Data-Driven Prediction and Optimization of Liquid Wettability of an iCVD-Produced Fluoropolymer

Daniel Schwartz[1], Tien Nguyen[2], Zhengtao Chen[2], Kenneth K. S. Lau[2], Michael C. Grady[3], Ali Shokoufandeh[1], and Masoud Soroush[2*]

[1]College of Computing and Informatics, Drexel University, Philadelphia, PA 19104, USA

[2]Department of Chemical and Biological Engineering, Drexel University, Philadelphia, PA 19104, USA

[3]Axalta Coating Systems, Philadelphia, PA 19103-7044, USA

February 23, 2022

FINAL VERSION

Submitted for Publication in the *AIChE Journal* Special Issue on AI in Chemical Engineering

Keywords: Machine learning, data-driven, iCVD polymers, liquid repellency, processing-condition-property relationship

*Corresponding author. soroushm@drexel.edu

Abstract

Initiated chemical vapor deposition (iCVD) is a reactive process that creates polymeric materials on a surface from vapor-phase monomers and thermal initiators. Our iCVD synthesis of poly(perfluorodecyl acrylate) (PPFDA) resulted in the growth of micro- and nano-worms normal to the surface. The micro- and nanostructures of the worms directly depend on iCVD process conditions. They in turn influence bulk properties such as their liquid wettability. The current absence of a physiochemical model that can explain the relationships between iCVD process conditions and bulk properties of the polymers motivates the use of data-driven modeling to capture and describe the relationships. In this work, we report iCVD data (contact angles of heptane, octane, and water on PPFDA and process conditions) from 49 batches and use artificial neural networks to model the relationships. The models are then used to determine the optimal iCVD process conditions that maximize the contact angles on PPFDA.

Introduction

Wettability of a liquid with respect to a solid is the tendency of the liquid to spread over or adhere to the surface of the solid in the presence of other immiscible fluids.[1] It can be expressed in terms of the angle of contact at the liquid-solid surface, which is the angle formed by the liquid at the three-phase boundary where the liquid, gas, and solid intersect. As the contact angle decreases, the wettability of the liquid increases. Thus, a zero contact-angle indicates complete wettability, while a contact angle of 180° is indicative of complete nonwetting. It is generally accepted that a solid with contact angles of 60° to 90° repels the liquid. The solid surface roughness affects the contact angle; surfaces with the same chemical composition but different roughness have different contact angles.

Initiated chemical vapor deposition (iCVD) is a reactive process, which creates a polymeric film on a substrate surface, from vapor-phase monomer(s) and initiator(s). It usually produces contiguous polymer thin films. However, in rare cases, it can instead form discrete micro- and nano-structures. For example, our iCVD synthesis of poly(perfluorodecyl acrylate) (PPFDA) resulted in the growth of micro- and nano-worms, which tend to grow perpendicular to the surface.[2,3]. Due to the low surface energy of the fluorinated polymer and the heterogeneous worm-like surface, liquid wetting can be substantially reduced by entering the Cassie-Baxter state. The micro- and nanostructures of the worm-like features directly depend on iCVD operating conditions. They in turn influence bulk properties such as repellency of liquids. With the contact angle being dependent on iCVD operating conditions, it would be advantageous to create a model that captures the dependence.

Using conventional reaction rate equations and parameters, the reaction kinetics can be modeled with enough accuracy[4]. However, the mechanisms that determine the growth of the worm-like structures are currently not fully understood.[3] Given this currently-inadequate chemical and physical understanding, an alternative to first-principles (physical) modeling to relate product (polymer) properties, such as contact angles of liquids on the polymer surface, to iCVD processing conditions is to use an empirical (data-driven) modeling approach such as neural networks. A typical neural network model can contain numerous fitting parameters, many times greater than the number of observations available to fit the model with. This relatively large number of fitting parameters provides a lot of flexibility for shaping the response surface of the target (output) variable(s).

Traditional machine learning techniques require a significant number of training examples to achieve high accuracy in model training. We experiment with different machine learning models to score high accuracy with the limited training data available for modeling iCVDconditions. Research has shown that, in general, training a classifier from scratch on small datasets does not work well, Recently, Barz *et al.* showed that using a cosine loss function provides better performance than traditional objective functions on smaller datasets[26]. Our first experiment is to verify the validity of the hypothesis of using cosine loss on a multi-layer perceptron trained on few data samples. Specifically, we experiment by modeling the iCVD data using a multi-layer perceptron with a cosine loss function and evaluate the performance due to the limited amount of data to train. We show the results of this experiment in the experimental section below. Our second hypothesis is to use convolutional layers to generate higher order features to improve the predictions. We verify this hypothesis by using a Convolutional Neural Network (CNN) with convolutional layers as feature generation layers. We verify this hypothesis and ration that the use of convolutional layers to generate higher-order features improves the quality of classification. While in principle, the CNN developed to find the optimal reacting conditions could be optimized using general optimization and extrapolation methods (i.e., Stochastic Gradient Descent), our final contribution focuses on relaxing the problem to a convex setting. We implemented a convexified convolutional neural network that approximates the best possible CNN and can estimate the optimal conditions in a more efficient and explanatory method using convex optimization.

In this paper. for the first time we: (i) report iCVD data (contact angles of heptane, octane, and water on PPFDA and process conditions) from 49 batches; (ii) capture and describe the relationships using artificial neural networks; and (iii) use the models to determine the optimal iCVD process conditions that maximize the contact angles on PPFDA. The ability to create surfaces with high contact angles and liquid repellency is important for a broad range of applications, such as in oil-water separations, self-cleaning surfaces, and microfluidics.[5]

iCVD Process

iCVD is a technique used to deposit polymer thin films under vacuum. Monomer and initiator vapors flow into the reactor. Inside the chamber, the initiator molecules are heated by a series of heated filament wires, leading to the generation of free radicals. Here also, the monomer molecules are adsorbed onto a temperature-controlled substrate surface. The free radicals initiate the surface polymerization of the adsorbed monomer (**Figure 1**). The advantages of iCVD include its much lower filament temperature (~200–400 °C) compared to other hot wire CVD techniques (>800 °C) as well as the relative chemical purity and physical uniformity of polymer films produced on surfaces. In addition, the substrate can be kept at near room temperature (~25 °C), which allows a wide range of substrates to be coated, like wafers, glass, metals, plastics, and even non-volatile liquids. Utilizing this approach, the entire substrate surface can be covered with polymer very efficiently[6,7]. It is a proven technique for numerous applications, including biomaterials,[8-10] solar cells,[11] sensors,[12] and various thin film coatings. Compared to other film deposition techniques, like spin coating, iCVD does not use liquid solvents during processing. This is particularly attractive as the use of liquid solvents during processing can often lead to solvent residue in the coatings and solvent incompatibilities with existing substrate surfaces. Furthermore, the iCVD process occurs in a single step, enabling simultaneous polymerization and coating deposition, and provides the necessary physical and chemical control needed for targeted applications.[13] In general, the surface polymerization process allows iCVD to deposit thin uniform coatings that conform to the substrate topology.[11,14] Besides planar substrates like silicon wafers, micro- and nano-scale porous materials can also be conformally coated by iCVD.[6,15-18] By controlling the relative rates of reactant diffusion and polymerization, uniform growth can be achieved.

iCVD Deposition of PPFDA Polymer

The monomer, 1H,1H,2H,2H-perfluorodecyl acrylate (99.6% Fluoryx), and the initiator, di-*tert*-butyl peroxide (99% Acros Organics), were used without further purification for the polymerization of PPFDA using iCVD (**Figure 2)**. The polymer was deposited on silicon substrates in a stainless steel custom-built vacuum reactor as described previously.[19] The monomer was heated in a source container to 80 °C, in order to attain sufficient vapor pressure, while the initator was kept at room temperature. For iCVD of PPFDA, vapors of the monomer and the initiator were sent into the reactor at set flow rates using precision needle valves. In some cases, an inert nitrogen gas flow was also added that was metered into the reactor using a mass flow controller (MKS Instruments). The precursor flowrates were set relative to the reactor leak rate; i.e., the inherent flow from the surrounding atmosphere into the vacuum chamber without any active precursor flow. Reactor pressure was maintained at a desired setpoint by using a pressure controller (MKS Instruments) in a feedback loop with a capacitance manometer (MKS Instruments) that adjusted the opening of a downstream throttle valve (MKS Instruments) to a rotary mechanical pump (Edwards). The reactor pressure was set relative to the reactor base pressure, i.e., the inherent pressure inside the vacuum chamber without any reactant flow and with the throttle valve fully open. A Chromaloy (Goodfellow) filament wire array was used to resistively heat the reaction chamber to a desired temperature using a DC power supply (Sorensen) that controlled the electrical voltage and current through the wires. The deposition substrate was placed on a thermoelectric cooling stage (Custom Thermoelectric) positioned below the array of filament wires, and the stage was cooled by backside contact with a thermal fluid flowing through a recirculating chiller (Thermo Scientific). To minimize polymer deposition elsewhere in the reactor, the walls and the reactor's glass lid were heated. Filament, substrate, and reactor wall temperatures were monitored by K-type thermocouples (Omega Engineering). A wide range of iCVD process parameters were studied, controlled, and monitored for the deposition of PPFDA polymer worms, as shown in **Table 1**.

After deposition, the PPFDA polymers were characterized for their composition, morphology, and liquid wettability. In all deposition runs, Fourier transform infrared spectroscopy (FTIR) and X-ray photoelectron spectroscopy (XPS) reveal the chemical composition and stoichiometry expected of PPFDA, results of which have been shown previously.[17] In terms surface morphology, scanning electron microscopy (SEM) show the formation of micro- and nano-scale worms. Depending on the iCVD process conditions (**Table 1**), worm morphology can change in terms of the height, shape, and orientation. **Figure 3** shows representative images of PPFDA worm morphology that can be produced, which range from straight and narrow worms to highly curled and tapered worms. The mechanism of worm formation and growth is not entirely understood, but there is evidence to show that the reaction kinetics and crystallization of PPFDA drives polymer chain and worm assembly.[17] The resulting changes in surface morphology directly impacts liquid wetting on the surface. To assess wettability, static contact angles of water and oil (heptane, octane) droplets were measured for each corresponding deposited sample. **Table 2** lists iCVD reactor operating conditions used to synthesize the PPFDA samples and contact angles of the samples. Contact angles measurements were performed using an automated contact angle goniometer (ramé-hart) with droplet volumes in the range of 3–14 mL. In general, by coupling greater surface roughness and a low surface energy fluorinated polymer like PPFDA, super hydrophobicity can be achieved, where water contact angle exceeds 150°.[20] However, to achieve superoleophobicity where oil contact angle exceeds 150° is a greater challenge. Surface roughness alone is not a sufficient criterion, but instead specific surface re-entrant structures, like overhangs, umbrellas, and mushrooms, are needed to prevent the wetting of low surface tension oils. Given the current lack of a physical model for understanding worm development of iCVD PPFDA, a data-driven model that aims to capture the impact of a multitude of iCVD processing parameters on PPFDA surface wettability would be valuable in helping us attain superoleophobicity.

Machine Learning

Machine Learning Problem

The machine learning problem to find the relationships between the input vector Z and $y_1,\ y_2,$ and $y_3,$ where

$$Z = [x_2\ \ x_4\ \ x_5\ \ x_6\ \ x_9\ \ x_{12}\ \ (x_{13} - x_{14})\ \ x_{15}x_{16}\ \ \ z_9\ \ \ z_{10}\ \ \ z_{11}\ \ \ z_{12}\ \ \ z_{13}\ \ \ z_{14}] \in \mathbb{R}^{14},$$

z_9 and z_{10} are the flowrate of the inert gas during the first period and the duration of the first period, respectively; z_{11} and z_{12} are the flowrate of the inert gas during the second period and the duration of the second period, respectively; z_{13} and z_{14} are the flowrate of the inert gas during the third period and the duration of the third period, respectively. $z_9,\ z_{10}, z_{11}, z_{12},\ z_{13}$, and z_{14} describe the flowrate of the inert gas, x_{19}, during the reaction time of x_{18}.

Machine Learning Methods

The first innovation in mimicking learning of human brains was the Artificial Neural Network (ANN)[21]. An ANN consists of a collection of connected nodes, artificial neurons, that loosely model the neurons in a biological brain. The connections between the nodes follow the likeness of synapses in the biological brain, which propagate signals to other neurons. In the ANN, these signals are represented as Real numbers, which are then transformed at the outputs of each neuron by some non-linear function of the sum of its inputs. The connections between the nodes, or edges, have a weight associated with them to adjust the learning process. These weights are like that of the natural threshold activations as seen in the biological brain. In general, the main objective of training an ANN is to learn the proper weights such that when inputs are passed through the complete collection of neurons and associated weights, the correct output is generated. ANNs can be constructed in many ways; architectures can consist of ensembles of neurons together, a layer, and more complicated models can consist of multiple layers connected to each other with outputs of one layer passed as inputs to the next layer.

One of the simplest ANNs, is the Multilayer-Perceptron (MLP), a feedforward ANN[22]. A MLP is composed of at least three layers of nodes: an input layer, a hidden layer, and an output layer. The input layer is simply the raw data; the hidden layer is as described above, an ensemble of neurons that compute the product of the inputs and the weights, generates a summation, and then applies a non-linear activation function as output. While the bare minimum for a MLP is a single hidden layer, there can be an infinite number of hidden layers stacked on top of one another. Finally, all MLPs are completed with an output layer that combines the last hidden layer and generates a single output of the desired dimension. The most common non-linear activation function applied to summations is the Rectified Linear Unit (ReLU) function, a piecewise linear function that outputs the input if it is positive, and zero otherwise[23].

$$f(x) = \max(0, x)$$

MLPs and most if not all ANNs are trained through the backpropagation algorithm[24] that require an activation function to look and act like a linear function but is non-linear to allow learning of more complex relationships. As a result, the ReLU function is a perfect tool for achieving this, since it is linear for all positive values, and is non-linear in the sense that it outputs all negative values as zero.

The intuition for deploying an MLP with multiple hidden layers derives from the history of using a single-layer neural network to represent linearly separable functions. As a result, simple problems that must classify between two classes can easily be separated by a line; yet, most problems, especially the problem of predicting iCVD reactor conditions are not necessarily linearly separable. Therefore, MLPs with many hidden layers have been used to represent convex regions such that they can learn to represent shapes around examples in some high-dimensional space. This in turn can outperform the limitation of a network that can only solve linearly separable classes. To select the number of nodes for each layer in a MLP, we systematically experiment to find the best performing solution. We gradually decrease the number of neurons within each layer and check the quality of the model prediction until we arrive at the simplest model that has satisfactory prediction accuracy. The number of epochs is chosen in a similar fashion in a method to determine the best accuracy without overfitting, we chose the number of epochs that converged in the smallest number without overfitting in a series of trials. When choosing the hyperparameters for the CNN architecture, we follow the principle that a choice of a small filter corresponds to a good compromise between computational complexity and quality of the resulting convolution. In disciplines such as computer vision, this compromise is most illustrative in noise reduction, while preserving the sharpness for a median filter. Thus, we experiment with three different sizes of the kernel saturating around the median values between 0 and the input size, 14. In general, the notion of filters is to capture patterns; and the best effort to capture patterns for a small input size such as 14 is to try to extract them around 4-8. In view of these, we propose an MLP architecture with 4 hidden layers: 48, 24, 12, and 6 nodes respectively, as shown in Figure 4. The model takes as input the 14 features as described before and outputs the predicted values for the three properties: water, heptane, and octane contact angles of the resulting polymer.

ANNs are known to overfit quickly on a training dataset with few samples; thus, the model fails to generalize on inputs that the model has never seen before. As a result, to improve the model's ability to generalize, Dropout layers were included following each of the hidden layers[25]. A common solution employed is the notion of randomly dropping nodes from the network as a form of regularization that approximates training many ANNs with different architectures in parallel. Moreover, during training, a portion of layer outputs are randomly suppressed, dropped out, to effect making the layer seem as though it had fewer input nodes passed in. Each iteration of training, the learning algorithm in effect has a different perspective of the model in hopes to adapt to mistakes from previous layers, making the model more robust. In our experimentation, we modified the proportion of nodes to be dropped to explore the effect Dropout layers had on a MLP for this small dataset.

Considering the limited number of samples in Table 2, we experimented with an alternative learning algorithm; that is, Cosine Loss in an MLP. Previous work has shown that using a cosine loss function as an objective function in an ANN provides substantially better performance trained on fewer data samples than traditional objective functions like cross-entropy[26]. The intuition of this modification is that cosine loss can better integrate prior knowledge using class hierarchies and improve performance in classification tasks. We used the existing architecture shown in Figure

5 and modified its objective function to employ cosine loss instead of Mean Squared Error (MSE) as used in the rest of the experiments.

Within the Deep Learning domain, Convolutional Neural Networks (CNNs) have been the most effective architectures in computer vision tasks[27]. CNNs extract abstract features from images by convolving the image with filters. These filters are then subsampled via pooling layers, commonly using Max-pooling that partitions the image in regions and returns the maximum value within each region. Finally, the last element of a CNN is the fully connected layer where each node is directly connected to every node in the previous and next layer to pass forward the results from the previous layer. These CNNs can successfully capture the spatial and temporal dependencies within an image through the application of relevant filters. This experiment helps verify the hypothesis that the use of a CNN could generate higher order features to improve predictions. One of the most notable achievements in CNNs was displayed in the AlexNet architecture[28] that used a faster convolution operation implementation on Graphical Processing Units (GPUs) that achieved state-of-the-art results when it was introduced. The convolutional layers of CNN have parameters to specify the number of filters, strides, and size of the kernel. The number of filters defines the dimensionality of the output space or the number of output filters in the convolution. The strides define the stride length of the convolution, and the kernel size defines the length of the 1D convolutional window from which the data is sampled. We propose a 1D-CNN architecture represented in Figure 5, that takes the 14 input features, passes them through a 1D convolutional layer with 8 filters, a stride of 1, and a kernel size of 3. The output of the convolutional layer is flattened and passed to 3 output nodes.

Experimental Setup

Our experimental setup is run on a Linux machine with Intel(R) Xeon(R) E-2276M CPU @ 2.80GHz 2.81 GHz, 6 cores, and 32GB RAM. The code was run and interpreted by Python 3.8.3 using NumPy 1.19.2 and TensorFlow 2.2.0 packages using Anaconda 4.10.3 as a package manager.

Results and Discussion

Table 3 presents results of five experiments comparing the use of Dropout and Cosine Loss modifications in the MLP as described before as well as two experiments on CNN implementations. In these experiments, there are 14 input features because there are the 6 original input features and the flowrate conditions are spread across six new features, three describing the time in seconds, and three describing the flowrate of the inert gas, N_2. The first row highlights the MLP with Dropout layers of a dropout proportion of 20% (0.2) following each hidden layer. The second row is similar but with a dropout proportion of 40% (0.4), and the third row describes the MLP architecture but trained using Cosine Loss instead of traditional MSE. The fourth row illustrates the results of a CNN with 4 filters and the final row illustrates the results of a CNN with 8 filters as described in Figure 5. While the MLP trained with Cosine Loss performs the best for the training data set, this model fails to generalize well as advertised in its proposal, since this model had some of the worst performance in the test data set. As a result, we have concluded that the hypothesis of using Cosine Loss on our dataset consisting of few samples trained on, does not improve the performance of the model as much as other modeling approaches. Instead, the model with Cosine Loss overfits on the training data and performs poorly on the test data. This follows the intuition that Cosine Loss would try to overly concentrate on the few samples and when provided with new samples, it has trouble predicting correctly. However, the CNN with 8 filters qualitatively performed the best since it has low train error and two of the lowest test errors for heptane and octane contact angles, with the MLP model with 0.2 Dropout trailing closely behind. Clearly, the use of convolutional layers to generate higher-order features does improve the quality of the model and outperforms using Cosine Loss.

We also performed experiments without the separating the N_2 flowrate into the piecewise representation; instead, we just recorded the total time in seconds as a whole and ignored the N_2 flowrate. These results are recorded in Table 4 and follow the same structure as in Table 3, with the only difference being that there are 9 input features as opposed to 14 when the N_2 flowrate conditions are represented as a piecewise function. Across all experiments, the MLP with a dropout of 0.2 performed the best. Moreover, it had the lowest error in all sections except for predicting water contact angle in the test set in the experiments without recording the flowrate conditions. However, the CNN with 4 filters outperformed this model with the flowrate conditions; yet, across all experiments, this model, CNN with 4 filters, generalized the best and had the lowest errors for almost all test datasets.

Optimization: Operating Conditions that Maximize Contact Angles

We propose to optimize the function learned in the previous section to find the optimal process conditions that maximize the contact angles; that is, to find the maximizer of the function using constrained linear programming. As noted in the previous section, the CNN had the best success in learning the optimal iCVD conditions; however, to maximize the contact angles from the CNN is quite difficult due to the sheer number of parameters and structure of the network. We propose three different approaches to solving the optimization problem. We first attempt to optimize the function by defining a surrogate function that relaxes the original problem. Instead of maximizing the function directly, we relax the CNN model to a convexified setting using the method proposed by Zhang *et al.* in Convexified Convolutional Neural Networks[29]. The proposed algorithm is to relax a two-layer two-dimensional CNN to a convexified setting and use the Convexified Convolutional Neural Network (CCNN) to estimate kernel functions for each contact-angle function. This relaxation of the original Convolutional Neural Network to a convexified setting, a CCNN, results in a simplified and easy method to investigate the model and explore the operating conditions that maximize the contact angles. We alter the algorithm implemented in [30] slightly by using an even simpler model, a one-dimensional two-layer CNN instead of a two-dimensional one.

Optimization Using Convexification as a Surrogate Function

We first begin by extracting a collection of P vectors $\{z_p(x)\}_{p=1}^{P}$ of the input vector x. We define $P = 12$ since there are 14 distinct features and our CNN takes patches of size 3 with a stride of 1, resulting in 12 total patches. As a result, these patches consist of overlapping elements of x and each vector $z_p(x) \in \mathbb{R}^{d_1}$ where $d_1 = 3$. We then followed the same steps to approximate feature matrix Q, through a Cholesky decomposition of kernel K, the Radial Basis Function (RBF) kernel. Following, we trained the CCNN, represented by the feature matrix, Q, to learn the parameter matrix, A, through Projected Stochastic Gradient Descent[31] that solves the constrained optimization problem. The estimated parameter matrix, $\hat{A}$**,** along with the matrix consisting of patches, $Z(x)$ for input $x \in X$, define the general form function of the CCNN. The function is computed by taking the concatenation of products of the trace of each input and the parameter matrix learned such that $\hat{f}_{ccnn}(x) \coloneqq \left(\left(\mathrm{tr}\left(Z(x)\right)\hat{A}_1\right), \ldots, \left(\mathrm{tr}\left(Z(x)\right)\hat{A}_{d_2}\right)\right)$, where $d_2 = 3$. With that, we maximize the function, $\hat{f}_{ccnn}(x)$, across all inputs $x \in X$ and use a multi-objective optimizer to find the Pareto frontier[32].

To solve the problem of finding the optimal inputs that maximize our convexified function we used coordinate ascent, an iterative algorithm beginning at the local minimum that ascends by walking along the gradients of the function to arrive at the optima. Since the function we learned in the previous section has inputs on the magnitude of fourteen dimensions, the optimization procedure we used, works by maximizing the function along one direction at a time, in essence solving a simpler single-variable optimization problem at a time [33].

Optimization of Convexified Surrogate Function via Extrapolation Using Metric Sets

An alternative method of optimizing the convexified surrogate function in the previous section builds upon the work by Kryanev et al[34]. Since the convex surrogate function is computed by taking the concatenation of products of the trace of each input and the parameter matrix learned such that $\hat{f}_{ccnn}(x) \coloneqq \left(\left(\operatorname{tr}\left(Z(x)\right)\hat{A}_1\right), \ldots, \left(\operatorname{tr}\left(Z(x)\right)\hat{A}_{d_2}\right)\right)$, where $d_2 = 3$. With that, we attempt to optimize the function, $\hat{f}_{ccnn}(x)$, which is the summation of the trace of multiple matrices representing a multi-variable function. We then extrapolate the function by the means of metric analysis. We build upon the work by Kryanev et al. [34] and extrapolate the surrogate function that is a function of several variables. We use an interpolation scheme of metric analysis to solve the extrapolation of the function in two steps: use metric analysis to interpolate the points of the domain; and then apply an auto regression model as well as the use of metric analysis to predict the function values along the domain.

Optimization of Convolutional Neural Network via Coordinate Ascent

Like the first method of optimization, we start by extracting a collection of P vectors $\left\{z_p(x)\right\}_{p=1}^{P}$ of the input vector x and define $P = 12$ since there are 14 distinct features and our CNN takes patches of size 3 with a stride of 1, resulting in 12 total patches. As a result, these patches consist of overlapping elements of x, where each vector $z_p(x) \in \mathbb{R}^{d_1}$ and $d_1 = 3$. We then leverage the CNN directly to solve the problem of finding the optimal inputs that maximize the output of the CNN function. To find the optimal inputs that maximize the output of the CNN function, we use coordinate ascent, an iterative algorithm beginning at the local minimum that ascends by walking along the gradients of the function to arrive at the optima. As discussed in the previous section, this optimization procedure works by maximizing the function along one direction at a time due to the multi-variate nature of the objective function[33]. This follows a similar process as in the section on optimization using convexification except for the function that determines the direction of the coordinate ascent is the original CNN and not the convexified surrogate function.

Experimental Setup

As in the previous section, the same hardware is used, a Linux machine with Intel(R) Xeon(R) E-2276M CPU @ 2.80GHz 2.81 GHz, 6 cores, and 32GB RAM. The optimization code was also run and interpreted by Python 3.8.3 using NumPy 1.19.2 and TensorFlow 2.2.0 packages using Anaconda 4.10.3 as a package manager. To note, we adapted the code implemented in [30] to fit the constraints of convexifying a 1D CNN as described in the previous section.

Results and Discussion

Table 5 presents the optimal input values calculated using six distinct optimization methods:

(A) Coordinate ascent algorithm on the function defined by our CCNN model;

(B) Data sample with maximum values;

(C) Coordinate ascent algorithm for our CNN model with 8 filters;

(D) Coordinate ascent algorithm for our CNN model with 6 filters;

(E) Coordinate ascent algorithm for our CNN model with 4 filters; and

(F) Extrapolation of function of many variables by means of metric analysis.

Our investigation indicated that the same optimal input values maximize the contact angles of water, heptane, and octane. We then used the optimal input values listed in Table 5 for our six approaches, calculated the contact angles corresponding to the optimal input values (Table 6). We operated the iCVD process at the theoretically optimal process conditions reported in Table 5 and measured the contact angles of the produced polymer films. The measured contact angles are reported in Table 7. As shown in Figure 6, the contact angles predicted by the convex surrogate model with optimal input values derived by the coordinate ascent algorithm perform very well and are similar to the measured contact angles of samples produced by the iCVD process. Similarly, in Figure 7, the absolute error in predicting the contact angle for the convex surrogate model does very well in predicting water and heptane, but not as well in the case of octane. As Figures 6 and 7 illustrate, in 10 out of the 18 cases the absolute difference (error) between the theoretically predicted (by an ML model with the optimal operation conditions calculated using the same ML model as the input) and measured contact angles of samples (produced by the iCVD process operated at the theoretically found optimal operation conditions found by the same ML model) is less than 20°. In particular, the theoretically predicted optimum contact angles of water and heptane and measured ones for convex surrogate grid search optimization (A) and data sample of maximum values (B), as well as the theoretically predicted optimum contact angles of water, heptane and octane and measured ones for data sample with maximum output (F) are in good agreement with the measured ones. On the other hand, the theoretically predicted optimum contact angles and the measured ones for the remaining cases are significantly different. These large differences indicate that CNN (8 Filters) grid search optimization (C), CNN (6 Filters) grid search optimization (D), and CNN (4 Filters) grid search optimization (E) are unable to capture the complexity of the relationships between the process conditions and the contact angles.

Concluding Remarks

In this work, for the first time we: (i) reported iCVD data (contact angles of heptane, octane, and water on PPFDA and process conditions) from 49 batches; (ii) captured and described the relationships using artificial neural networks; (iii) used the models to determine the optimal iCVD process conditions that maximize the contact angles on PPFDA; and (iv) experimentally validated the optimal process conditions. We applied and tested several machine learning models to capture the relationships. A challenge is this machine learning problem was the low ratio of the number of input-output data points to the number of input features. We found that a CNN model with 4 filters yields the best prediction performance in terms of MSE.

We maximized the contact angles using constrained linear programming. Instead of maximizing the function directly, we relaxed the two-layer two-dimensional CNN to a convexified setting using the method proposed by Zhang *et al.* [29] and estimated kernel functions for each contact-angle function.

Acknowledgement

D.S., A.S., and M.S. acknowledge financial support from the U.S. National Science Foundation (NSF); this material is partially based upon work supported by the U.S. NSF under Grant No. CBET–1953176. Any opinions, findings, and conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Supplementary Information

Section SI1. Convexified Convolutional Neural Network

This section includes a Jupyter Notebook that contains the methods for the convexification of a CCNN applied to the input features of the iCVD reaction. The convexified model is then optimized via projected gradient descent to find the optimal parameters that fit the matrices such that when applied to the input features, it predicts the ground truth contact angles of water, heptane, and octane.

Section SI2. Convexified Convolutional Neural Network Optimization

This section includes a Jupyter Notebook that builds upon the optimized convexification of the CCNN applied to the input features of the iCVD reaction and attempts to optimize the function to find the reactor conditions that maximize the respective contact angles for water, heptane, and octane via coordinate descent and extrapolation using metric sets.

Section SI3. Convolutional Neural Network

This section includes a Python script that contains the methods for learning the weights of a CNN applied to the input features of the iCVD reaction. The CNN model is then optimized via Adam, an algorithm for first-order gradient-based optimization of stochastic objective functions to find the optimal weights that fit the input features to the ground truth contact angles of water, heptane, and octane.

Section SI4. Convolutional Neural Network Optimization

This section includes a Jupyter Notebook that builds upon the optimized CNN applied to the input features of the iCVD reaction and attempts to optimize the function to find the reactor conditions that maximize the respective contact angles for water, heptane, and octane via coordinate descent.

Section SI5. Data

This section presents an Excel sheet that includes the data of the iCVD process conditions used to synthesize the PPFDA samples and contact angles of the samples.

Section SI6. Multi-Layer Perceptron

This section includes a Python script that contains the methods for learning the weights of a MLP applied to the input features of the iCVD reaction. The MLP model is then optimized via Adam, an algorithm for first-order gradient-based optimization of stochastic objective functions to find the optimal weights that fit the input features to the ground truth contact angles of water, heptane, and octane.

References

1. Civan F. instrumental and laboratory techniques for characterization OF reservoir rock. *Reservoir Formation Damage*: Gulf Professional Publishing Burlington; 2007:154-173.

2. Lau KKSC, Z.; Nguyen, T. . Directed Vapor Deposition and Assembly of Polymer Micro- and Nanostructures. U.S. Patent Application 62941086. November 27, 2019 2019.

3. Chen ZN, T.; Lau, K.K.S. . One-Step Bottom-Up Growth of Highly Liquid Repellent Worm-Like Surfaces on Planar Substrates, ACS Nano (submitted). 2021.

4. Lau KK. Growth Mechanism, Kinetics, and Molecular Weight. *CVD Polymers: Fabrication of Organic Surfaces and Devices.* 2015:13-44.

5. Kota AK, Kwon G, Tuteja A. The design and applications of superomniphobic surfaces. *NPG Asia Materials.* 2014;6(7):e109-e109.

6. Nejati S, Lau KK. Pore filling of nanostructured electrodes in dye sensitized solar cells by initiated chemical vapor deposition. *Nano letters.* 2010;11(2):419-423.

7. Nejati S, Lau KK. Integration of polymer electrolytes in dye sensitized solar cells by initiated chemical vapor deposition. *Thin Solid Films.* 2011;519(14):4551-4554.

8. Bose RK, Lau KKS. Initiated chemical vapor deposition of poly(2-hydroxyethyl methacrylate) hydrogels. *Thin Solid Films.* 5/2/ 2011;519(14):4415-4417.

9. Bose RK, Lau KKS. Mechanical Properties of Ultrahigh Molecular Weight PHEMA Hydrogels Synthesized Using Initiated Chemical Vapor Deposition. *Biomacromolecules.* 2010/08/09 2010;11(8):2116-2122.

10. Bose RK, Lau KKS. Initiated CVD of Poly(2-Hydroxyethyl Methacrylate) Hydrogels: Synthesis, Characterization and In-vitro Biocompatibility. *Chemical Vapor Deposition.* 2009;15(4-6):150-155.

11. Nejati S, Lau KKS. Integration of polymer electrolytes in dye sensitized solar cells by initiated chemical vapor deposition. *Thin Solid Films.* 5/2/ 2011;519(14):4551-4554.

12. Tenhaeff WE, McIntosh LD, Gleason KK. Synthesis of Poly(4-vinylpyridine) Thin Films by Initiated Chemical Vapor Deposition (iCVD) for Selective Nanotrench-Based Sensing of Nitroaromatics. *Advanced Functional Materials.* 2010;20(7):1144-1151.

13. Nejati S, Lau KKS. Pore Filling of Nanostructured Electrodes in Dye Sensitized Solar Cells by Initiated Chemical Vapor Deposition. *Nano Letters.* 2011/02/09 2011;11(2):419-423.

14. Nejati S, Lau KKS. Pore Filling of Nanostructured Electrodes in Dye Sensitized Solar Cells by Initiated Chemical Vapor Deposition. *Nano Letters.* 2011/02/09 2010;11(2):419-423.

15. Laird ED, Bose RK, Wang W, Lau KK, Li CY. Carbon Nanotube-Directed Polytetrafluoroethylene Crystal Growth via Initiated Chemical Vapor Deposition. *Macromolecular rapid communications.* 2013;34(3):251-256.

16. Laird ED, Bose RK, Qi H, Lau KK, Li CY. Electric field-induced, reversible lotus-to-rose transition in nanohybrid shish kebab paper with hierarchical roughness. *ACS applied materials & interfaces.* 2013;5(22):12089-12098.

17. Chen Z, Lau KK. Suppressing crystallinity by nanoconfining polymers using initiated chemical vapor deposition. *Macromolecules.* 2019;52(14):5183-5191.

18. Hsieh CY, Lau KK. Growth of polyglycidol in porous TiO2 nanoparticle networks via initiated chemical vapor deposition: probing polymer confinement under high nanoparticle loading. *Advanced Materials Interfaces.* 2015;2(17):1500341.

19. Bose RK, Lau KK. Initiated CVD of Poly (2-Hydroxyethyl Methacrylate) Hydrogels: Synthesis, characterization and in-vitro biocompatibility. *Chemical Vapor Deposition.* 2009;15(4-6):150-155.

20. Quéré D. Non-sticking drops. *Reports on Progress in Physics.* 2005;68(11):2495.

21. Mitchell TM. Machine learning. 1997.

22. Goodfellow I, Bengio Y, Courville A. *Deep learning*: MIT press; 2016.

23. Hara K, Saito D, Shouno H. Analysis of function of rectified linear unit used in deep learning. Paper presented at: 2015 international joint conference on neural networks (IJCNN)2015.

24. Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. *nature.* 1986;323(6088):533-536.

25. Srivastava N. Improving neural networks with dropout. *University of Toronto.* 2013;182(566):7.

26. Barz B, Denzler J. Deep learning on small datasets without pre-training using cosine loss. Paper presented at: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision2020.

27. Albawi S, Mohammed TA, Al-Zawi S. Understanding of a convolutional neural network. Paper presented at: 2017 International Conference on Engineering and Technology (ICET)2017.

28. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems.* 2012;25:1097-1105.

29. Zhang Y, Liang P, Wainwright MJ. Convexified convolutional neural networks. Paper presented at: International Conference on Machine Learning2017.

30. van Schaik MJ. Applications of the convexified convolutional neural network. *Master's Thesis.* 2018.

31. Cohen K, Nedić A, Srikant R. On projected stochastic gradient descent algorithm with weighted averaging for least squares regression. *IEEE Transactions on Automatic Control.* 2017;62(11):5974-5981.

32. Deb K. Multi-objective optimization. *Search methodologies*: Springer; 2014:403-449.

33. Bezdek J, Hathaway R, Howard R, Wilson C, Windham M. Local convergence analysis of a grouped variable version of coordinate descent. *Journal of Optimization Theory and Applications*. 1987;54(3):471-477.

List of Figures

Figure 7. The absolute difference (error) between the theoretically predicted (by an ML model with the optimal operation conditions calculated using the same ML model as the input) and measured contact angles of samples (produced by the iCVD process with the theoretically found optimal operation conditions found by the same ML model). W: water; H: heptane; and O: octane.

List of Tables

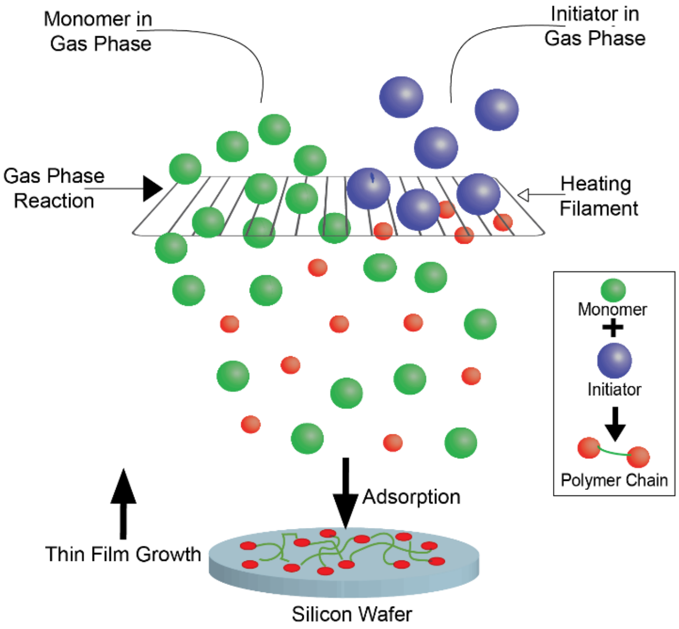

Figure 1. Schematic of the iCVD process. Monomer and initiator are introduced as vapors into the reaction chamber heated by filaments. The supplied thermal energy converts the thermal initiator to free radicals. Monomer and the free-radicals are absorbed onto a temperature-controlled substrate where surface polymerization between the free radicals and monomer occurs.

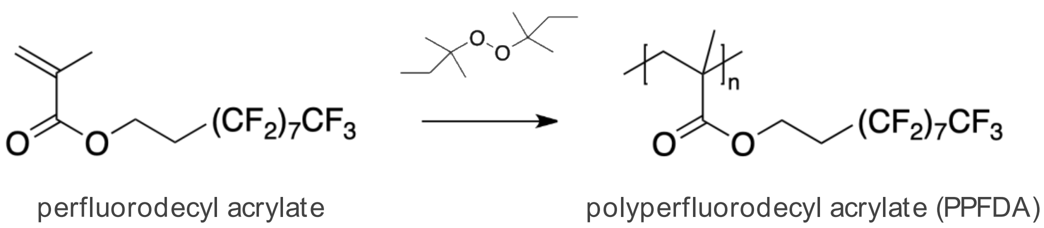

Figure 2. Polymerization chemistry of poly(perfluorodecyl acrylate) (PPFDA).

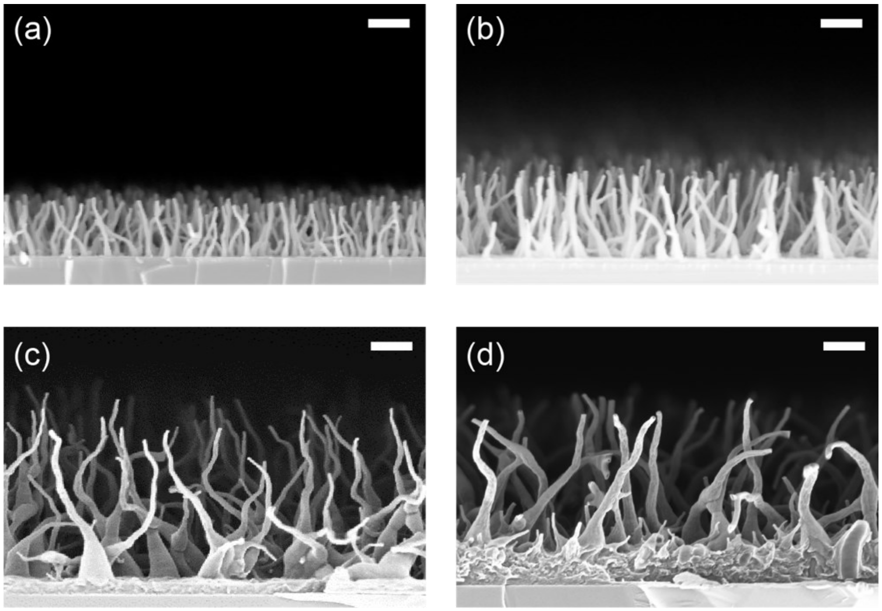

Figure 3. Cross-sectional SEM images showing PPFDA worms that are (a,b) narrower and straighter, and (c,d) more tapered and curled, depending on the specific iCVD growth conditions.

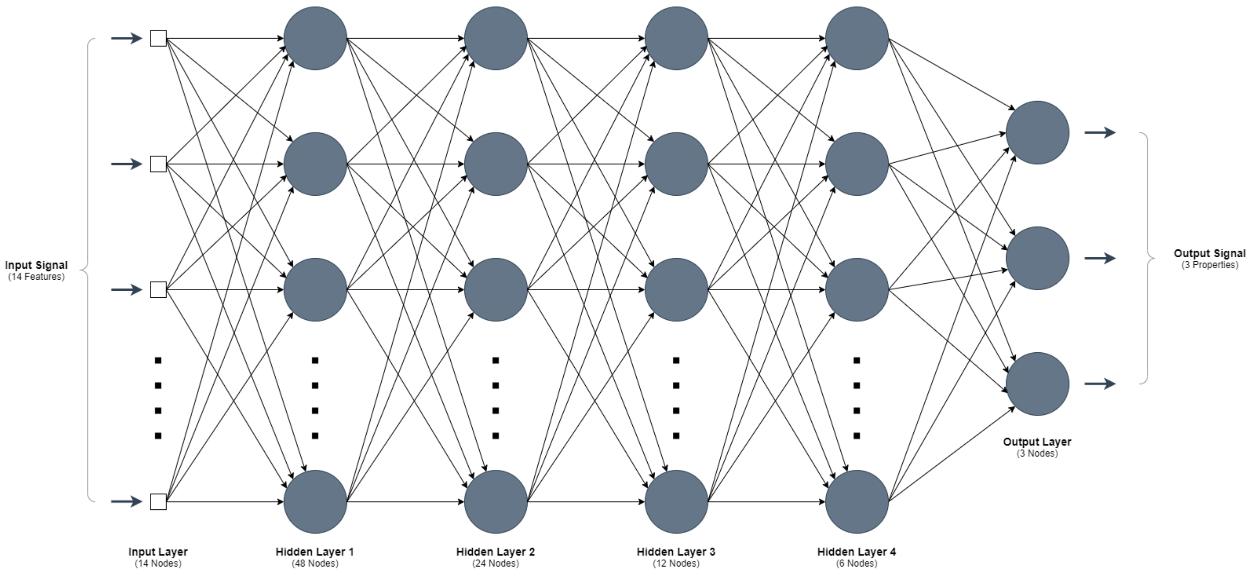

Figure 4. Overview of an MLP architecture.

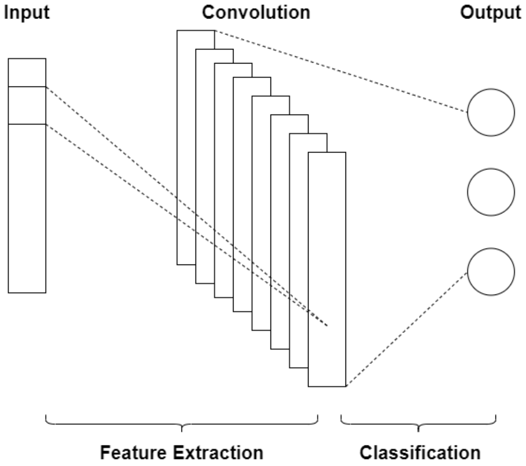

Figure 5. Overview of a CNN architecture.

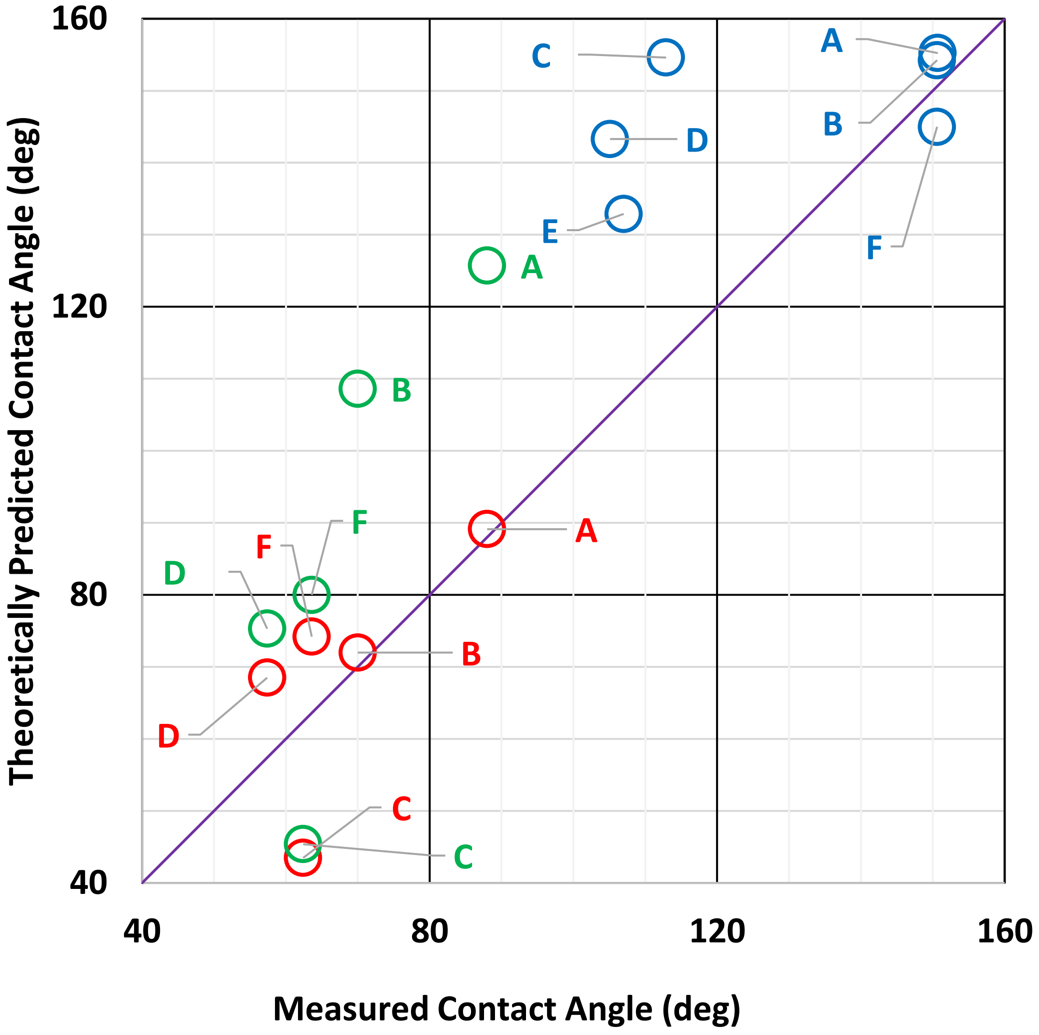

Figure 6. Theoretically predicted (by an ML model with the optimal operation conditions calculated using the same ML model as the input) vs. measured contact angles of samples (produced by the iCVD process with the theoretically found optimal operation conditions found by the same ML model). A: Convex Surrogate Grid Search Optimization; B: Data Sample of Maximum Values; C: CNN (8 Filters) Grid Search Optimization; D: CNN (6 Filters) Grid Search Optimization; E: CNN (4 Filters) Grid Search Optimization; and F: Data Sample with Maximum Output. Blue circles: water contact angles. Red circles: heptane contact angles. Green circles: octane contact angles.

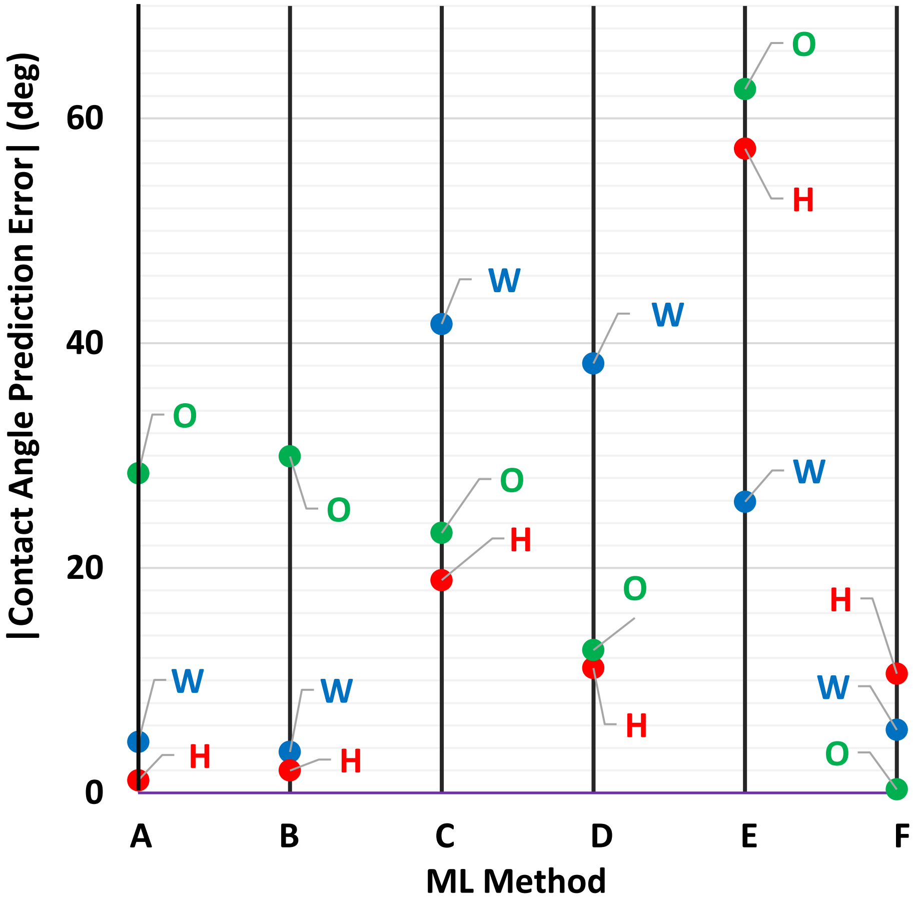

Figure 7. The absolute difference (error) between the theoretically predicted (by an ML model with the optimal operation conditions calculated using the same ML model as the input) and measured contact angles of samples (produced by the iCVD process with the theoretically found optimal operation conditions found by the same ML model). W: water; H: heptane; and O: octane.

Table 1. iCVD process conditions and measurements.

x_1 = filament resistance (Ω)	x_7 = initiator feed temperature (°C)	x_{13} = total pressure (torr)	x_{19} = N_2 flow rate (sccm)
x_2 = substrate temperature (°C)	x_8 = initiator feed flow rate setpoint	x_{14} = base pressure (torr)	y_1 = water contact angle (°)
x_3 = chiller temperature (°C)	x_9 = initiator feed flow rate (sccm)	x_{15} = filament current (A)	y_2 = heptane contact angle (°)
x_4 = wall temperature (°C)	x_{10} = monomer feed temperature (°C)	x_{16} = filament voltage (V)	y_3 = octane contact angle (°)
x_5 = Glass heater power (%)	x_{11} = monomer feed flow rate setpoint	x_{17} = filament temperature (°C)	
x_6 = leak rate (sccm)	x_{12} = monomer feed flow rate (sccm)	x_{18} = Reaction time (min)	

Table 2. iCVD process conditions used to synthesize the PPFDA samples and contact angles of the samples.

ID	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}	x_{19}	y_1	y_2	y_3
1	17.9	42	50	51.6	17	0.026	20	8.5	0.204	80	118	0.120	0.20	0.0100	1.1	19.1	211.2	45.00	0.00 for 45 min	144.7	70.8	77.6
2	17.6	44	50	51.6	17	0.026	20	8.8	0.204	80	180	0.118	0.20	0.0100	1.1	19.2	247.0	45.00	0.00 for 45 min	145	70.2	78.5
3	17.4	48	50	51.6	17	0.026	20	9.0	0.205	80	180	0.119	0.20	0.0099	1.1	18.9	253.7	45.00	0.00 for 45 min	126.5	69.0	72.4
4	17.3	46	50	51.6	17	0.026	20	10.9	0.201	80	130	0.121	0.20	0.0102	1.0	19.0	275.6	45.00	0.00 for 45 min	159.0	100.0	108.0
5	17.4	46	50	51.6	17	0.026	20	10.6	0.200	80	120	0.117	0.15	0.0099	1.0	18.9	285.7	25.00	0.00 for 25 min	147.0	86.0	100.0
6	17.4	46	50	51.6	17	0.026	20	10.5	0.198	80	220	0.116	0.15	0.0103	1.0	18.9	292.4	5.75	0.00 for 5.75 min	145.0	82.0	87.0
7	17.4	46	50	51.6	17	0.026	20	10.5	0.197	80	120	0.034	0.15	0.0107	1.0	18.9	-	23.50	0.00 for 23.50 min	149.0	72.0	86.0
8	17.4	46	50	51.6	17	0.026	20	10.6	0.198	80	128	0.119	0.10	0.0110	1.0	19.0	284.5	25.00	0.00 for 25 min	133.0	91.0	92.0
9	17.4	46	50	51.6	17	0.026	20	10.6	0.196	80	130	0.119	0.10	0.0103	1.0	18.9	270.9	25.00	0.00 for 25 min	137.0	89.0	93.0
10	17.5	46	50	51.6	17	0.026	20	10.6	0.203	80	157	0.117	0.10	0.0125	1.0	18.9	255.6	45.00	0.35 for 45 min	168.0	111.5	124.0
11	17.4	46	50	51.6	17	0.026	20	10.6	0.194	80	176	0.122	0.10	0.0116	1.0	18.9	272.2	45.00	0.35 for 45 min	169.2	104.5	117.7
12	17.6	46	50	51.6	17	0.026	20	10.6	0.198	80	140	0.118	0.10	0.0121	1.0	18.9	-	10.00	0.35 for 10 min	156.0	70.0	76.0
13	17.6	46	50	51.6	17	0.026	20	10.7	0.195	80	150	0.119	0.10	0.0120	1.0	18.9	273.0	20.00	0.35 for 20 min	160.0	91.2	98.6
14	17.6	46	50	51.6	17	0.026	20	10.7	0.198	80	150	0.119	0.10	0.0119	1.0	18.9	289.4	30.00	0.35 for 30 min	N/A	102.0	113.0
15	17.4	46	50	51.6	17	0.026	20	9.2	0.197	80	140	0.117	0.10	0.0119	1.0	18.9	-	5.00	0.35 for 5 min	123.0	64.0	69.0
16	16.9	46	50	51.6	17	0.026	20	10.1	0.202	80	150	0.117	0.10	0.0120	1.0	18.8	-	7.00	0.35 for 7 min	128.8	68.0	69.0
17	17.2	48	50	51.6	17	0.026	20	10.7	0.200	80	153	0.125	0.10	0.0118	1.0	18.9	-	30.00	0.35 for 30 min	175.0	97.0	107.0
18	17.2	44	50	51.6	17	0.026	20	10.7	0.200	80	153	0.125	0.10	0.0118	1.0	18.9		30.00	0.35 for 30 min	163.0	85.0	96.0
19	17.3	36	50	51.6	17	0.026	20	8.7	0.195	80	155	0.123	0.10	0.0121	1.1	18.8		30.00	0.35 for 30 min	162.0	88.0	104.9
20	17.3	33	43	51.6	17	0.026	20	10.7	0.196	80	165	0.120	0.10	0.0119	1.0	18.9	302.0	14.42	0.35 for 14.42 min	173.0	72.0	89.5
21	17.5	42	50	51.6	17	0.026	20	9.2	0.205	80	165	0.120	0.10	0.0128	1.0	18.9	255.8	45.00	0.00 for 45 min	145.0	68.0	76.3
22	17.3	46	50	51.6	17	0.026	20	8.5	0.196	80	145	0.124	0.10	0.0117	1.1	18.7	-	45.00	0.50 for 45 min	133.0	68.0	71.5
23	17.2	46	50	51.6	17	0.026	20	10.1	0.199	80	175	0.122	0.10	0.0100	1.1	18.8	-	45.00	0.10 for 45 min	155.0	108.0	121.5
24	17.4	46	50	51.6	17	0.026	20	10.5	0.200	80	175	0.121	0.10	0.0112	1.1	18.8	-	55.00	0.35 for 25 min; 0.10 for 20 min	160.0	109.0	116.0
25	17.4	46	50	51.6	17	0.026	20	11.0	0.203	80	160	0.125	0.10	0.0103	1.0	18.9	261.0	3.00	0.35 for 3 min	124.5	66.2	71.0
26	17.6	46	50	51.6	17	0.026	20	11.0	0.196	80	160	0.118	0.10	0.0108	1.0	18.8	253.0	48.00	0.35 for 48 min	162.5	111.4	125.0
27	17.4	46	50	51.6	17	0.026	20	11.3	0.199	80	160	0.122	0.10	0.0117	1.0	18.9	294.0	50.00	0.40 for 45 min; 0 for 7 min	163.0	103.0	109.5
28	17.7	46	50	51.6	17	0.026	20	11.3	0.205	80	175	0.120	0.10	0.0128	1.0	18.8	251.6	55.00	0.40 for 45 min; 0 for 7 min	156.6	104.0	114.3
29	17.6	46	50	51.6	17	0.026	20	11.3	0.198	80	175	0.115	0.10	0.0130	1.0	18.8	250.6	65.00	0.40 for 45 min;	145.0	74.2	80.0

																			0 for 7 min			
30	17.4	46	50	51.6	17	0.026	20	11.3	0.201	80	160	0.124	0.1-0.2	0.0122	1.0	18.9	227.0	55.00	0.40 for 45 min; 0 for 7 min	156.0	106.3	111.7
31	17.3	46	50	51.6	17	0.026	20	10.5	0.198	80	90	0.120	0.10	0.0131	1.0	18.9	-	45.00	0.35 for 45 min	167.0	105.0	111.5
32	17.2	46	50	51.6	17	0.026	20	10.3	0.196	80	95	0.124	0.10	0.0015	1.0	18.8	258.0	45.00	0.4 for 45 min	165.0	106.0	115.0
33	17.2	46	50	51.6	17	0.026	20	10.7	0.200	80	95	0.121	0.10	0.0019	1.1	18.9	193.7	52.00	0.40 for 45 min; 0 for 7 min	165.0	96.0	108.4
34	17.3	46	50	51.6	17	0.026	20	10.2	0.200	80	95	0.119	0.10	0.0019	1.0	18.8	284.7	63.00	0.35 for 45 min; 0 for 7 min; 0.35 for 10 min	165.0	105.0	110.0
35	17.2	46	50	51.6	17	0.026	20	11.3	0.196	80	80	0.125	0.09	1.8000	1.0	18.8	283.4	45.00	0.35 for 45 min	166.0	100.0	107.0
36	17.1	46	50	51.6	17	0.026	20	11.0	0.205	80	75	0.119	0.09	3.0000	1.0	18.9	219.7	45.00	0.50 for 45 min	159.6	109.0	121.0
37	30.4	46	50	51.6	17	0.026	20	8.1	0.196	80	70	0.124	0.09	4.3000	0.9	28.5	258.8	45.00	0.50 for 45 min	151.0	107.0	113.0
38	17.2	46	50	51.6	17	0.026	20	10.2	0.200	80	95	0.115	0.08	-0.7000	1.0	18.8	320.8	45.00	0.50 for 45 min	162.0	97.3	108.0
39	17.3	46	50	51.6	17	0.026	20	10.2	0.196	80	100	0.124	0.15	-1 .0000	1.0	18.8	269.4	15.00	0.00 for 15 min	167.0	88.0	99.0
40	22.2	46	50	51.6	17	0.026	20	10.5	0.200	80	110	0.118	0.09	-0.1000			257.4	45.00	0.35 for 45 min	148.0	112.0	115.0
41	17.2	46	50	51.6	17	0.026	20	10.5	0.205	80	85	0.120	0.20	-0.5000	1.0	18.7	262.0	15.00	0.00 for 15 min	143.8	75.0	85.0
42	17.2	46	50	51.6	17	0.026	20	10.7	0.197	80	110	0.122	0.09	-1.7000	1.0	18.7	308.4	45.00	0.50 for 45 min	157.9	95.0	101.0
43	17.4	46	50	51.6	17	0.026	20	9.1	0.118	80	90	0.124	0.08	-0.2000	1.0	18.8	233.3	45.00	0.50 for 45 min	171.0	75.0	86.8
44	17.2	46	50	51.6	17	0.026	20	9.5	0.118	80	100	0.118	0.08	0.8000	1.0	18.9	182.0	45.00	0.35 for 45 min	166.0	77.0	88.0
45	17.8	46	50	51.6	17	0.026	20	10.8	0.118	80	100	0.118	0.08	-0.7000	1.0	18.8	280.2	55.00	0.35 for 45 min; 0 for 10 min	156.5	95.5	103.7
46	17.0	46	50	51.6	17	0.026	20	10.9	0.197	80	110	0.118	0.08	-0.2000	1.0	18.8	246.5	60.00	0.35 for 45 min; 0 for 15 min	154.0	88.0	94.0
47	17.2	46	50	51.6	17	0.026	20	9.4	0.198	80	150	0.227	0.08	1. 0000	1.0	18.8	297.5	45.50	0.35 for 45.5 min	140.0	83.7	96.0
48	17.2	46	50	51.6	17	0.026	20	11	0.204	80	170	0.300	0.08	-0.3000	1.0	18.9	254.8	55.00	0.35 for 45 min; 0 for 10 min	166.0	87.3	95.0
49	17.2	46	50	51.6	17	0.026	20	11.1	0.197	80	180	0.228	0.08	1.5000	1.0	18.9	242.4	45.00	0.35	159.0	93.0	

Table 3. Comparison of model performances (MSEs) using piecewise N_2 flowrate conditions.

	Water		**Heptane**		**Octane**	
	Train	**Test**	**Train**	**Test**	**Train**	**Test**
MLP Dropout (0.2)	4.5043	13.6808	3.029	11.646	2.959	10.982
MLP Dropout (0.4)	11.3727	12.8414	8.9308	16.4888	9.3543	16.7297
MLP Cosine Loss	**0.6532**	17.9388	**1.385**	14.6387	**1.4134**	15.4653
CNN (4 Filters)	8.4781	**12.3324**	10.3334	8.1957	11.4962	8.3518
CNN (8 Filters)	8.3272	16.0255	9.6154	**7.9111**	10.5427	**7.8215**

Table 4. Comparison of model performances (MSEs) ignoring the N_2 flowrate.

	Water		**Heptane**		**Octane**	
	Train	**Test**	**Train**	**Test**	**Train**	**Test**
MLP Dropout (0.2)	4.247	11.808	3.120	9.158	3.135	8.475
MLP Dropout (0.4)	12.008	10.636	7.177	12.455	7.832	12.665
MLP Cosine Loss	11.930	9.703	13.163	10.292	14.889	10.604
CNN (4 Filters)	11.217	9.439	12.703	9.639	14.318	9.653
CNN (8 Filters)	9.836	9.604	11.219	11.977	12.327	11.608

Table 5. Theoretically optimal process conditions that maximize the contact angles.

	Substrate Temperature (°C)	Reactor Wall Temperature (°C)	Glass Heating (%)	Leak Flowrate (sccm)	Initiator Flowrate (sccm)	Monomer Flowrate (sccm)	P_total - P_base (torr)	I_Filament (A) x V_Filament (V)	Reaction Time 0 (sec)	N2 Flow 0 (sccm)	Reaction Time 1 (sec)	N2 Flow 1 (sccm)	Reaction Time 2 (sec)	N2 Flow 2 (sccm)
Convex Surrogate Grid Search Optimization	47.90	56.10	17.00	0.04	0.20	0.22	0.19	25.60	2880	0.45	1,200	0.10	600	0.30
Data Sample with Maximum Values	48.00	56.20	17.00	0.06	0.21	0.30	0.20	25.65	2,880	0.50	1,200	0.10	600	0.35
CNN (8 Filters) Grid Search Optimization	46.56	48.24	17.00	0.02	0.12	0.04	0.09	18.71	2,877	0.01	137	0.09	600	0.00
CNN (6 Filters) Grid Search Optimization	46.17	49.50	17.00	0.02	0.12	0.04	0.19	18.71	2,178	0.39	420	0.00	600	0.00
CNN (4 Filters) Grid Search Optimization	46.94	49.50	17.00	0.02	0.12	0.04	0.09	18.71	2,296	0.33	416	0.00	600	0.00
Data Sample with Maximum Output	46.00	52.50	17.00	0.04	0.20	0.12	0.09	18.80	2,700	0.35	1,200	0.00	0	0.00

Table 6. Contact angles predicted by the fived models with the inputs (optimal process conditions) reported in Table 5.

		Water	Heptane	Octane
A	**Convex Surrogate Grid Search Optimization**	155.2	89.1	125.7
B	**Data Sample of Maximum Values**	145.2	72.0	108.6
C	**CNN (8 Filters) Grid Search Optimization**	154.6	43.5	45.4
D	**CNN (6 Filters) Grid Search Optimization**	143.3	68.5	75.3
E	**CNN (4 Filters) Grid Search Optimization**	132.9	82.8	98.6
F	**Data Sample with Maximum Output**	145.0	74.2	80.0

Table 7. Measured contact angles of the polymer films produced by the iCVD process operated with the optimal process conditions reported in Table 5.

	ML Modeling/Optimization Method	**Water**	**Heptane**	**Octane**
A	**Convex Surrogate Grid Search Optimization**	150.7	88.0	97.3
B	**Data Sample of Maximum Values**	150.6	70.0	78.7
C	**CNN (8 Filters) Grid Search Optimization**	112.9	62.4	68.5
D	**CNN (6 Filters) Grid Search Optimization**	105.1	57.4	62.6
E	**CNN (4 Filters) Grid Search Optimization**	107.0	25.5	36.0
F	**Data Sample with Maximum Output**	150.6	63.6	79.7