

Received May 17, 2020, accepted May 28, 2020, date of publication June 4, 2020, date of current version June 26, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2998988

# A Novel Hybrid Machine Learning Algorithm for Limited and Big Data Modeling With Application in Industry 4.0

HAMID KHAYYAM<sup>1</sup>, (Senior Member, IEEE), ALI JAMALI<sup>2</sup>,  
ALIREZA BAB-HADIASHAR<sup>1</sup>, (Senior Member, IEEE), THOMAS ESCH<sup>3</sup>,  
SEERAM RAMAKRISHNA<sup>4</sup>, (Senior Member, IEEE), MAHDI JALILI<sup>1</sup>, (Senior Member, IEEE),  
AND MINOO NAEBE<sup>5</sup>

<sup>1</sup>School of Engineering, RMIT University, Melbourne, VIC 3000, Australia

<sup>2</sup>Faculty of Mechanical Engineering, University of Guilan, Rasht 41447-84475, Iran

<sup>3</sup>Mechanical Engineering, FH Aachen University of Applied Sciences, 52066 Aachen, Germany

<sup>4</sup>Department of Mechanical Engineering, National University of Singapore, Singapore 119077

<sup>5</sup>Institute for Frontier Materials, Deakin University, Burwood, VIC 3217, Australia

Corresponding author: Hamid Khayyam (hamid.khayyam@rmit.edu.au)

**ABSTRACT** To meet the challenges of manufacturing smart products, the manufacturing plants have been radically changed to become smart factories underpinned by industry 4.0 technologies. The transformation is assisted by employment of machine learning techniques that can deal with modeling both big or limited data. This manuscript reviews these concepts and present a case study that demonstrates the use of a novel intelligent hybrid algorithms for Industry 4.0 applications with limited data. In particular, an intelligent algorithm is proposed for robust data modeling of nonlinear systems based on input-output data. In our approach, a novel hybrid data-driven combining the Group-Method of Data-Handling and Singular-Value Decomposition is adapted to find an offline deterministic model combined with Pareto multi-objective optimization to overcome the overfitting issue. An Unscented-Kalman-Filter is also incorporated to update the coefficient of the deterministic model and increase its robustness against data uncertainties. The effectiveness of the proposed method is examined on a set of real industrial measurements.

**INDEX TERMS** Industry 4.0, big data modeling, limited data modeling, multi-objective optimization.

## NOMENCLATURE

AC	Actor-Critic	KNN	K-Nearest Neighbor
CNN	Convolutional Neural Network	LDA	Linear Discriminant Analysis
DL	Deep Learning	MAB	Multi-Armed Bandit
DNN	Deep Neural Networks	ML	Machine Learning
D&PL	Distributed and Parallel Learning	MCS	Monte Carlo Simulation
DBN	Deep Belief Networks	NN	Neural Networks
DDPG	Deep Deterministic Policy Gradient	NSGA	Nondominated Sorting Genetic Algorithm
DOE	Design of Experiences	PCA	Principal Component Analysis
DQN	Deep Q-Network	PDF	Probability Density Function
DT	Decision Tree	PCA	Principal Component Analysis k-Means
GMDH	Group Method of Data Handling	PG	Policy Gradient
GA	Genetic Algorithm	QDA	Quadratic Discriminant Analysis
GPR	Gaussian Process Regression	ResNet	Residual Network
IL	Incremental Learning	RNN	Recurrent Neural Network
JK	JackKnife	RF	Random Forest
		RL	Reinforcement Learning
		RVM	Relevance Vector Machine
		SARSA	State-Action-Reward-State-Action

The associate editor coordinating the review of this manuscript and approving it for publication was Qichun Zhang<sup>1</sup>.

SVD	Singular Value Decomposition
SOM	Self Organised Maps
SVM	Support Vector Machine
SVR	Support Vector Regression
TL	Transfer Learning
UKF	Unscented Kalman Filter

## I. INTRODUCTION

Current key challenges of manufacturing processes can be summarized as (i) adoption of advanced manufacturing technologies, (ii) growing importance of manufacturing of high value-added products, (iii) increasing process complexity, uncertainty and dynamism, (iv) utilizing advanced knowledge, data science, and AI systems [1]–[3]. Among their challenges, the process of data collection could be more challenging and resource intensive, due to being costly, time consuming and compute-intensive. As such, the amount of data needed to build accurate models is often limited. System identification, decision making, and predictive analytics based on limited data, may reduce the production yields, increases the production costs or decreases enterprise competitiveness. In the case of limited data, the size of the dataset is not enough to train a reliable model. For example, when the size of the training dataset is less than the number of unknown parameters of the model, e.g. the unknown weights of an artificial neural network architecture, we face the limited data challenge and conventional learning tasks might not property work. In such case, one still needs to develop appropriate data models with small variance of forecasting error and good accuracy based on these small data sets.

On the other hand, in some cases one has to deal with big data, where the data produced by the system has big volume, variety, veracity and velocity. Big data is an ambiguous term to define in data sizes that are difficult to manage, observe, acquisition, store, process and analyses, using prevalent database tools. These processes are often too complex with highly dynamical uncertainties, demanding heavy computational effort to find a simple model. Machine Learning (ML) offers effective solutions to solve challenging issues in various industrial applications [4]. ML includes computer algorithms and statistical methods required for data-driven control, estimation, prediction, classification, or clustering. Although ML is effective in many ways, some of the existing ML techniques have some limitations, such as over-fitting, under-fitting due to the nature of the data, poor generalizability and poor long-term prediction ability [5], [6]. Hybrid ML technique can potentially capture more characteristics of complex systems to overcome these limitations. Hybrid ML works based on developing algorithms to couple model-based and data-driven learning system. Despite research in hybrid data-driven ML techniques, they are not yet widely used mainly because of high computational complexity [7], [8]. This manuscript aims to provide a review of common data modeling techniques for limited and big data based on ML approaches with list of advantages and disadvantages. It also provides an overview of some recent research works on data

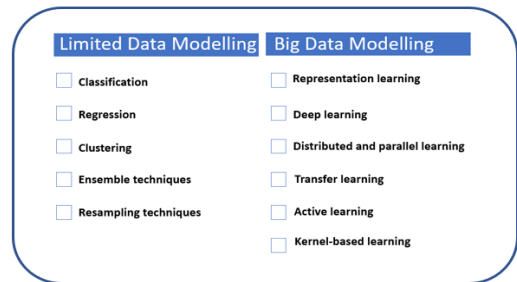


FIGURE 1. Common modeling techniques for limited or big data scenarios.

modeling techniques with limited or big data constraint for various industrial applications as illustrated in Fig.1.

Based on the advantages and limitations of the ML, the paper introduces an intelligent hybrid data-driven algorithm that is robust against limited and big data constraints to reduce the computational cost and increase the estimation accuracy. This manuscript has two main parts. The first part provides brief review of some well-known ML techniques. Then, we introduce a hybrid algorithm and apply it to a modeling task with limited data constraints.

This manuscript is organized as follows. Section 2 and 3 present modeling techniques for limited and big data scenarios. Section 4 introduces a novel intelligent algorithm for robust modeling of nonlinear process. Section 5 shows an industrial case study that can be modeled by the proposed intelligent algorithm. Conclusion remarks are provided in section 6.

## II. MACHINE LEARNING TECHNIQUES TO MODEL LIMITED OR BIG DATA

One of the distinctions of industry 4.0 concept is data-informed decision making. In some applications, data collection can be a challenging and expensive task, leading to limited data problem. The level of required modeling accuracy depends on sample size. To make a reliable statistical test under small sample size, Design of Experiments (DOE) methods have been proposed, such as Response Surface, Taguchi and Factorial [9]. Development of DOE based on Taguchi methods to reduce industrial experimental tests requires reliable data modeling [10], [11]. In some other industrial applications, large-scale data is produced, and sophisticated machine learning needs to be employed to process the data. In the following sections, we review a number of machine learning techniques, from conventional to recent, that are often used in industrial applications.

To process, analyze, predict and support decision-making based on limited or big data, Machine Learning (ML) techniques can be used. ML techniques are able to observe, store and model data with high nonlinearity and uncertainty. ML can easily identify trends and patterns associated with black-box (or gray-box) in complex systems. There are different techniques in ML, and the choice of the method depends on factors, such as the nature of the dataset, the scope of problem, and the desired outcomes. Usual ML tasks include regression, modeling, prediction, classification and clustering. Generally,

there are three major ML categories: supervised, unsupervised and reinforcement learning. Recently, semi-supervised learning methods are also increasingly developed and used in many applications.

Supervised learning works in preparing a model through labeled training data until the model achieves a desired level of accuracy on the training data. Supervised learning is usually performed when the final values of the output variable or the class labels are known, and one can produce an error function between the output of the model and that of the system. Supervised learning is used in classification (predicting a label) or regression classification (predicting a quantity). Some supervised learning techniques are: rule-based systems, regularization, Bayesian, ensemble, Neural Networks (NNs), instance-based, decision trees and explicit regression. Many of these supervised learning techniques are used in both regression and classification in industry cases.

Unsupervised learning is used to find patterns or hidden structures in datasets that have not been categorized or labeled. Unsupervised learning typically focuses on exploratory analysis, dimensionality reduction methods, and feature extraction. Examples of unsupervised learning include: Principal Component Analysis (PCA), k-Means, Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), and Self Organized Maps (SOM). Reinforcement Learning (RL) differs from supervised and unsupervised learning and it works with data from a dynamic environment. The goal of reinforcement learning is to find the superlative sequence of actions that will generate the optimal outcome and controlling but not to cluster or label data. The way reinforcement learning solves this problem is by allowing a software called agent to explore, interact with, and learn from the environment (system). The idea is a trade-off between exploration and exploitation. Within the agent, there is a function that takes in state observations (the inputs) and maps them to actions (the outputs). This is the single function that will take the place of all of the individual subcomponents of the control system. In the RL nomenclature, this function is called the policy. Given a set of observations, the policy decides which action to take. The policies in RL algorithms can be divided by off-policy: PG, SARSA and on-policy: DQN, AC, DDPG and MAB.

Some of popular techniques of clustering, classification, regression and advanced machine learning are discussed in the following.

### A. CLASSIFICATION

Classification is one of the most popular methods in ML with many potential applications in industry settings. Classification is a data mining technique to predict categorical class labels based on the observations. Appropriate models need to be constructed to define imperative data classes in classification. In order to build a classifier, one often divides the data into three parts: training data, test and validation. The training dataset is used to find unknown parameters of the model, which is verified using test data and validated using validation data.

If  $\mathbf{X} = X_{i=1}^N \subset R^q$  and a corresponding set of labels  $\mathbf{Y} = Y_{i=1}^N \subset R$ , then  $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$  is training set of  $q$  dimensional patterns.

The classifier can assign an appropriate label to all unlabeled patterns, i.e. allocate them to the most appropriate class. In order to improve the classifier robustness and generalizability, a number of approaches have been developed. Dimensionality reduction through feature selection is an effective technique to improve generalizability of classification tasks. Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA) are two frequently used feature extraction methods [12], [13]. There are many classification approaches each with their own pros and cons. One of the frequently used simple yet powerful classification methods is K-Nearest Neighbor (KNN) classifier [14]. Support Vector Machine (SVM) is a novel supervised machine learning which can be used for both classification or regression challenges. SVM builds a separating hyperplane that maximizes the margin between the two classes [15]. SVM have high prediction accuracy, non-parametric, robust to outliers and low prediction time [5]. However, some limitation of SVM are: high computational cost, poor uncertainty management ability and requires cross validation procedure to determine hyper-parameters [5]. Relevance Vector Machine (RVM) employs a Bayesian framework to infer the weights, with which the Probability Density Function (PDF)s of the outputs instead of point estimates can be obtained. RVM provides performance comparable to SVM, while utilizing arbitrary kernel functions with high sparsity and offering probabilistic predictions [16]. High sparsity means that a significant number of weights are zero, leading to more computationally efficient models. Advantages of RVM include ability of generating PDF directly, being non-parametric, and ability of realizing high sparsity and avoiding cross validation process. It has however some limitations including large volumes of data is required for modeling, huge time and memory are consumed during the training process, easily falling into a local optimum and potentially causing over-fitting [6].

A Decision Tree (DT) has a tree structure (flowchart) with numerous nodes and branches. It is a fast and easy method with decent performance in many classification tasks. Bayesian classification is a statistical model and learns the distributions of instance to predict class membership probabilities.

### B. REGRESSION

The regression technique is very close to classification technique and the difference is to find a pattern to determine numerical values. The regression modeling tries to find the relationship between a response or dependent variable  $y$  and independent variables  $x_1, \dots, x_k$ . In other words, the regression modeling aims is to develop a model to create a prediction of the response variable(s), based on independent variables. Linear regression has been massively used in the literature for fitting a measurable response variable as a function of one or more independent predictor variables. The least square regression [17] method is the most commonly

used approach for fitting a regression line. NNs, as a tool for nonlinear regression, is considered as a well-developed technique that can be applied to all kinds of areas in process identification, control, and model prediction [18]. Support Vector Regression (SVR) is an extension of SVM to build a nonlinear regression model [9]. SVR transfers the data into a higher dimensional feature space and fits it to a linear function with minimum complexity to the feature space [19]. Kernel-based probabilistic and nonparametric models such as Gaussian Process Regression (GPR) can create a stochastic function as the regression output with some advantages: provide covariance to generate uncertainty level, non-parametric and being flexible [20]. However, some of GPR limitation are: performance is highly affected by kernel functions and high computational cost [21]. Thin Plate Spline (TPS) is a prevalent technique and insensitive to noise that can be used in data fitting and prediction. TPS choose a function that minimizes an integral that represents the bending energy of a surface [22]. The advantages to using TPSs, do not require any a priori knowledge of the functional form of the data or the relationship of interest. Just as complex data visualization is a key strength, it is also a limitation of using thin plate splines. The three-dimensionality of thin plate splines makes for the inclusion of confidence intervals difficult as the visual may become too complex for interpretation [23]. A popular nonlinear model that is appropriate technique in interpolation and extrapolation curve fitting is Taylor Polynomial (TP) [22].

### C. CLUSTERING

Clustering technique can segment data into groups, based on data similarity. It is using to identify outliers and resulting groups may be the matter of interest. Clustering can be achieved by various algorithms and it is an iterative process (involving trial and error). Some of popular techniques clustering are: K-means, Fuzzy K-means, Hierarchical, NN, Gaussian Mixture. K-means is a partitioning method to partitions data into K exclusive clusters. Each cluster has a centroid (or center) and sum of distances from all objects to the center is minimized. Example neural network architectures for clustering are: (i) self-organizing maps, (ii) competitive layers. Gaussian Mixture is good when clusters have different sizes and are correlated and assume that data is drawn from a fixed number K of normal distributions. In general, clustering technique: (i) dose no method is perfect for data modeling (depends on data), (ii) process is iterative; explore different algorithms, (iii) beware of local minima (global optimization can help).

### D. ENSEMBLE TECHNIQUES

Ensemble techniques are mixture of numerous models to create a novel learning method with better performance than the individual classifiers. In this technique, the unseen data (test data) is passed to individual classifier, returning some votes. The ensemble technique revenues the final class prediction based on the popular of classification models votes (Fig. 2). This technique is appropriate when there is not enough data

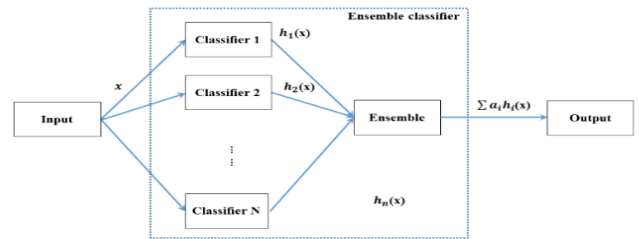


FIGURE 2. An example of ensemble learning [26].

available for presenting the data distribution. The technique is a decent option for uncertainty of selecting the computational model. Ensemble techniques are also used when the classifier is not able to solve complex problems. The technique is used in many industrial applications, such as intrusion detection, malware fraud and remote sensing, speech, and identity recognition. Random Forest (RF) is a well-known ensemble technique that is a cluster of many decision trees that any tree is built by sampling with replacement [24], [25]. A recently proposed ensemble technique is Adaptive Boosting (AdaBoost) algorithm that can be used for regression and/or classification regression industrial problems.

### E. RESAMPLING TECHNIQUES

Resampling techniques are very prevalent methods because of their accuracy, robustness, simplicity and high generalizability.

Resampling generate new data by using difference methods without being correlated to theoretical distribution. This method used when data distribution is very limited or unknown [8]. Resampling generate many times new data with or without replacement. Bootstrapping and randomization methods are examples of resampling technique with and without replacement, respectively. Some popular resampling techniques are JackKnife (JK), Monte Carlo Simulation (MCS) and exact test methods. MCS is a repeated random sampling based on many possible scenarios to obtain numerical results and estimation.

### F. REPRESENTATION LEARNING

Representation learning is a technique to predict or classify unstructured data, which is useful in various ML tasks, such as dimensionality reduction. Representation learning determines a lower dimensional representation of capturing several input configurations from original dataset. It can offer a solution for big data through facilitating significant improvements in statistical and computational efficiency. The hidden representations of representation nodes inside the dataset search to recall core information for ML statistical processes to model full properties of the data, such as vertex content and topological structure. Following this, modeling and analytics tasks can be effortlessly used through vector-based and conventional machine learning algorithms.

### G. DEEP LEARNING

Deep Learning (DL) can automatically find hierarchical representations of data sets, which works on deep architectures in supervised and/or unsupervised learning strategies. By using

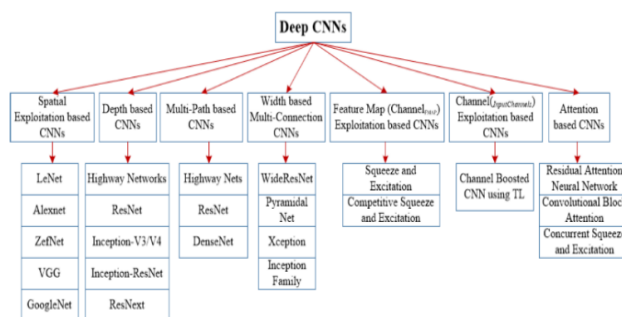
these strategies, DL has the potential to capture complicated patterns and highly nonlinear big data with large feature spaces. Different DL architectures, such as Deep Neural Networks (DNNs) and Recurrent Neural Networks (RNNs), can characterize the multi-layering of commonly narrow algorithms to contain numerous processing data layers [27].

DNN requires much more parameters than traditional systems, which brings huge cost during online evaluation. A new effort on DNN aiming at reducing the model size while keeping the accuracy improvements is using Singular Value Decomposition (SVD). SVD works on the weight matrices in DNN and then restructure the model based on the inherent sparseness of the original matrices. After restructuring we can reduce the DNN model size significantly with negligible accuracy loss [28].

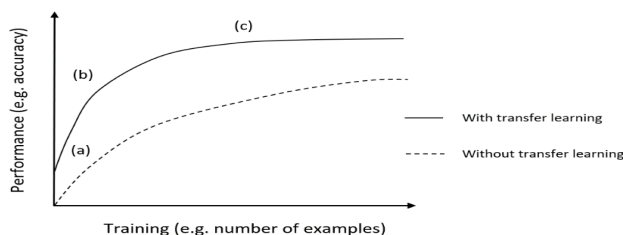
Recent architectures proposed for DL, such as Convolutional Neural Networks (CNNs), Deep Belief Networks (DBNs) and Group Method of Data Handling-type Neural Networks (GMDH-type NN) can process difficult learning algorithm for classification and regression tasks. Different improvements in CNN architecture can be categorized as parameter optimization, regularization, and structural reformulation. Depending upon the type of architectural modification, CNN can be broadly categorized into seven different classes namely; (i) spatial exploitation, (ii) depth, (iii) multi-path, (iv) width, (v) feature map exploitation, (vi) channel boosting, and (vii) attention as shown in Fig.3. [29]. Some other DL include Inception [30], deep Residual Network (ResNet) [31], and VGG16 [32]. Szegedy et al. [30] proposed Inception-v4 that combine Inception architecture with residual connection to accelerate the network training. The core idea of ResNet is recommending a so-called “identity shortcut connection” that skips one or more layers [31]. Moreover, VGG16 is a convolutional neural network model proposed by Simonyan and Zisserman [32] to improve AlexNet by replacing large kernel-sized filters with multiple smaller kernel-sized filters one after another. GMDH is a family of inductive algorithms for computer-based mathematical modeling of multi-parametric datasets that features fully automatic structural and parametric optimization of models. GMDH-type NN is a way of using self-organizing networks and shown to result in successful applications in a broad range of areas [33].

**H. DISTRIBUTED AND PARALLEL LEARNING**

Distributed and Parallel Learning (D&PL) technique uses a parallel processing through distributed network, assigning learning processes to be efficient of ML algorithms. The technical limitation of classical ML can be addressed by D&PL technique, which naturally need the whole datasets to be set within a local memory. D&PL techniques use the divided data in vertical (by features within the training set) or horizontal (by instances within the training set) fashions. In most of the cases the division are worked based on horizontally, as it is the furthest natural selection of the applications. Although D&PL architectures can be leveraged in classification and



**FIGURE 3. Recent taxonomy of deep CNN architectures [29].**



**FIGURE 4. Transfer learning technique: (a) model with a higher initial performance, (b) improvement with a greater rate of performance (c) a faster training timeframe [28].**

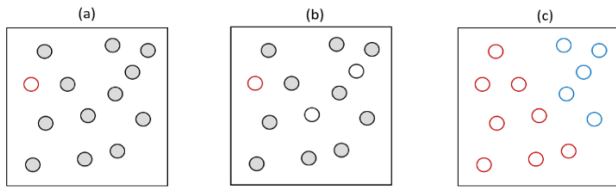
regression tasks, classification has found further wide-spread application due to simply way for implementation.

**I. TRANSFER LEARNING**

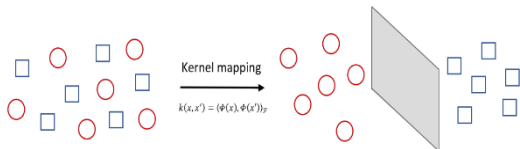
Transfer Learning (TL) influences heterogeneity of the dataset, which is typical for big data processing and analytics. This is determined by velocity characteristics of big data, and training of new models can use important resource and time. A wide range of models can be trained using TL in a more efficient manner through classifying a group of domains. This approach can be used for regression, classification or clustering tasks. Fig. 4 shows a typical set of training performance accuracy, for ML models with and without TL. The model with TL not only starts at a higher performance, also shows faster converges towards an optimal solution. This can be attributed to the old model being closer to the solution as a starting point, rather than using random initializations, so that through the process of updating via gradient descent through back-propagation, or other applicable algorithms, fewer iterations are required to converge to the solution for the new task.

**J. ACTIVE LEARNING**

Active Learning (AL) searches to shift ML strategies for big data from huge volumes of unlabelled data, to that which uses labelled data. AL can be used for broad applications of supervised and semi-supervised ML scenarios. Obtaining labelled data is time consuming, and AL can reduce the cost associated through classifying a subsection of points in the original data distribution. Fig. 5 provides an intuitive example



**FIGURE 5.** Active learning technique: (a) class 1 (red) labelled instance belonging, with the other points depicted in grey. By using active learning, two points are recognized in (b) white circles and labelled as class 1 (red) and class 2 (blue). (c) using k-Means machine learning techniques is used to more easily classify the other points.



**FIGURE 6.** Kernel mapping process.

diagram of AL. The process of AL classifies key points in a dataset, which if labelled. Similar techniques approaches can be taken for regression modeling.

**K. KERNEL-BASED LEARNING**

Kernel-based learning techniques use some nonlinear kernel functions, along with well-known methods such as SVMs, to transfer the input-space of a dataset to a higher-dimensional feature-space. This method permits for better expressive power and better chance to perform various analyses with the same dataset [34]. However, this often requires significantly higher computational complexity as compared with the cases with linear kernels. It may appear counterintuitive to make a higher dimensional illustration for a big dataset. However, these methods have been shown successful in many applications. One often must make a trade-off for maximizing the computational efficiency and minimizing the impact of increasing the size of the dataset. Fig. 6 illustrates the kernel mapping process to simple prediction models.

**L. MULTI-OBJECTIVE LEARNING**

Learning algorithms in ML can be divided into three categories: single-objective learning, scalarized multi-objective learning, and Pareto-based multi-objective learning [35]. There are two main weaknesses of scalarized multi-objective learning: (i) the determination of an appropriate hyperparameter  $\lambda$  that properly reflects the purpose of the user is not trivial, and (ii) only a single solution can be obtained, from which little insight into the problem can be gained [36]. Most of the efforts on solving multi-objective ML problems can be solved using Pareto-based multi-objective optimization methodology particularly due to the great success of multi-objective optimization using evolutionary algorithms and other population-based stochastic search methods [37]. Pareto-based multi-objective learning approaches are more

powerful compared to learning algorithms with a scalar cost function in addressing various topics of ML tasks [36]. One gains a deeper insight into the learning problem by analysing the Pareto front composed of multiple Pareto-optimal solutions [38]. Unlike single objective optimization, multi-objective optimization reduces the chances of falling in a local minimum. In other word, in multi-objective optimization framework, improving one of the objective functions might worsen the other objective function(s); thus, there is not an optimum point to optimize all the objective functions. Multi-objective optimization algorithms based on evolutionary algorithms [39] have been proposed in the literature, e.g. NSGA-II [16], NSGA-III [40], multi-objective uniform-diversity differential evolution (MUDE) [41], and on-line variable-fidelity meta-model assisted Multi-Objective Genetic Algorithm (OLVFM-MOGA) [42].

**M. HYBRID DATA-DRIVEN LEARNING**

As stated in pervious sections, ML techniques have some limitations, and a hybrid ML technique can potentially capture more characteristics of complex systems to overcome these limitations. Recent studies of hybrid models combining different ML techniques have shown promising results [22], [37]. There are various types of frameworks to develop hybrid models, and it is unknown that which hybrid model can perform the best in data-driven learning. Some recent hybrid data-driven methods are: a combination of NN and EKF, GPR technique with ARD kernel [21], SVM and SVR models optimized by GA [43], RVM with incremental learning [44]. ANNs have become popular practical solutions of engineering problems. In the following we introduce a novel algorithm of this type and apply it to an ML task under limited data constrains.

**III. A NOVEL ALGORITHM FOR ROBUST MODELING OF NONLINEAR PROCESSES**

In industry 4.0 applications, an offline approximate model needs to be developed that can express the relationship between inputs and outputs of industrial process. In fact, offline modeling represents an initial identification model of the complex nonlinear system. This model is then updated using limited data obtained from the online process. The online adaptation requires real-time algorithms while high runtime algorithms, such as evolutionary optimization methods can only be used for offline modeling [15]. To address the needs of both offline and online parts, we propose a new algorithm that builds a robust model based on the input-output data using an offline deterministic model and online updating parts.

In the first part, the structure (topology) of the NN and the initial values of its coefficients are extracted based on deterministic input-output data. In other words, the uncertainties included in the observed data are not considered and the modeling is done based on nominal values of input-output data. It is obvious that there are many sources of uncertainties, such as human error, laboratory equipment errors, or errors

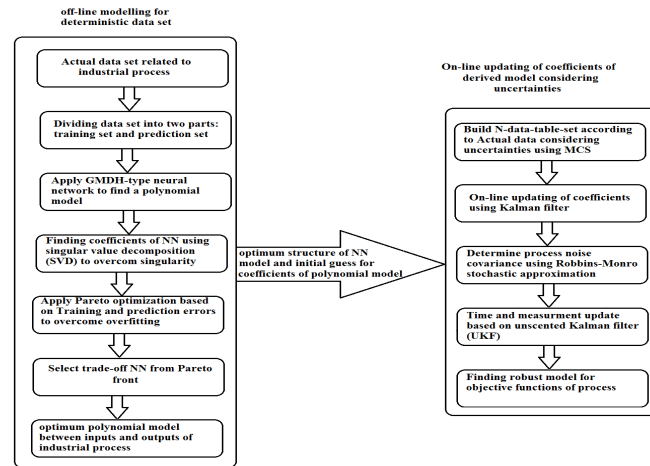


FIGURE 7. A novel algorithm to create a robust model for industrial processes with available input-output data.

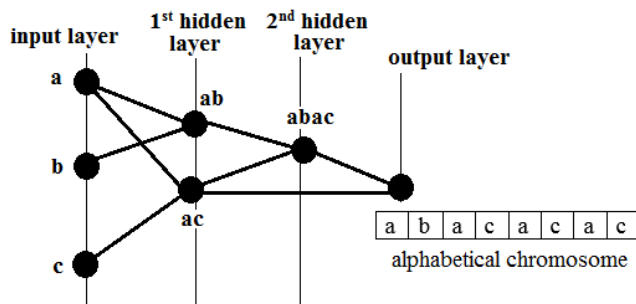


FIGURE 8. The alphabetical chromosome representing GS-GMDH.

due to changes in environmental factors, which can affect the data. In order to obtain a robust model, all these sources of uncertainties must be considered as a percentage of the variation around the reported nominal value [45]. In the second part, the coefficients of the obtained model are updated using MCS in combination with UKF to achieve a robust model. In the following, we give details of these steps.

**A. OFF-LINE MODELING BASED ON DETERMINISTIC DATA**

In the off-line deterministic modeling part, first the modified Taguchi DOE is used to make a reliable statistical test under small sample size [16]. Then, a polynomial model is trained explaining the relationship between the inputs and outputs of the industrial process. To this end, one can use GMDH-type NNs with multi-objective optimization and SVD to overcome both overfitting and singularity. In the proposed algorithm, we used the fuzzy adaptive mutation proposed by some authors in to reach global optimum solutions [46]. As shown in Fig. 7, in the first part (left), a primary model is extracted, and the derived model is used as an input to the second part that builds a more robust model. This model is not sensitive to properties of data and can have reasonable performance with both limited and big data.

As shown in Fig. 7, the input-output data are divided into two sets of training and prediction. The training set, which consists of 60% of all inputs–output data pairs, is used for training the neural networks model. The prediction set consists of 40% of the data that are indeed unforeseen input–output data samples during the training process and are used for testing the performance of the trained model in correctly capturing relationship between the inputs and outputs of the process. For an acceptable performance of GMDH type NN, the topology and the polynomial coefficients of each neuron should be properly determined. Here, SVD is used to determine the polynomial coefficients of each neuron. Multi-objective Genetic Algorithm (GA) is used to find optimal topology of GMDH-type NN. In multi-objective optimization, both modeling and prediction errors are simultaneously considered as objectives. Using Nondominated Sorting Genetic Algorithm (NSGA)-II [15], Pareto optimum non-dominated models are obtained from the point of view of these two objective functions. An alphabetical chromosome is used to coding the structure and topology of general structure of GMDH (GS-GMDH). In the conventional GMDH, each neuron is building using combination of two neurons in adjacent layer, while in the GS-GMDH, all neurons in the all previous layers used to build a new neuron. The alphabetical coding of such GS-GMDH is shown in Fig. 8. In a GS-GMDH neural network, neuron *ac* in the first hidden layer is connected to the output layer by directly going through the second hidden layer. Therefore, it is now very easy to notice that the name of output neuron (network’s output) includes *ac* twice as *acac*. In other words, a virtual neuron named *ac* has been constructed in the second hidden layer and used with *abac* in the same layer to make the output neuron *abacacac* as shown in Fig.8. The evolutionary process starts by randomly generating an initial population of alphabetical chromosome, each as a candidate solution. Then, using the crossover and mutation and tournament selection, the entire population of symbolic strings improves gradually based on training and prediction errors.

Indeed, in order to achieve high modeling accuracy, the polynomial degree is usually increased, which reduces the generalizability (or prediction capability) of the model, often due to overfitting. Finally, the designer chooses the trade-off obtained from Pareto non-dominated solutions by compromising between these two objective functions. A polynomial model among the inputs and outputs of the industrial process is developed by using the selected GMDH structure. The derived topology of NN is used as the input of the second part of the proposed algorithm.

**B. ONLINE UPDATE TO MAKE THE MODEL ROBUST**

In the second part (right side of Fig. 7) of the proposed algorithm, the obtained model in the first part is modified to enhance its robustness. To this end, the coefficients of the derived model are updated using UKF to capture the uncertainties in the input-output data. In order to obtain a robust model, all sources of uncertainties are considered as a percentage of the variation around the nominal values of main data table. To take into account the uncertainties,  $N$  data table sets are built around the nominal values using MCS. The network’s error on each data table is calculated according to

$$E_j = \frac{\sum_{i=1}^k (y_{model} - y_{actual})^2}{k}, \quad j = 1, 2, \dots, N \quad (1)$$

The goal of UKF is to obtain the network coefficients that minimize the mean and variance of network error. Indeed, minimizing the mean of error ( $mean(E_j)$ ) attempts to minimize the total network error for all tables. Also, by minimizing the variance of error ( $var(E_j)$ ), the error variation is minimized relative to the average value. Mean and variance of error is combined according to equation (2), and  $F$  is minimized in a recursive process.

$$F = Mean(E_j) + var(E_j) \quad (2)$$

The UKF filter equations for determining the GMDH-type neural network coefficients are as follows:

(i) The weight vector and its covariance matrix in network are initialized with:

$$\hat{a}_0 = E[a_0] \quad (3)$$

$$P_0 = E[(a_0 - \hat{a}_0)(a_0 - \hat{a}_0)^T] \quad (4)$$

where  $a = \{a_1, a_2, \dots, a_s\}$  is the vector of coefficients of neural network and  $s$  is the number of coefficients.

(ii) The time-update equation are:

$$\hat{a}_k^- = \hat{a}_{k-1} \quad (5)$$

$$P_{a,k}^- = P_{a,k-1} + R_{r,k-1} \quad (6)$$

where  $k$  indicates the time.

(iii) The sigma points and measurement update according basic equations of UKF.

Algorithm 1 show the pseudo-code of the proposed algorithm and summarizes the steps.

**Algorithm 1** The Proposed Algorithm for Robust Modeling of Data

**Part A: offline modeling:**

A1. Enter matrix of experimental data set  $D[X(m,n) Y(m,1)]$  %  $m$  and  $n$  are the number of experimental data set and number of inputs, respectively.

A2.  $T=D(1:t,n)$  and  $V=D(t+1:m,n)$  % Making Training ( $T$ ) and validation ( $V$ ) data sets

A3.  $Pa=[Y_1 Y_2 \dots Y_k]$  % Pareto optimum design of GMDH and finding  $k$  non-dominated solutions ( $Y_1 \dots Y_k$ )

A4.  $Y_{model} = F(X)$  % Select trade-off model from Pareto front and finding polynomial model

**Part B: on-line updating:**

B1.  $DT = [D_1 D_2 \dots D_N]$  % Building  $N$  data table sets around the nominal values of  $D$  using MCS.

B2.  $E_j = \frac{\sum_{i=1}^k (y_{model} - y_{actual})^2}{k}$  % calculation error of  $Y_{model}$  on all data tables ( $j=1 \dots N$ )

B3.  $F = Mean(E_j) + var(E_j)$  % finding cost function based of mean and variance of derived errors

B4.  $\hat{a}_k^- = \hat{a}_{k-1}$   
 $P_{a,k}^- = P_{a,k-1} + R_{r,k-1}$  % Update coefficients of  $Y_{model}$  using UKF

B5. if ( $F > \epsilon$ ) go to B2 %  $\epsilon$  is threshold value of  $F$  else go to B5 (equation 2)

B5.  $Y_{robust} = G(X)$  % finding robust model

**IV. INDUSTRIAL CASE STUDY**

To examine the performance of the algorithm proposed above, an industry case study is considered. To this end, 27 samples (L27) modified Taguchi DOE method (Table 1) was constructed at Carbon Fiber production line. The PAN fiber inputs were chosen as the controlling parameters based on the feasible processing window of the pilot plant: temperatures of 227, 230, 233, and 236°, space velocity of 20, 25, 30 and 35 m/h, and stretching-ratio of 1.0, 2.0, 3.0 and 4.0%. The measured output was a physical property. To reduce the number of experiments, the Taguchi design was modified by adding some marginal operating parameters as listed in Table 1.

NSGA-II and SVD are employed to design optimal topology and to find optimal values of the polynomial coefficient of GMDH, respectively. It is clear from this limited input-



**TABLE 1.** input-output experimental data of carbon fiber production line.

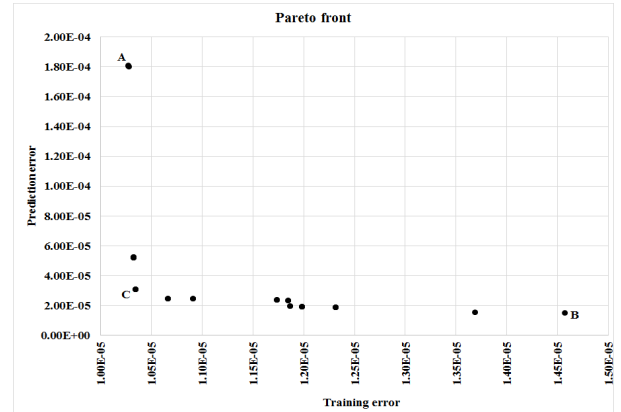
Experiment No.	Inputs			Output Physical property
	$x_1$ Temperatures	$x_2$ Space velocity	$x_3$ Stretching-ratio	
1	227	20	1	1.2446
2	227	20	4	1.2438
3	227	25	3	1.2417
4	227	30	3	1.2359
5	230	20	2	1.2574
6	230	25	1	1.2417
7	230	30	4	1.2341
8	230	35	1	1.2335
9	233	20	3	1.257
10	233	25	1	1.2611
11	233	25	2	1.2457
12	233	25	3	1.2465
13	233	30	1	1.2429
14	233	30	3	1.2421
15	236	20	1	1.2707
16	236	20	4	1.271
17	236	30	2	1.2547
18	236	35	1	1.2504
19	227	35	1	1.2339
20	230	20	4	1.2588
21	227	25	2	1.25
22	227	35	4	1.2244
23	230	25	3	1.2464
24	230	35	3	1.2317
25	233	20	4	1.2601
26	233	25	4	1.2565
27	233	35	2	1.2363
28	236	25	3	1.263
29	236	35	4	1.2474
30	233	35	3	1.2372

output data that the inputs are not in the same order. For example,  $x_1$  is in order  $10^3$ , while order of  $x_3$  is  $10^0$ . Therefore, to keep the impact of each data set unchanged, using a linear mapping equation (7), all inputs are mapped to the interval [1, 2].

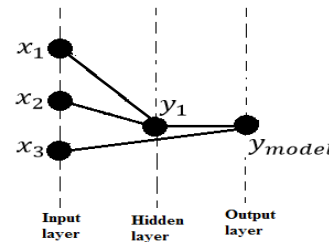
$$x_i^M = 1 + \frac{(x_i - x_i^{min})}{(x_i^{max} - x_i^{min})}, \quad i = 1, 2, 3 \quad (7)$$

where  $x_i^M$  and  $x_i$  are the mapped and actual values of the input variables, respectively.  $x_i^{max}$  and  $x_i^{min}$  are the maximum and minimum values of each input variables, and  $i = 1, 2, 3$ . It is clear that the other mapping methods such as logarithmic normalization could not be employed here since it will change the impact of each data set.

Two different sets of training and validation have been used in order to show the prediction capability of the optimized GMDH-type neural networks. The training set, which is used for training, is composed of 18 out of 30 input–output data pairs. The validation set, consisting of 12 unforeseen input–output data samples during the training process, is solely used for checking the performance of the trained model. GMDH-type neural networks are employed to fit a polynomial curve to the output of the model as a function of effective input parameters. Considering two objective functions (namely, Training Error, TE, and Prediction Error, PE),



**FIGURE 9.** Final front including non-dominated solutions in plane of objective functions.



**FIGURE 10.** Optimized structure of GMDH-type neural network of design point C.

**TABLE 2.** The and pe of optimum design point obtained in fig 7.

	A	B	C
Training Error	1.027E-05	1.4565E-05	1.0344E-05
Prediction error	18.1181E-05	1.5409E-05	3.0979E-05

NSGA-II is used for the Pareto multi-objective optimization of the GMDH-type neural network. The values of 80 for population size, 0.9 for crossover probability, 0.1 for mutation probability, and 300 for number of generations, were found to be suitable using a trial and error study. The resultant Pareto front, representing a set of non-dominated superior solutions, is shown in Fig. 9. In this figure, points A and B designate solutions with the best PE and TE, respectively. As compared with solution ‘A’, Point C shows a very small increase in the TE value (about 0.7%) but a substantial improvement (about 83%) in PE. Consequently, when the two objectives are simultaneously taken into consideration, point ‘C’ may be considered as a best trade-off solution.

The structure of the neural network corresponding to the design point C obtained by the genetic algorithm NSGA-II is shown in Fig. 10. Reasonable behavior of GMDH-type neural networks model in the training and validation data are shown in Fig. 11. Moreover, Fig. 12 shows a correlation coefficient of 92.36% between the actual value and the predicted one, representing a reasonably accurate model. for the obtained model. Table 2 shows the values of TE and PE corresponding to points A, B, and, C.

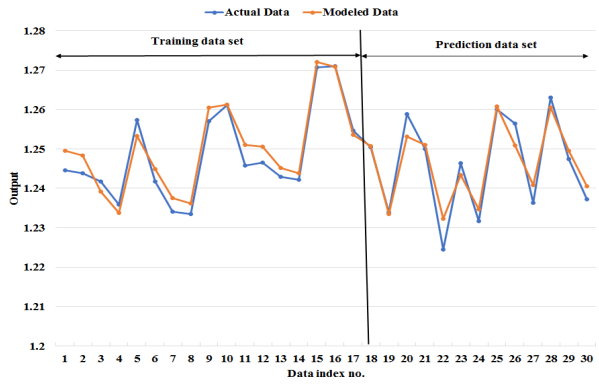


FIGURE 11. Performance of the optimized GMDH model on both actual and validation data sets corresponding to optimum point C.

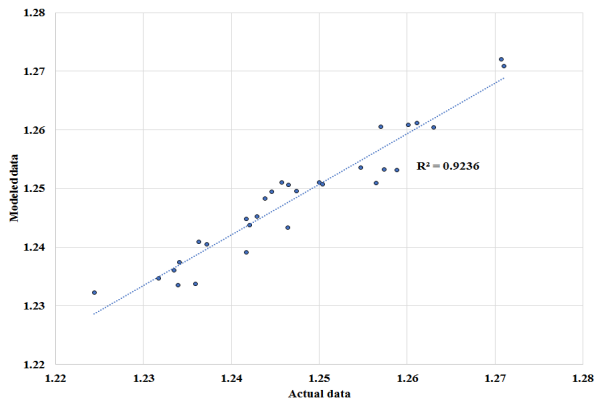


FIGURE 12. Correlation coefficient between the actual values and predicted ones corresponding to optimum point C.

To check the robustness of the obtained model, a  $\pm 5\%$  variation around the nominal values reported in Table 1 are considered and 1000 random data tables are constructed using MCS. Fig. 13 shows the upper and lower bounds of the predicted values using this model. Extensive changes relative to the nominal values show poor robustness of derived model. UKF is used to improve the robustness of the derived model. It must be noted that the structure of GMDH as shown in Fig. 10 is employed and only the coefficients of equations are update by UKF. The equations governing the neurons of this model derived by UKF are as follows:

$$y_1 = 1.29 - 0.01x_1 - 0.0546x_2 + 0.0135x_1^2 + 0.0156x_2^2 - 0.00718x_1x_2 \quad (8a)$$

$$y_{model} = -4.24 + 7.787y_1 - 0.0129x_3 - 2.71y_1^2 + 0.003x_3^2 + 0.0022y_1x_3 \quad (8b)$$

Fig. 14 shows the upper and lower bound of the predicted values. It is seen that employing UKF is effective in reducing the tolerance margins. The intelligent algorithm results assisted the carbon fiber production line in alleviating the reduced number of experiments, cost and limited data. It should be noted that the same procedure can be used on big data to find a robust model among inputs and outputs of any nonlinear industrial process. Also, DL methods, such as CNNs, DBNs and RNNs that mentioned in section II can be

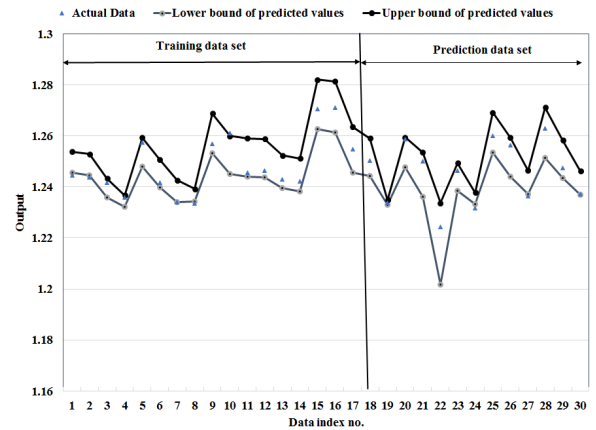


FIGURE 13. Upper and lower bound of predicted values by deterministic model on 1000 input-output random tables.

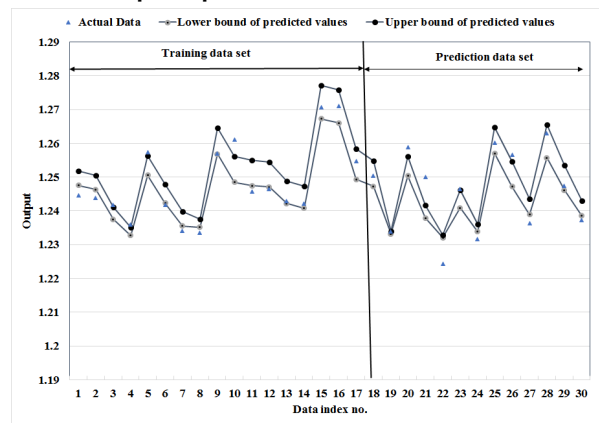


FIGURE 14. Upper and lower bound of predicted values by robust model on 1000 input-output random tables.

used for very large (terabyte) data sets. One of the most immediate and impactful outcomes of technological evolution is the vast advancement in automation through data modeling. The intelligent algorithm for robust data modeling continues to accelerate, so will automation. Although our model aimed to increase the estimation accuracy, the computational effort is also reduced by 2 (ms) in offline modeling and 3 (ms) in online modeling.

## V. CONCLUSION

In this paper, some of existing machine learning techniques that can deal with issues commonly observed in many industrial applications and arising from having limited or big data constraints, were reviewed. These techniques can be used to efficiently address various challenges, such as increased complexity, uncertainty, and dynamism, related to data processing and analytics of manufacturing processes. Machine learning is a powerful tool for many industrial applications, and its importance is further enhanced due to increased use of data collection and sensor technologies as part of industry 4.0 implementation, leading to generation of valuable data sources. We proposed a new intelligent algorithm that can model limited industrial data. We applied the proposed algorithm to a real industry case study with data collected

from a Carbon Fiber production line. Pareto multi-objective optimization including two objective functions was employed to design the topology of GMDH and overcome overfitting. Using the proposed algorithm, a deterministic model was obtained that showed a very accurate fit to the actual data. We further used UKF approach to improve robustness of the model against uncertainties. The intelligent algorithm assisted the carbon fiber production line in alleviating the reduced number of experiments, cost and limited data. The proposed algorithm has the potential to be used in any other industrial settings for which the aim is to obtain a reliable and robust model between its inputs and outputs.

## REFERENCES

- [1] T. Wuest, D. Weimer, C. Irgens, and K.-D. Thoben, "Machine learning in manufacturing: Advantages, challenges, and applications," *Prod. Manuf. Res.*, vol. 4, no. 1, pp. 23–45, Jan. 2016.
- [2] H. Khayyam, R. N. Jazar, S. Nunna, G. Golkarnarenji, K. Badii, S. M. Fakhrohoseini, S. Kumar, and M. Naebe, "PAN precursor fabrication, applications and thermal stabilization process in carbon fiber production: Experimental and mathematical modelling," *Prog. Mater. Sci.*, vol. 107, Jan. 2020, Art. no. 100575.
- [3] J. Wan, S. Tang, D. Li, S. Wang, C. Liu, H. Abbas, and A. V. Vasilakos, "A manufacturing big data solution for active preventive maintenance," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 2039–2047, Aug. 2017.
- [4] M. Binkhonain and L. Zhao, "A review of machine learning algorithms for identification and classification of non-functional requirements," *Expert Syst. Appl.*, X, vol. 1, Apr. 2019, Art. no. 100001.
- [5] H. Khayyam, G. Golkarnarenji, and R. N. Jazar, "Limited data modelling approaches for engineering applications," in *Nonlinear Approaches in Engineering Applications*, R. N. Jazar, Ed. Cham, Switzerland: Springer, 2018.
- [6] Y. Li, K. Liu, A. M. Foley, A. Zülke, M. Berecibar, E. Nanini-Maury, J. Van Mierlo, and H. E. Hoster, "Data-driven health estimation and lifetime prediction of lithium-ion batteries: A review," *Renew. Sustain. Energy Rev.*, vol. 113, Oct. 2019, Art. no. 109254.
- [7] B. Crawford, H. Khayyam, A. S. Milani, and R. N. Jazar, "Big data modeling approaches for engineering applications," in *Nonlinear Approaches in Engineering Applications*, R. N. Jazar, Ed. Cham, Switzerland: Springer, 2019.
- [8] K. Liu, K. Li, Q. Peng, Y. Guo, and L. Zhang, "Data-driven hybrid internal temperature estimation approach for battery thermal management," *Complexity*, vol. 2018, Jul. 2018, Art. no. 9642892.
- [9] J. P. Davim, *Design of Experiments in Production Engineering*. Cham, Switzerland: Springer, 2015.
- [10] H. Khayyam, M. Naebe, A. Bab-Hadiashar, F. Jamshidi, Q. Li, S. Atkiss, D. Buckmaster, and B. Fox, "Stochastic optimization models for energy management in carbonization process of carbon fiber production," *Appl. Energy*, vol. 158, pp. 643–655, Nov. 2015.
- [11] G. Golkarnarenji, M. Naebe, K. Badii, A. Milani, R. Jazar, and H. Khayyam, "Production of low cost carbon-fiber through energy optimization of stabilization process," *Materials*, vol. 11, no. 3, p. 385, Mar. 2018.
- [12] A. Sharma and K. K. Paliwal, "Linear discriminant analysis for the small sample size problem: An overview," *Int. J. Mach. Learn. Cybern.*, vol. 6, no. 3, pp. 443–454, Jun. 2015.
- [13] A. Ilin and T. Raiko, "Practical approaches to principal component analysis in the presence of missing values," *J. Mach. Learn. Res.*, vol. 11, no. 1, pp. 1957–2000, 2010.
- [14] X. Peng, Y. Cai, Q. Li, and K. Wang, "Control rod position reconstruction based on K-Nearest neighbor method," *Ann. Nucl. Energy*, vol. 102, pp. 231–235, Apr. 2017.
- [15] G. Golkarnarenji, M. Naebe, K. Badii, A. S. Milani, A. Jamali, A. Bab-Hadiashar, R. N. Jazar, and H. Khayyam, "Multi-objective optimization of manufacturing process in carbon fiber industry using artificial intelligence techniques," *IEEE Access*, vol. 7, pp. 67576–67588, 2019.
- [16] G. Golkarnarenji, M. Naebe, K. Badii, A. S. Milani, R. N. Jazar, and H. Khayyam, "A machine learning case study with limited data for prediction of carbon fiber mechanical properties," *Comput. Ind.*, vol. 105, pp. 123–132, Feb. 2019.
- [17] A. Rasmuson, B. Andersson, L. Olsson, and R. Andersson, *Mathematical Modeling in Chemical Engineering*. New York, NY, USA: Cambridge Univ. Press, 2014.
- [18] D. Hunter, H. Yu, M. S. Pukish, III, J. Kolbusz, and B. M. Wilamowski, "Selection of proper neural network sizes and architectures—A comparative study," *IEEE Trans. Ind. Informat.*, vol. 8, no. 2, pp. 228–240, May 2012.
- [19] K. Badii, G. Golkarnarenji, A. S. Milani, M. Naebe, and H. Khayyam, "A comprehensive chemical model for the preliminary steps of the thermal stabilization process in a carbon fibre manufacturing line," *Reaction Chem. Eng.*, vol. 3, no. 6, pp. 959–971, 2018.
- [20] K. Liu, X. Hu, Z. Wei, Y. Li, and Y. Jiang, "Modified Gaussian process regression models for cyclic capacity prediction of lithium-ion batteries," *IEEE Trans. Transport. Electrification*, vol. 5, no. 4, pp. 1225–1236, Dec. 2019.
- [21] K. Liu, Y. Li, X. Hu, M. Lucu, and W. D. Widanage, "Gaussian process regression with automatic relevance determination kernel for calendar aging prediction of lithium-ion batteries," *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 3767–3777, Jun. 2020.
- [22] H. Khayyam, M. Naebe, O. Zabihi, R. Zamani, S. Atkiss, and B. Fox, "Dynamic prediction models and optimization of polyacrylonitrile (PAN) stabilization processes for production of carbon fiber," *IEEE Trans. Ind. Informat.*, vol. 11, no. 4, pp. 887–896, Aug. 2015.
- [23] H. Khayyam, S. M. Fakhrohoseini, J. S. Church, A. S. Milani, A. Bab-Hadiashar, R. N. Jazar, and M. Naebe, "Predictive modelling and optimization of carbon fiber mechanical properties through high temperature furnace," *Appl. Thermal Eng.*, vol. 125, pp. 1539–1554, Oct. 2017.
- [24] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
- [25] M. V. Datla, "Bench marking of classification algorithms: Decision trees and random forests—a case study using R," in *Proc. Int. Conf. Trends Autom., Commun. Comput. Technol. (I-TACT)*, Dec. 2015, pp. 1–7.
- [26] J. Xu, L. Yao, and L. Li, "Argumentation based joint learning: A novel ensemble learning approach," *PLoS ONE*, vol. 10, no. 5, May 2015, Art. no. e0127281.
- [27] Y. LeCun, Y. Bengio, and G. E. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.
- [28] J. Xue, J. Li, Y. Gong, "Restructuring of deep neural network acoustic models with singular value decomposition," in *Proc. Int. Speech Commun. Assoc. (INTERSPEECH)*, 2013, pp. 2365–2369.
- [29] A. Khan et al., "A survey of the recent architectures of deep convolutional neural networks," *Artif. Intell. Rev.*, pp. 1–62, Apr. 2020, doi: 10.1007/s10462-020-09825-6.
- [30] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," presented at the Proc. 31 st AAAI Conf. Artif. Intell. (AAAI), 2016.
- [31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," presented at the IEEE Conf. Comput. Vis. Pattern Recognit., 2016.
- [32] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," presented at the 3rd Int. Conf. Learn. Represent. (ICLR), San Diego, CA, USA, 2015.
- [33] S. Jeddi and S. Sharifian, "A hybrid wavelet decomposer and GMDH-ELM ensemble model for network function virtualization workload forecasting in cloud computing," *Appl. Soft Comput.*, vol. 88, Mar. 2020, Art. no. 105940.
- [34] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 181–201, Mar. 2001.
- [35] H. Khayyam and A. Bab-Hadiashar, "Adaptive intelligent energy management system of plug-in hybrid electric vehicle," *Energy*, vol. 69, pp. 319–335, May 2014.
- [36] Y. Jin and B. Sendhoff, "Pareto-based multiobjective machine learning: An overview and case studies," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 38, no. 3, pp. 397–415, May 2008.
- [37] M. Ramezankhani, B. Crawford, H. Khayyam, M. Naebe, R. Seethaler, and A. S. Milani, "A multi-objective Gaussian process approach for optimization and prediction of carbonization process in carbon fiber production under uncertainty," *Adv. Compos. Hybrid Mater.*, vol. 2, no. 3, pp. 444–455, Sep. 2019.
- [38] K. Liu, C. Zou, K. Li, and T. Wik, "Charging pattern optimization for lithium-ion batteries with an electrothermal-aging model," *IEEE Trans. Ind. Informat.*, vol. 14, no. 12, pp. 5463–5474, Dec. 2018.
- [39] H. Khayyam, A. Jamali, H. Assimi, and R. N. Jazar, "Genetic programming approaches in design and optimization of mechanical engineering applications," in *Nonlinear Approaches in Engineering Applications*, R. N. Jazar, Ed. Cham, Switzerland: Springer, 2020, pp. 367–402.

- [40] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 602–622, Aug. 2014.
- [41] I. Gholaminezhad and A. Jamali, "A multi-objective differential evolution approach based on  $\epsilon$ -elimination uniform-diversity for mechanism design," *Struct. Multidisciplinary Optim.*, vol. 52, no. 5, pp. 861–877, 2016.
- [42] L. Shu, P. Jiang, Q. Zhou, X. Shao, J. Hu, and X. Meng, "An on-line variable fidelity metamodel assisted multi-objective genetic algorithm for engineering design optimization," *Appl. Soft Comput.*, vol. 66, pp. 438–448, May 2018.
- [43] G. Golkarnarenji, M. Naebe, K. Badii, A. S. Milani, R. N. Jazar, and H. Khayyam, "Support vector regression modelling and optimization of energy consumption in carbon fiber production line," *Comput. Chem. Eng.*, vol. 109, pp. 276–288, Jan. 2018.
- [44] D. Liu, J. Zhou, D. Pan, Y. Peng, and X. Peng, "Lithium-ion battery remaining useful life estimation with an optimized relevance vector machine algorithm with incremental learning," *Measurement*, vol. 63, pp. 143–151, Mar. 2015.
- [45] A. Jamali, M. Ghamati, B. Ahmadi, and N. Nariman-zadeh, "Probability of failure for uncertain control systems using neural networks and multi-objective uniform-diversity genetic algorithms (MUGA)," *Eng. Appl. Artif. Intell.*, vol. 26, no. 2, pp. 714–723, Feb. 2013.
- [46] A. Jamali, R. Mallipeddi, M. Salehpour, and A. Bagheri, "Multi-objective differential evolution algorithm with fuzzy inference-based adaptive mutation factor for Pareto optimum design of suspension system," *Swarm Evol. Comput.*, vol. 54, May 2020, Art. no. 100666.



**HAMID KHAYYAM** (Senior Member, IEEE) received the B.Sc. degree (Hons.) from the University of Isfahan, the M.Sc. degree from the Iran University of Science and Technology, and the Ph.D. degree in mechanical (mechatronics) engineering from Deakin University, Australia. He has worked in automation and productivity of process lines in various industrial companies for more than ten years. In his previous position, he was leading the efforts on modeling and optimization of energy systems in the carbon fiber production line at Deakin University. He is currently a Senior Lecturer with RMIT University, Australia. He has published more than 90 peer reviewed journal articles, conference papers and book chapters. His research activities focus on system identification, modeling, control, simulation-based optimization of complex systems, applications of artificial intelligence techniques, machine learning algorithms, and optimization methods for engineering.



**ALI JAMALI** received the M.Sc. (Eng.) degree in mechanical engineering from the Iran University of Science and Technology, Iran, in 2004, and the Ph.D. degree in solid mechanical engineering from Guilan University, Iran, in 2008. He has served as the Head of Mechanical Engineering Department, and the Deputy of research and education of the Faculty of Mechanical Engineering, University of Guilan. He is currently an Associate Professor of evolutionary computation and control engineering with the University of Guilan. He also visits Kyungpook National University, South Korea, as a Visiting Associate Professor (March–June 2019). His research interests are multi-disciplinary Pareto optimization, game theory, robust design, reliability and risk analysis, neural networks, machine learning, genetic programming, and fuzzy logic, which have been either developed or deployed to optimization, and modeling and controlling of complex systems or processes.



**ALIREZA BAB-HADIASHAR** (Senior Member, IEEE) received the B.Sc. degree in mechanical engineering from Tehran University, in 1988, the M.E.S. degree in mechanical and mechatronics engineering from The University of Sydney, in 1993, and the Ph.D. degree in computer vision from Monash University, in 1997. He is currently a Professor with the School of Engineering, RMIT University, Melbourne, VIC, Australia. He has published numerous highly cited articles in optical flow, robust segmentation, and model selection techniques. His main research interests are the development of robust estimation and segmentation techniques for computer vision applications and intelligent and robust sensory systems for new generation vehicle technologies.



**THOMAS ESCH** was born in Mayen, in 1958. He studied process engineering (and power engineering) at the RWTH University of Aachen (and the UNLV Las Vegas), where he received the Ph.D. degree from the Institute for Applied Thermodynamics with a dissertation in the field of tribology in internal combustion engines, in 1992. After working at FEV Motorentechnik in Aachen as the Department Head of Engine Mechanics, he joined, in September 1993, as a full-time Professor of Thermodynamics and Combustion Technology at the Aachen University of Applied Sciences. Since 2007, he has been working as a course Director of the Master study course Vehicle Powertrain Technology. Besides numerous publications in journals and patents, he wrote as a coauthor of the book *Commercial Vehicle Technology*. His research interest is on internal combustion engines and synthetic fuels, fuel cell systems for aviation and autonomous flying, electric compressors and drive systems, aviation certifications for piston engines and the climate impact adaptation in the automotive industry. He is a member of the board of directors of the IEF (International Energy Foundation) and a board member of the Solar Institute Jülich, since August 2009, AIAA, VDI, SAE, IEF, and DGLR.



**SEERAM RAMAKRISHNA** (Senior Member, IEEE) is the Chair of Circular Economy Taskforce with the National University of Singapore (NUS), which is ranked among the top 11 universities in the world. He is named among the World's Most Influential Minds and the Top 1% Highly Cited Researchers by Thomson Reuters and Clarivate Analytics. He has coauthored 1,300 SCI indexed international journal articles and eight authored books, which received 104,494 citations and 152 H-index. He is a member of Enterprise Singapore NMC on ISO/TC323 on Circular Economy. He is a Fellow of the Royal Academy of Engineering (FREng). He is an Elected Fellow of major professional societies and academies of Singapore, India, U.K., and USA. He chairs the Future of Manufacturing technical committee at the Institution of Engineers Singapore. He is an advisor to the Singapore National Environmental Agency's CESS events.



**MAHDI JALILI** (Senior Member, IEEE) received the B.S. degree in electrical engineering from Tehran Polytechnic, Tehran, Iran, in 2001, the M.S. degree in electrical engineering from the University of Tehran, Tehran, in 2004, and the Ph.D. degree in synchronization in dynamical networks from the Swiss Federal Institute of Technology Lausanne, Lausanne, Switzerland. He then joined the Sharif University of Technology as an Assistant Professor. He is currently a Senior

Lecturer with the School of Engineering, RMIT University, Melbourne, VIC, Australia. His research interests include network science, dynamical systems, social networks analysis and mining, and data analytics. Dr. Jalili was a recipient of the Australian Research Council DECRA Fellowship and the RMIT Vice-Chancellor Research Fellowship. He is an Associate Editor of the IEEE CANADIAN JOURNAL OF ELECTRICAL AND COMPUTER ENGINEERING and an Editorial Board Member of the *Mathematical Problems in Engineering* and *Complex Adaptive Systems Modeling*.



**MINOO NAEBE** is currently an Associate Professor and a Theme Leader with the Carbon Nexus, Institute for Frontier Materials (IFM), Deakin University, Australia. Her research interests include development of high-performance carbon based fibres and composites. The current focus of her team is on development of low-cost carbon fibres from lignin, textile PAN and coal as well as industrial application of graphene and its composites. Her research to date has resulted in over 100 publications and her projects are well supported by government and numerous industrial organizations. She was a recipient of multiple awards, including the recent Chemchina Patent Excellence Award.

Her research to date has resulted in over 100 publications and her projects are well supported by government and numerous industrial organizations. She was a recipient of multiple awards, including the recent Chemchina Patent Excellence Award.

...