ELSEVIER

Chemical Engineering Journal 97 (2004) 115–129

Chemical Engineering Journal

www.elsevier.com/locate/cej

Hybrid process modeling and optimization strategies integrating neural networks/support vector regression and genetic algorithms: study of benzene isopropylation on Hbeta catalyst

Somnath Nandi[a], Yogesh Badhe[a], Jayaram Lonari[a], U. Sridevi[b], B.S. Rao[c], Sanjeev S. Tambe[a,*], Bhaskar D. Kulkarni[a]

[a] *Chemical Engineering Division, National Chemical Laboratory, Pune 411 008, India*
[b] *Department of Chemical Engineering, University of Notre Dame, Notre Dame, IN 46556, USA*
[c] *Catalysis Division, National Chemical Laboratory, Pune 411 008, India*

Received 22 February 2003; accepted 30 May 2003

Abstract

This paper presents a comparative study of two artificial intelligence based hybrid process modeling and optimization strategies, namely ANN-GA and SVR-GA, for modeling and optimization of benzene isopropylation on Hbeta catalytic process. In the ANN-GA approach [Ind. Eng. Chem. Res. 41 (2002) 2159], an artificial neural network model is constructed for correlating process data comprising values of operating and output variables. Next, model inputs describing process operating variables are optimized using genetic algorithms (GAs) with a view to maximize the process performance. The GA possesses certain unique advantages over the commonly used gradient-based deterministic optimization algorithms. In the second hybrid methodology, a novel machine learning formalism, namely *support vector regression* (SVR), has been utilized for developing process models and the input space of these models is optimized again using GAs. The SVR-GA is a new strategy for chemical process modeling and optimization. The major advantage of the two hybrid strategies is that modeling and optimization can be conducted exclusively from the historic process data wherein the detailed knowledge of process phenomenology (reaction mechanism, rate constants, etc.) is not required. Using ANN-GA and SVR-GA strategies, a number of sets of optimized operating conditions leading to maximized yield and selectivity of the benzene isopropylation reaction product, namely cumene, were obtained. The optimized solutions when verified experimentally resulted in a significant improvement in the cumene yield and selectivity.

Keywords: Process modeling; Process optimization; Artificial neural networks; Support vector regression; Genetic algorithms; Benzene isopropylation; Cumene synthesis; Hbeta catalyst

1. Introduction

Availability of a process model is a prerequisite to process optimization. Conventionally, two approaches namely *phenomenological* (first principles) and *empirical*, are employed for chemical process modeling. In phenomenological modeling, the detailed knowledge of the reaction kinetics and associated heat and mass transport phenomena are required to represent mass, momentum, and energy balances. The advantages of a phenomenological model are: (i) since it represents physicochemical phenomenon underlying the process explicitly, it provides a valuable insight into the process behavior, and (ii) it possesses extrapolation ability. Owing to the complex nature of many chemical processes, the underlying physicochemical phenomenon is seldom fully understood. Also, collection of the requisite phenomenological information is costly, time-consuming and tedious, and therefore development of phenomenological process models poses considerable practical difficulties. Moreover, nonlinear behavior being common in chemical processes, it leads to complex nonlinear models, which in most cases are not amenable to analytical solutions; thus, computationally intensive numerical methods must be utilized for obtaining solutions. Difficulties associated with the construction and solution of phenomenological models necessitates exploration of alternative modeling formalisms. Modeling using empirical (regression) methods is one such alternative. In conventional empirical modeling, appropriate linear or nonlinear models are constructed exclusively from the process input–output data without invoking the process phenomenology. A fundamental deficiency of the

[*] Corresponding author. Tel.: +91-20-5893095; fax: +91-20-5893041.
E-mail address: sstambe@che.ncl.res.in (S.S. Tambe).

1385-8947/$ – see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/S1385-8947(03)00150-5

Nomenclature

C	cost function
$\mathcal{D}$	training data set
E_{trn}	RMSE for training set
E_{tst}	RMSE for test set
$\hat{f}$	single aggregated objective function
f_{k}	function correlating *k*th output with inputs
$\mathcal{F}$	feature space
k	index for output variable; also number of folds in cross-validation
K	number of output variables (equal to number of nodes in output layer of network)
l_n	length of *n*th binary segment
L	number of nodes in hidden layer of neural network model
N	number of input variables (equal to number of nodes in input layer of network)
N_{pop}	population size
$N_{\mathrm{gen}}^{\mathrm{max}}$	maximum number of allowable generations for GA
p_{cross}	probability of cross-over
p_{mut}	probability of mutation
P	no. of input–output example patterns used in training
$P(\cdot)$	probability density function
RMSE	root mean squared error
R_{emp}	empirical risk
R_{reg}	regression risk
S_n	decimal equivalent of *n*th binary segment
$\mathbf{x}$	*N*-dimensional input vector
$\mathbf{x}^*$	optimal decision vector
$x_n^{\mathrm{L}}, x_n^{\mathrm{U}}$	lower and upper bounds of *n*th input variable
$\hat{y}_{i,k}$	desired value of *k*th output
$y_{i,k}^{\mathrm{p}}$	neural network predicted *k*th output
$\mathbf{w}$	weight vector
$\mathbf{w}_k$	*k*th function's parameter vector
$\hat{w}_k$	weighting coefficient
$\|\|\mathbf{w}\|\|^2$	Euclidean norm

Greek symbols

α_{EBP}	momentum coefficient
$\boldsymbol{\alpha}, \boldsymbol{\alpha}^*$	vectors of Lagrange's multiplier
ε	precision parameter
$\varepsilon_{\mathrm{loss}}$	loss function parameter
$\varepsilon_{\mathrm{tol}}$	tolerance for termination criterion
η	learning rate
λ	regularization constant
ξ, ξ^*	slack variables
$\hat{\xi}_j$	fitness value of *j*th candidate solution
σ	width of kernel of radial basis function
Φ	function termed feature

conventional empirical modeling approach is that the structure (functional form) of the data-fitting model must be specified a priori. Satisfying this requirement, especially for nonlinearly behaving processes is a cumbersome task since it involves selecting heuristically an appropriate nonlinear model structure from numerous alternatives.

In the last decade, *artificial neural networks* (ANNs) and more recently *support vector regression* (SVR) have emerged as two attractive tools for nonlinear modeling especially in situations where the development of phenomenological or conventional regression models becomes impractical or cumbersome. The most widely utilized ANN paradigm is the *multi-layered perceptron* (MLP) that approximates nonlinear relationships existing between an input set of data (causal process variables) and the corresponding output (dependent variables) data set. The advantages of an ANN-based model are: (i) it can be constructed solely from the historic process input–output data (example set), (ii) detailed knowledge of the process phenomenology is unnecessary for the model development, (iii) a properly trained model possesses excellent generalization ability owing to which it can accurately predict outputs for a new input data set, and (iv) even multiple input–multiple output (MIMO) nonlinear relationships can be approximated simultaneously and easily. Owing to their several attractive characteristics, ANNs have been widely used in chemical engineering applications such as steady state and dynamic process modeling, process identification, yield maximization, nonlinear control, and fault detection and diagnosis (see, e.g., [2–6]). There exists a number of algorithms—each possessing certain positive characteristics—to train an MLP network, for example, the most popular *error-back-propagation* (EBP) [7], *Quickprop* [8] and *resilient back-propagation* (RPROP) [9]. Training of an ANN involves minimizing a nonlinear error function (e.g., root mean squared error, RMSE) that may possess several local minima. Thus, it becomes necessary to employ a heuristic procedure involving multiple training runs to obtain an optimal ANN model whose parameters (weights) correspond to the global or the deepest local minimum of the error function.

In recent years, SVR [10–12], which is a statistical learning theory based machine learning formalism is gaining popularity due to its many attractive features and promising empirical performance. The salient features of SVR are: (i) like ANNs, SVR is an exclusively data-based nonlinear modeling paradigm, (ii) SVR-based models are based on the principle of structural risk minimization (SRM), which equips them with greater potential to generalize, (iii) parameters of an SVR model are obtained by solving a quadratic optimization problem, (iv) the objective function in SVR being of quadratic form it possesses a single minimum thus avoiding the heuristic procedure involved in locating the global or the deepest local minimum on the error surface, and (v) in SVR, the inputs are first nonlinearly mapped into a high-dimensional feature space wherein they are correlated linearly with the output. Although the foundation of the SVR paradigm was laid down in mid-1990s, its chemical engineering applications such as fault detection [13,14] have emerged only recently.

Once an ANN or SVR-based process model is developed, it can be used for process optimization to obtain the optimal values of the process input variables that maximize or minimize a specified objective function. Thus, it is possible to obtain the optimal values of process operating variables, which, for instance, maximize reactant conversion and selectivity of the desired products, or minimize reactor temperature and the selectivity of undesired by-products. Conventionally, various deterministic gradient-based methods [15] are used for optimizing a process model. Most of these methods however require that the objective function should be smooth, continuous, and differentiable. The ANN or SVR models cannot be guaranteed to be smooth especially in regions wherein the input–output data (training set) used in model building are located sparsely. Hence, gradient-based methods cannot be used efficiently for optimizing the input space of an ANN or SVR model. In such situations, an efficient optimization formalism known as *genetic algorithms* (GAs), which is lenient towards the form of the objective function, can be used. The GAs are stochastic optimization formalisms originally developed as the genetic engineering models mimicking population evolution in natural systems [16–18]. GAs follow the "survival-of-the-fittest" and "genetic propagation of characteristics" principles of biological evolution for searching the solution space of an optimization problem. The principal advantages of the GAs are: (i) they are zeroth order optimization method requiring only the scalar values—and not the second and/or the first order derivatives—of the objective function, (ii) capability of handling nonlinear and noisy objective functions, (iii) they perform global search and thus are most likely to arrive at or near the globally optimum solution, and (iv) unlike most gradient-based deterministic optimization methods, GAs do not impose preconditions, such as smoothness, differentiability, and continuity, on the form of the objective function. Due to their several attractive properties, GAs have been extensively used in chemical engineering (see, e.g., Refs. [19–24] and reviews [25–27]).

In the present paper, ANN and SVR formalisms are integrated with GAs to arrive at two hybrid process modeling and optimization strategies. The strategies (henceforth referred to as "ANN-GA" and "SVR-GA") use an ANN or SVR as the nonlinear process modeling paradigm, and the GA for optimizing the input space of the ANN/SVR model such that an improved process performance is realized. To our knowledge, the hybrid involving SVR and GA is being used for the first time for chemical process modeling and optimization. In this study, the ANN-GA [1] and SVR-GA hybrid strategies have been used to model and optimize the pilot plant scale process involving benzene isopropylation using the Hbeta catalyst. The optimized operating conditions leading to maximized yield and selectivity of the desired reaction product (cumene) as given by the two strategies have been compared. The best sets of operating conditions obtained thereby when subjected to experimental validation indeed resulted in significant enhancements in cumene yield and selectivity.

The paper is organized as follows. Section 2 describes process modeling using ANN and SVR methods; the optimization of the models using GAs is explained in Section 3. Usage of ANN-GA and SVR-GA strategies for optimizing the benzene isopropylation process along with the results of experimental verification are described in Section 4. Finally, Section 5 gives a summary of the study.

2. Hybrid process modeling and optimization formalisms

The process optimization objective under consideration is stated as: given catalytic process data comprising values of process operating (input) variables and the corresponding values of process output (response) variables, find the optimal values of input variables such that the pre-specified measures of process performance are simultaneously maximized.

This optimization problem can be formulated as:

$$\text{Maximize } y_k = f_k(\mathbf{x}, \mathbf{w}_k), \quad k = 1, 2, \dots, K \tag{1}$$

where y_k denotes the kth output variable, $\mathbf{x} = [x_1, x_2, \dots, x_N]^T$ is an N-dimensional vector of process operating variables, f_k the function correlating kth output variable with the inputs, and $\mathbf{w}_k$ the parameter vector of function f_k. Eq. (1) describes a multi-objective (MO) optimization problem since it involves simultaneous maximization of K outputs, $\{y_k\}$, $k = 1, 2, \dots, K$. Using aggregation principle (also known as "weighting objective method"), the MO optimization task can be converted into a single objective (SO) optimization by defining

$$\text{Maximize } \hat{f} = \sum_{k=1}^{K} \hat{w}_k y_k = \sum_{k=1}^{K} \hat{w}_k f_k(\mathbf{x}, \mathbf{w}_k) \tag{2}$$

where $\hat{f}$ denotes the single aggregated objective function and $\hat{w}_k$ the weighting coefficient ($0 \le \hat{w}_k \le 1$, $\sum_k \hat{w}_k = 1$). The weighting coefficient, $\hat{w}_k$, signifies the relative importance of kth function in the MO optimization (Eq. (1)). The hybrid strategies fulfill the SO optimization task in two steps. In the first step, an ANN or SVR-based process model, $y_k = f_k(\mathbf{x}, \mathbf{w}_k)$, is developed and in the second step, the input space ($\mathbf{x}$) of the process model is optimized using GAs with a view of maximizing the single aggregated objective function defined in Eq. (2).

2.1. ANN-based modeling

For process modeling, the commonly used feed-forward ANN architecture namely MLP may be employed. The MLP network approximates the nonlinear input–output relationships as defined by, $y_k = f_k(\mathbf{x}, \mathbf{w}_k)$, $k = 1, 2, \dots, K$, where

$\mathbf{w}_k$ is the vector defining network weights. The MLP network usually consists of three layers. The layers described as *input*, *hidden*, and *output* layers comprise *N*, *L*, and *K* number of processing nodes, respectively. Each node in the input (hidden) layer is linked to all the nodes in the hidden (output) layer using weighted connections. The MLP architecture also houses a bias node (with fixed output of +1) in its input and hidden layers; the bias nodes are also connected to all the nodes in the subsequent layer. Usage of bias nodes helps the MLP-approximated function to be positioned anywhere in the *N*-dimensional input space; in their absence, the function is forced to pass through the origin of the *N*-dimensional space. The *N* number of nodes in the input layer is equal to the number of process operating variables, whereas the number of output nodes (*K*) equals the number of process outputs. However, the number of hidden nodes (*L*) is an adjustable parameter whose magnitude is determined by issues such as the desired approximation and generalization performance of the network model. In order that the MLP network accurately approximates the nonlinear relationship existing between the process inputs and the outputs, it needs to be trained in a manner such that a pre-specified error function is minimized. In essence, the MLP training procedure aims at obtaining an optimal weight set $\{\mathbf{w}_k\}$ that minimizes a pre-specified error function. The commonly employed error function is the RMSE defined as

$$\mathrm{RMSE} = \sqrt{\sum_{i=1}^{P}\sum_{k=1}^{K}(\hat{y}_{i,k} - y_{i,k}^{\mathrm{p}})^2} \tag{3}$$

where *P* refers to the number of input–output example patterns used in training, *i* the index of the example pattern (vector) and, $\hat{y}_{i,k}$ and $y_{i,k}^{\mathrm{p}}$ the desired (target) and MLP predicted values of the *k*th output node, respectively. The widely used formalism for the RMSE minimization is the EBP algorithm [7] utilizing a gradient-descent technique known as the *generalized delta rule* (GDR) for iterative updation of weights. The details of the heuristic procedure involved in obtaining an optimal network model possessing good prediction and generalization capabilities can be found in, e.g., Freeman and Skapura [28], Bishop [29], Tambe et al. [2] and Nandi et al. [30]. The EBP training algorithm makes use of two adjustable (free) parameters namely, the learning rate, η $(0 < \eta \le 1)$ and the momentum coefficient, α_{EBP} $(0 < \alpha_{\mathrm{EBP}} \le 1)$. The magnitudes of both these parameters need to be optimized heuristically.

2.2. SVR-based modeling

SVR is an adaptation of a recent statistical learning theory based classification paradigm, namely *support vector machines* [11]. The SVR formulation follows SRM principle, as opposed to the empirical risk minimization (ERM) approach which is commonly employed within statistical machine learning methods and also in training ANNs. In SRM, an upper bound on the generalization error is minimized as opposed to the ERM, which minimizes the prediction error on the training data. This equips the SVR with a greater potential to generalize the input–output relationship learnt during its training phase for making good predictions for new input data. The SVR is a linear method in a high-dimensional feature space, which is nonlinearly related to the input space. Though the linear algorithm works in the high-dimensional feature space, in practice it does not involve any computations in that space, since through the usage of kernels, all necessary computations are performed directly in the input space. In the following, the basic concepts of SVR are introduced. A more detailed description of SVR can be found in Vapnik [10,12], Burges [31], Smola et al. [32] and Schölkopf et al. [33].

Consider a training data set $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_P, y_P)\}$, such that $\mathbf{x}_i \in \mathcal{R}^N$ is a vector of input variables and $y_i \in \mathcal{R}$ the corresponding scalar output (target) value. Here, the modeling objective is to find a regression function, $y = f(\mathbf{x})$, such that it accurately predicts the outputs $\{y\}$ corresponding to a new set of input–output examples, $\{(\mathbf{x}, y)\}$, which are drawn from the same underlying joint probability distribution, $P(\mathbf{x}, y)$, as the training set. To fulfill the stated goal, SVR considers the following linear estimation function:

$$f(\mathbf{x}) = \langle \mathbf{w}, \Phi(\mathbf{x})\rangle + b \tag{4}$$

where $\mathbf{w}$ denotes the weight vector, *b* a constant known as "bias", $\Phi(\mathbf{x})$ a function termed *feature*, and $\langle \mathbf{w}, \Phi(\mathbf{x})\rangle$ the dot product in the feature space, $\mathcal{F}$, such that $\Phi : \mathbf{x} \to \mathcal{F}$, $\mathbf{w} \in \mathcal{F}$. In SVR, the input data vector, $\mathbf{x}$, is mapped into a high-dimensional feature space, $\mathcal{F}$, via a nonlinear mapping function, Φ, and a linear regression is performed in this space for predicting *y*. Thus, the problem of nonlinear regression in lower-dimensional input space $\mathcal{R}^N$ is transformed into a linear regression in the high-dimensional feature space, $\mathcal{F}$. Accordingly, the original optimization problem involving nonlinear regression is transformed into finding the flattest function in the feature space $\mathcal{F}$ and not in the input space, $\mathbf{x}$. The unknown parameters $\mathbf{w}$ and *b* in Eq. (4) are estimated using the training set, $\mathcal{D}$. To avoid over fitting and thereby improving the generalization capability, following regularized functional involving summation of the empirical risk and a complexity term $\|\mathbf{w}\|^2$, is minimized [31]:

$$R_{\mathrm{reg}}[f] = R_{\mathrm{emp}}[f] + \lambda\|\mathbf{w}\|^2 = \sum_{i=1}^{P} C(f(\mathbf{x}_i) - y_i) + \lambda\|\mathbf{w}\|^2 \tag{5}$$

where R_{reg} and R_{emp} denote the regression and empirical risks, respectively, $\|\mathbf{w}\|^2$ the Euclidean norm, $C(\cdot)$ a cost function measuring the empirical risk, and $\lambda > 0$ a regularization constant. For a given function, *f*, the regression risk (test set error), $R_{\mathrm{reg}}(f)$, is the possible error committed by the function *f* in predicting the output corresponding to a new (test) example input vector drawn randomly from the same sample probability distribution, $P(\mathbf{x}, y)$, as the training

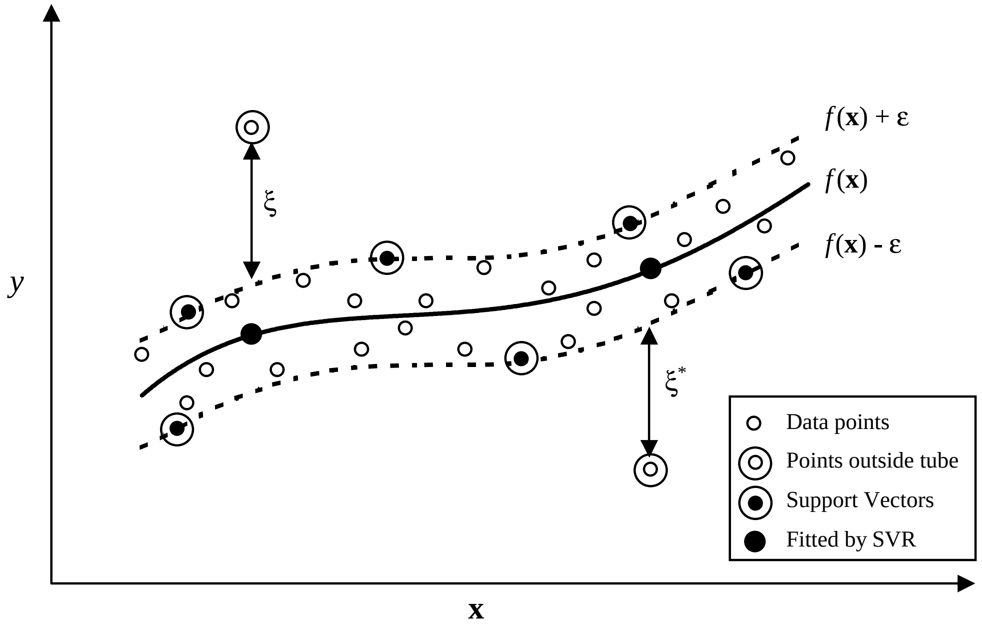

Fig. 1. A schematic illustration of SVR using ε-sensitive loss function.

set. The empirical risk $R_{\text{emp}}(f)$, represents the error (termed "training set error") committed in predicting the outputs of the training set inputs. Minimization task described in Eq. (5) involves: (i) minimization of the empirical loss function $R_{\text{emp}}(f)$ and, (ii) obtaining as small $\mathbf{w}$ as possible, using the training set $\mathcal{D}$. The commonly used loss function is the "ε-insensitive loss function" given as [12]

$$C(f(\mathbf{x}) - y) = \begin{cases} |f(\mathbf{x}) - y| - \varepsilon & \text{for } |f(\mathbf{x}) - y| \geq \varepsilon, \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where ε is a precision parameter representing the radius of the tube located around the regression function (see Fig. 1); the region enclosed by the tube is known as "ε-intensive zone". The SVR algorithm attempts to position the tube around the data as shown in Fig. 1. The optimization criterion in Eq. (6) penalizes those data points whose y values lie more than ε distance away from the fitted function, $f(\mathbf{x})$. In Fig. 1, the size of the stated excess positive and negative deviations are depicted by ξ and ξ^*, respectively, which are termed "slack" variables. Outside the $[-\varepsilon, \varepsilon]$ region, the slack variables assume non-zero values. The SVR fits $f(\mathbf{x})$ to the data in a manner such that: (i) the training error is minimized by minimizing ξ_i and ξ_i^* and, (ii) $||\mathbf{w}||^2$ is minimized to increase the flatness of $f(\mathbf{x})$ or penalize over complexity of the fitting function. Vapnik [7] showed that the following function possessing finite number of parameters can minimize the regularized function in Eq. (5):

$$f(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\alpha}^*) = \sum_{i=1}^{P} (\alpha_i - \alpha_i^*) K(\mathbf{x}, \mathbf{x}_i) + b \tag{7}$$

where α_i and α_i^* ($\geq$0) are the coefficients (Lagrange multipliers) satisfying $\alpha_i \alpha_i^* = 0$, $i = 1, 2, \ldots, P$, and $K(\mathbf{x}, \mathbf{x}_i)$ denotes the so called 'kernel' function describing the dot product in the feature space. The kernel function is defined in terms of the dot product of the mapping function as given by

$$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle \tag{8}$$

The advantage of this formulation (Eqs. (7) and (8)) is that for many choices of the set $\{\Phi_i(\mathbf{x})\}$, including infinite-dimensional sets, the form of K is analytically known and very simple [34]. Accordingly, the dot product in the feature space $\mathcal{F}$ can be computed without actually mapping the vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ into that space (i.e., computation of $\Phi(\mathbf{x}_i)$ and $\Phi(\mathbf{x}_j)$). There exist several choices for the kernel function K; the two commonly used kernel functions, namely, radial basis function (RBF) and nth degree polynomial are defined below in Eqs. (9) and (10), respectively:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\frac{-||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma^2}\right) \tag{9}$$

$$K(\mathbf{x}_i, \mathbf{x}_j) = [1 + (\mathbf{x}_i, \mathbf{x}_j)]^n \tag{10}$$

In Eq. (7), the coefficients α and α* are obtained by solving following quadratic programming problem:

$$\text{Maximize } R(\boldsymbol{\alpha}^*, \boldsymbol{\alpha}) = -\frac{1}{2} \sum_{i,j=1}^{P} (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) K(\mathbf{x}_i, \mathbf{x}_j) - \varepsilon \sum_{i=1}^{P} (\alpha_i^* + \alpha_i) + \sum_{i=1}^{P} y_i (\alpha_i^* - \alpha_i) \tag{11}$$

subject to constraints $0 \leq \alpha_i, \alpha_i^* \leq C \,\forall\, i$ and $\sum_{i=1}^{P} (\alpha_i^* - \alpha_i) = 0$. Having estimated $\boldsymbol{\alpha}, \boldsymbol{\alpha}^*$ and b, using a suitable quadratic programming algorithm, the SVR-based regression function takes the form

$$f(\mathbf{x},\mathbf{w}) = f(\mathbf{x},\boldsymbol{\alpha},\boldsymbol{\alpha}^*) = \sum_{i=1}^{P}(\alpha_i^* - \alpha_i)K(\mathbf{x}_i,\mathbf{x}) + b \qquad (12)$$

where vector **w** is described in terms of the Lagrange multipliers $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}^*$. Owing to the specific character of the above-described quadratic programming problem, only some of the coefficients, $(\alpha_i^* - \alpha_i)$, are non-zero and the corresponding input vectors, $\mathbf{x}_i$, are called support vectors (SVs). The SVs can be thought of as the most informative data points that compress the information content of the training set. The coefficients α_i and α_i^* have an intuitive interpretation as forces pushing and pulling the regression estimate $f(\mathbf{x}_i)$ towards the measurements, y_i [35].

In Eq. (12), the bias parameter, b, can be computed as follows:

$$b = \begin{cases} y_i - f(\mathbf{x}_i)_{b=0} - \varepsilon & \text{for } \alpha_i \in (0, C), \\ y_i - f(\mathbf{x}_i)_{b=0} + \varepsilon & \text{for } \alpha_i^* \in (0, C) \end{cases} \qquad (13)$$

where $\mathbf{x}_i$ and y_i, respectively, denote the ith SV and the corresponding target output. In the SVR formulation, C and ε are two user-specified free parameters; while C represents the trade-off between the model-complexity and the approximation error, ε signifies the width of the ε-insensitive zone used to fit the training data. The stated free parameters together with the specific form of the kernel function control the accuracy and generalization performance of the regression estimate. The procedure of judicious selection of C and ε is explained by Cherkassky and Ma [36].

3. GA-based optimization of ANN and SVR models

The optimization objective underlying the GA-based optimization of an ANN or SVR model is defined as: find the N-dimensional optimal decision variable vector, $\mathbf{x}^* = [x_1^*, x_2^*, \ldots, x_N^*]^{\mathrm{T}}$, representing optimal process conditions such that it simultaneously maximizes process outputs, y_k, $k = 1, 2, \ldots, K$. The corresponding SO function $\hat{f}$ to be maximized by the GA is defined in Eq. (2). In the GA procedure, the search for an optimal solution (decision) vector, $\mathbf{x}^*$, begins from a randomly initialized population of probable (candidate) solutions. The solutions, usually coded in the form of binary strings (chromosomes), are then tested to measure their fitness in fulfilling the optimization objective. Subsequently, a main loop comprising following operations is performed: (i) selection of better (fitter) parent chromosomes, (ii) production of an offspring solution population by crossing over the genetic material between pairs of the fitter parent chromosomes, and (iii) mutation (bit-flipping) of the offspring strings. Implementation of this loop generates a new population of candidate solutions, which as compared to the previous population, usually fares better at fulfilling the optimization objective. The best string that evolves after repeating the above-described loop till convergence, forms the solution to the optimization problem [1,30]. The stepwise procedure for the GA-based optimization of an ANN or SVR model is provided in Appendix A (also see the flowchart in Fig. 2).

4. Modeling and optimization of benzene isopropylation over Hbeta catalyst process

Isopropylation of benzene is an important alkylation reaction in the petrochemical industry for the synthesis of cumene, which is the chief starting material in phenol production. In the last decade, several modifications of the zeolite beta were explored as potential catalysts in cumene synthesis [37–40]. More recently, isopropylation of benzene over Hbeta (protonic form of beta catalyst) was investigated by Sridevi et al. [41]. Beta is a crystalline alumino-silicate catalyst with high silica content and its important characteristic is that it is the only large pore zeolite with chiral pore intersections. It consists of 12-membered rings interconnected by cages formed by intersecting channels. The linear channels have pore opening dimensions of 5.7 Å × 7.5 Å, whereas the tortuous channels with intersections of two linear channels have approximate dimensions of 5.6 Å × 6.5 Å. The catalyst has pore volume of ≈0.2 cm^3/g. In the study by Sridevi et al. [41], a phenomenological model for benzene isopropylation reaction was developed based on the isopropyl alcohol conversion in a continuous down-flow differential packed bed reactor taking into account the secondary reactions such as the dehydration of alcohol. This model however was restricted to the lower conversion (<30%) of the limiting reactant, i.e., isopropyl alcohol, wherein heat and mass transfer resistances in the differential bed were assumed to be negligible. For maximizing yield and selectivity of cumene in the vapor phase alkylation of benzene with isopropyl alcohol over Hbeta catalyst, experiments were also conducted in a pilot plant scale reactor. Isopropylation of benzene involves a main reaction producing cumene and multiple side reactions as described below:

- Main reaction:

  $$\text{Benzene} + \text{isopropyl alcohol} \rightarrow \text{cumene} + \text{water} \qquad \text{(benzene alkylation)}$$

- Secondary reactions:

  $$\text{Cumene} + \text{isopropyl alcohol} \rightarrow p\text{-di-isopropyl benzene} + \text{water} \qquad \text{(cumene alkylation)}$$

  $$p\text{-Di-isopropyl benzene} \rightarrow m\text{-di-isopropyl benzene} \qquad \text{(isomerization)}$$

  $$\text{2-Isopropyl alcohol} \rightarrow \text{di-isopropyl ether} + \text{water} \qquad \text{(alcohol dehydration)}$$

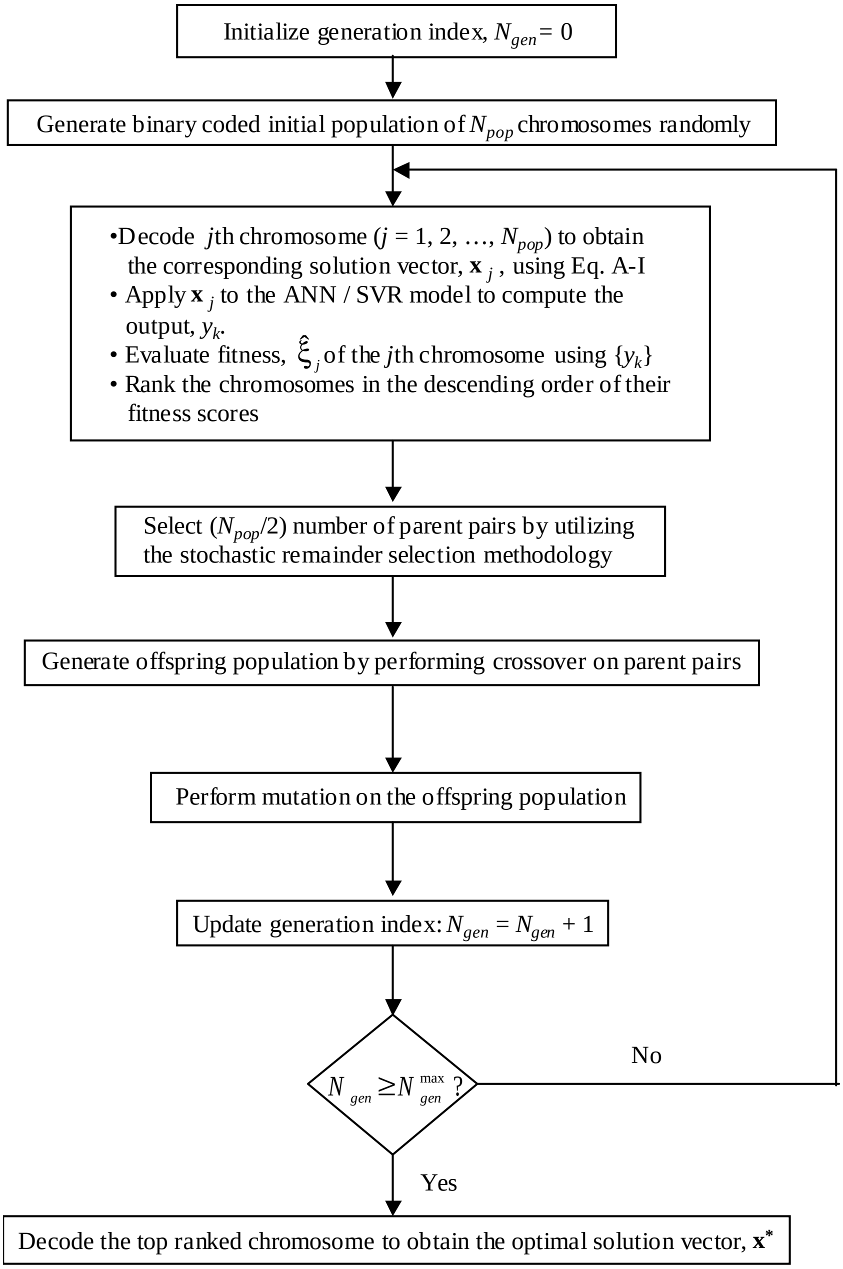
Initialize generation index, $N_{gen} = 0$
Generate binary coded initial population of N_{pop} chromosomes randomly
•Decode jth chromosome ($j = 1, 2, \ldots, N_{pop}$) to obtain the corresponding solution vector, $\mathbf{x}_j$, using Eq. A-I
• Apply $\mathbf{x}_j$ to the ANN / SVR model to compute the output, y_k.
• Evaluate fitness, $\hat{\xi}_j$ of the jth chromosome using $\{y_k\}$
• Rank the chromosomes in the descending order of their fitness scores
Select ($N_{pop}/2$) number of parent pairs by utilizing the stochastic remainder selection methodology
Generate offspring population by performing crossover on parent pairs
Perform mutation on the offspring population
Update generation index: $N_{gen} = N_{gen} + 1$
$N_{gen} \geq N_{gen}^{max}$?
No
Yes
Decode the top ranked chromosome to obtain the optimal solution vector, $\mathbf{x}^*$

Fig. 2. Flowchart of GA-based optimization of ANN/SVR model.

4.1. Materials

Beta catalyst (1.5 mm extrudates with 20% binder) in its active protonated form with Si–Al ratio of 15 was obtained from M/s UCIL, India, and utilized in the reaction; benzene and isopropyl alcohol (isopropanol) were of “analytical reagent” grade.

4.2. Experimental set-up

Vapor phase isopropylation of benzene was carried out in a pilot plant scale stainless steel reactor (see Fig. 3) with a preheater in its upstream and a condenser in the down-stream. The reactor specifications are as follows—material of construction: SS 316, internal diameter (ID): 25 mm, wall thickness: 6 mm, reactor length: 33 cm and catalyst bed height: 10–15 cm. Heating coils are wound around the reactor to provide proper heating and maintain temperature; the reactor is also jacketed with insulation to minimize the heat loss. During reactor operation, the liquid mixture of benzene and isopropyl alcohol was fed to the reactor by a positive displacement pump; hydrogen was used as the carrier gas. The condensed products collected were analyzed with a flame ionization detector (FID) using a “Xylene Master” capillary column fitted to a Shimadzu 15A gas chromatograph (GC).

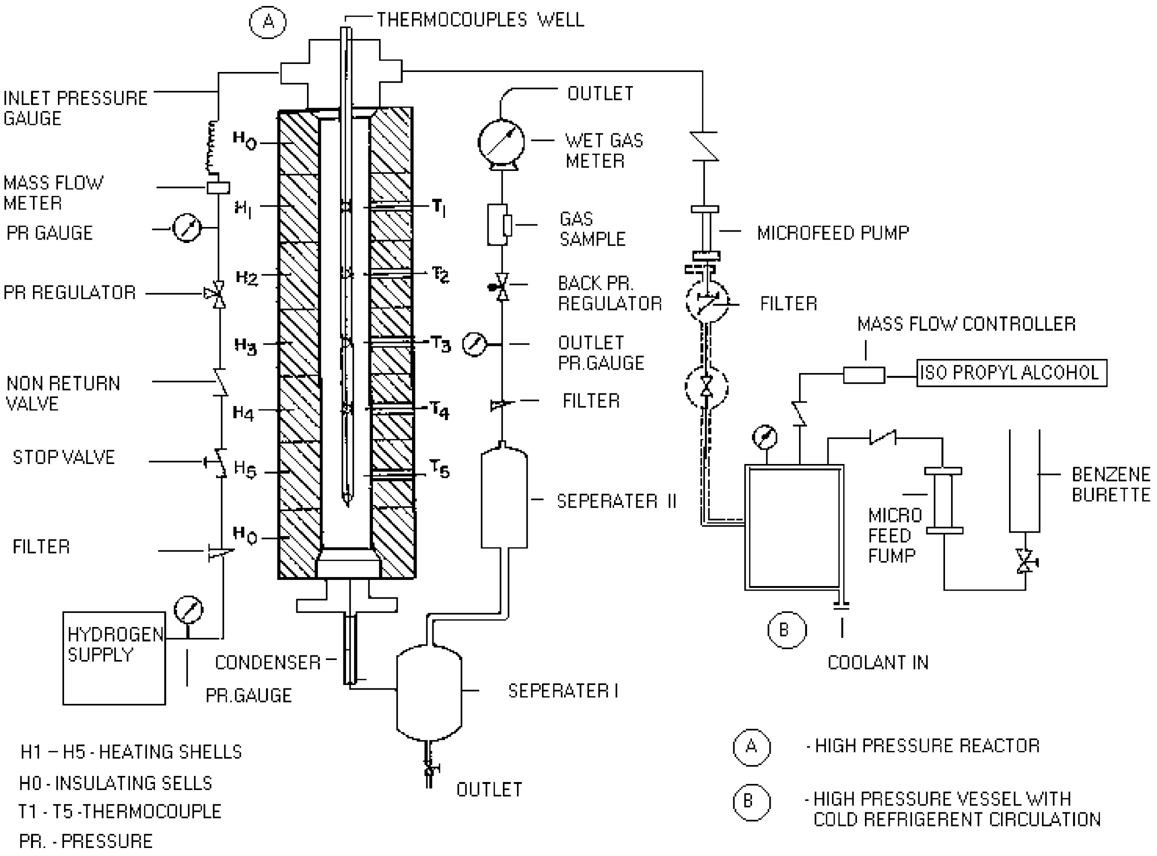

Fig. 3. Schematic illustration of the reactor set-up.

4.3. Modeling and optimization of isopropylation process

The objective of the present case study is to model and optimize the isopropylation pilot plant process with a view to simultaneously maximize yield and selectivity of cumene. Though the formation of cumene via isopropylation of benzene is the main reaction, a series of other components are also produced via side reactions. For developing a phenomenological model for the pilot plant scale integral reactor, it is necessary to consider the detailed kinetics of the stated multiple reactions in the conservation equations. Due to the tedious procedure involved in obtaining the requisite kinetic information, the exclusively data-based ANN-GA and SVR-GA methods were chosen for simultaneously maximizing the yield and selectivity of cumene. Accordingly, four reactor operating variables namely, reaction temperature (x_1), pressure (x_2), benzene to isopropyl alcohol mole ratio (x_3) and weight hourly space velocity (WHSV) (x_4), form the input space of the ANN and SVR-based reactor models. Cumene yield and selectivity defined as y_1 and y_2, respectively, are the output variables and they are evaluated as

$$y_1 = \frac{100 \times \text{weight of cumene formed per unit time}}{\text{weight of isopropyl alcohol fed per unit time}} \quad (14)$$

$$y_2 = \frac{100 \times \text{weight of cumene formed per unit time}}{\text{weight of total aromatics produced per unit time}} \quad (15)$$

A total of 42 experiments (see Table 1) were conducted to seek the effect of varying values of four operating condition variables (x_1–x_4) on cumene yield and selectivity. The lower (x_n^{L}) and upper (x_n^{U}) limits over which the four input variables were varied are: (i) temperature (°C) $(x_1) = [100.0, 280.0]$; (ii) pressure (MPa) $(x_2) = [0.101, 2.533]$; (iii) benzene to isopropyl alcohol mole ratio $(x_3) = [1.0, 10.0]$; (iv) WHSV (h^{-1}) $(x_4) = [2.5, 13.0]$. It can be easily noticed from the reaction stoichiometry that no change in the number of moles occurs and hence pressure will have no significant effect. Low operating pressure should then be favored from a practical consideration. Accordingly, majority of the experiments were conducted at a low pressure (0.101 MPa) with the remaining ones covering the pressure range of 0.404–2.533 MPa. The maximum cumene yield and selectivity values obtained in the 42 experiments were: $y_1 = 22.1$ wt.% ($x_1 = 210\,^{\circ}\mathrm{C}, x_2 = 2.533\,\mathrm{MPa}, x_3 = 6, x_4 = 5\,\mathrm{h}^{-1}$) and, $y_2 = 93.8$ wt.% ($x_1 = 230\,^{\circ}\mathrm{C}, x_2 = 2.533\,\mathrm{MPa}, x_3 = 6, x_4 = 5\,\mathrm{h}^{-1}$), respectively. For maximizing cumene yield and selectivity simultaneously, the SO optimization problem (Eq. (2)) was solved wherein ANN and SVR-based models were built and optimized separately using the GA. This way, it is possible

Table 1
Process data used for development of ANN and SVR-based models

Expt. no.	Temperature (°C)	Pressure (MPa)	Benz/IPA (mole ratio)	WHSV (h^{-1})	Yield (wt.%)	Selectivity (wt.%)
1	110	0.101	8	3.3	0.07	77.03
2[a]	145	0.101	8	3.3	11.6	58.75
3	180	0.101	8	3.3	15.78	79.93
4	210	0.101	8	3.3	17.365	90.72
5	215	0.101	8	3.3	16.09	91.95
6	150	0.404	8	3.3	12.2	65.74
7	135	0.404	8	3.3	12.99	74.58
8[a]	110	0.404	8	3.3	0.71	80.82
9	100	0.404	8	3.3	0.19	75.02
10	110	0.101	10	3.3	0.55	67.74
11	110	0.101	8	3.3	0.24	54.85
12[a]	110	0.101	6	3.3	0.37	53.63
13	110	0.101	3	3.3	0.2	32.13
14	110	0.101	1	3.3	0.14	21.62
15	110	0.101	8	6.8	0.24	54.85
16	110	0.101	8	8	0.15	44.64
17	110	0.101	8	9.5	0.13	37.38
18	110	0.101	8	10.5	0.08	39.3
19[a]	110	0.101	8	12	0.09	39.13
20	110	0.101	8	13	0.07	39.1
21	105	0.101	8	6.8	0.3	70.38
22	110	0.101	8	6.8	0.24	54.85
23	115	0.101	8	6.8	0.35	48.25
24	130	0.101	8	6.8	4.61	76.68
25	185	0.101	8	6.8	9.2	59.23
26	210	0.101	6.5	3.3	20.04	91.8
27	155	0.101	6.5	3.3	16.93	77.4
28[a]	180	0.101	6.5	3.3	20.27	90.9
29	210	0.101	6.5	3.3	19.86	91.9
30	225	0.101	6.5	3.3	19.1	89.3
31	250	0.101	6.5	3.3	17.89	85.2
32	275	0.101	6.5	3.3	17.29	83.1
33	230	0.101	6.5	2.5	20.33	91.1
34	215	0.101	7	5	19.86	91.9
35[a]	215	1.013	7	5	19.54	92
36	215	1.824	7	5	18.68	89.1
37	215	2.533	7	5	17.74	86.8
38	195	2.533	6	5	18.92	85.6
39	210	2.533	6	5	22.1	93.7
40	230	2.533	6	5	22.02	93.8
41	250	2.533	6	5	21.35	90.7
42	280	2.533	6	5	20.48	86.2

[a] These data were used as test set.

to compare the modeling and optimization performance of the ANN-GA and SVR-GA hybrid formalisms.

4.4. *ANN-based modeling of benzene isopropylation process*

An advantage of the ANN-based modeling is that unlike SVR, a comprehensive MIMO model can be constructed for both the process outputs y_1 and y_2. To develop such a model, a three-layered MLP architecture was used. For conducting network training, the experimental data set (see Table 1) was partitioned into training (36 patterns) and test (6 patterns) sets. While the training set was utilized for the EBP based iterative updation of the network weights, the test set was used for simultaneously monitoring the generalization ability of the MLP model. The MLP architecture comprised four input ($N = 4$) and two output ($K = 2$) nodes. For developing an optimal MLP model, its structural parameter, namely the number of hidden nodes (L) and the EBP algorithm-specific parameters, viz., learning rate (η) and momentum coefficient (α_{EBP}) were varied systematically; the effect of random initialization of the network weights also was examined by changing the seed values of the pseudo-random number generator [1]. For choosing an overall optimal network model, the criterion used was least RMSE for the test set. The optimal MLP model that satisfied this criterion has five hidden nodes ($L = 5$), $\eta = 0.7$ and $\alpha_{\mathrm{EBP}} = 0.02$. The values of training set RMSE (E_{trn}) and the test set RMSE (E_{tst}) along with the corresponding values of correlation coefficient (CC) are listed in Table 2.

Table 2
Performance indicators of ANN and SVR models

	ANN-based model				SVR-based model			
	Yield		Selectivity		Yield[a]		Selectivity[b]	
	E_{trn}	E_{tst}	E_{trn}	E_{tst}	E_{trn}	E_{tst}	E_{trn}	E_{tst}
RMSE	0.492	0.438	4.641	4.678	0.842	0.712	6.595	4.986
CC	0.998	0.999	0.974	0.976	0.995	0.999	0.955	0.962

[a] ε-SVR parameters—yield model: $C = 264$; $\gamma = 1/2\sigma^2 = 0.9$; $\varepsilon_{loss} = 0.00001$; $\varepsilon_{tol} = 0.00001$.

[b] ε-SVR parameters—selectivity model: $C = 132$; $\gamma = 1/2\sigma^2 = 0.8$; $\varepsilon_{loss} = 0.00001$; $\varepsilon_{tol} = 0.00001$.

The low and comparable E_{trn} and E_{tst} values indicate good prediction and generalization ability of the trained network model. Good prediction and generalization performance of the model is also evident from the high and comparable CC values corresponding to both the outputs of training and test sets. Fig. 4 (panels a and b) depicts a comparison of the outputs as predicted by the MLP model and their target values.

4.5. SVR-based modeling of isopropylation reaction

Here, two SVR models for cumene yield and selectivity, respectively, were developed using the same training set as used to obtain the ANN-based model. The generalization performance of the SVR models was evaluated using the respective test sets. In the present study, an SVR-implementation known as "ε-SVR" in the LIBSVM software library [42], has been used to develop the two SVR-based models. The LIBSVM package utilizes a fast and efficient method known as sequential minimal optimization (SMO) [43,44] for solving large quadratic programming problems and thereby estimating function parameters $\boldsymbol{\alpha}$, $\boldsymbol{\alpha}^*$ and b (see Eq. (12)). To obtain an optimal SVR model, it is necessary to examine the effects of kernel function and other algorithm-specific parameters; the three kernel functions that were tested are polynomial, RBF and sigmoid. Among these, RBF resulted in the least RMSE values for the training and test sets of the outputs, y_1 and y_2. The number of SVs used by the SVR algorithm for fitting the yield and selectivity models were 30 and 33, respectively. The optimal values of the four SVR-specific parameters namely, width of RBF kernel (σ), cost coefficient (C), loss function parameter (ε_{loss}) and tolerance for termination criterion (ε_{tol}) that minimized the E_{trn} and E_{tst} corresponding to yield and selectivity models are listed in Table 2; also listed are the values of CCs for the training and test set predictions along with the corresponding RMSE values for both the models. A comparison of the SVR model predicted and the corresponding target values of cumene yield and selectivity is depicted in panels c and d of Fig. 4.

In the above-described modeling simulations, the size of the training set (36 patterns) was closer to its statistical limit (30 patterns) below which any prediction made by the model is considered arbitrary. To avoid the possibility of arbitrary predictions arising out of insufficient training data, a method known as "k-fold cross-validation (CV)" is commonly used to select an optimal model. In this approach, the training set is first divided randomly into k equal sized subsets. Next,

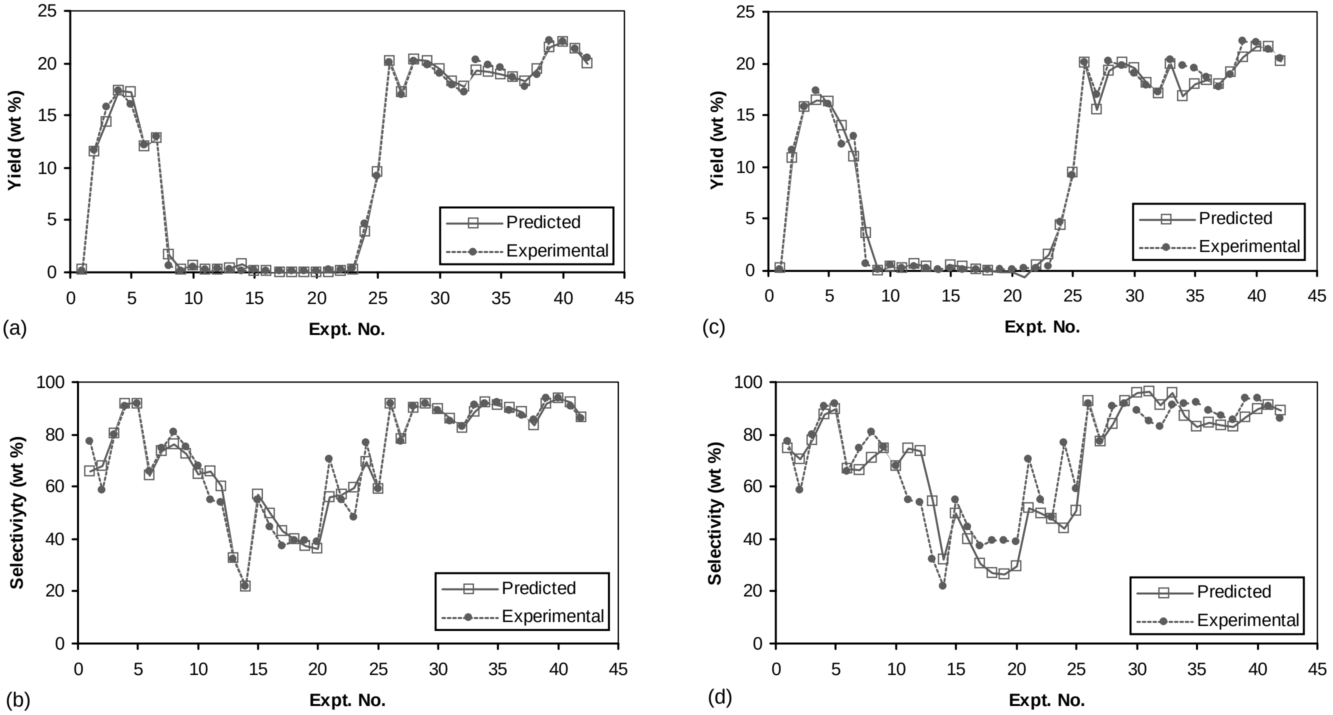

Fig. 4. Yield and selectivity values predicted by the ANN (panels a and b) and SVR (panels c and d) models.

k number of models are constructed by leaving out a different subset each time, with the remaining $(k-1)$ subsets collectively representing the training set. An average of the RMSEs corresponding to the left-out subsets, known as "CV error (E_{CV})" gives an estimate of the model performance if a large-sized data set was available for building the model. Accordingly, in a separate exercise numerous SVR models for cumene yield and selectivity were developed by changing the SVR's algorithm-specific parameters and simultaneously using the CV strategy. For performing CV, the training set was divided into six subsets ($k = 6$) each comprising six patterns. It was observed from these simulations that the SVR-based yield and selectivity models described earlier (see Table 2) also minimized the corresponding CV errors. The minimum E_{CV} values for the cross-validated yield and selectivity models were 0.753 and 4.847, respectively. A comparison of the test set RMSEs (0.712 and 4.986) pertaining to the yield and selectivity models (see Table 2) with the corresponding CV errors (0.753 and 4.847) reveals that they are in close agreement. Such a close match between the CV and test set RMSEs indicate that the size of the training data was adequate for building the SVR models.

4.6. GA-based optimization of the ANN and SVR models

While performing optimization of the input space of the ANN and SVR models, the best values of following GA-specific parameters were chosen heuristically: population size $(N_{\mathrm{pop}}) = 25$, cross-over probability $(\mathrm{p}_{\mathrm{cross}}) = 0.82$, mutation probability $(p_{\mathrm{mut}}) = 0.05$, and maximum number of generations $(N_{\mathrm{gen}}) = 100$. In order to obtain the best set of operating conditions, GA runs were replicated several, i.e., 50 times, using different random number generator seeds. A different seed value generates a dissimilar population of initial candidate solutions, thus assisting in the exhaustive search of the solution space and thereby locating the globally optimum solution. For computing fitness values $(\hat{\xi}_j)$ of the candidate solutions, following fitness function was employed:

$$\hat{\xi}_j = \frac{1}{100}(\hat{w}_1 y_1 + \hat{w}_2 y_2) = \frac{1}{200}(y_1 + y_2),$$
$$\hat{w}_1 = \hat{w}_2 = 0.5, \quad j = 1, 2, \ldots, N_{\mathrm{pop}} \qquad (16)$$

The three best operating condition sets given by the GA-based optimization of the ANN and SVR models are tabulated in Table 3. It is seen from the tabulated optimized values that the ANN-GA method has yielded an overall optimal solution ($x_1^* = 271.5, x_2^* = 0.342, x_3^* = 3.69, x_4^* = 12.83$) maximizing both the yield (24.88%) and selectivity (99.04%) of cumene. Moreover, the best set of operating conditions given by the SVR-GA method ($x_1^* = 270.1, x_2^* = 0.380, x_3^* = 3.55, x_4^* = 13.27$) is similar to that obtained using the ANN-GA method. This set though results in approximately same value (24.8%) of cumene yield as given by the ANN-GA method, the corresponding selectivity value (95.76%) is marginally inferior to that from the ANN-GA method (99.04%). Similar observation can also be made from the other two solutions given by the SVR-GA method. The small differences in the selectivity values maximized by the SVR-GA method are possible if: (i) the GA has found a locally—and not globally—optimum solution, and (ii) the function fitted by the SVR differs from that fitted by the ANN. To test first possibility, input space of the SVR-based selectivity model was searched rigorously using GAs, by restricting the search in the neighborhood of the best solution given by the ANN-GA strategy. However, these simulations showed no significant improvement over the best solution given by the SVR-GA method. This result clearly suggests that the GA has indeed searched the global or the tallest local maximum corresponding to the SVR-based model.

A comparison of the training set RMSE values and the corresponding CC magnitudes in respect of ANN and SVR models (see Table 2) reveals that there exists a small difference in the values. It can thus be inferred that the functions fitted by the ANN and SVR are indeed different. These differences however affect the optimized solutions given by the two methods only marginally. The differences in the functions fitted by the ANN and SVR can arise owing to very different fitting strategies employed by the two methods—while the ANNs approximate the nonlinear input–output relationships directly, in SVR the inputs are first mapped into a high-dimensional feature space and then correlated linearly with the output. It is also possible that ANN and SVR react differently to the noise, which is inherent to the experimental data.

4.7. Experimental verification of GA-optimized solutions

It is noticed from the optimized reactor conditions listed in Table 3 that solutions 1 and 3 given by the ANN-GA method and the three solutions provided by the SVR-GA method fall in a narrow range. However, ANN-GA-based solutions result in slightly higher (up to 3.31%) selectivity values when compared to the SVR-GA-based best solution. To verify their validity, all the three sets of optimized operating conditions given by the ANN-GA method were subjected to experimental verification and the results obtained thereby are listed in Table 4. As can be observed from the tabulated values, the experimental results match the predictions of the ANN-GA method with excellent accuracy. In fact, except the yield value in the second verification experiment, which shows 4.41% error from its GA-maximized value of 24.84%, all other yield and selectivity values validate the corresponding GA-maximized values within 1% accuracy. It is possible to explain the occurrence of 4.41% error on the basis of sparseness in the experimental data. As noted previously, most experiments were conducted at a low pressure (0.101 MPa) and therefore experimental data are sparsely located in the higher pressure (>0.101 MPa) range. In regions of sparse data, it is possible that the function fitted by the ANN or SVR shows minor deviations from the

Table 3
Optimized process conditions given by ANN-GA and SVR-GA methodologies

Soln. no.	ANN-GA						SVR-GA					
	Optimized inputs				Maximized outputs		Optimized inputs				Maximized outputs	
	Temperature (°C) (x_1^*)	Pressure (MPa) (x_2^*)	Benz/IPA (mole ratio) (x_3^*)	WHSV (h^{-1}) (x_4^*)	Yield (wt.%) (y_1^*)	Selectivity (wt.%) (y_2^*)	Temperature (°C) (x_1^*)	Pressure (MPa) (x_2^*)	Benz/IPA (mole ratio) (x_3^*)	WHSV (h^{-1}) (x_4^*)	Yield (wt.%) (y_1^*)	Selectivity (wt.%) (y_2^*)
1	271.5	0.343	3.69	12.83	24.88	99.04	270.1	0.380	3.55	13.27	24.8	95.76
2	267.2	0.1588	4.05	12.83	24.84	98.90	266.1	0.392	3.95	12.93	24.0	93.8
3	270.08	0.366	4.05	11.76	24.82	98.74	266.1	0.368	3.85	12.35	24.63	93.96

Table 4
Results of experimental verification

Expt. no.	Experimental conditions				Yield (output 1)			Selectivity (output 2)		
	Temperature (°C)	Pressure (MPa)	Benz/IPA (mole ratio)	WHSV (h^{-1})	GA-maximized value (wt.%)	Exptl. value (wt.%)	Error (%)	GA-maximized value (wt.%)	Exptl. value (wt.%)	Error (%)
1	271.5	0.344	3.7	12.8	24.88	24.69	0.77	99.04	98.98	0.06
2	267.2	0.162	4.0	12.8	24.84	23.79	4.41	98.90	98.70	0.20
3	270.0	0.365	4.0	11.8	24.82	24.58	0.98	98.74	98.65	0.09

true trend line. Consequently, the solution searched by the GA also deviates from its true optimum. In essence, the sparseness of the experimental data in the pressure range of 0.101–0.404 MPa may have resulted in the 4.41% error for the second validation experiment, wherein GA-optimized pressure value was 0.162 MPa. From the experimental data used for building the ANN and SVR methods, it is seen that the maximum values of the cumene yield and selectivity were 22.1% (expt. no. 39) and 93.8% (expt. no. 40), respectively. A comparison of these values with those from the verification experiments reveal that the GA-based optimized conditions have simultaneously improved the cumene yield and selectivity by 2.59 and 5.1%, respectively.

5. Conclusion

In this paper, two hybrid process modeling and optimization strategies integrating ANNs/SVR with the GAs have been employed for modeling and optimization of benzene isopropylation pilot plant scale catalytic process. The SVR is a novel machine learning based nonlinear modeling paradigm possessing certain unique features such as its formulation involves solution of a quadratic programming problem possessing a single minimum. Thus, the rigorous heuristic involved in locating the global minimum (as in ANNs) is avoided in SVR-implementation. In the two hybrid strategies, a process model is developed using an ANN or SVR method following which the input space of that model is optimized using GAs such that the process performance is maximized. The hybrid approach involving SVR and GAs is a new method wherein, similar to the ANN-GA formalism, process modeling and optimization can be conducted exclusively using historic process data. Using ANN-GA and SVR-GA strategies separately, a number of optimized sets of operating conditions, which simultaneously maximize the yield and selectivity of cumene (desired product of isopropylation reaction) were obtained. It was observed that the best sets of process operating conditions given by the two strategies were similar. Finally, the optimized solutions given by the ANN-GA method when subjected to experimental verification, matched the maximized cumene yield and selectivity values with excellent accuracy.

Acknowledgements

SN thanks Council of Scientific and Industrial Research (CSIR), the Govt. of India, New Delhi, for a Senior Research Fellowship (SRF).

Appendix A. Stepwise procedure for the GA-based optimization of an ANN or SVR model

- *Step* 1 (initialization): Set generation index (N_{gen}) to zero and generate a population of N_{pop} binary strings (chromosomes) randomly; each string consisting of l_{chr} bits is divided into N segments equal to the number of decision (input) variables to be optimized.
- *Step* 2 (fitness computation): Decode jth binary-coded chromosome ($j = 1, 2, \ldots, N_{pop}$) to obtain its equivalent decimal-valued solution vector ($\mathbf{x}_j$) using

$$x_{j,n} = x_n^{\mathrm{L}} + \frac{(x_n^{\mathrm{U}} - x_n^{\mathrm{L}})S_n}{2^{l_n} - 1}, \qquad \sum_{n=1}^{N} l_n = l_{\mathrm{chr}} \qquad \text{(A.1)}$$

  where x_n^{L} and x_n^{U}, respectively, refer to the lower and upper bounds on x_{n}, l_n the length of nth binary segment and S_n the decimal equivalent of the nth binary segment. Next, depending upon the model to be optimized, vector $\mathbf{x}_j$ is used to compute the output of an ANN or SVR model; this output is subsequently used to calculate the fitness value ($\hat{\xi}_j$) of the jth candidate solution (see Eq. (16)). Upon computing the fitness scores of N_{pop} candidate solutions in the current population, the solutions are ranked in the decreasing order of their fitness scores.
- *Step* 3 (parent selection): From the current population, choose N_{pop} number of parent chromosomes to form the mating pool. The members of this pool, which are used to produce offspring population, possess relatively high fitness scores. The commonly used parent selection techniques are the *Roulette–Wheel* (RW) method or its more stable variant known as the *stochastic remainder selection* (SRS) [18].
- *Step* 4 (cross-over): Randomly select $N_{pop}/2$ number of parent pairs from the mating pool and perform cross-over operation on each pair with probability equal to p_{cross} ($0 < p_{cross} \leq 1$). In cross-over, parent strings are cut at

the same randomly chosen cross-over point to obtain two substrings per parent. The substrings are then mutually exchanged between the parents and combined to form two offspring chromosomes. This cross-over operation is performed on all the parent pairs in the mating pool to obtain an offspring population of the size of the mating pool.

- *Step* 5 (mutation): Flip (mutate) the bits of the offspring strings where the probability of a bit getting flipped (zero to one or vice versa) is equal to p_{mut}. The recommended range of p_{mut} is [0.01–0.05].
- *Step* 6: Increment the generation index: $N_{\text{gen}} = N_{\text{gen}} + 1$.
- *Step* 7: Repeat steps 2–6 on the newly generated offspring strings until convergence is achieved. The criteria for the convergence are: N_{gen} exceeds its maximum limit ($N_{\text{gen}}^{\max}$), or the fitness score of the best (fittest) string in the offspring population undergoes a very small or no change over successive generations. After convergence, the string possessing highest fitness value is decoded to obtain the optimized decision variable vector, $\mathbf{x}^*$.

References

[1] S. Nandi, P. Mukherjee, S.S. Tambe, R. Kumar, B.D. Kulkarni, Reaction modeling and optimization using neural networks and genetic algorithms: case study involving TS-1 catalyzed hydroxylation of benzene, Ind. Eng. Chem. Res. 41 (2002) 2159–2169.

[2] S.S. Tambe, B.D. Kulkarni, P.B. Deshpande, Elements of Artificial Neural Networks with Selected Applications in Chemical Engineering, and Chemical and Biological Sciences, Simulations and Advanced Controls, Louisville, KY, 1996.

[3] A.B. Bulsari (Ed.), Neural Networks for Chemical Engineers, Elsevier, Amsterdam, 1995.

[4] A.B. Bulsari, Applications of artificial neural networks in process engineering, J. Syst. Eng. 4 (1994) 131–170.

[5] K. Huang, X.-L. Zhan, F.-Q. Chen, D.-W. Lü, Catalyst design for methane oxidative coupling by using artificial neural network and hybrid genetic algorithm, Chem. Eng. Sci. 58 (2003) 81–87.

[6] G. Stephanopoulos, C. Han, Intelligent systems in process engineering: a review, Comp. Chem. Eng. 20 (1996) 743–791.

[7] D. Rumelhart, G. Hinton, R. Williams, Learning representations by backpropagating errors, Nature 323 (1986) 533.

[8] S.E. Fahlman, An empirical study of learning speed in back-propagation networks, Technical Report CMU-CS-88-162, Computer Science Department, Carnegie-Mellon University, Pittsburgh, PA, 1988.

[9] M. Riedmiller, H. Braun, A direct adaptive method for faster backpropagation learning: the RPROP algorithm, in: Proceedings of the IEEE International Conference On Neural Networks, San Francisco, CA, March 28–April 1, 1993.

[10] V. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 1995.

[11] V. Vapnik, S. Golowich, A. Smola, Support vector method for function approximation, regression estimation and signal processing, Adv. Neural Inform. Process. Syst. 9 (1996) 281–287.

[12] V. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.

[13] M. Agarwal, A.M. Jade, V.K. Jayaraman, B.D. Kulkarni, Support vector machines: a useful tool for process engineering applications, Chem. Eng. Prog. 99 (1) (2003) 57–62.

[14] L.B. Jack, A.K. Nandi, Fault detection using support vector machines and artificial neural networks augmented by genetic algorithms, Mech. Syst. Signal Process. 16 (2002) 372–390.

[15] T.F. Edgar, D.M. Himmelblau, Optimization of Chemical Processes, McGraw-Hill, Singapore, 1989.

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

[17] L. Davis (Ed.), Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York, 1991.

[18] K. Deb, Optimization for Engineering Design: Algorithms and Examples, Prentice-Hall, New Delhi, 1995.

[19] S.P. Ramanathan, S. Mukherjee, R.K. Dahule, S. Ghosh, I. Rahman, S.S. Tambe, D.D. Ravetkar, B.D. Kulkarni, Optimization of continuous distillation columns using stochastic optimization approaches, Trans. Inst. Chem. Eng. 79 (2001) 310–321.

[20] U. Rodemerck, D. Wolf, U.V. Buyevskaya, P. Claus, S. Senkan, M. Baerns, High throughput synthesis and screening of catalytic materials case study of the search for a low temperature catalyst for oxidation of low concentration propane, Chem. Eng. J. 82 (2001) 3–11.

[21] V.S. Sumanwar, V.K. Jayaraman, B.D. Kulkarni, H.S. Kusumakar, K. Gupta, J. Rajesh, Solution of constrained optimization problems by multiobjective genetic algorithms, Comp. Chem. Eng. 26 (2002) 1481–1492.

[22] C.D. Immanuel, F.J. Doyle III, Open loop control of particle size distribution in semi-batch emulsion copolymerization using genetic algorithms, Chem. Eng. Sci. 57 (2002) 4415–4427.

[23] J.M. Nougués, M.D. Gran, L. Puigjaner, Parameter estimation with genetic algorithms in control of fed-batch reactor, Chem. Eng. Process. 41 (2002) 303–309.

[24] K. Yee, A.K. Ray, G.P. Rangaiah, Multiobjective optimization of an industrial styrene reactor, Comp. Chem. Eng. 27 (2003) 111–130.

[25] C.B. Lacasius, G. Kateman, Understanding and using genetic algorithms. Part I. Concepts, properties and context, Chem. Intell. Lab. Syst. 19 (1993) 1–33.

[26] C.B. Lucasius, G. Kateman, Understanding and using genetic algorithms. Part II. Representation, configuration and hybridization, Chem. Intell. Lab. Syst. 25 (1994) 99–145.

[27] V. Venkatasubramanian, A. Sundaram, Genetic algorithms: introduction and applications in: Encyclopaedia of Computational Chemistry, Wiley, Chichester, UK, 1998.

[28] J.A. Freeman, D.M. Skapura, Neural Networks: Algorithms, Applications and Programming Techniques, Addision-Wesley, Reading, MA, 1991.

[29] C.M. Bishop, Neural networks and their applications, Rev. Sci. Instrum. 65 (1994) 1803.

[30] S. Nandi, S. Ghosh, S.S. Tambe, B.D. Kulkarni, Artificial neural-network-assisted stochastic process optimization strategies, AIChE J. 47 (2001) 126–141.

[31] C. Burges, A tutorial on support vector machines for pattern recognition, Data Mining Knowl. Disc. 2 (2) (1998) 1–47.

[32] A. Smola, B. Schölkopf, K.R. Müller, The connection between regularization operators and support vector kernels, Neural Netw. 11 (1998) 637–649.

[33] B. Schölkopf, J.C. Platt, J. Shawe-Taylor, A.J. Smola, R.C. Williamson, Estimating support of a high-dimensional distribution, Neural Comput. 13 (2001) 1443–1471.

[34] S. Mukherjee, E. Osuna, F. Girosi, Nonlinear prediction of chaotic time series using support vector machines, in: Proceedings of the IEEE Workshop on Neural Networks for Signal Processing 7 (IEEE NNSP'97), 1997, pp. 511–519.

[35] K.R. Müller, A. Smola, G. Ratsch, B. Schölkopf, J. Kohlmorgen, V. Vapnik, Predicting time series with support vector machines, in: W. Gerstner, A. Germond, M. Hasler, J.-D. Nicoud (Eds.), Proceedings of the Conference on Artificial Neural Networks—ICANN'97, Lecture Notes on Computer Science, vol. 1327, Springer, Berlin, 1997, pp. 999–1004.

[36] V. Cherkassky, Y. Ma, Practical selection of SVM parameters and noise estimation for SVM regression, Neurocomputing (special issue on SVM), 2002, submitted for publication

http://www.ece.umu.edu/users/cherkass/N2002-SI-SUM-13-whole.pdf.

[37] C. Perego, G. Pazzuconi, G. Girotti, G. Terzoni, Process for the preparation of cumene, Eur. Pat. Appl. EP629599 A1 (December 21, 1994).

[38] F. Cavani, G. Girotti, V. Arrigoni, G. Terzoni, Alkylation catalyst for aromatic compounds for lower olefins, US Patent 5650547 A (July 22, 1997).

[39] A. Geatti, M. Lenarda, L. Storaro, R. Ganzerla, M. Perissinotto, Solid acid catalysts from clays: cumene synthesis by benzene alkylation with propene catalyzed by cation exchanged aluminium pillared clays, J. Mol. Catal. A 121 (1997) 111–118.

[40] G.R. Meima, Advances in cumene production, CATTECH, June 5–12, 1998.

[41] U. Sridevi, B.K.B. Rao, N.C. Pradhan, S.S. Tambe, C.V. Satyanarayana, B.S. Rao, Kinetics of isopropylation of benzene over Hbeta catalyst, Ind. Eng. Chem. Res. 40 (2001) 3133–3138.

[42] C. Chang, C. Lin, LIBSVM: a library for support vector machines, 2001. http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

[43] T. Joachims, Making large-scale SVM learning practical, in: B. Schölkopf, C.J.C. Burges, A.J. Smola (Eds.), Advances in Kernel Methods—Support Vector Learning, MIT Press, Cambridge, MA, 1998.

[44] J.C. Platt, Fast training of support vector machines using sequential minimal optimization, in: B. Schölkopf, C.J.C. Burges, A.J. Smola (Eds.), Advances in Kernel Methods—Support Vector Learning, MIT Press, Cambridge, MA, 1998.