly 7 ($\sim 1300/190$) times faster.
3. The emergence of multi layered networks as results of the extensive study on ANNs, justifies the need for exploring the potential of multi-layered perceptron networks. This result justifies the elimination of the assumption based on heuristic to consider only single hidden layered architectures.

The ANN surrogate models obtained for both the outputs are then allowed to replace the original sintering 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 10. For the sake of obtaining a clear cut qualitative estimation of the result observed in Fig 10, the inputs of the PO points obtained using the ANN surrogate based optimization are sent to the original sintering model and the corresponding outputs are compared to measure the RMSE values. An average RMSE (averaged over output 1 and 2) of 0.05 was obtained.

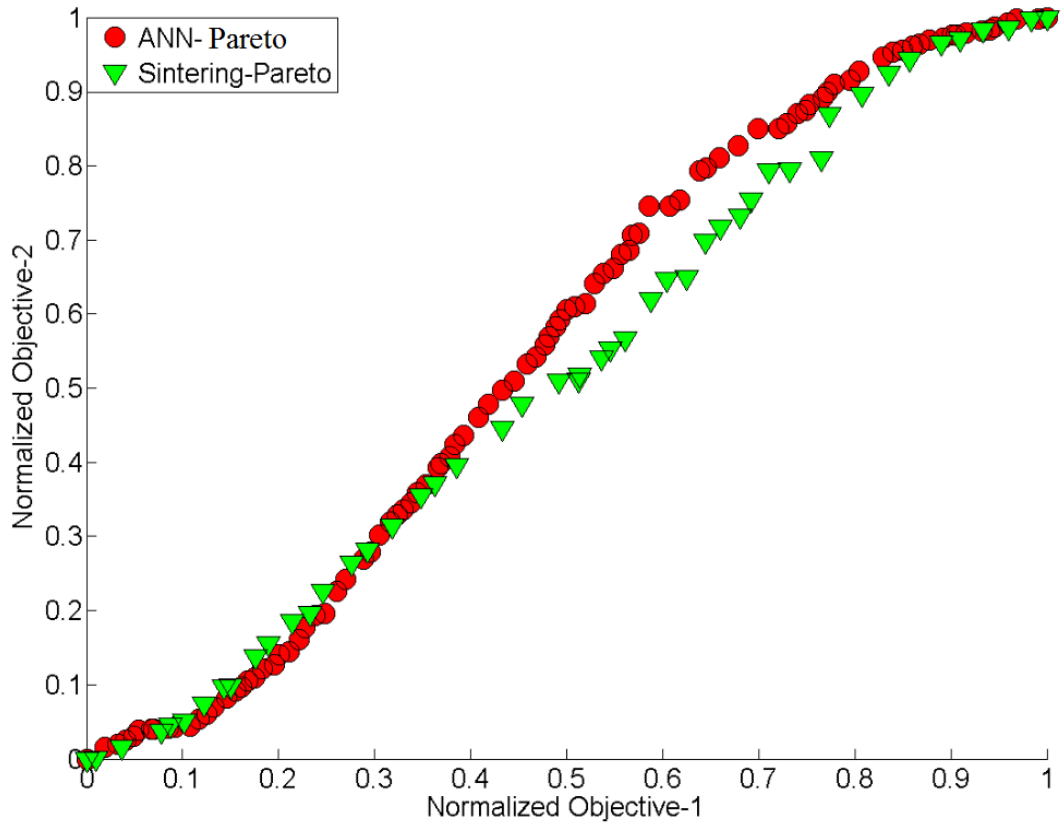


Fig. 10: PO front comparison of optimization using ANN surrogate built by HC based sampling method and original first principle Sintering model

A significant work reported previously in the literature [28] suggested using the combination of ANN surrogates along with the original model in NSGA II algorithm while optimizing the sintering model. This hybrid based optimization resulted in nearly 45-60% savings in computational time. Since the ANN surrogates were built only during the optimization runs [28], the resultant ANN surrogates, despite predicting with higher accuracies, emulate the original model in only a specified zone guided by the optimizer in the feasible search space. However, the proposed surrogate models in this work, emulate the first principle model over the

entire domain specified by the training set. Thus the ANN models in the current work are generic enough as opposed to the simpler models built in [28] which are highly specific to only optimization algorithm. However, the fact that these models are not the best possible ANN surrogate models for Sintering process cannot be denied. The work which essentially used the hybrid ANN – GA based optimization resulted in 700 function calls thereby concurring to the fact that the surrogate models obtained in the current work, despite being superior in terms of parsimonious predictions, are nearly 2 times faster than the former ones. All the simulations were carried out in Intel(R) Xeon(R) CPU E5-2690 0 @ 2.90GHz (2 processors) 128 GB RAM machine.

Chapter 4

Conclusion

The author in this work has presented a comprehensive research over design of ANN surrogate models for enabling the optimization of complex industrial 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. 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 ANN surrogate building process has been studied. It was found that the parameters viz. (a) architecture of ANN, (b) sample size required by the ANN, (c) maximum possible accuracy of prediction, (d) a robust sampling plan and (e) transfer function choice for node activation 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 unseen data the multi layered perceptron networks might overpower the single layered network which have enjoyed the monopoly till the present time due to the heuristic based assumptions. Therefore, it has been suggested that there is a need for a novel parameter free ANN surrogate building algorithm, which can estimate all the parameters automatically, thus

eliminating the human intervention ANN design. The prime objective of this study is to understand the functioning of the ANNs and lay out a blue print for the intelligent design of Neural networks. The ANN surrogate models are utilized to emulate a complicated nonlinear sintering model used for successful operation of the blast furnace in the steel plants. The results of the surrogate based optimization revealed that the surrogate based optimization methods were 4 to 7 times faster than the conventional method. The ANN-surrogate based optimization reduced the function evaluations by a dramatic 70% clearly making way for real time optimization of the complex industrial sintering model.

Future Work

1. To develop a novel parameter free ANN surrogate building algorithm for intelligent design of neural networks along with simultaneous estimation of parameters such as sample size, sample plan and transfer function to enable it to emulate with maximum accuracy without being over-fitted.
2. To apply the proposed surrogate building algorithms to build ANN 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 ANN models for an experimental setup and ensure the successful working of the proposed ANN surrogate building algorithm with experimental setups.
4. To apply the surrogate building algorithms to construct recurrent neural networks to emulate dynamic systems such as those of systems biology and several other real time experimental and industrial models.
5. To successfully eliminate the entire human intervention and heuristic based inputs in ANN design and implementation with the help of the proposed parameter free 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 and K. Mitra, “Kriging surrogate based multi-objective optimization of bulk vinyl acetate polymerization with branching”, *Mat. and Man. Proc., Genetic Algorithms special issue*, vol. 30, pp 394-402, 2015.
- [3] K. Deb, “Multi-objective Optimization using Evolutionary Algorithm”, 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 coarse-to-fine grain modelling”, In *Proceedings of the PPSN Workshop on Evolutionary Multi-objective Optimization*, 2002.
- [6] B Yegnanarayana, “Artificial neural networks for pattern recognition”, *Sadhana* vol. 19, no.2, pp 189-238, 1994.
- [7] R. Esmaeili and M. R. Dashtbayazi, “Modelling and optimization for microstructural properties of Al/SiC nanocomposite by artificial neural network and genetic algorithm”, *Exp. Sys. with App.*, vol. 41, pp. 5817-5831, 2014.
- [8] M. Badel, S. Angorani and M. S. Panahi, “The application of median indicator

kriging and neural network in modeling mixed population in an iron ore deposit”, *Comp. and geosci.*, vol. 37, pp. 530 -540, 2011.

[9] H. Karimi and M. Ghaedi, “Application of artificial neural network and genetic algorithm to modelling and optimization of removal of methylene blue using activated carbon”, *Jour. of Ind. and Engg. Chem.*, vol. 20, pp. 2471 – 2476, 2014.

[10] D. M. Himmelblau, “Applications of artificial neural networks in chemical engineering”, *Kor. Jour. of Chem. Engg.*, vol. 17, no. 4, pp. 373-392, 2000.

[11] E. Betiku and A. E. Taiwo, “Modelling and optimization of bioethanol production from breadfruit starch hydrolyzate vis-_a-vis response surface methodology and artificial neural network”, *Ren. Ener.*, vol. 74, pp. 87 – 94, 2015.

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

[13] D. R. Jones, “A taxonomy of global optimization methods based on response surfaces”, *Jour. of Glob. Opt.*, vol. 21, pp 345-383, 2001.

[14] R. H. Myers, A. I. Khuri and W. H. Carter, “Response surface methodologies”, *Technometrics*, vol. 31, no. 2, pp. 137-157, 1989.

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

[16] 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.

[17] 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.

[18] I. J. Forrester, A. Sóbester and A. J. Keane, “Engineering Design via Surrogate Modelling, A practical guide”, Wiley, 2008.

- [19] M. D. Morris, “Factorial sampling plans for preliminary computational experiments”, *Technometrics*, vol. 33, no. 2, pp. 161–174, 1991.
- [20] B. Tang “Orthogonal array-based latin hypercubes”, *Jour. of the Amer. Statis. Asso.*, vol. 88, pp. 1392–1397, 1993.
- [21] 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.
- [22] M. D. Morris and T. J. Mitchell, “Exploratory designs for computational experiments”, *Jour. of Statis. Plan. and Inf.*”, vol. 43, pp. 381–402, 1995.
- [23] 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.
- [24] 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
- [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] M. E. Johnson, L. M. Moore and D. Ylvisaker, “Minimax and maximin distance designs”, *Jour. of Statis. Plan. and Inf.*, vol. 26, pp 131–148, 1990.
- [28] K. Mitra, “Evolutionary surrogate optimization of an industrial sintering process”, *Mat. and Man. Proc.*, vol. 28, no. 7, pp. 768-775, 2013.

[29] N. K. Nath and K. Mitra, "Mathematical modeling and optimization of two-layer sintering process for sinter quality and fuel efficiency using genetic algorithm", *Mat. and Man. Proc.*, vol. 20, no. 3, pp. 335-349, 2005.