Hari P. N. Nagarajan[1]
Mechanical Engineering and Industrial Systems (MEI),
Tampere University of Technology,
P.O. Box 589,
Tampere 33101, Finland
e-mail: hari.nagarajan@tut.fi

Hossein Mokhtarian
Mem. ASME
Mechanical Engineering and Industrial Systems (MEI),
Tampere University of Technology,
P.O. Box 589,
Tampere 33101, Finland;
G-SCOP Laboratory,
CNRS,
University Grenoble Alpes,
Grenoble 38000, France
e-mail: Hossein.mokhtarian@tut.fi

Hesam Jafarian
Mechanical Engineering and Industrial Systems (MEI),
Tampere University of Technology,
P.O. Box 589,
Tampere 33101, Finland
e-mail: hesam.jafarian@tut.fi

Saoussen Dimassi
Mechanical Engineering and Industrial Systems (MEI),
Tampere University of Technology,
P.O. Box 589,
Tampere 33101, Finland
e-mail: swndimassi@gmail.com

Shahriar Bakrani-Balani
LGP-ENIT-INPT & Institut Clément Ader,
CNRS UMR 5312,
University of Toulouse,
47th Avenue d'Azereix,
Tarbes Cedex BP1629-65016, France
e-mail: sbakrani@enit.fr

Azarakhsh Hamedi
Mechanical Engineering and Industrial Systems (MEI),
Tampere University of Technology,
P.O. Box 589,
Tampere 33101, Finland
e-mail: Azarakhsh.hamedi@tut.fi

Eric Coatanéa
Mem. ASME
Mechanical Engineering and Industrial Systems (MEI),
Tampere University of Technology,
P.O. Box 589,
Tampere 33101, Finland
e-mail: eric.coatanea@tut.fi

G. Gary Wang
Mem. ASME
School of Mechatronics Systems Engineering,
Simon Fraser University,
250-13450 102 Avenue,
Surrey, BC V3A0A3, Canada
e-mail: gary_wang@sfu.ca

Karl R. Haapala
Mem. ASME
School of Mechanical, Industrial and
Manufacturing Engineering (MIME),
Oregon State University,
204 Rogers Hall,
Corvallis, OR 97331
e-mail: karl.haapala@oregonstate.edu

Knowledge-Based Design of Artificial Neural Network Topology for Additive Manufacturing Process Modeling: A New Approach and Case Study for Fused Deposition Modeling

Additive manufacturing (AM) continues to rise in popularity due to its various advantages over traditional manufacturing processes. AM interests industry, but achieving repeatable production quality remains problematic for many AM technologies. Thus, modeling different process variables in AM using machine learning can be highly beneficial in creating useful knowledge of the process. Such developed artificial neural network (ANN) models would aid designers and manufacturers to make informed decisions about their products and processes. However, it is challenging to define an appropriate ANN topology that captures the AM system behavior. Toward that goal, an approach combining dimensional analysis conceptual modeling (DACM) and classical ANNs is proposed to create a new type of knowledge-based ANN (KB-ANN). This approach integrates existing literature and expert knowledge of the AM process to define a topology for the KB-ANN model. The proposed KB-ANN is a hybrid learning network that encompasses topological zones derived from knowledge of the process and other zones where missing knowledge is modeled using classical ANNs. The usefulness of the method is demonstrated using a case study to model wall thickness, part height, and total part mass in a fused deposition modeling (FDM) process. The KB-ANN-based model for FDM has the same performance with better generalization capabilities using fewer weights trained, when compared to a classical ANN. [DOI: 10.1115/1.4042084]

Keywords: additive manufacturing, fused deposition modeling, dimensional analysis, empirical learning, knowledge-based artificial neural networks

[1] Corresponding author.

Contributed by the Design for Manufacturing Committee of ASME for publication in the Journal of Mechanical Design. Manuscript received July 7, 2018; final manuscript received November 5, 2018; published online December 20, 2018. Assoc. Editor: Mian Li.

1 Introduction

Major technological and industrial advancements in manufacturing (e.g., additive manufacturing (AM), cloud computing, nanomanufacturing, and advanced materials) have brought about great paradigm shifts in the way products are designed and manufactured. Additive manufacturing research has enabled the growth of innovative techniques and functional products, framing AM as a feasible alternative to subtractive and formative techniques [1]. AM processes are being adopted at an ever-increasing pace for mainstream manufacturing. Particularly, polymer extrusion technology, such as fused deposition modeling (FDM) are among the most well researched and most widely used AM processes. The FDM process involves successive melting, extrusion, deposition, and solidification of thermoplastic polymer melts [2]. Typical FDM equipment consists of a material delivery system or extrusion system, heating system, build plate, and filament feeder. The process begins with the generation of layer profile information using a rapid prototyping (RP) software for any given 3D CAD model. The FDM equipment then deposits semiliquid molten polymer beads onto a heated build plate following the layer information provided from the RP software [2]. This process remains a source of innovation; new technologies are being developed using this approach for metal printing using a metal and polymer matrix, for example, see Refs. [3] and [4]. The FDM process involves storage of thermal energy in the molten material, distribution of this energy into the part through a thermal conduction process, and energy dissipation from the part by convection cooling. The redistribution of the thermal energy ensures the bonding between layers. Several methods exist for thermoplastic delivery in the process, namely, use of liquefiers for self-extruding filament, fluid metering rotary pumps, and high-pressure plunger systems [5,6]. The liquefier-based material delivery method is dominant in most FDM machines. In this research, material delivery using a liquefier, which employs a self-extruding filament, is modeled.

For FDM parts, the cross section of a deposited layer is shaped through the direct flow of polymer melt between the previous layer and the printing nozzle. This results in shapes having the form of flattened ellipsoids. Since the 1980s, process models have been developed for understanding the complex phenomena taking place in FDM, such as thermal transfer, layer creation, and bonding processes [7,8]. Existing research on FDM modeling has focused on the cooling of single and multiple filaments, thermal behavior of the liquefier, analysis of melt front location, degree of cooling in the nozzle and impact of its design on operational stability, temperature distribution across different part design configurations, and impact of the build file [5,8,9]. This available knowledge provides a set of dispersed submodels supporting the understanding of localized phenomena, but does not provide an overall system perspective nor a global process model of FDM. Thus, it raises two main questions for qualification of FDM technology in mainstream manufacturing: (i) are the part requirements achievable with current FDM technology? and (ii) what are the optimal manufacturing parameters that need to be selected to achieve required part specifications? Further, existing localized models cannot be used effectively in closed-loop control of FDM machines. Thus, metamodeling approaches can be evaluated for effective modeling and control of FDM processes [10].

Artificial neural networks (ANN) have been widely used as a modeling strategy to approximate complex functions. In this context, ANNs can be considered as one type of metamodeling approach [11–14]. ANNs are utilized in numerous domains and form the backbone of deep learning algorithms. The main challenge of developing and implementing an ANN is that it demands a large number of training data. Moreover, the architecture of an ANN is problem-dependent, and it requires extra training to explore and progressively generate a suitable architecture via the weights allocated to each of the edges [15]. After training, ANNs are often difficult to interpret. Hence, ANNs have lost their luster as a metamodeling approach over the past two decades [16]. This is specifically the case when a limited amount of training data is available, or if the system to be modeled is subject to large variability due to its complexity. Deep learning approaches can be used for such large complex systems but the duration of the training can be extremely long (up to several years) and extremely costly [16–18]. The amount of training data required sometimes implies the need of resources often only available in large companies. In addition to data challenges, an ANN topology has to be specified before the training, and available system knowledge is often not considered in designing such topologies. However, in engineering design and manufacturing, one needs to understand system behaviors in detail in order to produce better systems or products.

Toward this goal, a methodology to design a modular ANN topology by integrating existing knowledge is proposed in this research. The modular ANN is composed of zones where system-related knowledge is available and synapses/weights of neurons in the ANN can be precomputed without training. In addition, detected zones where knowledge about the system is insufficient to precompute weights, classical ANNs are trained using experimental data. This proposed methodology is applied to an FDM process to elucidate how the topology of a modular ANN is derived. The approach helps to understand how a modular ANN structure composed of a mixture of known zones with precomputed weights and unknown zones requiring training can improve the performance of ANNs compared to the classical ANN approach. The existing process knowledge is integrated into this research using the dimensional analysis conceptual modeling (DACM) framework [19]. The DACM framework recently has been applied in various engineering domains, including reverse engineering, early stage design, multidisciplinary optimization, and in this research is applied to artificial intelligence. This conjoint modeling approach, combining DACM and classical ANNs, results in a new type of modular knowledge-based ANN (KB-ANN), differing greatly from existing KB-ANN methods. The expected outcome of this article is to benefit from the existing knowledge of a system and encode this knowledge in the form of causal graphs linking different variables of a system. Specifically, in this research, variables are termed as neurons and causal graphs generated using DACM are considered as an ANN. Training datasets from experiments are required only for the zones in the KB-ANN where existing knowledge is limited, nonexistent, or difficult to extract. For zones with sufficient pre-existing knowledge, the training process is replaced by precomputed weights for neurons using the DACM methodology. This conjoint experimental and modeling approach using KB-ANN is used to predict wall thickness, height, and mass of a part produced using the FDM process. The developed KB-ANN model is compared to a classical, fully connected ANN model under prescribed performance metrics.

The research is organized as follows: Sec. 2 describes the experimental procedure used in the study and the approaches considered to encode knowledge for designing the ANN topology. Section 3 presents the case study and the application of the developed methodology for the case study. Section 4 discusses the key results of the study, and Sec. 5 concludes the work and briefly describes future development efforts.

2 Background

2.1 Design of Experiments. Performing experiments by varying one-factor-at-a-time is cost intensive. Thus, design of experiments (DOE) proposes a set of principles to maximize the efficiency of experiments by minimizing the number of experiments to be conducted. One of those principles is the use of factorial experiments. Full factorial DOE explores all the possible combinations of factors and levels [20]. In AM, the number of factors influencing the part quality is potentially large, and it is impractical and difficult to explore all the potential combinations of these possible factors. Sampling, which is the use of a subset of the experimental space, is consequently required to explore this space at an acceptable cost [21,22]. Currently, there are multitude of sampling methods available to explore the experimental space, such as stratified sampling, probability sampling, and sequential sampling [20,22–25]. In this research, DOE plays a role in collecting training datasets for zones of nonexistent knowledge. Thus, the sampling method must explore the experimental space in those zones to ensure good generalization for the ANN training. Plackett–Burman design or Taguchi's orthogonal arrays are proven to be useful in evaluating a small number of sample points considering interaction between the studied variables [20,26,27]. However, the Taguchi method has come under scrutiny due to its many weaknesses in terms of confusing signal-to-noise ratio statistics, nonadaptive and nonsequential approach to experimentation, and ignorance of randomization, and old data analytical approaches [28]. Hence, some of the classical Taguchi tools such as signal-to-noise ratio are not utilized in this study. Nevertheless, randomization is considered via the selection of Taguchi tables, and the analytics are developed as part of the KB-ANN approach. The Taguchi method is employed in this study mainly for its simplicity. In addition, optimizing the experimental design for AM is not the central idea of the study, but such optimization could further improve the accuracy of the developed process models in this research.

2.2 Metamodeling Using Artificial Neural Networks. Several metamodeling approaches, such as Kriging models, polynomial models, and neural network models, exist in the literature for modeling complex systems. A metamodeling approach using ANNs can provide numerous advantages for the FDM process and AM, in general. ANN enables the development of global predictive models integrating numerous parameters. Furthermore, ANNs can support the implementation of a closed-loop control system to improve part quality and process repeatability. In other metamodeling approaches, such as Kriging or Gaussian process regression, modeling is performed as black boxes built over a designed set of experiments. This means that existing knowledge of the process or system (e.g., process physics) is not used. The current effort is using this untapped knowledge of the process or system to enable a gray box or white box metamodeling approach. The proposed approach differs from classical surrogate methods in using existing knowledge; thus, the number of required experiments is reduced. The experiments are not used to train the entire model, but to train only parts of the model. Through knowledge extraction and management, we can limit the need for experimental data sets by integrating the existing system knowledge available for the observed process into the ANN [19,29]. Nevertheless, existing knowledge in the literature is represented in multiple forms and lacks interoperability [30]. For this reason, the DACM framework is utilized to integrate different knowledge to be coherent with each other, and to visualize the cause–effect relationships in the form of causal graphs.

2.3 Dimensional Analysis Conceptual Modeling Framework. Coatanea et al. [19] developed a method to extract and encode knowledge associated with system architectures, equations, and measuring units. The encoded knowledge is represented in the form of causal graphs. DACM can be an efficient approach to the creation of surrogate models and for adaptively training an ANN. Modeling starts with designation of the system boundary and definition of the model's objectives. Functional representation is used to represent the sequence of functions taking place in the system of interest. Those sequences of functions describe the different behavior of the observed system. DACM transforms the initial function model into a generic functional model formed around a limited set of fundamental functions and uses the causal rules extracted from bond graph theory [31–33]. The dimensional analysis (DA) is applied to each node of the graph to form behavioral equations. A color pattern is applied to different variables to highlight their design nature. The primary result of this modeling is a colored hypergraph with a list of governing equations for the system of interest. The model can then be used for qualitative or quantitative simulations, and to search for contradicting variables during optimization. Figure 1 visualizes different steps in the DACM Framework; the process ends when a computable model of the system of interest is available with the required level of detail.

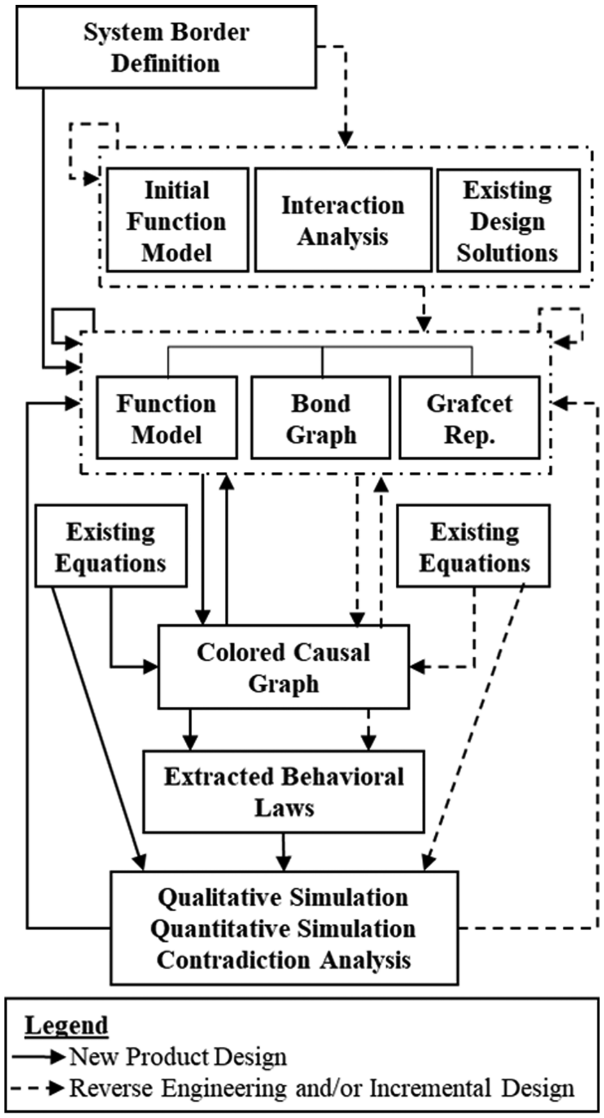
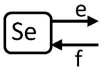
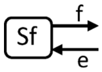
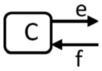
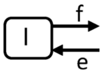
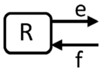
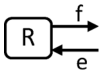
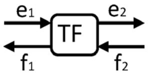
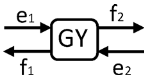
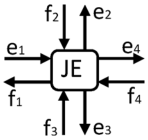
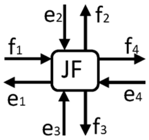


Fig. 1 Visual representation of the DACM framework

Generic functions represented by bond graph organs are used as an intermediate level between the classical functional models and the final causal graphs [32,33]. To facilitate the systematic assignment of variables to the generic functional representation, regardless of the energy domain, all variables are classified into five generalized categories, namely, flow, effort, momentum, displacement, and connecting [34].

The mathematical relationship between generic variables describes how the variables relate to each other. The sequence of functions in the functional model provides initial insight into the global causality. Mapping functions to the generic functional elements enables the extraction of the causality among the variables characterizing those functions. Table 1 summarizes the causal rules in the DACM approach. Figure 2 represents a causal extraction algorithm used to automate the DACM modeling process.

Table 1 Causality for generic functions and associated bond graph elements [52]

Bond graph element	Schematic view	Bond graph element	Schematic view
Source of effort (Se) fixed effort-out causality		Source of flow (Sf) fixed flow-out causality	
Capacitor (C) fixed effort-out causality		Inertia (I) fixed flow-out causality	
Resistor (R) preferable effort-out causality (Resistive)		Resistor (R) preferable flow-out causality (Conductive)	
Transformer (TF) maintain incoming causality (two-port element)		Gyrator (GY) switch incoming causality (two-port element)	
Effort junction (JE) or (0) (multiport element) $e_1=e_2=e_3=e_4 f_1+f_2+f_3+f_4=0$		Flow junction (JF) or (1) (multiport element) $f_1=f_2=f_3=f_4 \;\; e_1+e_2+e_3+e_4=0$	

First, the algorithm checks if a generic functional organ is defined for each functional box. Then, the algorithm explores each functional box of the model from start to end, to verify that there is no conflict in the coherence of the generic functional representation in terms of causality.

Finally, according to categories of assigned variables and using the causal rules (Table 1), the cause–effect relationships between variables are established [19,35]. The causal graph generated using DACM is used to define the topology of ANNs during process modeling.

2.4 Empirical Learning Using Artificial Neural Networks. Machine learning methods are empirical learning techniques. Empirical learning systems inductively generalize from specific examples. They usually require little theoretical knowledge about the problem domain. This advantage is compensated by the need for a large training data set. ANNs have proven to be equal, or superior, to other empirical learning systems over a wide range of domains, when evaluated in terms of their generalization ability [36,37]. ANNs are usually comprised of layers (k) with nodes (j), where each node sums up i weighted outputs of the nodes from the previous layer as per the following equation:

$$s_{j,k} = \sum\nolimits_i w_{ij,k} x_{i,k-1} + w_{0,k} \tag{1}$$

In Eq. (1), $s_{j,k}$ represents the weighted sum of node j at layer k, w_{ij} represents the weight of ith output at node j, $w_{o,k}$ represents the initial weight of layer k at the first node. This summation is passed through a nonlinear activation function, the output of which acts as input for the next layer. A common choice for the activation function is the sigmoid function, which is also called the continuous unit step function

$$h(s_{j,k}) = \frac{1}{1+\exp(-s_{j,k})} \tag{2}$$

The computational power attributed to these networks originates from these nonlinear functions $h(s_{j,k})$ of the weighted sums. However, the nonlinearity also makes it difficult to mathematically analyze these networks at a deeper level and requires a large set of training data to capture the desired relationship. In ANNs, a state p_i of a neural network can be uniquely described by $\{w_1 \ldots w_n\}_p$ where w_i represents a weight within the network. During the training process, the network goes through a subset of the state space (p) continuously improving the model performance. The hypothesis for the following investigation is that the total number of states p and the total number of weights n of an ANN can be reduced by incorporating prior knowledge about the system. Thus, this approach can increase the efficiency of the model, while reducing the computational cost.

The initial weights allocated to the network can greatly affect how well ANNs can learn [38]. The initial weights allocated is also the central source of the well-known vanishing gradient problem associated with ANNs [39], which is present when training ANNs with gradient-based learning methods and backpropagation. According to Hochreiter et al. [39], in such methods, each of the ANN's weights receives an update proportional to the gradient of the error function with respect to the current weight in each iteration of training. In some cases, the gradient will be vanishingly small, preventing the weight from changing its value. In the worst case, this may completely stop the ANN from training. This problem is more likely when too many hidden layers of neurons are used in an ANN. Some heuristic rules can be used to constrain the potential size of an ANN [40]. This implies that ANN designs must be small in terms of the number of inputs, number of outputs, size of hidden layers, and number of hidden layers.

2.5 Artificial Neural Network-Based Process Modeling. Artificial neural networks with numerous architectures and training algorithms are utilized in process modeling and forecasting output variables. ANNs with the assistance of data standardization, data preprocessing, and model performance optimization have become a key enabler in modeling different processes. The main advantages of ANNs in modeling when compared to other process modeling methods are, (i) its ability to handle noisy and ambiguous data, (ii) lower cost of implementation than other approaches, (iii) and their suitability for accurate representation of dynamic problems [41,42]. However, it is only possible to perform black box modeling using classical ANNs, resulting in limited information about the hidden layers and relations between the layers. This lack of process information during ANN topology design can result in overfitting models due to the empirical nature of ANNs [42]. Thus, research must be focused on designing ANN architectures that are transparent and require less computation to improve cost-effectiveness.

2.6 Relevance of Designing an Artificial Neural Network Topology for Manufacturing. In manufacturing, several problems are associated with capturing and using existing knowledge. This knowledge can be efficiently used to reduce the size and complexity of engineering models and be applied to the design of

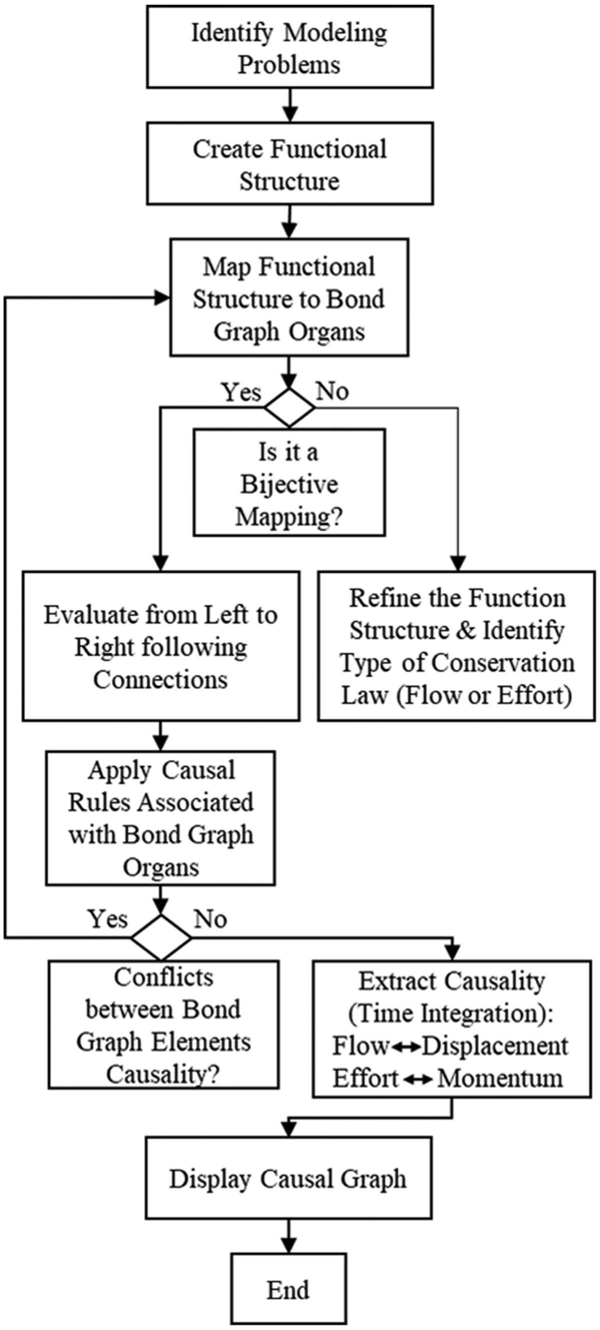

Identify Modeling Problems
Create Functional Structure
Map Functional Structure to Bond Graph Organs
Yes
No
Is it a Bijective Mapping?
Evaluate from Left to Right following Connections
Refine the Function Structure & Identify Type of Conservation Law (Flow or Effort)
Apply Causal Rules Associated with Bond Graph Organs
Yes
No
Conflicts between Bond Graph Elements Causality?
Extract Causality (Time Integration): Flow↔Displacement Effort↔Momentum
Display Causal Graph
End

Fig. 2 The causal relationship extraction algorithm

ANN topologies. Dimensional analysis offers a way to reduce problem dimensionality by combining variables. For example, inputs of an ANN can be combined into dimensionless groups. This transformation directly affects the number of weights to be trained and, intuitively, should have a positive effect on ANN performance. Similarly, the DACM method generates causal graphs that can be seen as elementary ANN topologies. Such ANNs can quickly grow in size and face the problems presented above, such as, vanishing gradient issue and overfitting. For this reason, a modular approach to model the target variables is presented. For example, causal graphs such as the one presented in Appendix A can be seen as an ANN topology having multiple zones. For zones where sufficient process knowledge is available, weights can be precomputed without the need for training datasets. A small portion of the causal graph in Appendix A is shown in Fig. 3 as an example of a knowledgeable zone. The bubbles in the figure

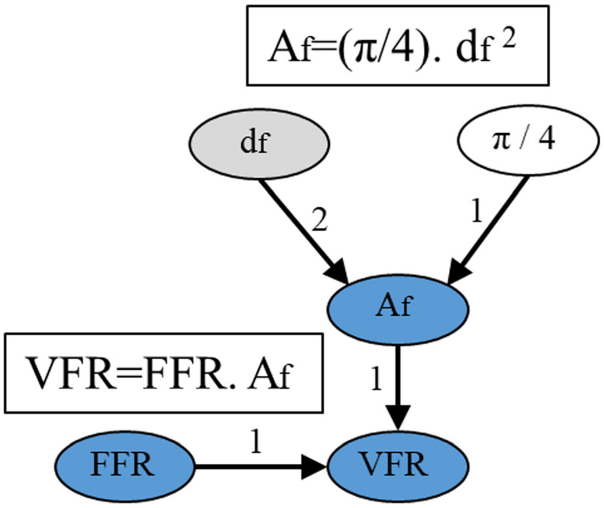


Fig. 3 Knowledgeable zone in the causal graph with precomputed weights

represent different variables within a knowledgeable zone. The precomputed weights are shown on the connectors between these variables.

For the zones where knowledge is nonexistent, a combination of modules consisting of classical ANNs are modeled. The modules are smaller ANNs that can be trained separately. The intermediate blue nodes represent, in an explicit manner, the locations where sensors could be implemented in the AM process to collect data required to train the local ANNs. If sensing in these locations is not possible, the intermediate data will need to be simulated or otherwise predicted. The central concept of a knowledge-based ANN remains, i.e., to use existing knowledge of a specific problem to develop a topology supporting faster training and better performance using smaller datasets [43]. This approach is a "hybrid learning system" because it combines empirical learning and domain theory learning [44]. Experimental training examples are used for empirical learning and domain theory learning is completed by encoding existing knowledge.

As described above, a KB-ANN is a hybrid-learning network that uses both theoretical knowledge and empirical data to construct a model of a physical system. Knowledge extraction and encoding in a KB-ANN can enable superior interpolation and extrapolation to estimate unmeasurable parameters. The main aim of the KB-ANN is to apply, transfer, and translate pre-existing knowledge into a hybrid neural network [45]. This allows for consolidation of knowledge to develop a global model of the system. Traditional KB-ANN development algorithms for a system to learn from both existing knowledge and empirical examples are shown in Fig. 4. The hybrid-learning approach starts with the conversion of existing knowledge to symbolic rules using the rules-to-network algorithm. These rules are used to construct and initialize a neural network that performs as a classifier that adheres to the rules upon which it is built. The next step involves using the network-training algorithm to train the classifier (initial ANN structure) using empirical examples to obtain a final trained ANN. Hence, the traditional KB-ANN method involves training all nodes within the developed ANN.

However, the KB-ANN approach developed herein follows one central objective: the significant reduction in the size of training data. The resulting approach is unique, because a significant portion of the KB-ANN produced using this approach does not require training. Training can be eliminated for portions of the network by using the DACM framework to encode existing knowledge in the form of an ANN architecture. This initial architecture, which forms the backbone of the KB-ANN

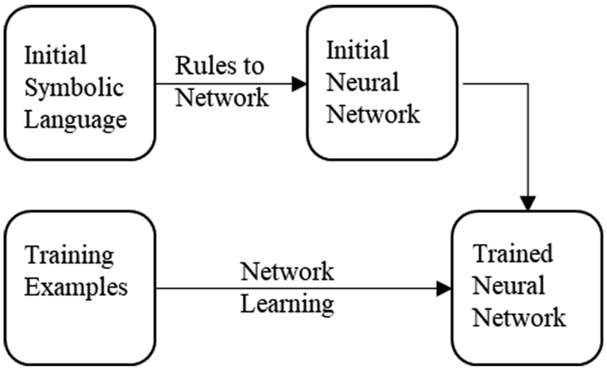


Fig. 4 Traditional KB-ANN development algorithm [44]

structure, is connected to smaller classical ANNs that represent the zones of the model where knowledge is nonexistent. Thus, the proposed approach differs from existing KB-ANN methods in terms of initial structure development and training; however, the proposed approach is similar to classical ANNs in terms of training knowledge-limited zones. Hence, using the approach described in the foregoing, classical ANNs and a KB-ANN are developed to model an FDM process, as described next.

3 Fused Deposition Modeling Case Study

In this section, it is demonstrated that exploration of the manufacturing space can be effectively performed using KB-ANN modeling to improve AM part quality, while keeping the number of required training data sets low. The printed FDM part used in this case study is shown in Fig. 5.

The FDM part has a wall thickness $e = 0.5 \pm 0.05$ mm and height $H_t = 12 \pm 0.05$ mm, constant for the entire profile. The concurrent modeling and experimental process applied in this study are decomposed into six steps summarized below (Fig. 6).

Step 1: Four initial printing tests are completed using preselected printing process parameters proposed by the slicing software (Repetier).

Step 2: The printing process parameters are analyzed and process parameters that will potentially affect the targets, namely, wall thickness (e), part height (H_t), and part mass (M_t) are extracted.

Step 3: A rapid evaluation of the effects of process parameters on the targets is performed with a few supplementary experiments, implying simultaneous variation of the parameters using orthogonal arrays. This evaluation is performed to find high latency variables that can later be fixed at a level minimizing their effect on the targets.

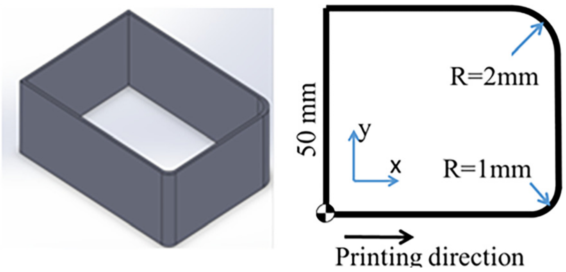


Fig. 5 Solid model (left) and dimensions (right) for the sample part [46]

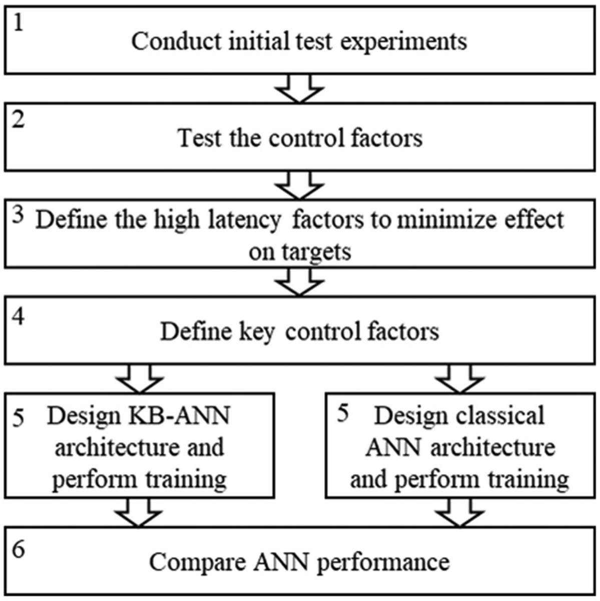

1 Conduct initial test experiments
2 Test the control factors
3 Define the high latency factors to minimize effect on targets
4 Define key control factors
5 Design KB-ANN architecture and perform training
5 Design classical ANN architecture and perform training
6 Compare ANN performance

Fig. 6 Concurrent experimental and modeling process

Step 4: The most significant parameters to achieve the expected thickness $e = 0.5 \pm 0.05$ mm and height $H_t = 12 \pm 0.05$ mm are selected for developing a predictive model.

Step 5: The prediction model for thickness, height, and mass are built for the remaining control factors: nozzle travel speed (TS), layer height (h_l), and extruder temperature (T_{set}). A colored causal graph is first developed using the DACM approach to encode the knowledge. The variables are classified into four main classes (i.e., colors). *Exogenous* variables (shown in black) are outside the system boundary and part of the surrounding environment of the system. They cannot be modified by the designer, but are imposed on the system. *Independent* design variables (shown in green) are not influenced by any other variables in the system, and their value can be modified by designers (examples include the nozzle travel speed, extruder temperature, and layer height). *Dependent* design variables (shown in blue) are influenced by other variables such as exogenous and independent variables, and are difficult to modify and control. *Performance* variables (shown in red) are the objective variables (selected by the designers to evaluate the performance of a system) and are usually dependent variables. In this case, wall thickness, part height, and mass are the dependent variables selected as the performance variables. Finally, the developed causal graph is translated into an ANN topology, which is designed for maximum compactness to maintain all the connections in the causal graph. Three classical ANNs and a KB-ANN topology are evaluated in this study. The KB-ANN topology developed in the form of modular ANNs is shown in the causal graph in Appendix B.

Step 6: The three classical ANNs and the KB-ANN are compared for performance and prediction capability to evaluate the relative utility of the selected approaches.

4 Results and Discussion

In steps 1 and 2, initial experimental prints were created and the process parameters that could affect printed part quality were detected. In step 3, the most influencing factors were taken into consideration, i.e., the layer height (h_l) in mm, the extruder temperature (T_{set}) in °C, the nozzle TS in mm/s, and the fan speed (Fan) in rpm. In step 4, these four input parameters influencing the wall thickness, part height, and mass were analyzed at three possible levels. An L27 standard orthogonal array was adopted,

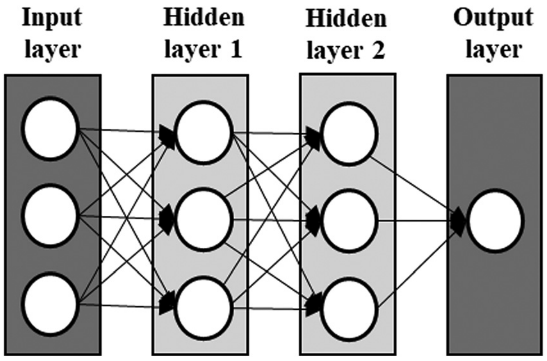


Fig. 7 A classical ANN model for FDM

and each of the necessary 27 experiments was replicated once to ensure repeatability of the FDM machine. In step 4, fan speed variations were removed from the model because of the latency of its effects on the three performance variables. The fan speed parameter was fixed to a value of ON at 50%.

In step 5, the prediction model for thickness, height, and mass was developed. The causal graph developed using the DACM framework for FDM is presented in Appendix A. The developed causal graph is simplified and represented in the form of modular ANNs for designing the KB-ANN topology (Appendix B). In both causal graphs, the independent variables (TS, h_i, and T_{set}) are represented in green. The dependent variables of the system are represented in blue. The performance, or target, variables are represented in red. The nodes of the causal graph from DACM are connected using black leader lines, where the arrows represent the direction of causality and the numbers represent precomputed weights for knowledgeable zones. Classical ANNs are presented in zones where knowledge is nonexistent. From the developed causal graph structure, two types of ANNs are developed, namely, a classical fully connected ANN and a KB-ANN.

4.1 Classical Artificial Neural Network.

Three classical ANNs (Fig. 7) are designed to model the three outputs, namely, wall thickness, part height, and mass using three inputs: layer height, travel speed, and extruder temperature. The ANNs are designed with two hidden layers consisting of three (3) neurons each and one output layer with one (1) neuron. The performance of the network is measured in terms of mean squared error (MSE). The Levenberg–Marquardt algorithm was chosen as the training function, and the tangent sigmoid function was chosen for the transfer function [47]. The input data for the ANN was divided, using 70% for training, 15% for validation, and 15% for testing.

Typical performance graphs contain three curves, namely, a training curve, a validation curve, and a test curve, which together indicate the mean square error of a training process. The performance curves indicate the quality of the training in terms of error reduction, under-fitting (bad training), and overfitting. For a good fit performance, the three curves must follow a downward trend indicating low MSE. In addition, the curves must be smooth and must follow the pattern of training and testing curves at the very bottom, followed by a validation curve.

The performance curves (Figs. 8(*a*)–8(*c*)) for the fully connected classical ANNs are shown in Fig. 8. The MSE value for best performance was found to be 5.43×10^{-4} after nine iterations (Fig. 8(*a*)), 1.15×10^{-4} after ten iterations (Fig. 8(*b*)), and 2.01×10^{-3} after 23 iterations (Fig. 8(*c*)) for wall thickness, part height, and part mass, respectively.

The performance curves obtained from the classical ANN (Figs. 8(*a*)–8(*c*)) are compared to the performance curve (Fig. 9) of a standard function, $z = \sin(x)\cos(y)$, modeled as a best fit performance using a classical ANN for 100 training samples. It is seen from Fig. 8(*a*) (wall thickness) and Fig. 8(*c*) (mass) that the training curve follows a downward trend, while the testing and validation curves follow a downward slope until the lowest MSE value achievable; it then trends slightly upward, indicating a low generalization to inputs with values lying outside the range of the

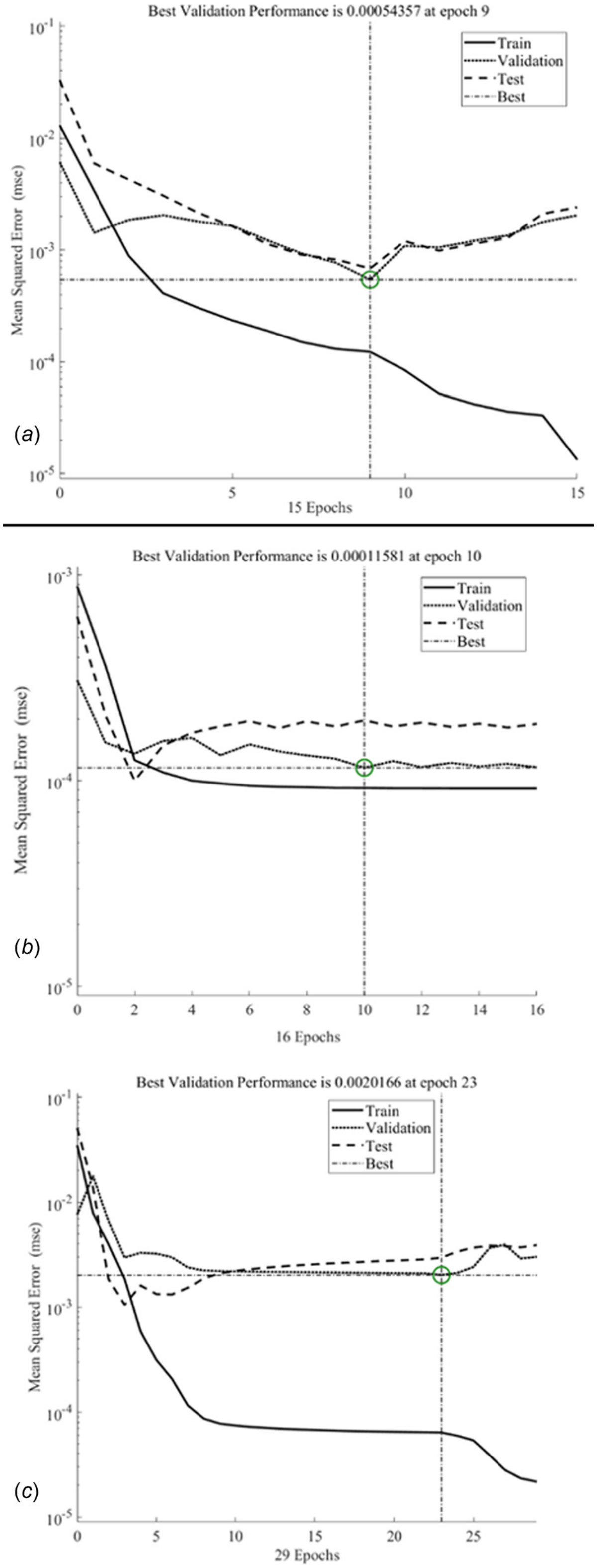


Fig. 8 Performance curves for classical fully connected ANN to model: (*a*) part wall thickness, (*b*) part height, and (*c*) part mass

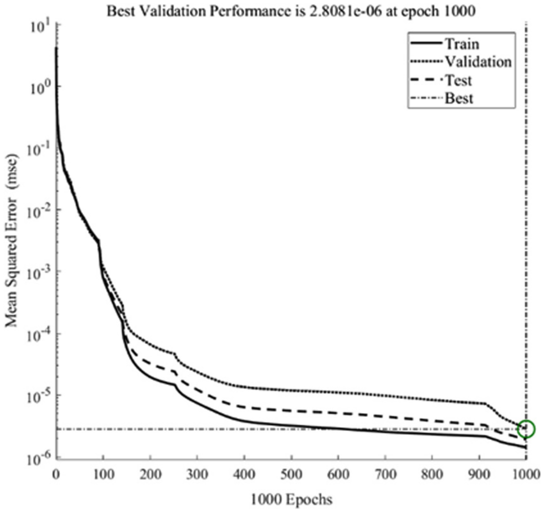


Fig. 9 Performance curve for best-fit scenario (standard function $z = \sin(x).\cos(y)$)

training data. In addition, the upward trend of the validation and testing curves against the continuous downward trend of the training curve indicate the possibility of overfitting. In Fig. 8(*b*), curves are steady at a fixed MSE value with the validation curve trending below the testing curve, indicating a poor fit to the provided data samples and a low level of generalization for inputs that lie outside the training state.

All performance curves show MSE value at low number of iterations indicating that the ANNs could not find a better fit or any reduction in MSE past that iteration point. This is opposed to the best-fit scenario shown in Fig. 9, which obtained best performance MSE value at 1000 iterations.

4.2 Knowledge-Based Artificial Neural Network. The KB-ANN was designed as four modular ANNs following the simplified causal graph, as shown in Appendix B. The first modular ANN is designed for one output: ratio of viscosity (μ) of molten polymer at extrusion temperature to the viscosity (μ_i) of molten polymer at a reference temperature (175 °C). The filament feed rate and extruder temperature (T_{set}) are used as inputs. Here, the output of the modular ANN 1 is an intermediate (blue) variable, which cannot be directly measured and, hence, has to be estimated using numerical simulations. A direct solution to the simulation of all the missing data is not provided in this study. In some cases, these data can be computed using the DACM method or numerical simulation, or can be directly measured using sensors. In this research, numerical simulation of viscosity was carried out using the CFD Module of COMSOL MULTIPHYSICS software. The Navier–Stokes equation (Eq. (3)) and continuity equation (Eq. (4)) are solved for the conservation of the momentum and conservation of the mass [48].

$$\rho\frac{\partial u}{\partial t} = \nabla.\left[-pI + \mu\left(\nabla u + (\nabla u)^{\mathrm{T}}\right] + \rho g + F_{\text{st}} + F \right. \tag{3}$$

$$\rho(u.\nabla u) = 0 \tag{4}$$

Viscosity of the fluid has been considered using the Carreau model in the numerical simulation (in the following equation):

$$\mu = \mu_{\text{inf}} + (\mu_0 - \mu_{\text{inf}})\left[1 + (\lambda\dot{\gamma})^a\right]^{\frac{n-1}{a}} \tag{5}$$

The viscosity of polylactic acid has been previously determined by measurement from a parallel-plate rheometer for frequency range 100–0.1 s^{-1}. Tests were carried out at four different temperatures: 175 °C, 185 °C, 195 °C, and 205 °C [49]. The viscosity curves were fitted with the Carreau–Yasuda equation using ORIGIN software. The terms of the Carreau–Yasuda equation for the studied temperatures are shown in Table 2. These terms have been implemented in the ORIGIN software in order to determine the flow properties in the liquefier.

Table 2 Values for Carreau–Yasuda viscosity curve fitting

	175 °C	185 °C	195 °C	205 °C
μ_0 (Pa·s)	5169 ± 5	2480 ± 14	1945 ± 16	726 ± 6
μ_{inf} (Pa·s)	0	0	0	0
λ	0.048 ± 0.02	0.09 ± 0.5	0.08 ± 0.02	0.05 ± 0.01
a	0.82 ± 0.3	1.6 ± 0.8	1.931 ± 0.5	2.60 ± 0.01
n	0.52 ± 0.3	0.7 ± 0.3	0.693 ± 0.2	0.79 ± 0.11

The second modular ANN is designed for the output, wall thickness. To reduce the dimensionality of the ANN, the inputs to predict wall thickness were represented in the form of Pi numbers (dimensionless primitives) using DA [50]. A widely used theory in DA is the Vashy–Buckingham π theorem [51,52], which identifies the number of these independent dimensionless primitives (Pi numbers) that characterize a given physical problem. The dimensionless primitives are the invariants of the problem, where the term "invariant" is understood here as a relationship deeply connected with the behavior of certain aspects of a phenomenon. The DA method offers a way to simplify complex problems by grouping variables into dimensionless primitives. Every law which takes the form $y_{\text{o}} = f(x_1, x_2, x_3, \ldots, x_n)$ can take the alternative form shown in Eq. (6), where π_{i} (for $i = 1$ to n) is the dimensionless product for the variable x_i and π_0 is the dimensionless product of variable y_0

$$\pi_0 = f(\pi_1, \pi_2, \ldots, \pi_n) \tag{6}$$

Equation (6) is the final form of the dimensional analysis and is the consequence of the Vashy–Buckingham theorem for the variable x_i, which takes the form shown in the following equation:

$$\pi_y = y_i x_j^{\alpha_{ij}} x_l^{\alpha_{il}} x_m^{\alpha_{im}} \tag{7}$$

Here, x_j, x_l, and x_m are called repeating variables, y_i is the performance variable, and the α values are exponents ensuring the dimensional homogeneity of the relation.

The third modular ANN was designed for part height (H_t) as the output, with layer height (h_i) and number of layers (n) as the inputs. The fourth modular ANN was designed for mass (M_t) as output, with wall thickness (e), height (H_t), and density of the material (ρ) used as the inputs. ANN performance was measured using MSE. The Levenberg–Marquardt algorithm was chosen as the training function and tangent sigmoid function was chosen for the transfer function [46]. The input data for the ANN were divided, using 70% for training, 15% for validation, and 15% for testing. The modular ANNs are designed with one hidden layer consisting of three nodes each. The performance curves for the four modular ANNs are shown in Figs. 10(*a*)–10(*d*).

Figure 10(*a*) shows that the modular ANN 1 was able to obtain the best validation performance at the 53rd iteration with an MSE of 7.7186 × 10^{-5}. The performance curves show training, testing, and validation following each other in a downward trend, indicating a good fit and good generalization capability. The downward trend also implies that a better model could be obtained by increasing the number of training samples. The results for modular ANN 2 for wall thickness (e) are shown in Fig. 10(*b*). The observed MSE was found to be 9.30 × 10^{-5} after 93 iterations. The curves show overlap during the first ten iterations, but soon smoothen and follow a uniform trend. This shows that the ANN

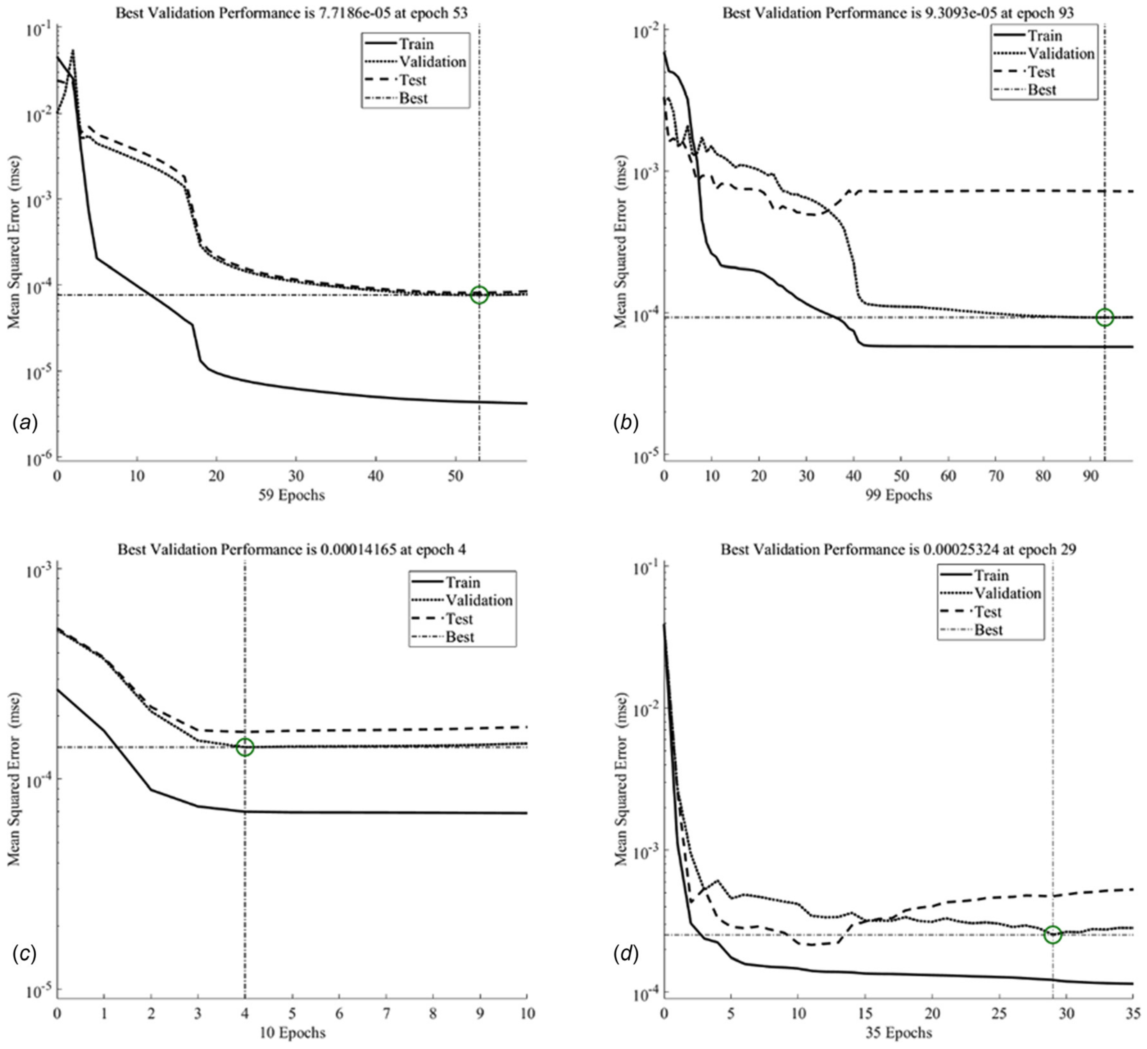


Fig. 10 Performance curves for: (*a*) modular ANN 1 (viscosity) in the KB-ANN, (*b*) modular ANN 2 (thickness) in the KB-ANN, (*c*) modular ANN 3 (height) in the KB-ANN, and (*d*) modular ANN 4 (mass) in the KB-ANN

was able to train for 93 iterations without failure, indicating a good fit to the training data. The modular ANN 3 results for part height shown in Fig. 10(*c*) have best-fit performance with an MSE of 1.41×10^{-4} after only four iterations. The curves are smooth and follow each other in the graph; however, the ANN achieved the best performance at four iterations, indicating a mediocre fit to the training data. Finally, the results of modular ANN 4 for part mass (M_t) are shown in Fig. 10(*d*). The observed MSE is 2.54×10^{-4} after 23 iterations. It was seen that the performance curves follow a downward trend with the validation curve below or at par with the testing curve. This indicates an average fit to the provided data samples, but with the possibility of overfitting.

4.3 Validation. The validation of the developed models was carried out with nine experimental tests. The values for the independent input variables (layer thickness, extruder temperature, and travel speed) were chosen at random. The range of values for the independent variables are as follows, layer thickness (0.1–0.4 mm), extruder temperature (175–215 °C), and travel speed (5–19 mm/s). From validation, the standard prediction errors for thickness, height, and mass using the KB-ANN were found to be 0.1627, 0.3647, and 0.4621, respectively. Similarly, the prediction errors for the fully connected classical ANN were found as 0.1376 (thickness), 0.5898 (height), and 0.4667 (mass). The propagated global error of the KB-ANN model was found to be 0.5220.

It can be noted that the KB-ANN global model error is propagated due to the output of modular ANN 1 (viscosity) acting as input for modular ANN 2, and similarly, the output of modular ANN 2 (thickness) and ANN 3 (height) acting as input for modular ANN 4 (mass). The MSE and standard error calculated after validation for the two types of networks are compared in Fig. 11. It is seen that the errors for the KB-ANN are in the same range as the prediction error of the classical ANN.

4.4 Model Comparison. In the case presented above, the KB-ANN method performed better than the classical fully connected ANN in terms of fit to the provided experimental data. Specifically, the prediction error for the KB-ANN method was found to be nearly the same as the classical approach for wall thickness and part mass, while lower for part height. This prediction error was largely the result of lost information when streamlining the complete causal graph (Appendix A) to the simpler version (Appendix B). In particular, the regression fit for the height using the KB-ANN method was poor largely due to the absence of adequate knowledge or models to represent the phenomena that influence part height. For instance, the cooling effect of the fan may affect the solidification rate

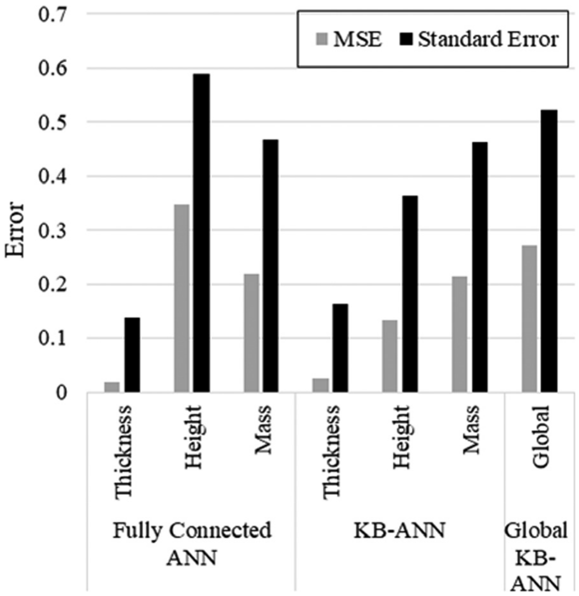


Fig. 11 Comparison of validation error for the fully connected classical ANN and the developed KB-ANN

of the molten polymer, resulting in tight bonding or sparse bonding of layers, which would have a direct impact on part height. The addition of key variables to the study through simulation results or experimental estimates would improve the training and reduce the prediction error. The key difference between the classical ANN model and KB-ANN model lies in the total number of weights that define the state space for each model. In this example, the KB-ANN model had 12 fewer weights than its classical counterpart. Further, the hidden layer of the KB-ANN model operates within the dimensionally homogenous space. The combination of these factors (number of weights trained and dimensionally homogenous hidden layer) results in improved efficiency during fit to training and similar MSE performance for the same number of training (27) samples. This increase in efficiency is visible in the training and its robustness. The KB-ANN method provides more robust generalization compared with the classical ANN approach. Nevertheless, the small number of data in the training set used in this effort, limit the conclusions that can be drawn from the training, validation, and testing. The higher number of epochs required for training in the KB-ANN method demonstrate that the difference in training epochs will increase with bigger datasets. It also indicates that KB-ANN can provide better results for smaller datasets.

5 Conclusions

This research developed a KB-ANN approach to limit the amount of required experiments for training and validating ANNs for characterizing a manufacturing system. The approach was applied to develop a metamodel capable of dynamically predicting control factors of fused deposition modeling. Benefits were gained from causal graph representation, which enabled the design of KB-ANN as modular ANNs with reduced dimensionality. The results demonstrated the superiority of the KB-ANN approach over classical full-connected ANNs in terms of fit and regularization for the same performance and same number of training samples. The case study was limited to the prediction of three target variables in comparison to the large number of target variables that essentially need to be modeled for a complex FDM system. In reality, more intermediate variables and phenomena (Appendix A) need to be modeled to represent the FDM process holistically. The work reported herein demonstrates an initial proof of concept for the techniques and approaches that can be used to combine knowledge in a modular manner and to reduce dimensionality of complex problems using knowledge extraction, representation, and integration techniques such as dimensional analysis.

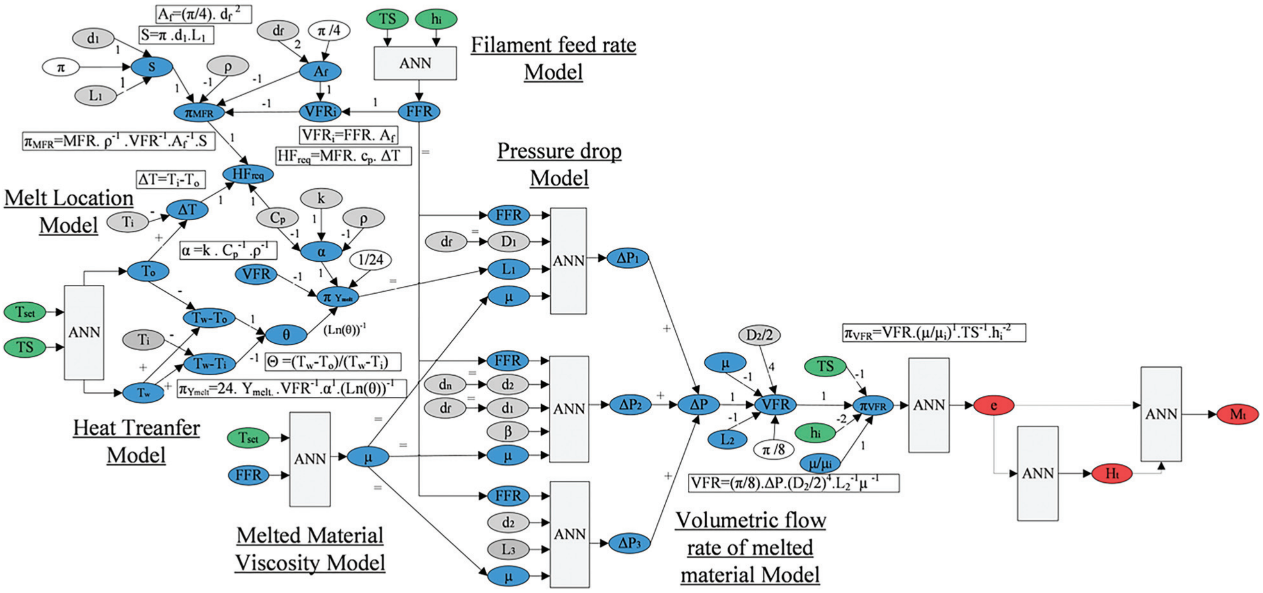
6 Future Work

The authors are expanding the case study analysis to model a larger set of target variables for FDM using the current methodology to obtain a holistic FDM process model. The developed model will be usable for real production simulations and process parameter tuning. In addition, the DACM method in this study generated causal graphs that were used to define elementary ANN topologies. Such elementary ANNs when expanded for a larger set of variables can grow quickly in size and be faced with problems such as, high training time, high training cost, requirement of large training data, high probability of model fitting issues (under-fitting or over-fitting), and vanishing gradient issue. Thus, a multilevel hierarchy approach to ANN topology creation as well as the use of regulators for the flow of values similar to long short-term memory neural networks is being investigated. This would allow for the development of a hierarchy of variables ranked based on importance and modular ANNs for variables in sequence following the hierarchy. This method could help in prioritizing variables and constrain the size of each modular ANN in the holistic KB-ANN model reducing training time and cost. The approach would also improve the accuracy of the ANN training and in turn reduce the prediction error of the holistic FDM model.

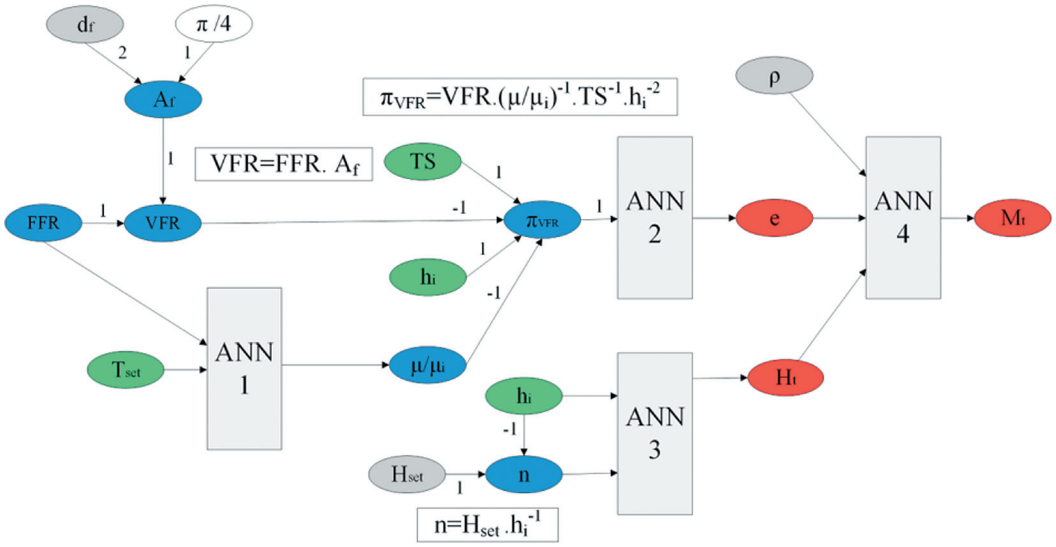
Nomenclature

$a =$	dimensionless parameter describing the transition between the first Newtonian plateau and the power law zone
$A_f =$	cross-sectional area of filament (mm^2)
$c_p =$	heat capacity (J/kg K)
$d_f =$	diameter of filament (mm)
$d_i =$	diameter at ith section of liquefier nozzle (mm)
$dx/dt =$	nozzle velocity in x direction for dx (mm/s)
$dy/dt =$	nozzle velocity in y direction for dy (mm/s)
$e =$	intended wall thickness (mm)
$F_{\mathrm{st}} =$	force resulting from surface tension (N)
$h_i =$	layer height (mm)
$H_t =$	part height (mm)
$k =$	coefficient of conduction (W/m K)
$L_i =$	length at ith section of liquefier geometry (mm)
$M_t =$	part mass (g)
$n =$	power index
$T_i =$	initial filament temperature (°C)
$T_{\mathrm{ref}} =$	reference temperature (°C)
$T_w =$	wall temperature (°C)
$T_0 =$	output temperature (°C)
TS $=$	nozzle travel speed (mm/s)
VFR $=$	volumetric flow rate of filament (mm^3/s)
$\beta =$	conical angle of liquefier geometry
$\gamma =$	shear rate (s^{-1})
$\Delta P_i =$	pressure drop (Pa) at ith section of nozzle
$\Delta V =$	change in nozzle travel velocity (mm/s)
$\theta =$	dimensionless temperature
$\lambda =$	relaxation time index
$\mu =$	viscosity of polymer filament (Pa·s)
$\mu_i =$	kinematic viscosity (m^2/s) at reference temperature
$\mu_{\mathrm{inf}} =$	viscosity at the infinite shear rate (Pa·s)
$\mu_0 =$	viscosity of fluid at zero shear rate (Pa·s)
$\rho =$	filament density (kg/m^3)

Appendix A: Detailed Causal Graph for Fused Deposition Modeling Using Dimensional Analysis Conceptual Modeling



Appendix B: Simplified Causal Graph With Modular KB-ANN Topology for Fused Deposition Modeling Using Dimensional Analysis Conceptual Modeling



References

[1] Bourell, D. L., Leu, M. C., and Rosen, D. W., 2009, "Roadmap for Additive Manufacturing: Identifying the Future of Freeform Processing," University of Texas at Austin Laboratory of Freeform Fabrication Advance Manufacturing Center, Austin, TX.

[2] Gibson, I., Rosen, D., and Stucker, B., 2014, *Additive Manufacturing Technologies: 3D Printing, Rapid Prototyping, and Direct Digital Manufacturing*, Springer, New York.

[3] Wang, X., Jiang, M., Zhou, Z., Gou, J., and Hui, D., 2017, "3D Printing of Polymer Matrix Composites: A Review and Prospective," Compos. Part B Eng., **110**, pp. 442–458.

[4] Rabinovich, J. E., 2002, "Rapid Manufacturing System for Metal, Metal Matrix Composite Materials and Ceramics," U.S. Patent No. 6,459,069.

[5] Batchelder, J. S., Curtis, H. W., Goodman, D. S., Gracer, F., Jackson, R. R., Koppelman, G. M., and Mackay, J. D., 1994, "Model Generation System Having Closed-Loop Extrusion Nozzle Positioning," U.S. Patent No. 5,303,141.

[6] Hilmas, G. E., Lombardi, J. L., Hoffman, R. A., and Stuffle, K., 1996, "Recent Developments in Extrusion Freeform Fabrication (EFF) Utilizing Non-Aqueous Gel Casting Formulations," Solid Freeform Fabrication Symposium (SFF), Austin, TX, Aug. 12–14, pp. 443–450.

[7] Bellehumeur, C., Li, L., Sun, Q., and Gu, P., 2004, "Modeling of Bond Formation Between Polymer Filaments in the Fused Deposition Modeling Process," J. Manuf. Process., **6**(2), pp. 170–178.

[8] Atif Yardimci, M., and Güçeri, S., 1996, "Conceptual Framework for the Thermal Process Modelling of Fused Deposition," Rapid Prototyp. J., **2**(2), pp. 26–31.

[9] Yardimci, M. A., Guceri, S. I., Agarwala, M., and Danforth, S. C., 1996, "Part Quality Prediction Tools for Fused Deposition Processing," International Solid Freeform Fabrication Symposium (SFF), Austin, TX, Aug. 12–14, pp. 539–548.

[10] Witherell, P., Narayanan, A., and Lee, J., 2011, "Using Metamodels to Improve Product Models and Facilitate Inferencing," Fifth IEEE International Conference on Semantic Computing (ICSC), Palo Alto, CA, pp. 506–513.

[11] Papadrakakis, M., Lagaros, N. D., and Tsompanakis, Y., 1998, "Structural Optimization Using Evolution Strategies and Neural Networks," Comput. Methods Appl. Mech. Eng., **156**(1–4), pp. 309–333.

[12] Varadarajan, S., Chen, W., and Pelka, C. J., 2000, "Robust Concept Exploration of Propulsion Systems With Enhanced Model Approximation Capabilities," Eng. Optim., **32**(3), pp. 309–334.

[13] Atashkari, K., Nariman-Zadeh, N., Gölcü, M., Khalkhali, A., and Jamali, A., 2007, "Modelling and Multi-Objective Optimization of a Variable Valve-Timing Spark-Ignition Engine Using Polynomial Neural Networks and Evolutionary Algorithms," Energy Convers. Manage., **48**(3), pp. 1029–1041.

[14] Magnier, L., and Haghighat, F., 2010, "Multiobjective Optimization of Building Design Using TRNSYS Simulations, Genetic Algorithm, and Artificial Neural Network," Build. Environ., **45**(3), pp. 739–746.

[15] Yegnanarayana, B., 2009, *Artificial Neural Networks*, PHI Learning, New Delhi, India.

[16] Wang, G. G., and Shan, S., 2007, "Review of Metamodeling Techniques in Support of Engineering Design Optimization," ASME J. Mech. Des., **129**(4), pp. 370–380.

[17] Schmidhuber, J., 2015, "Deep Learning in Neural Networks: An Overview," Neural Networks., **61**, pp. 85–117.

[18] Bengio, Y., Goodfellow, I. J., and Courville, A., 2015, "Deep Learning," Nature, **521**, pp. 436–444.

[19] Coatanéa, E., Roca, R., Mokhtarian, H., Mokammel, F., and Ikkala, K., 2016, "A Conceptual Modeling and Simulation Framework for System Design," Comput. Sci. Eng., **18**(4), pp. 42–52.

[20] Montgomery, D. C., 2017, *Design and Analysis of Experiments*, Wiley, Hoboken, NJ.

[21] Dowdy, S., Wearden, S., and Chilko, D., 2011, *Statistics for Research*, Wiley, Hoboken, NJ.

[22] Levy, P. S., and Lemeshow, S., 2013, *Sampling of Populations: Methods and Applications*, Wiley, Hoboken, NJ.

[23] Schillewaert, N., Langerak, F., and Duharnel, T., 1998, "Non-Probability Sampling for WWW Surveys: A Comparison of Methods," Mark. Res. Soc. J., **40**(4), pp. 1–13.

[24] Marshall, M. N., 1996, "Sampling for Qualitative Research," Fam. Pract., **13**(6), pp. 522–526.

[25] Robbins, H., 1985, "Some Aspects of the Sequential Design of Experiments," *Herbert Robbins Selected Papers*, Springer, Berlin, pp. 169–177.

[26] Plackett, R. L., and Burman, J. P., 1946, "The Design of Optimum Multifactorial Experiments," Biometrika, **33**(4), pp. 305–325.

[27] Roy, R. K., 2001, *Design of Experiments Using the Taguchi Approach: 16 Steps to Product and Process Improvement*, Wiley, New York.

[28] Maghsoodloo, S., Ozdemir, G., Jordan, V., and Huang, C.-H., 2004, "Strengths and Limitations of Taguchi's Contributions to Quality, Manufacturing, and Process Engineering," J. Manuf. Syst., **23**(2), pp. 73–126.

[29] Efthymiou, K., Sipsas, K., Mourtzis, D., and Chryssolouris, G., 2015, "On Knowledge Reuse for Manufacturing Systems Design and Planning: A Semantic Technology Approach," CIRP J. Manuf. Sci. Technol., **8**, pp. 1–11.

[30] Witherell, P., Feng, S., Simpson, T. W., Saint John, D. B., Michaleris, P., Liu, Z.-K., Chen, L.-Q., and Martukanitz, R., 2014, "Toward Metamodels for Composable and Reusable Additive Manufacturing Process Models," ASME J. Manuf. Sci. Eng., **136**(6), p. 061025.

[31] Hirtz, J., Stone, R. B., McAdams, D. A., Szykman, S., and Wood, K. L., 2002, "A Functional Basis for Engineering Design: Reconciling and Evolving Previous Efforts," Res. Eng. Des., **13**(2), pp. 65–82.

[32] Karnopp, D. C., Margolis, D. L., and Rosemberg, R. C., 2012, *System Dynamics: Modeling, Simulation, and Control of Mechatronic Systems*, Wiley, Hoboken, NJ.

[33] Shim, T., 2002, "Introduction to Physical System Modelling Using Bond Graphs," University of Michigan-Dearborn, Dearborn, MI.

[34] Coatanéa, E., 2005, "Conceptual Modelling of Life Cycle Design: A Modelling and Evaluation Method Based on Analogies and Dimensionless Numbers," Ph.D. dissertation, Helsinki University of Technology, Espoo, Finland.

[35] Mokhtarian, H., Coatanéa, E., Paris, H., Ritola, T., Ellman, A., Vihinen, J., Koskinen, K., and Ikkala, K., 2016, "A Network Based Modelling Approach Using the Dimensional Analysis Conceptual Modeling (DACM) Framework for Additive Manufacturing Technologies," ASME Paper No. DETC2016-60473.

[36] Shavlik, J. W., Mooney, R. J., and Towell, G. G., 1991, "Symbolic and Neural Learning Algorithms: An Experimental Comparison," Mach. Learn., **6**(2), pp. 111–143.

[37] Atlas, L., Cole, R., Muthusamy, Y., Lippman, A., Connor, J., Park, D., El-Sharkawai, M., and Marks, R. J., 1990, "A Performance Comparison of Trained Multilayer Perceptrons and Trained Classification Trees," Proc. IEEE, **78**(10), pp. 1614–1619.

[38] Ahmad, S., and Tesauro, G., 1989, "Scaling and Generalization in Neural Networks: A Case Study," Advances in Neural Information Processing Systems (NIPS), Denver, CO, Nov. 27–30, pp. 160–168.

[39] Hochreiter, S., Bengio, Y., Frasconi, P., and Schmidhuber, J., 2001, "Gradient Flow in Recurrent Nets: The Difficulty of Learning Long-Term Dependencies," *A field Guide to Dynamical Recurrent Neural Networks*, IEEE Press, New York.

[40] Panchal, G., Ganatra, A., Kosta, Y. P., and Panchal, D., 2011, "Review on Methods of Selecting Number of Hidden Nodes in Artificial Neural Network," Int. J. Comput. Theory Eng., **3**(2), pp. 332–337.

[41] Dawson, C., and Wilby, R., 2001, "Hydrological Modelling Using Artificial Neural Networks," Prog. Phys. Geogr., **25**(1), pp. 80–108.

[42] Tu, J. V., 1996, "Advantages and Disadvantages of Using Artificial Neural Networks Versus Logistic Regression for Predicting Medical Outcomes," J. Clin. Epidemiol., **49**(11), pp. 1225–1231.

[43] Ingrassia, S., and Morlini, I., 2005, "Neural Network Modeling for Small Datasets," Technometrics, **47**(3), pp. 297–311.

[44] Towell, G. G., and Shavlik, J. W., 1994, "Knowledge-Based Artificial Neural Networks," Artif. Intell., **70**(1–2), pp. 119–165.

[45] Psichogios, D. C., and Ungar, L. H., 1992, "A Hybrid Neural Network-First Principles Approach to Process Modeling," AIChE J., **38**(10), pp. 1499–1511.

[46] Mokhtarian, H., Coatanéa, E., Paris, H., Mbow, M. M., Pourroy, F., Marin, P. R., Vihinen, J., and Ellman, A., 2018, "A Conceptual Design and Modeling Framework for Integrated Additive Manufacturing," ASME J. Mech. Des., **140**(8), p. 081101.

[47] Moré, J. J., 1978, "The Levenberg-Marquardt Algorithm: Implementation and Theory," *Numerical Analysis*, Springer, Berlin, pp. 105–116.

[48] Anderson, J. D., and Wendt, J., 1995, *Computational Fluid Dynamics*, Springer, Berlin.

[49] Bakrani Balani, S., Chabert, F., Valerie, N., Cantarel, A., and Christian, G., 2017, "Toward Improvement of the Properties of Parts Manufactured by FFF (Fused Filament Fabrication) Through Understanding the Influence of Temperature and Rheological Behaviour on the Coalescence Phenomenon," AIP Conf. Proc., **1896**(1), p. 040008.

[50] Bridgman, P., 1922, "Dimensional Analysis," Philos. Mag., **2**(12), pp. 1263–1266.

[51] Szirtes, T., 2007, *Applied Dimensional Analysis and Modeling*, Butterworth-Heinemann, Oxford, UK.

[52] Mokhtarian, H., Coatanéa, E., and Paris, H., 2017, "Function Modeling Combined With Physics-Based Reasoning for Assessing Design Options and Supporting Innovative Ideation," AI Edam, **31**(4), pp. 476–500.