European Journal of Operational Research 264 (2018) 294–309

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Computational Intelligence & Information Management

TRANSFORM-ANN for online optimization of complex industrial processes: Casting process as case study

Srinivas Soumitri Miriyala[a], Venkat Subramanian[b], Kishalay Mitra[a,*]

[a] Department of Chemical Engineering, Indian Institute of Technology Hyderabad, Kandi, Sangareddy Telangana 502285, India
[b] Department of Chemical Engineering, University of Washington, 1410 NE Campus Parkway, Seattle, WA 98195, USA

ARTICLE INFO

Article history:
Received 17 July 2016
Accepted 11 May 2017
Available online 17 May 2017

Keywords:
Artificial Intelligence
Multiple objective programming
Neural Networks
Online optimization
Surrogate models

ABSTRACT

Artificial Neural Networks (ANNs) are well known for their credible ability to capture non-linear trends in scientific data. However, the heuristic nature of estimation of parameters associated with ANNs has prevented their evolution into efficient surrogate models. Further, the dearth of optimal training size estimation algorithms for the data greedy ANNs resulted in their overfitting. Therefore, through this work, we aim to contribute a novel ANN building algorithm called TRANSFORM aimed at simultaneous and optimal estimation of ANN architecture, training size and transfer function. TRANSFORM is integrated with three standalone Sobol sampling based training size determination algorithms which incorporate the concepts of hypercube sampling and optimal space filling. TRANSFORM was used to construct ANN surrogates for a highly non-linear industrially validated continuous casting model from steel plant. Multiobjective optimization of casting model to ensure maximum productivity, maximum energy saving and minimum operational cost was performed by ANN assisted Non-dominated Sorting Genetic Algorithms (NSGA-II). The surrogate assisted optimization was found to be 13 times faster than conventional optimization, leading to its online implementation. Simple operator's rules were deciphered from the optimal solutions using Pareto front characterization and *K*-means clustering for optimal functioning of casting plant. Comprehensive studies on (a) computational time comparisons between proposed training size estimation algorithms and (b) predictability comparisons between constructed ANNs and state of art statistical models, Kriging Interpolators adds to the other highlights of this work. TRANSFORM takes physics based model as the only input and provides parsimonious ANNs as outputs, making it generic across all scientific domains.

1. Introduction

Artificial Neural Networks (ANN), one of the efficient data modelling techniques, are finding extensive real world applications including operational research (Sermpinis, Theofilatos, Karathanasopoulos, Georgopoulos, & Dunis, 2013). Well known for their ability to capture nonlinear dynamics of complex data (Barrow & Kourentzes, 2016; Denton & Hung, 1996; Sexton, Dorsey, & Johnson, 1999; Kamini, Vadlamani, Prinzie, & Van denPoel, 2014), ANNs are advantageous over similar class of technologies such as Support Vector Machines (SVMs) and Response Surface Methods (RSMs). SVMs, like ANNs, implement supervised learning techniques to classify the given data. Belonging to the class of machine learning algorithms this method works by creating partition between data points such that the distance from the closest training point to the partition is maximised. Primarily designed for linear classification, this method is also extended for non-linear classification by transforming the working domain to a region of higher dimensions (Sermpinis, Theofilatos, Karathanasopoulos, Georgopoulos, & Dunis, 2017). RSMs on the other hand are statistical models, which try to regress lower order (commonly, second order) polynomials to build data based relationship between explanatory and response variables. Here, the optimizer finds the optimal response of objective function through a sequence of designed experiments. This method has found immense applications in engineering and operational research due to its extreme simplicity and ease in implementation (Shi, Shang, Liu, & Zuo, 2014). However, several disadvantages which creep in due to the heuristic estimation of parameters governing the neural networks create suspicion in their predictability (Wong & Hsu, 2006). This impression is further solidified with the trial and error based determination of training sample size without implementing a formal design of experiments or sample plan. Lack of an intelligent framework for ANN construction has put them in the back step behind robust statistical data modellers such as Kriging Interpolators (Mogilicharla, Mittal, Majumdar, & Mitra, 2015).

* Corresponding author.
E-mail addresses: kishalay@iith.ac.in, kishalay.mitra@gmail.com (K. Mitra).

http://dx.doi.org/10.1016/j.ejor.2017.05.026
0377-2217/

The motivation for the current work is to contribute in operational research a novel ANN building algorithm called TRANSFORM (*TR*ade-off between *A*ccuracy, *N*odes and **S**ample size *FOR M*eta-modelling). This intelligent framework, capable of estimating most of the ANN related parameters, thereby making it parameter free, determines the best configuration and optimal training sample size, simultaneously. While doing this, TRANSFORM ensures a balance between the aspects of over-fitting and prediction accuracy. Further, three novel sample size determination techniques designed using two potential concepts: hypercube (HC) sampling and space filling based single objective optimization (SOOP) formulation, are presented. TRANSFORM is fast enough to be implemented in real time and generic enough to be applied to any physics based model without constraints on dimensionality.

In this work, we aim to explore the scope of TRANSFORM-ANNs (TRANSFORM based ANNs) as surrogate model for online implementation of computationally expensive optimization processes. As an example, we considered a 7-input-2-output industrially validated highly non-linear continuous casting (*concast*) model from steel plants. The main reason behind our decision to use the *concast* model is to demonstrate the potential of TRANSFORM to construct ANNs capable of emulating the complex physics based models. Physics based models are highly robust and accurate owing to rigorous implementation of scientific principles behind the considered phenomena. Often this leads to the comprisal of several Differential Algebraic Equations (DAE), thereby increasing the computational expense in simulating the model (Mogilicharla, Chugh, Majumdar, & Mitra, 2014; Olaf, Barth, Freisleben, & Grauer, 2005; Ruud, Driessen, Hamers, & Hertog, 2005; Uğur, Karasözen, Schäfer, & Yapıcı, 2008). *Concast* is one such model containing a mix of partial different equations, ordinary differential equations and several algebraic empirical correlations, whose details are described in subsequent sections of this paper.

Optimal running of casting plant in steel industries is one of the prime targets of production plant managers. Incorporating the decisions of management, which are mainly driven by the volatile nature of markets, requires the plant wide optimization and control to be implemented in online fashion. In online optimization, the optimizer works in consolidation with a robust controller, thus together forming an effective Advance Process Control (APC) unit. Here, considering the practical changes to be implemented, the optimizer often solves a multiobjective optimization problem (MOOP) in real time. The optimal solutions are then provided to the controller as the set point. However, the working of APC unit in real time requires the inherent model to be computationally efficient. Thus, owing to the complex nature of *concast*, its optimization is always confined to the offline mode.

This enables the implementation of meta-models, also known as surrogate models, which are trained to emulate the physics based model accurately. These surrogates then replace the time consuming physics based model during their optimization to generate simulations fast enough to run it in online mode (Jin, 2011; Tabatabaei, Hakanen, Hartikainen, Miettinen, & Sindhya, 2015).

We thus implemented TRANSFORM in conjunction with the proposed sample size determination techniques to construct parsimonious ANNs capable of predicting *concast* with maximum accuracy. Before moving on to the ANN surrogate assisted online implementation, we optimized the casting model without surrogate using both classical and evolutionary optimizers. Further, a comparative performances between the TRANSFORM based ANN (TRANSFORM-ANN) and Kriging Interpolators is also presented. As it turns out the TRANFORM-ANN outperformed Kriging Interpolators in terms of sample size requirement and statistical accuracy. Significant reduction in computational time due to the implementation of TRANSFORM-ANN assisted optimization of *concast* lead to its online implementation.

Finally, we present a set of operator's rules using the concept of Pareto characterization and *K*-means clustering algorithm. These rules draw the mapping from complex mathematical realization of optimization studies such as Pareto to a set of simple linguistic instructions aiding the ground operators to enable optimal functioning of casting plant.

The rest of the paper is organized as follows – we first present the literature survey of several recent contributions in the field of research which forms the central theme of this paper. This is followed by the continuous casting model description and formulation of its optimization problem. Proposed algorithm TRANSFORM and novel size estimation algorithms are discussed next. The rest of the paper is devoted to surrogate building, comprehensive comparison studies, ANN surrogate assisted optimization, scope for online implementation and discussions on Pareto characterization, all summed up in the results section following which the novelty in current contribution is briefly summarized in conclusions.

2. Literature review

ANNs being potential classifiers can serve as ideal candidates for meta-modelling. However, they suffer with major disadvantages such as those listed below:

- Heuristic based design of architecture.
- No proper guidelines for choosing the transfer function of network.
- No sample plan and measure of optimal sample size.
- ANN model often gets over-fitted.

These drawbacks not only degrade the performance of ANNs but also prevent them from qualifying as potential surrogates for optimization. The objective of this work is to contribute an efficient ANN building algorithm capable of solving all the aforementioned problems simultaneously, within short time frame to ensure its streamlining with online optimization. For this purpose, we present TRANSFORM with two novel sample size estimation techniques. Our proposed methods for architecture estimation and size determination are not implemented yet. However, Dua (2010), reported solving a mixed integer nonlinear programming problem (MINLP) to determine the optimal configuration of ANN. The algorithm being robust, does not address the other issues of ANNs and further, the computational complexity involved in solving MINLP cannot be ignored (Dua, 2010). In another work, configuration of ANN was obtained by solving an optimization problem with the objective of maximizing the prediction accuracy of each of the outputs (Boithias, Mankibi, & Michel, 2012). If the number of outputs are large (say >3), the proposed method might take a huge time to converge (Deb, 2001). Carvalho, Ramos, and Chaves (2011) used the weighted combination of training error, validation error as a single objective function to resolve the problem of architecture design (Carvalho et al., 2011). However, a more generic metric such as the Akaike Information Criteria (AIC) (Akaike, 1971), representing parsimonious nature of the network, might have served as more suitable objective function. Moreover, weighted sum approach has been shown to fail for generating well-distributed Pareto optimal (PO) points while solving MOOPs (Deb, 2001).

The problem of sample size determination (SSD) for black box models in general, is broadly classified into two categories – adaptive and sequential sampling (Eason & Cremaschi, 2014). Sequential sampling is similar to forward marching problem where points are sampled sequentially until the surrogate is trained with desired accuracy. Many researchers contributed in this area using methodologies such as Delaunay triangulations (Davis & Ierapetritou, 2010), Voronoi tessellations (Crombecq, 2011), optimization based approaches and Monte Carlo based random sampling methods (Crombecq, 2011). On the other hand, the adaptive sampling

methods focus on sampling exclusively in the regions which exhibit extreme non-linearity. A classic contribution in this regard is by Jones 2001 for Kriging Interpolation (KI) surrogate models, a Gaussian regression based statistical procedure and the current state of art in surrogate literature (Jones, 2001). The advantage with this stochastic technique, is the availability of standard measure of prediction accuracy at any interpolated point. The sampling strategy to sample a new point is such that the improvement in the prediction error is maximized. Although this makes KI a robust technique, concern still lies with its ability to uniformly sample the input domain (Müller & Shoemaker, 2014). Of other adaptive sampling techniques reported in literature, the most prominent one is the technique which samples by minimizing the prediction variance (Kleijnen, 2016). A recent contribution reported three novel iterative methods for SSD (Eason & Cremaschi, 2014). First one, based on incremental Latin Hypercube Sampling (i-LHS) sampling strategy, fails to preserve the sample points utilized in previous iterations while the second method requires creation of large number of random subsamples for variance estimation. The third method is a combination of first and second.

SUMO toolbox built using MATLAB (Gorissen, Couckuyt, Demeester, Dhaene, & Crombecq, 2010) is one of the efforts towards enabling usage of various surrogate models under one platform. This package has wide applicability as it effectively deals with almost all the aforementioned surrogate techniques. Similarly, Müller and Shoemaker (2014) presented a framework, where the choice of surrogates and problem of training sample size is articulated. However, the SUMO toolbox does not provide a robust sample size determination algorithm which can avoid over-fitting, while the latter tries to utilize the combinatorial power of heterogeneous surrogates along with random sampling strategy.

3. Continuous casting model and its optimization problem formulation

Cost and energy effective process of continuous casting has enabled a steep rise in implementation of this process by the steelmakers across the globe. In this process, molten steel coming from a blast furnace is first cooled in a water cooled copper cast, held by a steel jacket. Subsequently the melt is further cooled by moving it across a series of seven water sprays. This ensures progressive cooling of the liquid steel within the desired temperature range. Recently, the casting mill is integrated with rolling mill to enable the steel makers to produce flat and thin strips of steel in cost effective manner. The casted steel bar is further sent to the hot mill where it is hot rolled to produce flat strips of steel. Continuous endeavour is there to increase the productivity of the steel using this process of casting into thin sheets.

Mitra and Ghosh (2008) suggested that increasing the caster speed could be one possible route by which the productivity can be improved (Mitra & Ghosh, 2008). However, increasing the caster speed results in severe fluctuations in the bulging profile of steel, thereby increasing the chances of deformations in the slab (Mitra & Ghosh, 2008). The fluctuations in the bulging profile can be suppressed by increasing the cooling rate of the slab. But random increase in flowrates of the series of sprays will reduce the exit temperature of the slab drastically, before it is sent to an in line furnace leading to the roll mill. In accordance with the effective industrial operating conditions, the temperature of the incoming steel slab into the reheating furnace should be maintained as high as possible to save upon the fuel and energy consumption. Thus, on the basis of these conflicting industrial requirements, a multiobjective optimization problem (MOOP-1) was formulated where the objectives considered were considered as minimizing the total bulging, maximizing the exit temperature and maximizing the castor speed, simultaneously. The flow rates of series of sprays (starting from Spray 2 to 6) and caster speed act as the decision variables (see Table 1). Due to the industrial constraint, the flowrate at nozzle Spray 1 was maintained constant.

Table 1
MOOP-1 formulation of continuous casting model.

Objective functions	Decision variables
Maximize slab exit temperature (T)	$S_2^L \leq \text{Spray}_2$ flowrate $\leq S_2^U$
Maximize casting speed	$S_3^L \leq \text{Spray}_3$ flowrate $\leq S_3^U$
Minimize bulging (B)	$S_4^L \leq \text{Spray}_4$ flowrate $\leq S_4^U$
	$S_5^L \leq \text{Spray}_5$ flowrate $\leq S_5^U$
	$S_6^L \leq \text{Spray}_6$ flowrate $\leq S_6^U$
	$S_7^L \leq \text{Spray}_7$ flowrate $\leq S_7^U$
	LB $\leq$ caster speed $\leq$ UB

In order to solve MOOP-1, a robust continuous casting model was built to map the 6 spray flowrates and caster speed with outlet temperature and an empirical measure of bulging in steel slab. The model involves partial differential equations for heat transfer coupled with empirical models for bulging phenomena. It is solved using control volume method with tri-diagonal matrix algorithm. The details of this model formulation are presented in Appendix A of supplementary file.

4. TRANSFORM: a novel parameter free ANN surrogate building algorithm

4.1. Idea behind TRANSFORM

The output of ANN is a summation across several layers of weighted inputs and biases activated using a specific function called transfer function. Apart from input and output layers, ANNs consists of a number of hidden layers made up of parallel processing units called nodes which provide sufficient handles for capturing nonlinear dynamics. Representing each hidden layer of an ANN as a hyper-plane which linearly segregates the sampled data (Hagen, Demuth, & Beale, 2002; Haykin, 1994), it is evident that multiple hidden layers are required for classifying linearly inseparable data. Since the nature of the training set is not known *a priori*, the exploration of architectures cannot be restricted to single layered networks. Thus for enhancing the classification ability and thereby prediction accuracy, ANNs need to have large number of hidden layers with sufficient number of nodes in each layer. However, consideration of a large arbitrary number of hidden layers is also not appropriate as it might lead to an outburst of parameters (e.g., weights and biases) causing over-fitting. This enables us to achieve a trade-off between prediction accuracy and total number of nodes in the network.

For a given architecture, initially the increment in number of training samples enhances its predictability. Although the extent of improvement also depends significantly on selected architecture and the underlying model, the aforementioned fact is justified under the purview of the basis on which adaptive and sequential sampling methods were developed. However, providing a large sample set over-fits the considered ANN model. Thus a trade-off between accuracy of predictions and training sample size can be inferred easily. The interdependency of accuracy, sample size and total nodes, results in third trade off which is between the sample size and total nodes of the network. This ideology pictorially represented in Fig. 1, leads to the formulation TRANSFORM.

4.2. Problem formulation

Given the conflicting nature of accuracy, sample size and total nodes (as shown in Fig. 1), a multiobjective optimization problem (MOOP-2) has been formulated with the objectives of finding an

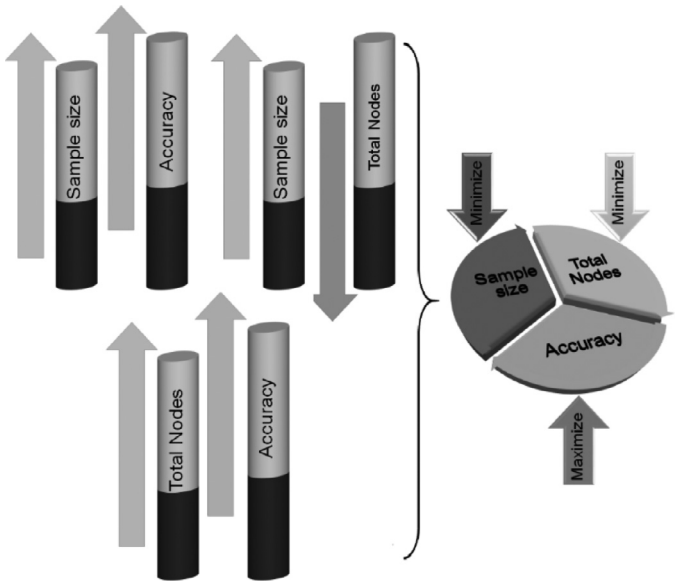

Fig. 1. Idea behind formulation of TRANSFORM.

Table 2
Multiobjective optimization problem (MOOP-2) formulation for TRANSFORM.

Objectives	Decision variables
Maximize accuracy in terms of R^2	Nodes in hidden layer 1: $1 \le N1 \le 8$
Minimize total number of nodes, N	Nodes in hidden layer 2: $0 \le N2 \le 7$
Minimize total sample size, n	Nodes in hidden layer 3: $0 \le N3 \le 7$
	Transfer function choice: N_TF 0 or 1

architecture having maximum prediction accuracy along with minimal associated nodes and minimum training points (as shown in Table 2). Exploration of multiple hidden layers was limited up to four layered architectures (3 hidden and one output layer), as further increase in number of layers would lead towards the dangers of over-fitting. The first three decision variables of MOOP-2 representing the number of nodes in the three hidden layers were varied from 1 to 8, 0 to 7 and 0 to 7, respectively, while, the fourth decision variable representing the transfer function choice was binary. A choice of 0 and 1 for the fourth decision variable, would enable the activation using log sigmoidal transfer function and tan sigmoidal function, respectively. The inclusion of 0 as the lower limit for the bounds of the second and third decision variables was to ensure the emergence of single and two hidden layered networks as candidates for optimization. An architecture presented as 7-3-4-2-1 signifies 7 inputs and 1 output as the first and last entry in the numeric expression, whereas there are three hidden layers in the architecture with 3, 4 and 2 number of nodes in the first, second and third hidden layers, respectively. The decision variable set describing the same architecture will be $\{N1 = 3, N2 = 4, N3 = 2, N\_TF = 0\}$.

The integral nature of the decision variables and nonlinear objective functions in the proposed MOOP-2 formulation gives it the status of Mixed Integer Non-Linear Programming Problem (MINLP) which are known to be extremely challenging to solve using the conventional optimization solvers. Further lack of gradient information and multi-objective nature of MINLP problem prevents the use of conventional classical methods to solve MOOP-2 because of which we implemented binary coded non dominated sorting Genetic Algorithm, NSGA II (Deb, 2001) in the current work to solve MOOP-2.

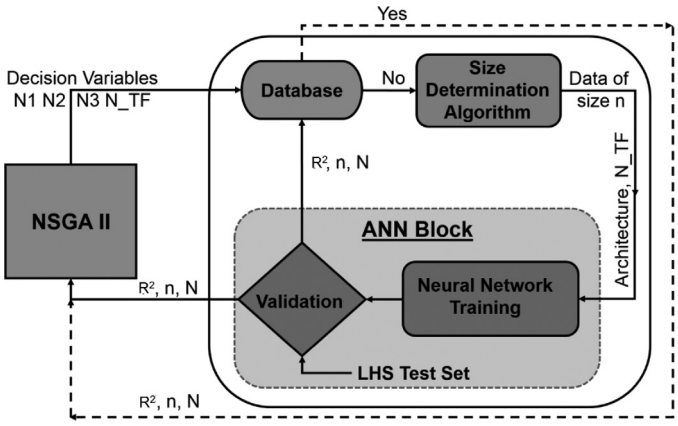

Fig. 2. TRANSFORM: parameter free ANN surrogate building algorithm.

4.3. Information flow in TRANSFORM

The decision variable set obtained from the optimizer is sent as input into the sample size determination algorithm (described later), which would determine the optimal size of training data. Once the training data is created, a set of 200 input–output points from the original complex model are further generated within the prescribed input bounds, using the best LHS plan (Forrester, Sobester, & Keane, 2008). As it will be described later in the section on sample size determination, the training set is obtained using a sampling scheme completely different from LHS. Thus, the 200 point test set generated exclusively for validation is entirely different from the data set used for training. The architecture, transfer function choice and sample set for training and validation are then sent into the multiple-input–single-output (MISO) ANN code built to enable parallel computing. The ANN code would then train the network and evaluate the validation accuracy in terms of a statistical measure called R^2 (see Eqs. (1)–(3)) which along with total nodes N and the sample size required for training n, is sent back to optimizer. The objectives along with architecture are saved in a database to check redundancy, thereby making TRANSFORM faster. The flow of TRANSFORM is depicted schematically in Fig. 2.

$$R^2 = \left(\frac{\text{cov}\ (y, \hat{y})}{\sqrt{\text{var}(y)\text{var}(\hat{y})}} \right)^2 \tag{1}$$

$$\text{cov}(y, \hat{y}) = n \sum_{i=0}^{n} y^{(i)} \hat{y}^{(i)} - \sum_{i=0}^{n} \hat{y}^{(i)} \sum_{i=0}^{n} y^{(i)} \tag{2}$$

$$\text{var}(y) = n \sum_{i=0}^{n} y^{(i)2} - \left(\sum_{i=0}^{n} y^{(i)} \right)^2 \tag{3}$$

Here, y is the original output and $\hat{y}$ is the predicted output.

5. Sample size determination algorithm

5.1. Sampling plan

Chi, Mascagni, and Warnock (2005) reported about the superiority of LHS plans over the low discrepancy sampling plans such as Sobol and Halton for the large dimensional models (Chi et al., 2005). However, the fact that LHS plan always generates a different set of points when prompted for a different sample size, discarding the previously generated sample points, cannot be neglected. This quality of sampling schemes to preserve and utilize the previously generated sample points, is of high prominence from the

Table 3
Quantitative analysis of sampling plan for 500 sample set in different input spaces.

Sampling plan and sample size	Number of inputs	PHI-LHS	PHI-Sobol
500	3	273.454	254.342
500	10	114.91	114.54
500	20	54.9	54.8

view point of saving computational time. Sampling plans such as Sobol and Halton (Diwekar & Kalagnanam, 1997) are equipped with this quality apart from being extremely fast when compared with the best-LHS plan (Forrester et al., 2008). The effectiveness of space filling can be quantitatively measured using the PHI metric (Forrester et al., 2008; Morris & Mitchell, 1995).

The PHI metric is reported to be one of the prominent measures of uniformity of sampling plans. Lower the PHI metric of the sampling plan, better is its uniform space filling ability. Given a set of N points $\{\mathbf{X}\} \subset \mathbb{C}^k$, where $\mathbb{C}^k$ is k dimensional unit cube, define the norm function $\mathbb{D}$ as follows:

$$\mathbb{D}\left(\mathbf{X_i}, \mathbf{X_j}\right) = \left\|\mathbf{X_i} - \mathbf{X_j}\right\|_2 = \left(\sum_{q=1}^{k}\left(\mathbf{X_{qi}} - \mathbf{X_{qj}}\right)^2\right)^{1/2}$$
$$\forall \quad i, j = 1, 2, \ldots N \mid i \neq j \tag{4}$$

Let $\mathcal{D}$ be the set of unique norm values $\mathbb{D}$ can take and let the cardinality of set $\mathcal{D} = \mathrm{M}$. We then define a counting function $\mathcal{T}(\mathcal{D}_m)$ as the number of $\{\mathbf{X_i}\ \mathbf{X_j}\}$ pairs which have same norm value. The PHI metric of the sampled set $\{\mathbf{X}\}$ is defined as,

$$\mathrm{PHI} = \left(\sum_{m=1}^{M}\frac{\mathcal{T}(\mathcal{D}_m)}{\mathcal{D}_m^2}\right)^2 \tag{5}$$

The PHI metric for both LHS and Sobol distributions with 500 sample points for 3, 10 and 20 dimensional input spaces has been measured and tabulated in Table 3. These results clearly show the efficiency of Sobol over LHS in terms of space filling.

5.2. Idea behind proposed SSD techniques

A robust model evaluation method known for preventing over-fitting, such as the K fold cross validation (Miriyala, Mittal, Majumdar, & Mitra, 2016), works by (i) dividing the given sample set P for training into K folds, (ii) training the model with data from all but 1 fold and validating it with the data from left over fold, (iii) identifying the model having minimum cross validation error, defined as the mean of K errors obtained by training the model in K different ways of step ii.

We propose three novel samples size determination algorithms, capable of preventing over-fitting without implementing the computationally expensive K times validation approach. These techniques, based on sequential sampling, are designed in such a way that they succeed in both the aspects of speed and accuracy. The principle behind these methods is to intelligently identify a validation set V out of the given sample set P such that, V provides a holistic representation of the input domain defined by P. Thus, training with the data points in the set $P\backslash V$ and validating the model with V, retains the significance of K fold cross validation based method. The proposed SSD techniques are dedicated to identify this validation set V.

5.3. Sample size determination based on optimization framework (SOOP)

The validation set V can be obtained such that it has maximum space filling ability among all the subsets of P of similar size. The size of this subset $V = n\_V$ is considered one thirds of the size of set P. A single objective optimization problem (SOOP) was formulated to identify V which has minimum PHI value (see Section 5.1). Due to the lack of gradient information, the SOOP formulation was solved using Genetic Algorithms (GA). This method is described sequentially in following steps:

Step 1. SSD starts with initial set P obtained from Sobol scheme.
Set $i = 1$. Cardinality of set P, $\mathrm{card}(P) = n$.
Iteration i starts
Step 2. Solve the SOOP for minimizing PHI and obtain the subset V from P.
Step 3. Define $T = P\backslash V = \{x : x \in P \text{ and } x \notin V\}$.
$\mathrm{card}(T) = n\_T$. Train ANN using T and obtain validation error ε_i.
Step 4. if $i = 1$
Go to step 6,
else
Calculate the slope ratio (SR), defined as:
$\mathrm{SR_i} = \frac{\mathrm{abs}(\varepsilon_i - \varepsilon_{i-1})}{(n_i - n_{i-1}) * \max(SR)}$
End
Step 5. Set $\propto = 0.01$ (user defined tolerance).
if $SR_i \leq \propto$
Final Sample size $n = n_i$ and terminate SSD.
else
Go to step 6.
end
Iteration i ends.
Step 6. Set $i = i + 1$ and go to step 2.

5.4. Sample size determination based on hypercube sampling (HC)

Step 1. SSD starts with initial set P obtained from Sobol scheme.
Set $i = 1$. Cardinality of set P, $\mathrm{card}(P) = n$.
Iteration i starts
Step 2. Divide the input space into of n_V (or greater) number of hyper-cubes of equal volume and randomly sample one data point from each to create V.
Step 3. Define $T = P\backslash V = \{x : x \in P \text{ and } x \notin V\}$.
$\mathrm{card}(T) = n\_T$. Train ANN using T and obtain validation error ε_i.
Step 4. if $i = 1$
Go to step 6,
else
Calculate the slope ratio (SR), defined as:
$\mathrm{SR}_i = \frac{\mathrm{abs}(\varepsilon_i - \varepsilon_{i-1})}{(n_i - n_{i-1}) * \max(SR)}$
end
Step 5. Set $\propto = 0.01$ (user defined tolerance).
if $SR_i \leq \propto$
Final Sample size $n = n_i$ and Terminate SSD.
else
Go to step 6.
end
Iteration i ends.
Step 6. Set $i = i + 1$ and go to step 2.

Although the SOOP based method provides a quicker alternative to the time consuming K-fold based method, it remains to be slower when brought into the realm of TRASNFORM. To facilitate faster selection of the n_V points out of n, which will form the validation set V, the input domain can be divided into smaller hypercubes of equal volumes of number $\geq n\_V$ and a random point is selected from each hyper-cube to form the validation set V. This will instil the space filling quality in the validation set in least possible

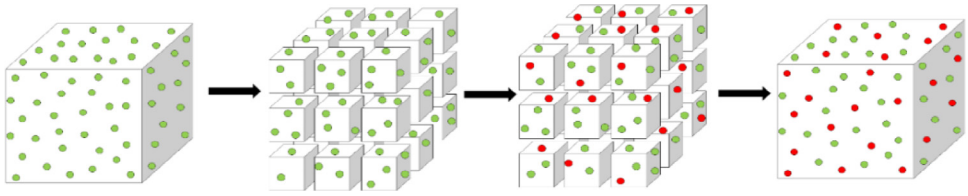

Fig. 3. HC Sampling technique for a three dimensional input space: lighter points in the leftmost cube indicate set *P*. This cube is split into smaller hyper-cubes and a representative from each hyper-cube (dark points) is randomly sampled, which collectively form *V*.

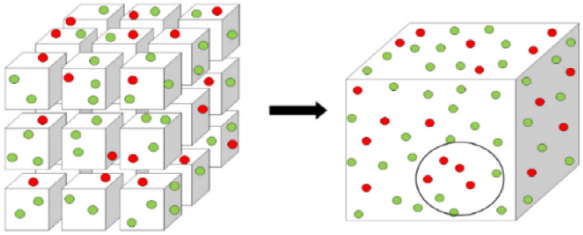

Fig. 4. Pictorial representation of problem associated with HC based SSD: random selection from each hyper-cube has led to the sampling of closer points-the encircled region depicts this.

computational time. The pictorial presentation of hypercube (HC) sampling technique is presented in Fig. 3.

The HC sampling based algorithm may face certain issues which are described below:

(a) After division into *n_V* hyper-cubes, some of them may remain empty, resulting in no selection of sample point from them.
(b) In case of high dimensional models, due to the random selection from each hyper-cube, there might be the cases where, although the points are selected from different adjacent hyper-cubes, they might still be closer to each other as depicted in Fig. 4.

To solve problem (a), we implemented the method where a progressive division of the input space is done, till exactly the *n_V* number of points are sampled. The minimal addition of computational burden because of this approach can be neglected when compared to the amount of time saved by implementing the HC sampling method. The problem (b) can be resolved in one way by taking mean of multiple (large number of) samples randomly drawn from each hyper-cube, but this will make the algorithm computationally intensive. In order to resolve this problem, we propose the following SSD technique where the goodness of both HC and SOOP based methods are combined

5.5. Sample size determination based on hybrid approach (HC+SOOP)

The validation set *V*, obtained within no time using the HC based method, is given as the initial guess in the population of GA for solving the SOOP based method to find the set having minimum PHI value. Since the initial guess is intelligently obtained using HC based method, the SOOP may converge quickly to the optimum and thus the number of populations and number of generations are kept low. This will, therefore, make the GA much faster when compared with the case where only SOOP formulation is implemented with random initial guess. Thus, this method is the best method out of all three SSD techniques as it has a theoretical justification for space filling and converges in very less time. However, HC based SSD technique remains to be fastest amongst all.

6. Kriging Interpolators

The standard Expected Improvement (EI) based sampling strategy to infill the sample points for training the KI model has been used. To start with, we sampled 10 data points using LHS sampling plan and constructed an initial KI model. Further number of sample points were found iteratively using EI approach, until KI predicted with desired accuracy. The principle and working of Kriging Interpolation along with the EI based sampling strategy is presented in detail in Supplementary text.

7. Results and discussions

This section has been segmented into subsections for better clarity and easy readability. Section 7.1 begins with the display of extent of non-linearity present in the considered casting model. The optimization formulation of casting model (MOOP-1) is then solved conventionally without surrogate using both evolutionary and classical methods of optimization, a brief comparison of which follows next. Section 7.2 presents time comparison between novel SSD techniques, results of TRANSFORM algorithm (MOOP-2) for constructing ANN surrogates and discussion on selection of ANN architecture to emulate the casting model while Section 7.3 provides the analysis of results obtained by TRANSFORM. Section 7.4 provides a comprehensive comparison of constructed ANN models and Kriging surrogate model in terms of accuracy and sample size required for training. In Section 7.5, the results of TRANSFORM-ANN assisted optimization of casting model, MOOP-1 are presented and compared with those obtained through conventional optimization as demonstrated in Section 7.1. Section 7.6 analyses the scope of practical implementation of online optimization of complex models using TRANSFORM, focussing on considered case study. Section 7.7 provides the results of Pareto characterization using *K*-means clustering method which provide with a set of simple operator's rules for optimal performance of casting industry. Section 7.8 describes the novelty in the current work in nutshell.

For surrogate model, inputs 1–6 are the six nozzle spray flowrates (see Table 1) in continuous casting model, while input 7 denotes the casting speed. Similarly, outputs 1 and 2 stand for the slab exit temperature and predicted value of bulging, respectively. Although similar results were obtained for both the outputs, to honour the space constraints, we present the results, for output 1 only. Results related to output 2 are included in the supplementary material. The proposed algorithms and codes are exclusively developed in MATLAB® (version 2015), without the use of any specific toolbox. All the simulations were run in Intel(R) Xeon(R) CPU E5-26900 @ 2.90 gigahertz (2 processors) 128 gigabytes RAM machine.

7.1. Continuous casting model and solving MOOP-2

7.1.1. Extent of nonlinearity

In order to assess the complexity of *concast*, certain number of sample points were obtained using the full factorial sampling plan to construct the tile plots (Forrester et al., 2008) as depicted in Fig. 5. Tile plots are specifically the contours of outputs with respect to two inputs taken at a time. They provide a qualitative measure of (i) non-linearity through contour diagrams and (ii) impact of inputs on considered output through intensity of shades. These plots

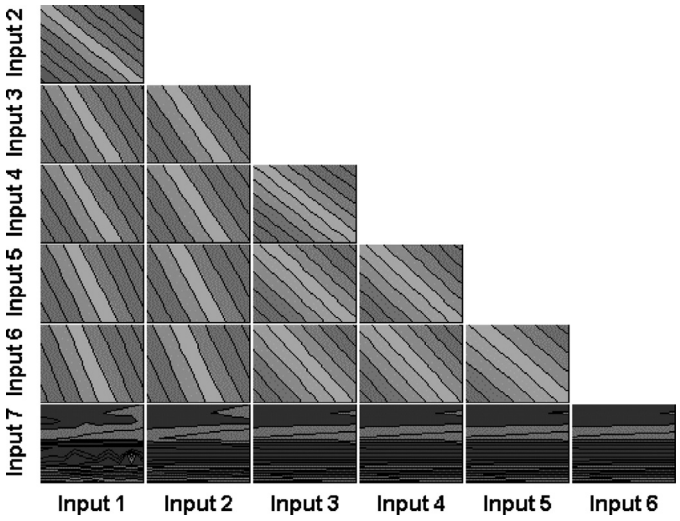

Fig. 5. Tile plot depicting the contours of output-1 with respect to all inputs.

Table 4
NSGA II credentials for solving the MOOP-1 formulation mentioned in Table 1.

Parameters	Values
Number of real variables	7
Population size	100
Number of generations	50
Crossover probability	0.9
Mutation probability	0.01

in Fig. 5 clearly reveal that *concast* is highly non-linear and of all the inputs, the input 7 – castor speed, has maximum effect on the output 1. A similar plot for output 2 is presented in Fig. S.F.1 of Supplementary text.

7.1.2. Solution of MOOP-1 using evolutionary and classical optimization methods

Concast was optimized in conventional manner using one of the robust evolutionary optimizers, NSGA II whose details are listed in Table 4. The total number of function evaluations required for that run was 5000 (50×100). Selection of this generation number was purely on the basis of convergence of candidate solutions on the Pareto front.

In order to justify the implementation of population based evolutionary solver, NSGA-II, MOOP-1 was also solved by the classical optimization techniques after reformulating it using the ε-constraint methodology (Deb, 2001). In this method, the multi objective problem is reformulated into a single objective optimization formulation by considering only one of the objectives and confining others within stipulated bounds ε_m. For a K dimensional MOOP problem containing M objectives, I inequality and E equality constraints the corresponding ε-constraint formulation is shown below.

General framework of MOOP formulation

$$\text{Minimize } f_m(x)\ \forall\ m = 1, 2, \ldots M$$
$$\text{subject to, } g_j(x) \geq 0\ \forall\ j = 1, 2, \ldots I$$
$$h_i(x) = 0\ \forall\ i = 1, 2, \ldots E$$
$$x_k^L \leq x_k \leq x_k^U\ \ \forall\ \ k = 1, 2, \ldots K$$

Equivalent ε-constraint SOOP formulation

$$\text{Minimize } f_n(x)\ |\ n \in [1\ M]$$
$$\text{subject to, } f_m(x) \leq\ \varepsilon_m \forall\ m = 1, 2, \ldots M;\ m \neq n$$
$$g_j(x) \geq 0\ \forall\ j = 1, 2, \ldots I$$
$$h_i(x) = 0\ \forall\ i = 1, 2, \ldots E$$
$$x_k^L \leq x_k \leq x_k^U\ \ \forall\ \ k = 1, 2, \ldots K$$

Table 5
MATLAB's fmincon credentials for solving the reformulated MOOP-1 formulation.

Parameters	Values
Fmincon algorithm	SQP
Number of real variables	7
Number of initial guesses	100
Maximum function evaluations	100,000
Maximum iterations	10,000
Constraint tolerance	1E-10
Function tolerance	1E-20

Here ε_m represents the upper bound on objective m. By selecting different values for ε_m and solving the corresponding SOOP problem, intermediate Pareto Optimal (PO) points are generated. This method is reported to be a better method than the weighted sum approach to solve MOOP formulations using classical optimization routines (Deb, 2001). In our problem we implement the ε-constraint method by considering the speed maximization as the only objective while the remaining two objectives, the Temperature and Bulging were converted into constraints. The lower and upper bounds on these two objectives (modified as constraints), that is the values of ε_m were obtained by identifying the anchor points of the Pareto front. The anchor points were obtained by solving one objective at a time and repeating this exercise for all three objectives. The anchor points provided the best and worst possible values of each objective function out of which we used the best and worst solutions of Temperature and Bulging as upper and lower bounds. This formulation was solved using the *fmincon* function of MATLAB®, where sequential quadratic programming (SQP) algorithm was implemented. For obtaining an unbiased comparative study of optimization, the population of zeroth generation of NSGA-II (100 in number) were given as initial conditions for *fmincon* and the reformulated optimization problem was solved using each one of these cases, one at a time. To ensure the convergence of *fmincon*, the values as mentioned in Table 5 were used.

The total number of function evaluations required by the classical technique for generating the Pareto Optimal front was 3600. The comparison of PO fronts obtained using NSGA II and fmincon is shown in Fig. 6. The PO front obtained by the classical optimization approach is found to be a local front as compared to the same achieved by NSGA II. Moreover, the spread of PO solutions obtained by the classical approach is not as uniform as the NSGA II solutions. Though the number of function evaluations is more, NSGA II has been preferred further as the choice of algorithm to solve the MOOPs due to the higher quality of PO solutions it provided.

7.2. Implementation of TRANSFORM to find best ANNs

TRANSFORM algorithm was implemented to generate the best ANNs for emulating *concast*. Details of NSGA II algorithm for solving MOOP-2 are listed in Table 6. To present an unbiased comparison, we incorporated HC and HC+SOOP based SSD techniques in TRANSFORM. However, in practise, prior to the ANN construction, one SSD technique needs to be finalized based on the

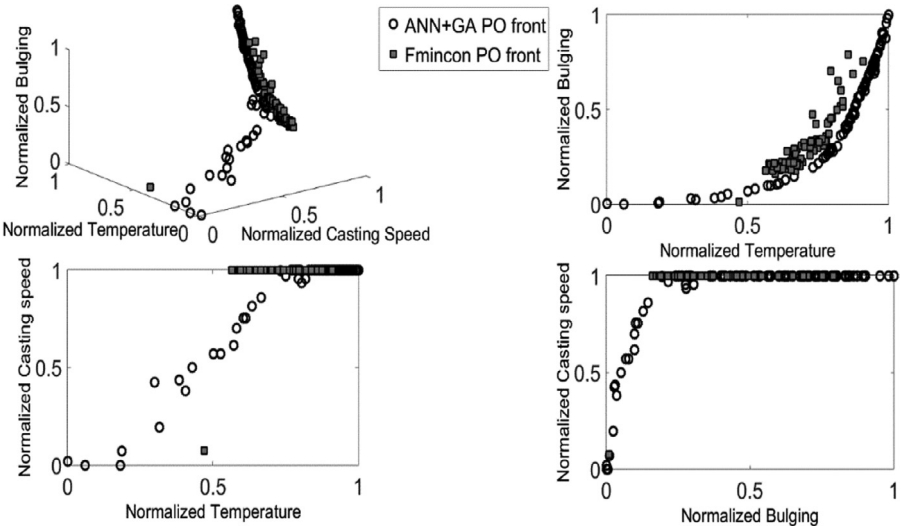

Fig. 6. Comparison of classical and evolutionary optimizer for solving the multiobjective optimization of industrial casting model from steel plants.

Table 6
Credentials of NSGA II for solving the MOOP-2 mentioned in Table 2.

Parameters	Values
Number of binary variables	4
Population size	200
Number of generations	100
Crossover probability	0.9
Mutation probability	0.01

Table 7
Computational time comparison between proposed SSD techniques for a fixed architecture [7–5–2–1–1] using tan sigmoidal activation for emulating output-1.

Technique	Architecture	*N_TF*	Computational time	Sample size
SOOP	7-4-3-3-1	1	1240 seconds	130
HC	7-4-3-3-1	1	500 seconds	130
HC+SOOP	7-4-3-3-1	1	555 seconds	130

computational constraints. The following subsection thus provides the computational time comparison between the proposed SSD techniques to enable the decision maker to select one based on the necessity.

7.2.1. Computational time comparison between proposed SSD techniques

The proposed SSD methods were implemented to determine the sample size of a complicated network of given configuration e.g., [7-4-3-1-1], which was arbitrarily chosen such that it contained 3 hidden layers. Tan sigmoidal activation function was chosen for emulating output-1 using the selected architecture. The results of this study are presented in Fig. 7 and Table 7. A sample size of 140, 130 and 130 were determined by SOOP, HC and HC+SOOP, respectively. Although the sample size determined by each of these methods for the architecture considered were different, the results are presented for a fixed sample size (130) to provide a common platform for comparing execution times of these methods. Since the choice of incrementing the sample size is purely based on the decision maker, any different smaller value of increment in sample size is encouraged, provided the resultant increase in computational time is acceptable to the decision maker. Also, the choice of termination criteria α is left to the decision maker. Since these two parameters (increment in sample size and termination criteria) play less significant role in ANN construction, they were not considered for estimation of optimal parameters in TRANSFORM.

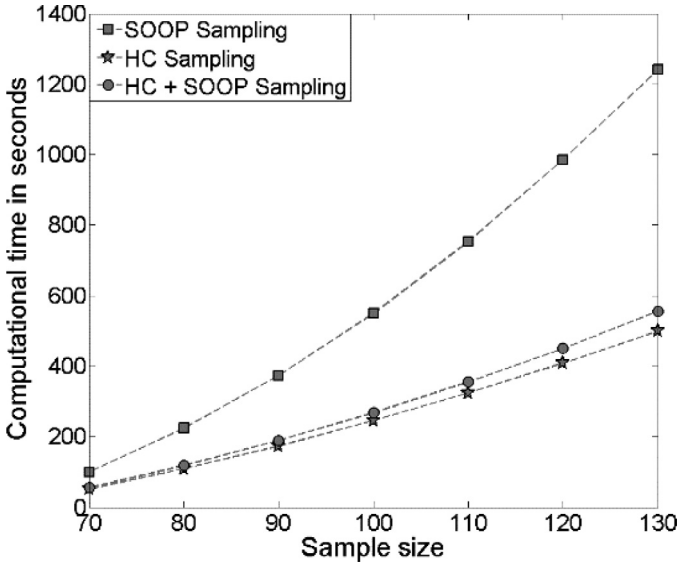

Fig. 7. Plot describing the computational time required by proposed techniques of SSD for architecture (7-4-3-1-1) for emulating output-1.

The results confirmed that the HC and HC+SOOP based sampling techniques are nearly 2.5 and 2.2 times faster than the SOOP based technique, respectively. The comparison in terms of accuracies can only be obtained after the entire optimization run is completed. Thus, considering the objective of this work to be the online implementation of optimization, TRANSFORM algorithm was run with the relatively faster HC and HC+SOOP based SSD techniques.

7.2.2. Implementation of TRANSFORM with HC and HC+SOOP based SSDs

The developed MATLAB© code being MISO in nature, two parallel simulations of TRANSFORM algorithm using HC based SSD technique were implemented for constructing ANNs corresponding to the two outputs of *concast*. Since there were three objectives (Table 2), the resulting solution of the MOOP-2 was a 3-dimensional PO front, as shown in Fig. 8(a). The corresponding PO points along with their sample size requirement, total nodes and accuracies are listed in Table 8. Based on a similar implemen-

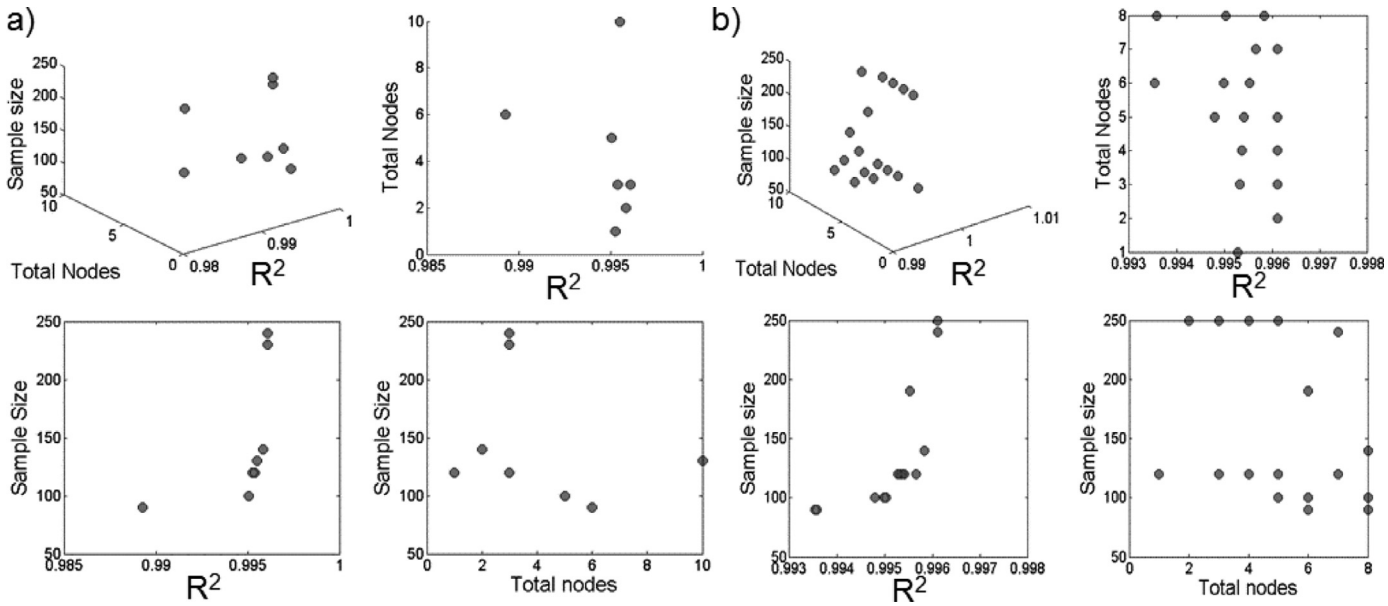

Fig. 8. 3 D PO front obtained from TRANSFORM using (a) HC based SSD and (b) HC+SOOP based SSD for emulating output-1.

Table 8
PO solutions obtained from TRANSFORM using HC SSD for emulating output-1.

*N*1	*N*2	*N*3	*N*_TF	R^2	*N*	*n*
1	0	1	1	0.995277	1	120
1	0	2	0	0.995277	1	120
1	2	0	0	0.995381	3	120
1	2	2	1	0.995073	5	100
1	5	4	0	0.995524	10	130
2	0	0	1	0.995839	2	140
2	0	0	0	0.995839	2	140
2	1	0	1	0.996098	3	230
2	1	0	0	0.996102	3	240
4	1	1	1	0.989285	6	90

Table 9
PO solutions obtained from TRANSFORM using HC+SOOP for emulating output-1.

*N*1	*N*2	*N*3	*N*_TF	R^2	*N*	*n*
1	0	0	0	0.995277	1	120
1	0	5	1	0.995277	1	120
1	1	4	0	0.994981	6	100
1	1	6	0	0.993575	8	90
1	2	0	0	0.995314	3	120
1	2	2	1	0.994795	5	100
1	2	5	0	0.995024	8	100
1	3	0	1	0.995359	4	120
1	3	1	1	0.995405	5	120
1	3	2	0	0.99353	6	90
2	0	0	0	0.996106	2	250
2	1	0	1	0.996109	3	250
2	1	0	0	0.99611	3	250
2	1	1	1	0.996111	4	250
2	1	1	0	0.996111	4	250
2	1	2	0	0.996111	5	250
2	1	4	0	0.996104	7	240
2	3	1	1	0.995526	6	190
2	4	1	1	0.99566	7	120
2	5	1	1	0.995826	8	140

tation of TRANSFORM, but this time with HC+SOOP based SSD, the results of MOOP-2 are presented in Fig. 8(b) and Table 9.

7.2.3. Selection of single ANN from set of PO solutions

In order to perform the surrogate assisted optimization of *concast*, a single ANN architecture for each output, is needed from the set of PO solutions of MOOP-2. Selection of a single selection from a set of PO solutions is performed by using some legitimate higher order information coming from the decision maker (Deb, 2001). We, being the decision makers in this case, have utilized a robust metric for model selection called AIC (Akaike, 1971), well known in literature for selecting the model which is least over-fitted (Basak, 2002; Dirick, Claeskens, & Baesens, 2015; Qi & Zhang, 2001). The readers are encouraged to implement any higher order information to select a single best ANN model for emulating the given physics based model. Of all the existing models (PO solutions), the one with least measure of AIC (see Eq. 6) is selected.

Table 10
ANN models to emulate the output-1 and output-2 of the casting model. Higher order information used is: AIC.

	Architecture [7-*N*1-*N*2-*N*3-1]	Sample size (*n*)	*N*_TF	R^2 (for 200 test set)
	HC			
Output-1	7-2-0-0-1	140	0	0.99
Output-2	7-2-1-0-1	200	0	0.98
	HC+SOOP			
Output-1	7-1-1-4-1	100	0	0.99
Output-2	7-2-1-0-1	190	1	0.98

$$\mathrm{AIC} = 2P + S\,\log(\mathrm{MSE}) \tag{6}$$

In Eq. (6), P is the number of parameters in the model, S is the training sample size and MSE is the mean square error of the predictions. There exists a subtle difference between these parameters and the entirely different set of parameters governing the ANN, considered for estimation in TRANSFORM. The parameters in Eq. (6) are the ones which are trained by a suitable training algorithm of the model to capture the variations in the training set. For ANNs, these are simply the total number of weights and biases in the network. Table 10 shows the selected architectures for emulating the outputs of *concast* in surrogate assisted optimization run. Fig. 9 shows the evolution of the selected architecture [7-1-1-4-1] with sample size increments in the HC+SOOP based SSD technique. In this figure, across a row, the left subfigure shows the distribution of Sobol points in specific three dimensions, the central subfigure describes the corresponding surface plots of ANN while the third

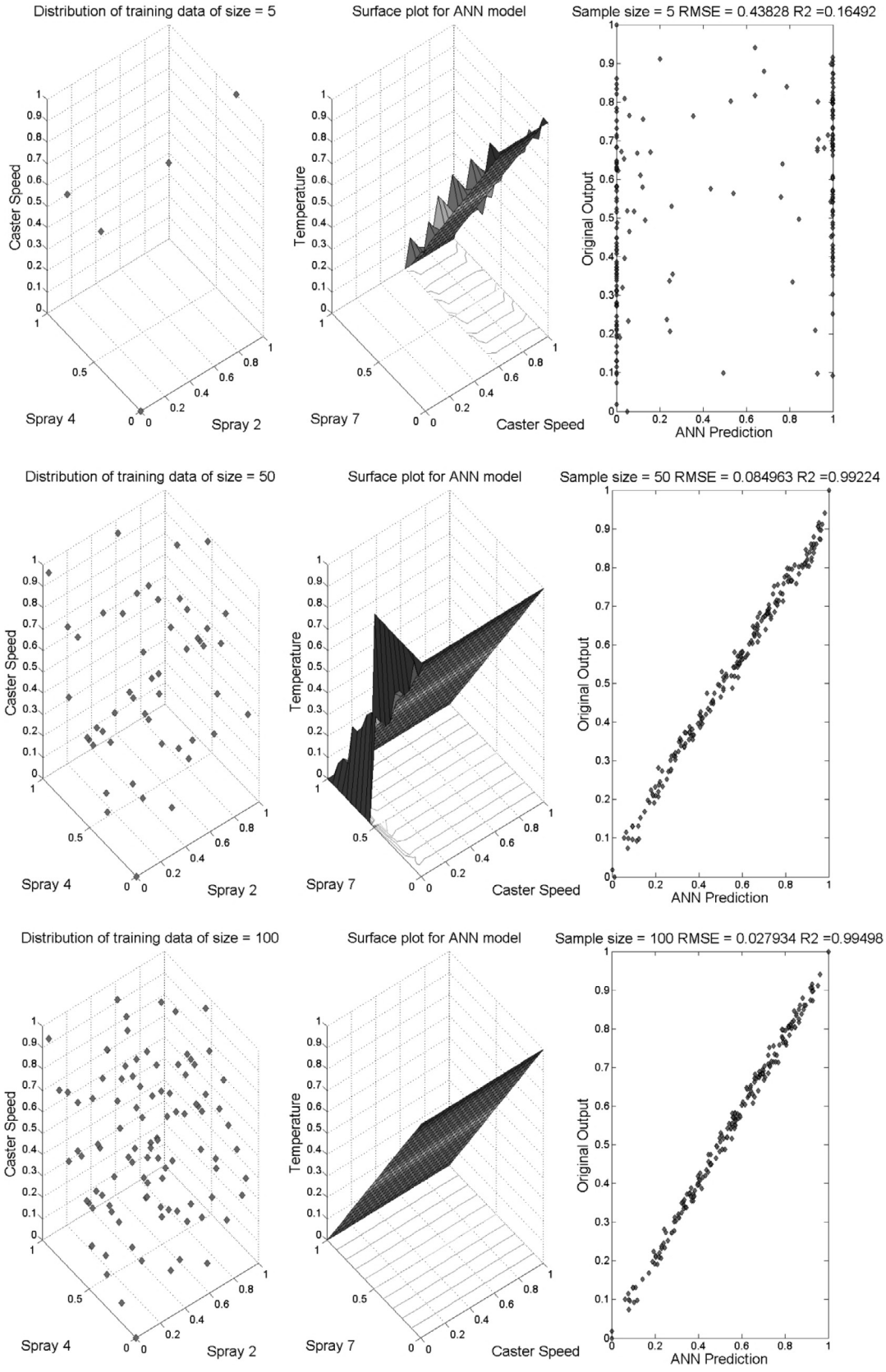

Fig. 9. Evolution of ANN model [7-1-1-4-1] in HC+SOOP SSD for emulating output-1.

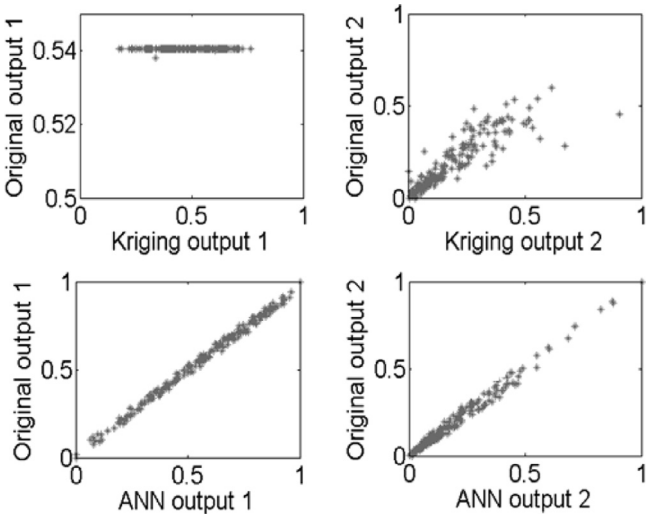

Fig. 10. Comparison of parity plots of KI (top) and TRANSFORM-ANN with HC+SOOP (bottom) for emulating output 1 (left) and output 2 (right) of casting model.

subfigure provides the parity between the original output and ANN predicted outputs.

7.3. Analysis of results obtained using TRANSFORM

- Emergence of multi-layered ANNs for emulating casting model, justifies the exploration for multi-layered perceptron through TRANSFORM and their ability to capture the highly non-linear dynamics with optimal (less) sample points for training.
- Appearance of architectures with both log and tan sigmoidal activation justifies their selection as crucial parameters governing the predictability of ANNs.
- Novel SSD techniques, robust in terms of both speed and accuracy, ensured rapid convergence of TRANSFORM in real time.
- The issue of over-fitting arising mainly due to large number of nodes and data greedy nature of ANNs has been resolved by manoeuvring the trade-off between efficiency and parsimony.
- ANNs built by TRANSFORM went through a robust and established model selection criteria, AIC, which further eliminated the threats of over-fitting.

7.4. Comparison of TRANSFORM-ANN and Kriging Interpolation

To further justify the robustness of the proposed TRANSFORM algorithm, Kriging surrogates, constructed for emulating *concast* using EI based sampling approach, are compared in this section with ANNs constructed using TRANSFORM. Since we have two different Kriging models built correspondingly for two outputs, their sample points are completely different. The final reported sample size in case of Kriging is the sum of individual sample sizes required for emulating two outputs. It should be noted here that ANNs consider the larger sample size out of both the outputs as the maximum sample size because of the efficient quality of Sobol scheme to preserve and maintain the sequence of sample points generated.

The parity plots (original output versus predicted output) obtained using Kriging and ANNs for both the outputs are compared in Fig. 10, while Table 11 presents an account of sample sizes and accuracies. The results clearly show that Kriging model was unable to capture the non-linearity even with large number of sample points, whereas ANN could successfully capture the same with high accuracy and comparatively very less number of samples.

Two inferences can be drawn from this comparative analysis:

Table 11
Comparison of TRANSFORM-ANN with HC+SOOP and Kriging Interpolators in terms of emulating the casting model: sample size and accuracies.

	ANN model			Kriging model		
	R^2	RMSE	Size	R^2	RMSE	Size
Output 1	0.99	0.02	100	0.004	0.14	730
Output 2	0.98	0.03	190	0.8	0.07	700
	Total function calls = 190			Total function calls = 1430		

(i) Reason for Kriging failure: EI based sampling primarily focuses on the region of maximum non-linearity. Thus more and more samples are dedicated to only a specific region. Thereafter focus shifts to the subsequent region having maximum error estimate. Thus, very huge set of samples are required to capture all the regions with non-linearity.
(ii) ANNs, on the other hand, through sequential sampling techniques, were able to uniformly sample the complete input domain. Thus, as it is evident from the aforementioned results, ANNs could emulate the complete non-linear region with very less sample points compared to Kriging. This can be accredited to the combination of robust Sobol based SSDs and the proposed intelligent framework – TRANSFORM which brought up the best ANN.

7.5. ANN Surrogate assisted optimization of continuous casting model: solving MOOP-1

The optimization of *concast* (MOOP-1) was performed next using NSGA-II with each of the selected ANN model (see Table 10). Although the number of population and generations were kept at 100 each, it was observed that the solutions converged around 50th generation. All function evaluations during optimization were obtained using the significantly faster ANN model, thus resulting in completion of the optimization run in very less time. This time advantage allows the decision maker to go for a larger size of population leading to PO front having denser spread of PO solutions, thereby providing more alternatives to the decision maker. The decision variables which form the final PO front were provided to the original time expensive casting model for obtaining the outputs for comparison. These comparisons for the cases of TRANSFORM-ANN with HC and HC+SOOP based SSDs are shown in Fig. 11(a) and (b), respectively. The average RMSE values (averaged over the three objective functions) were also calculated and reported to be less than 3% for both the cases. Fig. 11 shows that the ANN Pareto front obtained using the HC+SOOP based TRANSFORM-ANN results in a better solution compared to the original Pareto front (see top right sub-figure of Fig. 11(b) where the improvement in optimality using ANN surrogate is clearly visible). However, with HC based TRANSFORM-ANN in place, we obtain results similar to the conventional Pareto front. The advantage with HC based surrogate in place is it is faster than the HC+SOOP based surrogate. These results justify the trade-off between the computational speed and accuracy of predictions using surrogate models.

7.6. Scope of online optimization through TRANSFORM

With reference to Table 10, it is clear that the ANN model obtained using TRANSFORM with HC and HC+SOOP based SSD algorithm, consumed 200 and 190 sample points, respectively, for training the corresponding networks. Apart from the training set, each of the network was validated with a LHS sample set of size 200. Thus, the total sample size required by the ANN models is 390 for HC+SOOP and 400 for HC+SOOP ($190+200$ and $200+200$) and therefore only those many function evaluations of the original model were required for the entire optimization run. Comparing

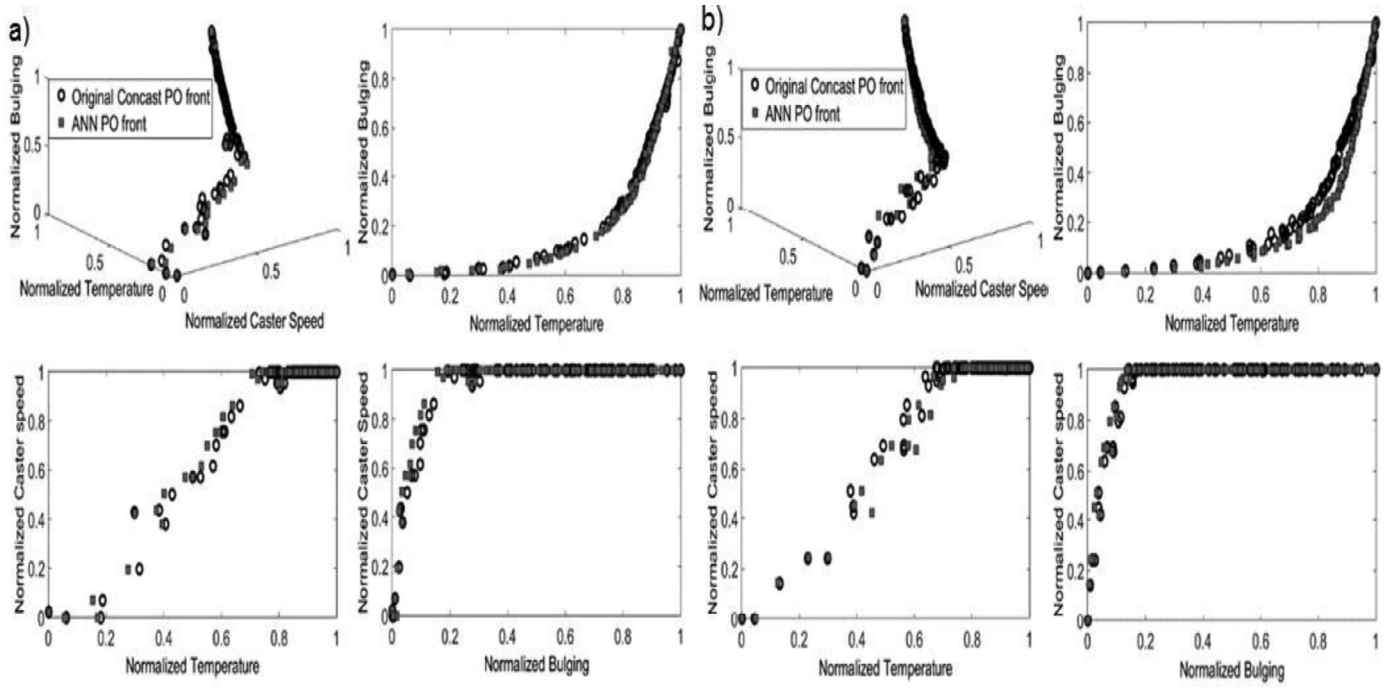

Fig. 11. Comparison between the PO fronts of conventional optimization (MOOP-1) using and Surrogate assisted optimization with (a) HC and (b) HC+SOOP SSD based TRANSFORM-ANN.

with the 5000 function evaluations of the original casting model when optimized conventionally without using the surrogates, the ANN based optimization is reported to be nearly 13 times efficient resulting in a saving of 92% function evaluations {100 ×(5000 – 390)/5000}. Although the surrogate construction has seen a maximum sample size of 250 (Tables 8 and 9), the margin of 60 points (250 – 190) does not significantly influence the speed of the surrogate based optimization. Also, on account of considering the fact that the sample size for validation can be safely reduced to a value as low as 80 from 200 (according to the generic thumb rule of considering one third of the size of training set to be the size of validation set), the surrogate building algorithm can actually be made much faster even after considering the maximum sample size obtained in the entire algorithm. The considered sample set for training is obtained such that, majority of the operation of the casting plant lies within the upper and lower bounds of the training set, thus ensuring only one time implementation of TRANSFORM. The ANNs once built need not be altered unless the plant operating conditions go beyond the limits of training set, which may be a rare case. It is important to mention that TRANSFORM requires considerable amount of time for constructing the best ANN model. However, once the model is trained, it is observed that ANN hardly requires a maximum of 1 second of CPU time for predicting single input–output data point. Thus, the ANN assisted optimization run (1 second per single simulation) will take less than {(5000*1)/(60*60)} = 1.5 hours, where 5000 is the function evaluations required by NSGA II to optimize *concast* conventionally without surrogate. The computational speed of TRANSFORM and thereby the scope for online optimization can be increased by several manifolds if the source-codes are ported from MATLAB to much faster high level languages such as FORTRAN or C. Since NSGA II works with population of solutions, the execution time for the optimization algorithm could have been further reduced by implementing parallel programming. The summary of the function evaluations is shown in Table 12 which clearly articulates that ANN assisted GA based optimization of continuous casting model can be made online.

Table 12
Comparison of function evaluations between classical optimization, modern evolutionary optimization and surrogate assisted optimization for checking the scope of online implementation.

Optimizer	PO front quality	Function evaluations	Online possibility?
fmincon	Local	3600	No
NSGA-II	Better than local	5000	No
ANN+NSGA-II	Better than local	390	Yes

The proposed methodology does not differentiate the simulation models with the experimental setups and thus can be used without bias for optimizing the operation of an experimental set up. All that is required by TRANSFORM is the experimental data sampled at the points obtained using the Sobol sampling plan. Although this study is restricted to feed forward perceptron networks, it can be easily extended to recurrent networks which have displayed immense capability in time series predictions, thus becoming one of the prominent candidates for dynamic data modelling.

7.7. Pareto characterization

In general after solving a multiobjective optimization problem, we obtain a set of decision variables called PO solutions which lead corresponding non-dominated realizations in objective function space, called PO front. It is often a general practise to select a single point from the PO front using some higher order information and discard the other solutions (Deb, 2001). Although effective, this procedure might not provide the complete information about the solution set. However, applying the data analytic techniques on the PO front may help in identifying the hidden pattern or information in the PO front about the corresponding decision variables (Mitra & Ghosh, 2008). We call this procedure as Pareto Characterization where we clustered the PO front into segments to retrieve the characteristic features of corresponding decision variables.

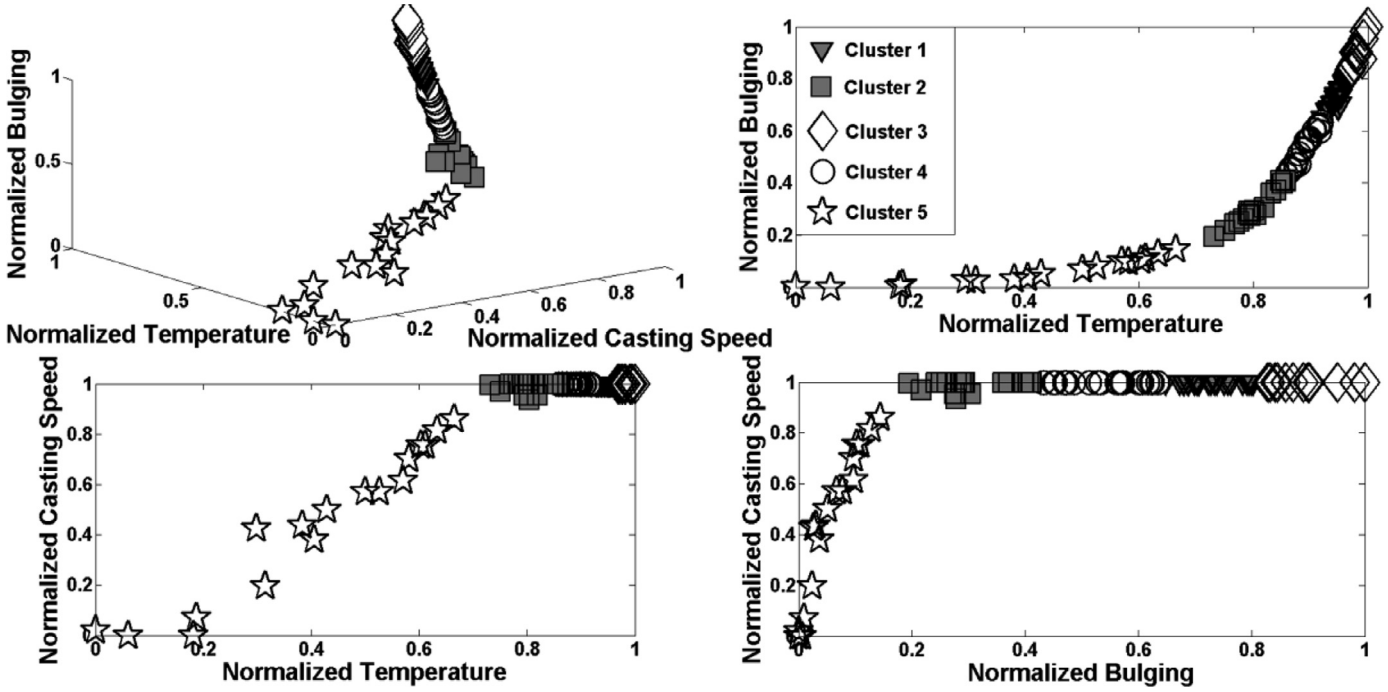

Fig. 12. *K*-means clustering of PO points obtained using surrogate assisted optimization.

Since the objectives of MOOP-1 were to maximize the caster speed, thereby leading to high productivity, maximize the exit temperature, thereby leading to more energy savings with less operational cost and minimize the bulging, thereby leading to high quality of the product, the PO points obtained from surrogate based optimization were further clustered using *K*-means method to identify credible information about the corresponding decision variables. Such information might be of significant importance to an industrial operator in order to successfully operate the plant in desired optimal fashion. The readers are referred to Section S.4 of Supplementary text for a brief description about *K*-means clustering algorithm (Hartigan & Wong, 1979).

Since the working of any clustering algorithm, such as *K*-means method, is based on a mathematical formulation involving the norm of points from the cluster centre (Hartigan & Wong, 1979), it is important to decipher a physical meaning out of the clusters before we proceed to observe any trend in corresponding decision variables. With this objective, the number of clusters in *K*-means method was varied from 2 to 10 till an optimal value of 5 clusters was identified. The clusters were obtained such that, we have five distinct operating conditions leading to five classes of products each varying in terms of productivity, operation cost and product quality. The clustered PO points in the objective function space are shown in Fig. 12. The corresponding decision variables, i.e., the flowrates at six nozzle sprays starting from Spray 2 to Spray 7 (see Table 1) of these clustered PO points (addressed from now on as PO flowrates) are plotted in Fig. 13 along with corresponding objective function classification.

We observed an interesting trend in the PO flowrates, at each nozzle spray for each cluster governing the optimal operation of casting process. Although the flowrates at each nozzle spray were allowed to vary anywhere between the lower and upper bounds (depicted by normalized values 0 and 1, respectively, in Fig. 13) the PO flowrates, however, confined themselves to particular region in the decision variable space. The distribution of PO flowrates at each spray in each cluster is presented in Fig. S.F.4 of Supplementary text. Based on this observation, a simple rule set is formulated and presented in Table 13. This rule set tells to the end user on how to vary the flowrates at six nozzle sprays such that casting operation can be performed with high productivity and high quality with least operational cost.

For instance, Rule 1 in Table 13 reads, *In order to perform the casting of steel such that the caster speed is maintained at its maximum, exit temperature of the slab is maintained at its maximum and quality of the product is maintained at mid-level of the desired lower and upper qualifications, the plant needs to be run with the flowrate of second nozzle spray near its maximum, flowrate at third nozzle spray ranging between its lower bound and central value, while the flowrates at all the remaining nozzle sprays are ensured at their minimal values.* The set of five rules are also presented pictorially in Fig. 14 and in Figs. S.F.5–S.F.8 of Supplementary text, respectively.

This result based on Pareto Characterization is interesting and of high practical importance. The trend in PO decision variables guides the end user (here the casting plant operator) to run his/her process (here the casting operation) in optimal fashion by following a set of simple instructions which do not necessitate any imperative prerequisite about the knowledge of process optimization and control. Although we have used Pareto Characterization to derive the general framework of operators' rules, the most efficient way to obtain a single solution from a set of PO solutions is to use the desired higher order information. However, in absence of higher order information, such as the situation in the current case study, the readers may implement Pareto Characterization to capture a trend if present in the PO solutions. Further, the selection of clustering method entirely depends on the perspective of the reader. We used *K*-means in this work to since we believe it to be well known to the readers and exploit its advantages such as simplicity and easy accessibility. There is no prejudice in selecting *K*-means as the clustering method and the readers are encouraged to use a more efficient clustering method.

7.8. Novelty in TRANSFORM

This section presents in brief, the novelty of TRANSFORM when compared with one of our previous works in Miriyala et al. (2016).

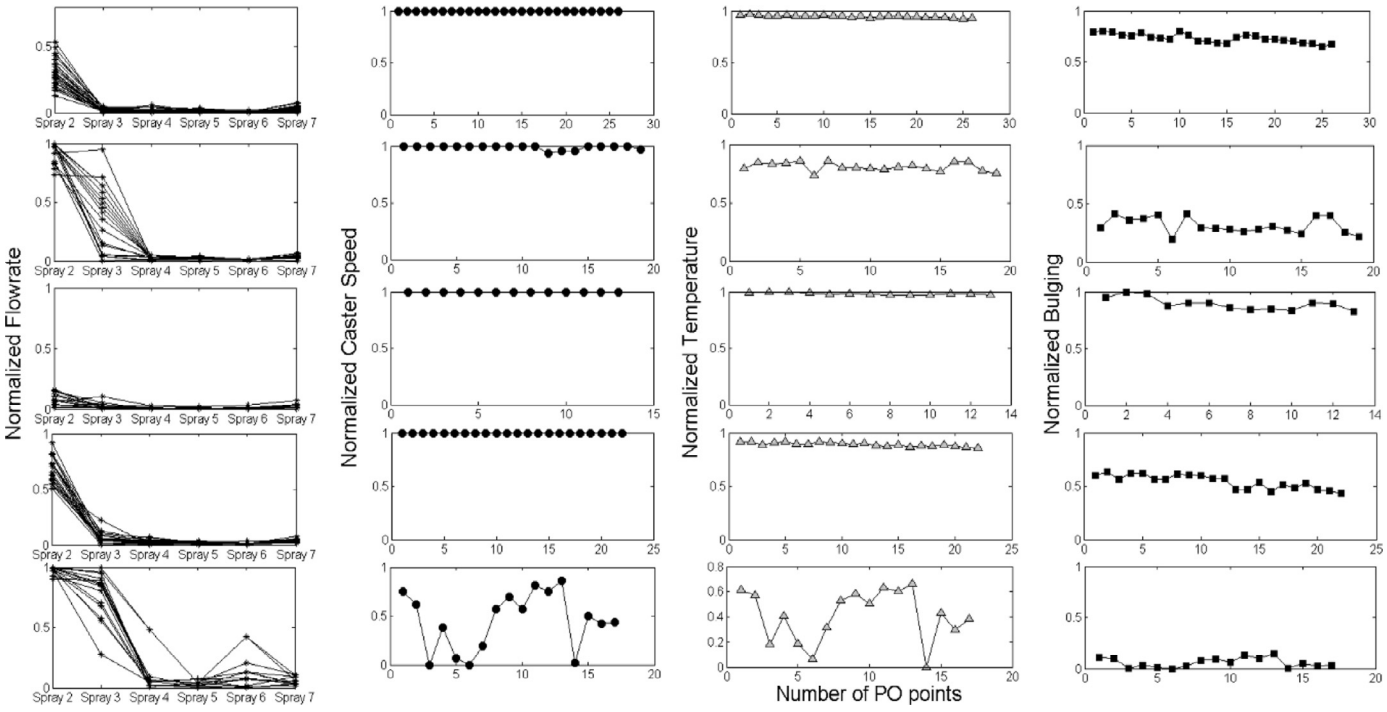

Fig. 13. Distribution of decision variables corresponding to the PO clusters.

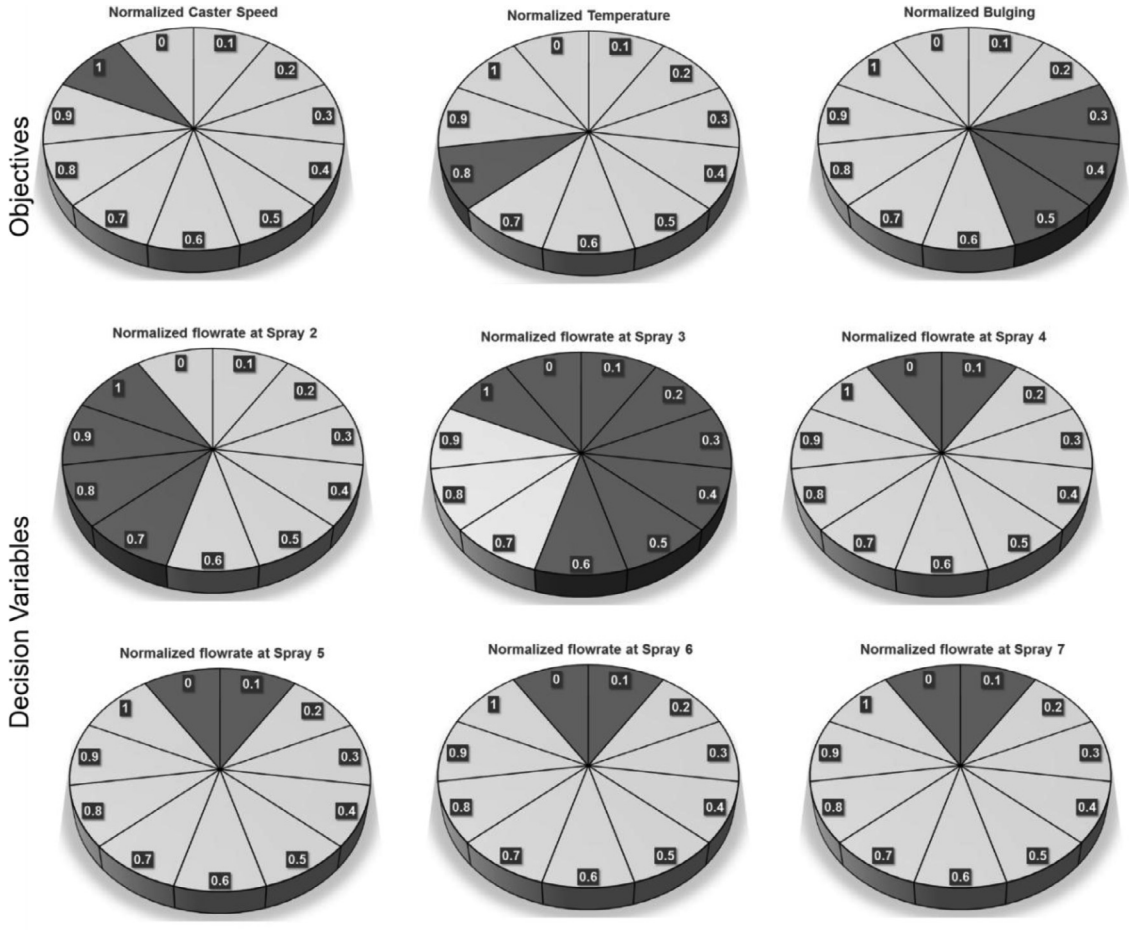

Fig. 14. Pictorial representation of operator rule-1.

Table 13
Set of operators rules, deduced after clustering the PO solutions obtained from surrogate based optimization, for optimal functioning of the continuous casting process.

Rules	Desired normalized objectives			Operating region of normalized decision variables (nozzle flowrates)					
	Caster speed	Temp.	Bulging	Spray 2	Spray 3	Spray 4	Spray 5	Spray 6	Spray 7
1	1	0.8	0.3–0.5	0.7–1	0–0.6 and 1	0–0.1	0–0.1	0–0.1	0–0.1
2	1	0.8	0.4–0.6	0.5–0.9	0–0.3	0–0.1	0–0.1	0–0.1	0–0.1
3	1	1	0.8–0.9	0–0.2	0–0.1	0–0.1	0–0.1	0–0.1	0–0.1
4	1	0.9	0.7–0.8	0.2–0.5	0–0.1	0–0.1	0–0.1	0–0.1	0–0.1
5	0–0.9	0–0.6	0.3–0.5	0.9–1	0.3–0.4 and 0.7–1	0–0.2	0–0.2	0–0.4	0–0.4

- The previous work focused on optimal architecture estimation, whereas TRANSFORM incorporated a three objective optimization problem with thrust on minimizing the sample size, maximizing accuracy and parsimony along with optimal architecture estimation.
- The current work proposes three novel sample size determination algorithms, which compared to the *K*-fold based method of our previous work, are nearly one order of magnitude faster.
- All the three novel SSD algorithms proposed in the current work are found to be working successfully for high dimensions whereas the work published earlier was confined to a three dimensional problem.
- Pareto Characterization provided with simple rules for optimal functioning of process industry.
- Comparative studies between evolutionary and classical optimization algorithms are included.

8. Conclusions

Optimization and control of real world processes is performed offline owing to large computational time required by evolutionary optimizers to solve the inherent MOOP. In the present study, this problem has been resolved using surrogate assisted optimization by selecting ANNs as potential surrogate models. However, ANNs in general, are governed by certain parameters, such as the architecture and sample size for training, etc., whose heuristic assumption deteriorates the surrogate quality. Thus, in order to realise a logical framework which would intelligently estimate the associated parameters, a novel parameter free ANN building algorithm called TRANSFORM is proposed in this work. ANN models are then constructed for a highly nonlinear industrially validated continuous casting model which were then used for the optimization. The constructed ANN models were also compared with the state-of-the-art Kriging surrogates for accuracy and speed. The application of *K*-means clustering algorithm on PO points resulted in interesting trends followed by PO decision variables. This led to a set of simple operator's rules for optimal operation of casting plant. The conclusions of the proposed work are as follows:

- ANN based optimization of casting process is 13 times efficient, saving 92% function evaluations compared to the conventional methods, thus enabling its online implementation.
- Over-fitting of ANNs is successfully prevented by incorporating three equipotent fast sample size determination (SSD) techniques in TRANSFORM apart from robust AIC based model selection criteria.
- SSD techniques are motivated by the concepts of space filling and a novel hypercube based data classification method.
- A comparative study between these SSD techniques provided huge flexibility to the algorithm enabling it to adjust as per the choice of the decision maker.
- The failure of KI based method for emulating the considered complex model, provided justification to the robustness of the proposed method.
- All this was possible with TRANSFORM – a generic parameter free ANN building algorithm, requiring only a first principle based simulation model or experimental data as input and capable of providing best parsimonious ANN based surrogate through simultaneous estimation of all the parameters governing the process of surrogate building.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ejor.2017.05.026.

References

Akaike, H. (1971). Information theory and an extension of the maximum likelihood principle. In B. N. Petrov, & F. Csáki (Eds.), *Proceeding of the second international symposium on information theory* (pp. 267–281). Tsahkadsor, Armenia: USSR.

Barrow, D., & Kourentzes, N. (2016). The impact of special days in call arrivals forecasting: A neural network approach to modelling special days. *European Journal of Operational Research*. doi:10.1016/j.ejor.2016.07.015.

Basak, I. (2002). On the use of information criteria in analytic hierarchy process. *European Journal of Operational Research, 141*(1), 200–216.

Boithias, F., Mankibi, M., & Michel, P. (2012). Genetic algorithms based optimization of artificial neural network architecture for buildings' indoor discomfort and energy consumption prediction. *Building Simulation, 5*(2), 95–106.

Carvalho, A. R., Ramos, F. M., & Chaves, A. A. (2011). Metaheuristics for the feedforward artificial neural network (ANN) architecture optimization problem. *Neural Computing and Applications, 20*(8), 1273–1284.

Chi, H., Mascagni, M., & Warnock, T. (2005). On the optimal Halton sequence. *Mathematics and Computers in Simulation, 70*, 9–21.

Crombecq, K. (2011). Surrogate modeling of computer experiments with sequential experimental design, Doctoral Dissertation Ghent University.

Davis, E., & Ierapetritou, M. (2010). A centroid-based sampling strategy for Kriging global modeling and optimization. *The American Institute of Chemical Engineers, 56*, 220–240.

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. Chichester, UK: Wiley.

Denton, J. W., & Hung, M. S. (1996). A comparison of nonlinear optimization methods for supervised learning in multilayer feedforward neural networks. *European Journal of Operational Research, 93*(2), 358–368.

Dirick, L., Claeskens, G., & Baesens, B. (2015). An Akaike information criterion for multiple event mixture cure models. *European Journal of Operational Research, 241*(2), 449–457.

Diwekar, U. M., & Kalagnanam, J. R. (1997). Efficient sampling techniques for optimization under uncertainty. *The American Institute of Chemical Engineers, 43*(2), 440–447.

Dua, V. (2010). A mixed-integer programming approach for optimal configuration of artificial neural networks. *Chemical Eng. Res. and Des., 88*, 55–60.

Eason, J., & Cremaschi, S. (2014). Adaptive sequential sampling for surrogate model generation with artificial neural networks. *Chemical Engineering Research and Design, 68*, 220–232.

Forrester, I. J., Sobester, A., & Keane, A. J. (2008). *Engineering design via surrogate modelling, a practical guide*. Wiley.

Gorissen, D., Couckuyt, I., Demeester, P., Dhaene, T., & Crombecq, T. (2010). A surrogate modeling and adaptive sampling toolbox for computer based design. *The Journal of Machine Learning Research, 11*, 2051–2055.

Hagen, M. T., Demuth, H. B., & Beale, M. H. (2002). *Neural Network Design*. Boulder, Colorado: Campus Publishers Service.

Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A *K*-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics), 28*(1), 100–108.

Haykin, S. (1994). *Neural networks: A Comprehensive foundation*. New York: Macmillan College Publishing Company.

Jin, Y. (2011). Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computational, 1*(2), 61–70.

Jones, D. R. (2001). A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization, 21*, 345–383.

Kamini, V., Vadlamani, R., Prinzie, A., & Van denPoel, D. (2014). Cash demand forecasting in ATMs by clustering and neural networks. *European Journal of Operational Research, 232*(2), 383–392.

Kleijnen, J. P. C. (2016). Regression and Kriging metamodels with their experimental designs in simulation: A review. *European Journal of Operational Research, 256*(1), 1–16.

Miriyala, S. S., Mittal, P., Majumdar, S., & Mitra, K. (2016). Comparative study of surrogate approaches while optimizing computationally expensive reaction networks. *Chemical Engineering Science, 140*, 44–61.

Mitra, K., & Ghosh, S. (2008). Unveiling salient operating principles for reducing meniscus level fluctuation in an industrial thin slab caster using evolutionary multicriteria pareto optimization. *Materials and Manufacturing Processes, 24*(1), 88–99.

Mogilicharla, A., Chugh, T., Majumdar, S., & Mitra, K. (2014). Multi-objective optimization of bulk vinyl acetate polymerization with branching. *Materials and Manufacturing Processes, 29*, 210–217.

Mogilicharla, A., Mittal, P., Majumdar, S., & Mitra, K. (2015). Kriging Surrogate based multi-objective optimization of bulk vinyl acetate polymerization with branching. *Materials and Manufacturing Processes, 30*, 394–402 Genetic Algorithms Special Issue.

Morris, M. D., & Mitchell, T. J. (1995). Exploratory designs for computational experiments. *Journal Statistical Planning and Inference, 43*, 381–402.

Müller, J., & Shoemaker, C. A. (2014). Influence of ensemble surrogate models and sampling strategy on the solution quality of algorithms for computationally expensive black-box global optimization problems. *Journal of Global Optimization, 60*(2), 123–144.

Olaf, A., Barth, T., Freisleben, B., & Grauer, M. (2005). Approximating a finite element model by neural network prediction for facility optimization in groundwater engineering. *European Journal of Operational Research, 166*(3), 769–781.

Qi, M., & Zhang, G. P. (2001). An investigation of model selection criteria for neural network time series forecasting. *European Journal of Operational Research, 132*(3), 666–680.

Ruud, B., Driessen, L., Hamers, H., & Hertog, D. D. (2005). Constrained optimization involving expensive function evaluations: A sequential approach. *European Journal of Operational Research, 160*(1), 121–138.

Sermpinis, G., Stasinakis, C., Rosillo, R., & de la Fuente, D. (2017). European exchange trading funds trading with locally weighted support vector regression. *European Journal of Operational Research, 258*(1), 372–384.

Sermpinis, G., Theofilatos, K., Karathanasopoulos, A., Georgopoulos, E. F., & Dunis, C. (2013). Forecasting foreign exchange rates with adaptive neural networks using radial-basis functions and particle swarm optimization. *European Journal of Operational Research, 225*(3), 528–540.

Sexton, R. S., Dorsey, R. E., & Johnson, J. D. (1999). Optimization of neural networks: A comparative analysis of the genetic algorithm and simulated annealing. *European Journal of Operational Research, 114*, 589–601.

Shi, W., Shang, J., Liu, Z., & Zuo, X. (2014). Optimal design of the auto parts supply chain for JIT operations: Sequential bifurcation factor screening and multi-response surface methodology. *European Journal of Operational Research, 236*(2), 664–676.

Tabatabaei, M., Hakanen, J., Hartikainen, M., Miettinen, K., & Sindhya, K. (2015). A survey on handling computationally expensive multiobjective optimization problems using surrogates: Non-nature inspired methods. *Structural and Multidisciplinary Optimization, 52*(1), 1–25.

Uğur, Ö., Karasözen, B., Schäfer, M., & Yapıcı, K. (2008). Derivative free optimization methods for optimizing stirrer configurations. *European Journal of Operational Research, 191*(3), 855–863.

Wong, W- T, & Hsu, S- H (2006). Application of SVM and ANN for image retrieval. *European Journal of Operational Research, 173*(3), 938–950.