

On the Data-Driven Modeling of Reactive Extrusion

Ruben Ibañez ¹, Fanny Casteran ², Clara Argerich ¹, Chady Ghnatios ³ , Nicolas Hascoet ¹,
Amine Ammar ⁴, Philippe Cassagnau ² and Francisco Chinesta ^{1,*}

¹ PIMM, Arts et Métiers Institute of Technology, 151 Boulevard de l'Hôpital, 75013 Paris, France; Ruben.IBANEZ-PINILLO@ensam.eu (R.I.); clara.argerich_martin@ensam.eu (C.A.); nicolas.hascoet@ensam.eu (N.H.)

² Univ-Lyon, Université Lyon 1, Ingénierie des Matériaux Polymères, CNRS, UMR5223, 15 Boulevard André Lataret, 69622 Villeurbanne, France; fanny.casteran65@gmail.com (F.C.); philippe.cassagnau@univ-lyon1.fr (P.C.)

³ Notre Dame University-Louaize, P.O. Box 72, Zouk Mikael, Zouk Mosbeh, Lebanon; cghnatios@ndu.edu.lb

⁴ LAMPA, Arts et Métiers Institute of Technology, 2 Boulevard du Ronceray, 49035 Angers, France; Amine.AMMAR@ensam.eu

* Correspondence: Francisco.Chinesta@ensam.eu

Received: 2 June 2020; Accepted: 10 June 2020; Published: 15 June 2020



Abstract: This paper analyzes the ability of different machine learning techniques, able to operate in the low-data limit, for constructing the model linking material and process parameters with the properties and performances of parts obtained by reactive polymer extrusion. The use of data-driven approaches is justified by the absence of reliable modeling and simulation approaches able to predict induced properties in those complex processes. The experimental part of this work is based on the in situ synthesis of a thermoset (TS) phase during the mixing step with a thermoplastic polypropylene (PP) phase in a twin-screw extruder. Three reactive epoxy/amine systems have been considered and anhydride maleic grafted polypropylene (PP-g-MA) has been used as compatibilizer. The final objective is to define the appropriate processing conditions in terms of improving the mechanical properties of these new PP materials by reactive extrusion.

Keywords: reactive extrusion; data-driven; machine learning; artificial engineering; polymer processing; digital twin

1. Introduction

Initially, the industry adopted virtual twins in the form of simulation tools that represented the physics of materials, processes, structures, and systems from physics-based models. These computational tools transformed engineering science and technology to offer optimized design tools and became essential in almost all industries at the end of the 20th century.

Despite of the revolution that Simulation Based Engineering—SBE—experienced, some domains resisted to fully assimilate simulation in their practices for different reasons:

- Computational issues related to the treatment of too complex material models involved in too complex processes, needing a numerical resolution difficult to attain. Some examples in polymer processing concern reactive extrusion or foaming, among many others.
- Modeling issues when addressing materials with poorly known rheologies, as usually encountered in multi-phasic reactive flows where multiple reactions occur.
- The extremely multi-parametric space defined by both the material and the process, where the processed material properties and performances strongly depend on several parameters related, for example in the case of reactive extrusion, to the nature of the reactants or the processing parameters, like the flow rate and viscosity, the processing temperature, etc.

In these circumstances, the use of data and the construction of the associated models relating the material and processing parameters to some quantities of interest—QoI, by using advanced artificial intelligence techniques, seems an appealing procedure for improving predictions, enhancing optimization procedures and enabling real-time decision making [1].

1.1. Data-Driven Modeling

Engineered artificial intelligence—EAI—concerns different data-science functionalities enabling: (i) multidimensional data visualization; (ii) data classification; (iii) modeling the input/output relationship enabling quantitative predictions; (iv) extracting knowledge from data; (v) explaining for certifying, and (vi) creating dynamic data-driven applications systems.

The present work aims at creating a model able to relate material and processing parameters to the processed material properties and performances. For this reason, in that follows, we will focus on the description and use of different strategies for accomplishing that purpose.

In the past, science was based on the extraction of models, these being simply the causal relation linking causes (inputs) and responses (outputs). This (intelligent) extraction or discovery was performed by smart (and trained) human minds from the data provided by the direct observation of the reality or from engineered experimental tests. Then, with the discovered, derived, or postulated model, predictions were performed, leading to the validation or rejection of these models. Thus, physics-based models, often in the form of partial differential equations, were manipulated by using numerical techniques, with the help of powerful computers.

However, sometimes models are not available, or they are not accurate enough. In that case, the most natural route consists of extracting the model from the available data (a number of inputs and their associated outputs). When data are abundant and the time of response is not a constraint, deep-learning could constitute the best alternative. However, some industrial applications are subjected to: (i) scarce data and (ii) necessity of learning on-the-fly under stringent real-time constraints.

Some models, as those encountered in mechanics, are subjected to thermodynamic consistency restrictions. They impose energy conservation and entropy production. In our former works [2–5], we proved that such a route constitutes a very valuable framework for deriving robust models able to assimilate available data while fulfilling first principles. However, some models cannot be cast into a physical framework because they involve heterogeneous data, sometimes discrete and even categorical. Imagine for awhile a product performance that depends on four factors: (i) temperature of the oven that produces the part; (ii) process time; (iii) commercial name of the involved material, and (iv) given-name of the employee that processed it. It seems evident that processes, whose representing data-points (implying four dimensions in this particular example) are close to each other, do not imply having similar performances. In that case, prior to employing techniques performing in vector spaces, in general based on metrics and distances, data must be mapped into that vector space. For this purpose, we proposed recently the so-called *Code2Vect* [6] revisited later.

Nonlinear regressions, relating a given output with the set of input parameters, are subjected to a major issue: complexity. In other words, the number of terms of a usual polynomial approximation depends on the number of parameters and the approximation degree. Thus, D parameters and a degree Q imply the order of D power Q terms and, consequently, the same amount of data are needed to define it. In our recent works, we proposed the so-called sparse Proper Generalized Decomposition, sPGD [7] able to circumvent the just referred issue. It is based on the use of separated representations with adaptive degree of approximation, defined on unstructured data settings and sparse sensing to extract the most compact representations.

Assuming that the model is expressible as a matrix relating arrays of inputs and outputs (as standard Dynamic Mode Decomposition—DMD—performs with dynamical systems [8,9]) both expressible in a low-dimensional space (assumption at the heart of all model order reduction techniques), the rank of the matrix (discrete description of the model) is assumed to be low [10].

1.2. Reactive Polymers Processing

Reactive extrusion is considered to be an effective tool of continuous polymerization of monomers, chemical modification of polymers and reactive compatibilisation of polymer blends. In particular, co-rotating and contra-rotating twin-screw extruders have proven to be a relevant technical and economical solution for reactive processing of thermoplastic polymers. The literature dedicated to reactive extrusion shows that a very broad spectrum of chemical reactions and polymer systems has been studied [11–15].

The many advantages of using the extruder as a chemical reactor can be described as follows: (i) polymerization and/or chemical modifications can be carried out in bulk, in the absence of solvents, the process is fast and continuous (residence time of the order of a few minutes); (ii) if necessary, devolatilization is effective, leading to the rapid removal of residual monomers and/or reaction by-products; and (iii) the screw design is modular, allowing the implementation of complex formulations (fillers, plasticizers, etc.).

However, there are also some disadvantages in using an extruder as a chemical reactor such as: (i) the high viscosity of the molten polymers, which lead to self-heating and therefore to side reactions (thermal degradation for example); (ii) the short residence time which limits reactive extrusion to fast reactions; and (iii) the difficulty of the scale up to industrial pilot and plants.

In terms of modeling and simulation, various strategies [16] can be considered as it needs to deal with a large number of highly nonlinear and coupled phenomena. Actually, the strategy of modeling depends on the objectives in terms of process understanding, material development from machine design or process optimization, and control. For example, in the case of free radical grafting of polyolefins, a two-phase stochastic model to describe mass transport and kinetics based on reactive processing data was proposed in [17].

Regarding process optimization, a simple 1D simulation approach provides a global description of the process all along the screws, whereas 3D models allow a more or less accurate description of the flow field in the different full zones of the extruder. However, most of these simulations are based on simplified steady-state 1D models (e.g., Ludovicc© software [18]).

Actually, the main processing parameters such as residence time, temperature, and extent of the reaction are assumed homogeneously distributed in any axial cross section. The use of one-dimensional models allows significant reductions of the simulation effort (computing time savings). In any case, the flow model is coupled with reaction kinetics that impact the fluid rheology [19].

Thus, one-dimensional models are specially appropriate when addressing optimization or control in reactive extrusion. In particular, the model proposed in [20] predicts the transient and steady-state behaviors, i.e., pressure, monomer conversion, temperature, and residence time distribution in different operation conditions.

However, these simulations require several sub-models on establishing constitutive equations (viscosity, chemical kinetics, mass and temperature transfers). Actually, it takes time and the intuition and accumulated knowledge of experienced specialists. Furthermore, it is important to note that, despite the impressive effort spent by hundreds of researchers and thousands of published papers, no constitutive equation exists describing, for example, the behavior of complex polymer formulations such as reactive extrusion systems.

In summary, such a process is quite complex and would require a detailed study on the influence of the nature of polymers and chemical reactions (kinetics and rheology), processing conditions (temperature, screw speed, flow rate, screw profile). Nevertheless, a determinist answer to each of these parameters is out of consideration and actually we believe that the understanding of such a process is quite unrealistic from usual approaches.

1.3. Objectives of the Study

The present work aims at addressing a challenge in terms of industrial applications that is not necessarily based on improving the understanding of the process itself, but replacing the complex fluid

and complex flow by an alternative modeling approach able to extract the link between the process outputs and inputs, key for transforming experience into knowledge.

A model of a complex process could be envisaged with two main objectives: (i) the one related to the online process control from the collected and assimilated data; (ii) the other concerned by the offline process optimization, trying to extract the optimal process parameters enabling the target properties and performances. Even if the modeling procedure addressed in this work could be used in both domains, the present work mainly focuses on the second one, the process modeling for its optimization; however, as soon as data could be collected in real-time, with the model available, process control could be attained without major difficulties.

There are many works in which each one uses a different data-driven modeling technique, diversity that makes it difficult to understand if there is an optimal technique for each model, or if most of them apply and perform similarly. Thus, this paper aims at comparing several techniques first, and then, using one of them that the authors recently proposed, and that performs in the multi-parametric setting, address some potential uses.

2. Modeling

In this section, we revisit some regression techniques that will be employed after for modeling reactive extrusion. For additional details and valuable references the interested reader can refer to Appendix A.

In many applications like chemical and process engineering or materials processing, product performances depend on a series of parameters related to both, the considered materials and the processing conditions. The number of involved parameters is noted by D and each parameter by x_i , $i = 1, \dots, D$, all of them grouped in the array \mathbf{x} .

The process results in a product characterized by different properties or performances in number smaller or greater than D . In what follows, for the sake of simplicity and without loss of generality, we will assume that we are interested in a single scalar output noted by y .

From the engineering point of view, one is interested in discovering the functional relation between the quantity of interest—QoI— y and the involved parameters $x_1, \dots, x_D \equiv \mathbf{x}$, mathematically, $y = y(\mathbf{x})$ because it offers a practical and useful way for optimizing the product by choosing the most adequate parameters \mathbf{x}^{opt} .

There are many techniques for constructing such a functional relation, currently known as regression, some of them sketched below, and detailed in Appendix A where several valuable references are given.

2.1. From Linear to Nonlinear Regression

The simplest choice consists in the linear relationship

$$y = \beta_0 + \beta_1 x_1 + \dots + \beta_D x_D, \quad (1)$$

that if $D + 1$ data are available that is $D + 1$ couples $\{y_s, \mathbf{x}_s\}$, $s = 1, \dots, D + 1$, then the previous equation can be written in the matrix form

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{D+1} \end{pmatrix} = \begin{pmatrix} 1 & x_{1,1} & x_{2,1} & \cdots & x_{D,1} \\ 1 & x_{1,2} & x_{2,2} & \cdots & x_{D,2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{1,D+1} & x_{2,D+1} & \cdots & x_{D,D+1} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_D \end{pmatrix}, \quad (2)$$

where $x_{i,s}$ denotes the value of parameter x_i at measurement s , with $i = 1, \dots, D$ and $s = 1, \dots, D + 1$. The previous linear system can be expressed in a more compact matrix form as

$$\mathbf{y} = \mathbb{X}\boldsymbol{\beta}. \quad (3)$$

Thus, the regression coefficients β_0, \dots, β_D are computed by simple inversion of Equation (3)

$$\boldsymbol{\beta} = \mathbb{X}^{-1}\mathbf{y}, \quad (4)$$

from which the original regression form (1) can be rewritten as

$$y = \beta_0 + \mathbf{W}^T \mathbf{x}, \quad (5)$$

where $\mathbf{W}^T = (\beta_1 \cdots \beta_D)$.

When the number of measurements P becomes larger than the number of unknowns β_0, \dots, β_D , i.e., $P > D + 1$, the problem can be solved in a least-squares sense.

However, sometimes linear regressions become too poor for describing nonlinear solutions and in that case one is tempted to extend the regression (1) by increasing the polynomial degree. Thus, the quadratic counterpart of Equation (1) reads

$$y = \beta_0 + \sum_{i=1}^D \sum_{j \geq i}^D \beta_{ij} x_i x_j, \quad (6)$$

where the number of unknown coefficients (β_0 & β_{ij} , $\forall i, j$) is $(D^2 - D)/2$ that roughly scales with D^2 . When considering third degree approximations, the number of unknown coefficients scales with D^3 and so on.

Thus, higher degree approximations are limited to cases involving few parameters, and multi-parametric cases must use low degree approximations because usually the available data are limited due to the cost of experiences and time.

The so-called sparse-PGD [7] tries to encompass both wishes in multi-parametric settings: higher degree and few data. For that purpose, the regression reads

$$y = \sum_{i=1}^N \prod_{j=1}^D F_i^j(x_j), \quad (7)$$

where the different single-valued functions $F_i^j(x_j)$ are a priori unknown and are determined sequentially using an alternate directions fixed point algorithm. As at each step one looks for a single single-valued function, higher degree can be envisaged for expressing it into a richer (higher degree) approximation basis, while keeping reduced the number of available data-points (measurements).

2.2. Code2Vect

This technique deeply revisited in the Appendix A proposes mapping points \mathbf{x}_s , $s = 1, \dots, P$, into another space $\boldsymbol{\xi}_s$, such that the distance between any pair of data-points $\boldsymbol{\xi}_i$ and $\boldsymbol{\xi}_j$ scales with the difference of their respective outputs, that is, on $|y_i - y_j|$.

Thus, using this condition for all the data-point pairs, the mapping \mathbf{W} is obtained, enabling for any other input array \mathbf{x} compute its image $\boldsymbol{\xi} = \mathbf{W}\mathbf{x}$. If $\boldsymbol{\xi}$ is very close to $\boldsymbol{\xi}_s$, one can expect that its output y becomes very close to y_s , i.e., $y \approx y_s$. In the most general case, an interpolation of the output is envisaged.

2.3. iDMD, Support Vector Regression, and Neural Networks

Inspired by dynamic model decomposition—DMD—[8,9] one could look for \mathbf{W} minimizing the functional $\mathcal{F}(\mathbf{W})$ [10]

$$\mathcal{F}(\mathbf{W}) = \sum_{s=1}^P \left(y_s - \mathbf{W}^T \mathbf{x}_s \right)^2, \quad (8)$$

whose minimization results in the calculation of vector \mathbf{W} that at its turn allows defining the regression $y = \mathbf{W}^T \mathbf{x}$. Appendix A and the references therein propose alternative formulations.

Neural Networks—NN—perform the same minimization and introduce specific treatments of the nonlinearities while addressing the multi-output by using a different number of hidden neuron layers [21].

Finally, Support Vector Regression—SVR—share some ideas with the so-called Support Vector Machine—SVM [22], the last widely used for supervised classification. In SVR, the regression reads

$$y = \beta_0 + \mathbf{W}^T \mathbf{x}, \quad (9)$$

and the flatness is enforced by minimizing the functional $\mathcal{G}(\mathbf{W})$

$$\mathcal{G}(\mathbf{W}) = \frac{1}{2} \mathbf{W}^T \mathbf{W}, \quad (10)$$

while enforcing as constraints a regularized form of

$$|y_s - \beta_0 - \mathbf{W}^T \mathbf{x}_s| \leq \epsilon, \quad s = 1, \dots, P. \quad (11)$$

3. Experiments

The purpose of this project is the dispersion of a thermosetting (TS) polymer in a polyolefin matrix using reactive extrusion by in situ polymerisation of the thermoset (TS) phase from an epoxide resin and amine crosslinker. Here, Polypropylene (PP) has been chosen as the polyolefin matrix. A grafted PP maleic anhydride (PP-g-MA) has been used to ensure a good compatibility between the PP and the thermoset phases.

These studies were carried out as part of a project with TOTAL on the basis of a HUTCHINSON patent [23]. This patent describes the process for preparing a reinforced and reactive thermoplastic phase by dispersing an immiscible reactive reinforcing agent (e.g., an epoxy resin as precursor on the thermoset dispersed phase). This process is characterized by a high shear rate in the extruder combined with the in-situ grafting, branching, and/or crosslinking of the dispersed phase. These in situ reactions permit the crosslinking of the reinforcing agent as well as the compatibility of the blend with or without compatibilizer or crosslinker. The result of this process is a compound with a homogeneous reinforced phase with thin dispersion ($<5 \mu\text{m}$) leading to an improvement of the mechanical properties of the thermoplastic polymer. The experiments carried out in the framework of the present project are mainly based on some experiments described in the patent. However, new complementary experiments have been carried out to complete the study.

3.1. Materials

The main Polypropylene used as the matrix is the homopolymer polypropylene PPH3060 from TOTAL. Two other polypropylenes have been used to study the influence of the viscosity, and several impact copolymer polypropylenes have also been tested in order to combine a good impact resistance with the reinforcement brought by the thermoset phase. A PP-g-MA (PO1020 from Exxon) with around 1 wt% of maleic anhydride has been used as a compatibilizer between the polypropylene matrix and the thermoset phase. All the polypropylenes used are listed in Table 1 with their main characteristics.

Table 1. Nature, supplier, and melt flow index –MFI– (216 kg/230 °C/10 min) of PP polymers.

Name	Nature	Supplier	MFI (g)
PPH3060	Polypropylene homopolymer	TOTAL	1.8
PPH7060	Polypropylene homopolymer	TOTAL	12
PPH10060	Polypropylene homopolymer	TOTAL	35
PPC14642	Impact Copolymer Polypropylene	TOTAL	130
PPC10641	Impact Copolymer Polypropylene	TOTAL	44
PPC7810	Impact Copolymer Polypropylene	TOTAL	15
PPC7810C	Impact Copolymer Polypropylene	TOTAL	15
Exxelor PO1020	Polypropylene grafted 1 wt% of maleic anhydride	Exxon Mobil	430

Concerning the thermoset phase, three systems have been studied. As a common point, these three systems are based on epoxy resins that are DGEBA derivatives with two epoxide groups, two different resins (DER 667 and DER 671 from DOW Chemicals) have been used. The first two systems, named R1 and R2 here, are both constituted of an epoxy resin mixed with an amine at the stoichiometry. The first uses the DER 667 with a triamine (Jeffamine T403 from Huntsman) that is sterically hindered, whereas the second one uses the DER 671 with a cyclic diamine (Norbonanediamine from TCI Chemicals). Melamine has also been tested in one of the formulations. The third system, named R5 here, mixes the epoxy resin DER 671 with a phenolic hardener (DEH 84 from DOW Chemicals) that is a blend of three molecules: 70 wt% of an epoxy resin, a diol, and less than 1 wt% of a phenolic amine. These systems have been chosen in order to see the influence of the structure, molar mass, and chemical nature on the in-situ generation of the thermoset phase within our polyolefin matrix. Table 2 summarizes the systems studied.

Table 2. Composition at the stoichiometry of the systems studied.

Epoxy	Amine/Hardener	Abbreviation
DER 667	Jeffamine T403	R1
DER 671	DEH 84	R2
DER 671	Norbonane diamine	R5

The kinetics of these chemical systems have been studied from the variation of the complex shear modulus from a time sweep experiment with an ARES-G2 Rheometer (TA Instruments). The experiments have been performed for temperatures from 115 °C to 165 °C using a 25 mm plate-plate geometry, with a 1 mm gap, at the frequency $\omega = 10$ rad/s and a constant strain of 1%. The kinetics have been performed on a stoichiometric premix of the reactants. The gel times of the systems have thus been identified as the crossover point between the loss and storage modulus. Note that the reaction is too fast to be performed at temperatures beyond $T = 165$ °C. Consequently, an extrapolation according to an Arrhenius law allowed us to determine the gel time of the systems at $T = 200$ °C (Barrel temperature of the extruder). The results give a gel time lower than 10 s for the three systems ($t_{gel}(R1) = 4.5$ s, $t_{gel}(R2) = 10$ s, and $t_{gel}(R5) < 1$ s), so we made the hypothesis that the reaction time is much lower than 1 min and thus that the reaction is totally completed at the die exit of the extruder. Moreover, a Dynamic Mechanical Analysis (DMA) showed that the main mechanical relaxation T_α associated with the T_g of the thermoset phase is close to 80 °C, which is the T_g observed for TS bulk systems.

The influence of the addition of silica on the final properties has been studied with two different silicas (Aerosil R974 and Aerosil 200).

3.2. Methods

3.2.1. Extrusion Processing

The formulations have been fulfilled in one single step with a co-rotating twin screw extruder (Leistritz ZSE18, L/D = 60, D = 18 mm), with the screw profile described in Figure 1.

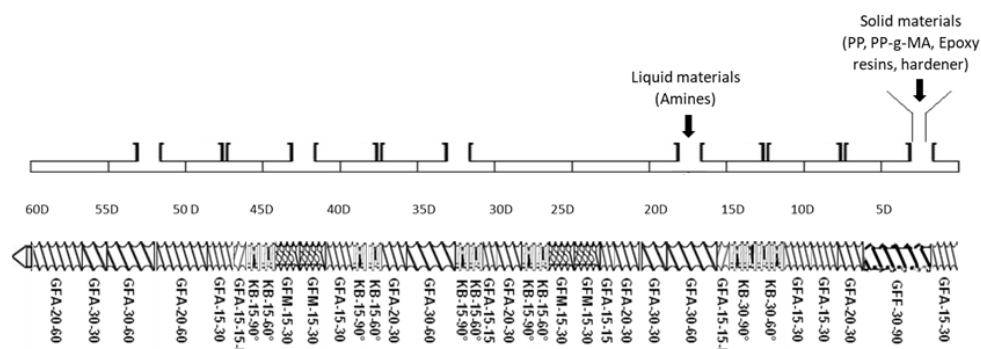


Figure 1. Diagram of the screw profile in the study.

Two different temperatures profiles have been used, one at 230 °C and the other one at 200 °C, both with lower temperatures for the first blocs to minimize clogging effects at the inlet. These temperature profiles are described in Figure 2.

230 °C Temperature profile												
N° Bloc	11	10	9	8	7	6	5	4	3	2	1	Alim
SetpointTemperature	230 °C	230 °C	230 °C	230 °C	230 °C	230 °C	230 °C	230 °C	215 °C	190 °C	180 °C	110 °C

200 °C Temperature profile												
N° Bloc	11	10	9	8	7	6	5	4	3	2	1	Alim
SetpointTemperature	200 °C	200 °C	200 °C	200 °C	200 °C	200 °C	200 °C	200 °C	200 °C	150 °C	150 °C	100 °C

Figure 2. Diagram of the temperature profile in the study.

Several screw rotation speeds and flow-rates have been used to study the influence of the process on the final materials ($N = 300, 600, 450, 800$ rpm; $\dot{w} = 3, 5, 6, 10$ kg/h).

The solid materials were mixed and introduced at the entrance by a hopper for the pellets and with a powder feeder for the micronized powders. As for the liquid reagents, they were injected over the third bloc with an HPLC pump. The formulations are cooled by air at the exit of the extruder and then pelleted.

3.2.2. Characterization

Tensile-test pieces (5A) and impact-test pieces have been injected with a Babyplast injection press at 200 °C and 100 bar. Young modulus has been determined by a tensile test with a speed of 1 mm/min and Stress at yield, Elongation at break, and Stress at break have been measured with a tensile speed of 50 mm/min. Impact strength has been measured by Charpy tests on notched samples at room temperature.

4. Data-Driven Modeling: Comparing Different Machine Learning Techniques

As previously mentioned, a model that links the material and processing parameters with the processed material properties is of crucial interest. By doing that, two major opportunities could be envisaged: the first one concerns the possibility of inferring the processed material properties for

any choice of manufacturing parameters; (second), for given target properties, one could infer the processing parameters enabling them.

In this particular case, process parameters are grouped in the six-entrees array \mathbf{x} :

$$\mathbf{x} = \begin{pmatrix} \text{Rotation Speed [rpm]} \\ \text{Exit Flow Rate [kg/h]} \\ \text{Temperature [}^{\circ}\text{C]} \\ \text{TS dispersed Phase [\%]} \\ \text{PP - g - MA/TS Ratio [-]} \\ \text{Ratio TS [-]} \end{pmatrix} \quad (12)$$

whereas the processed material properties are grouped in the five-entrees array \mathbf{y} , containing the Young modulus, the yield stress, the stress at break, the strain at break, and the impact strength,

$$\mathbf{y} = \begin{pmatrix} E \text{ [MPa]} \\ \sigma_y \text{ [MPa]} \\ \sigma_b \text{ [MPa]} \\ \epsilon_b \text{ [-]} \\ Re_{amb} \left[\frac{kJ}{m^2} \right] \end{pmatrix}. \quad (13)$$

As previously discussed, our main aim is extracting (discovering) the regression relating inputs (material and processing parameters) \mathbf{x} with the outputs (processed material properties) \mathbf{y} , the regression that can be written in the form

$$y_i = y_i(\mathbf{x}), \quad (14)$$

where $y_i()$ represents the linear or nonlinear regression associated with the i -output, or when proceeding in a compact form by creating the multi-valued regression relating the whole input and output data-pairs, as

$$\mathbf{y} = \mathbf{y}(\mathbf{x}). \quad (15)$$

The intrinsic material and processing complexity justify the nonexistence of valuable and reliable models based on physics, able to predict the material evolution and the process induced properties. For this reason, in the present work, the data-driven route is retained, from the use of regression techniques, as the ones previously summarized.

The available data come from experiments conducted, described in the previous section that consists of P pairs of arrays \mathbf{x}_s , and \mathbf{y}_s , $s = 1, \dots, P$, that is:

$$[\mathbf{x}_s, \mathbf{y}_s], \quad s = 1, \dots, P, \quad (16)$$

all them reported in Tables A1 and A2 included in the Appendix B (for the sake of completeness and for allowing researchers to test alternative regression procedures).

Table A1 groups the set of input parameters involved in the regression techniques. The hyper parameter *MaskIn* is a boolean mask indicating if the data are included in the training (*MaskIn*= 1) or it is excluded from the training to serve for quantifying the regressions performance (*MaskIn*= 0). On the other hand, Table A2 groups the responses, experimental measures, for each processing condition.

As indicated in the introduction section, one of the objectives of the present paper is analyzing if different machine learning techniques perform similarly, or their performances are significantly different. For this purpose, this section aims at comparing the techniques introduced in Section 2, whereas the next section will focus on the use of one of them.

In order to compare the performances of the different techniques, an error was introduced serving to compare the regressions prediction. In particular, we consider the most standard error, the Root

Mean Squared Error (RMSE). When applied to the different regression results, it offers a first indication on the prediction performances. Table 3 reports the errors associated with each regression when evaluating the output of interest that is the array \mathbf{y} for a given input \mathbf{x} , for all the data reported in Tables A1 and A2. Because the different outputs (the components of array \mathbf{y}) present significant differences in their typical magnitudes, Table 4 reports the relative errors, computed from the ratios between the predicted and measured data difference to the measured data.

Table 3. RMSE error for the different regression techniques.

RMSE ERROR (%)	E (MPa)	σ_{yield} (MPa)	σ_{break} (MPa)	ϵ_{break}	Re_{amb} (kJ/m)
NN	369.81	7.50	6.92	119.16	2.44
C2V	289.22	5.91	5.81	126.83	1.89
SPGD	229.69	5.34	4.53	127.36	2.05
SVM	229.69	5.34	4.53	127.36	2.05
D. TREE	268.61	5.39	5.53	111.86	2.21
iDMD	252.72	5.41	5.53	114.83	1.89

Table 4. Relative error for different regression techniques.

REL ERROR (%)	E (MPa)	σ_{yield} (MPa)	σ_{break} (MPa)	ϵ_{break}	Re_{amb} (kJ/m)
NN	23.43	24.05	34.96	100.22	43.38
C2V	15.06	16.41	22.22	105.54	37.05
SPGD	18.32	18.97	29.38	106.67	33.73
SVM	14.55	17.12	22.90	107.12	36.58
D. TREE	17.02	17.30	27.96	94.08	39.38
iDMD	16.01	17.35	27.96	96.58	33.70

Sparse PGD—sPGD—employed second degree Chebyshev polynomials and performed a regression for each of the quantities of interest according to Equation (14). The use of low degree polynomials avoided overfitting, being a compromise for ensuring a reasonable predictability for data inside and outside the training data-set. From a computational point of view, 20 enrichment (N in Equation (A16)) were needed for defining the finite sum involved in the separated representation that constitutes the regression of each output of interest y_i .

Code2Vect addressed the low-data limit constraint by imposing a linear mapping between the representation (data) and target (metric) spaces, avoiding spurious oscillations when making predictions on the data outside the training set.

Considering the iDMD because of the reduced amount of available data, the simplest option consisting of a unique matrix relating the input–output data pairs in the training set (linear model) was considered, i.e., with respect to Equation (15), it was assumed $\mathbf{F}(\mathbf{x}) = \mathbb{F}\mathbf{x}$, and matrix \mathbb{F} ensuring the linear mapping was obtained by following the rationale described in Section 2. The computed regression performs very well despite the fact of assuming a linear behavior.

The quite standard Neural Network we considered (among a very large variety of possible choices) presents a severe overfitting phenomenon in the low data-limit addressed here. This limitation is not intrinsic to NN, and could be alleviated by considering richer architectures and better optimizers, parameters, out of the scope of the present study.

The main conclusion of this section is the fact that similar results are obtained independently of the considered technique that seems quite good now from the point of view of engineering. Even if the errors seem quite high, it is important to note that: (i) the highest errors concern the variables exhibiting the largest dispersion in the measurements; (ii) the prediction errors are of the same order

as that of the dispersion amplitudes; and (iii) we only considered 35 data-points from the 59 available for the training (regression construction) while the reported errors were calculated by using the whole available data (the 59 data-points). The next section proves that the prediction quality increases with the increase of the amount of points involved in the regression construction (training).

5. Data-Driven Process Modeling

In view of the reported results, it can be stressed that all the analyzed techniques show similar performances and work reasonably well in the low-data limit (where only 60% out of 59 data points composed the training data-set were used in the regressions).

As it can be noticed, some quantities of interest such as the Young's modulus and the stress at break are quite well predicted when compared with the others on which predictions were less performant. There is a strong correlation between this predictability capability and the experimental dispersion noticed when measuring these other quantities, like the strain at break. That dispersion represents without any doubt a limit in the predictability that must be addressed within a probabilistic framework. All the mechanical tests were performed on five samples from the same experiment process on the extruder. The final value is the average of these five tests. The confidence interval is estimated: 10% for the Young modulus and Yield stress, 20% on elongation and stress at break.

Extracting a model of a complex process could serve for real-time control purposes, but also, as it is the case in the present work, for understanding the main tendencies of each quantity of interest with respect to each process or material parameter (the last constituting the regression inputs), enabling process and material optimization.

In order to perform that sensibility analysis, we consider a given quantity of interest and evaluate its evolution with respect to each of the input parameters. When considering the dependence with respect to a particular input parameter, all the others are fixed to their mean values, even if any other choice is possible. Figure 3 shows the evolution of σ_b with respect to the six input parameters, using the lowest order sPGD modes to extract the main tendencies.

From these AI-based metamodels, one should be able to identify the process conditions and the concentration of the TS phase in order to enhance a certain mechanical property. Thus, in order to increase the stress at break, increasing the content of thermoset seems a good option, with all the other properties (Young modulus, stress at yield, strain at break and impact strength being almost insensible to that parameter). A more detailed analysis, involving multi-objective optimization (making use of the Pareto front) and its experimental validation, constitutes a work in progress, out of the scope of the present work.

To further analyze the accuracy of the methodology and the convergence behavior, in what follows, we consider one of the regression techniques previously described and employed, the sPGD, and perform a convergence analysis, by evaluating the evolution of the error with respect to the size of the training data-set.

The training-set was progressively enriched, starting from 30 data points, and then considering 35, 41, 47, and finally 53 (that almost correspond to 50%, 60%, 70%, 80%, and 90% of the available data-set). The error was calculated again by considering both the training and test data-sets. Table 5 reports the results on the elastic modulus prediction, and clearly proves, as expected, that the prediction accuracy increases with the size of the training-set, evolving from around 15% to finish slightly below 10%.

It is important to note that one could decrease even more the error when predicting the training-set, but overfitting issues will occur and the error will increase tremendously out of the training set compromising robustness. The errors here reported are a good compromise between accuracy in and out of the training-set.

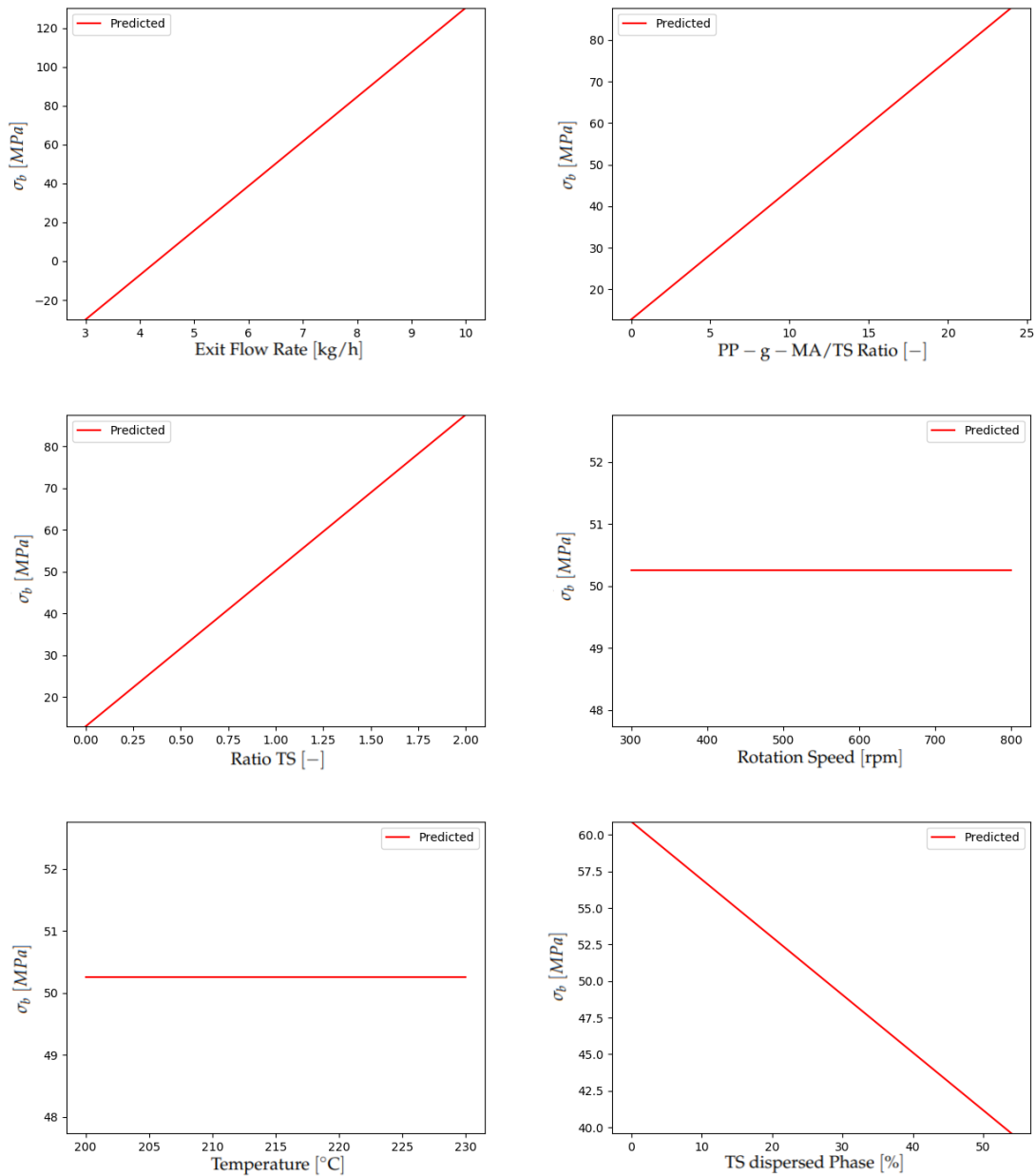


Figure 3. Evolution of the stress at break with: (top-left) Exit flow rate; (top-right) PP-g-MA/TS ratio; (middle-left) Ratio TS; (middle-right) Rotation speed; (bottom-left) Temperature; and (bottom-right) TS dispersed.

Table 5. Mean error for different numbers of points in the training-set.

Training-Set	Mean Error (%)
30	15.4
35	12.0
41	10.4
47	10.1
53	9.9

In order to facilitate the solution reproducibility, in what follows, we give the explicit form of the sPGD regression. As previously discussed, the sPGD makes use of a separated representation of the parametric solution, whose expression reads for a generic quantity of interest $u(\mathbf{x})$

$$u(\mathbf{x}) = \sum_{i=1}^N \prod_{j=1}^D F_i^j(x_j). \quad (17)$$

More explicitly, each univariate function $F_i^j(x_j)$ is approximated using an approximation basis,

$$u(\mathbf{x}) = \sum_{i=1}^N \prod_{j=1}^D \sum_{k=1}^Q T_k^j(x_j) \alpha_{ki}^j. \quad (18)$$

When approximating the elastic modulus, whose results were reported in Table 5, we considered six parameters, i.e., $D = 6$, a polynomial Chebyshev basis consisting of the functions $T_k^j(x_j)$ (needing for a pre-mapping of the parameter intervals into the reference one $[-1, 1]$ where the Chebyshev polynomials are defined). The number of modes (terms involved by the finite sum separated representation) and number of interpolation functions per dimension were set to $N = 10$ and $Q = 3$. The coefficients related to Equation (18) when applied to the elastic modulus approximation are reported in Appendix C.

An important limitation, inherent to machine learning strategies, is the fact that most likely other factors instead of the ones considered as inputs could be determinant for expressing the selected outputs. This point constitutes a work in progress.

6. Conclusions

We showed in this paper that different machine learning techniques are relevant in the low-data limit, for constructing the model that links material properties and process parameters in reactive polymer processing. Actually, these techniques are undeniably effective in complex processes such as reactive extrusion. More precisely, this work was based on the in situ synthesis of a thermoset phase during its mixing/dispersion with a thermoplastic polymer phase, which is certainly one of the most complex cases in the processing of polymers.

We proved that a variety of procedures can be used for performing the data-driven modeling, whose accuracy increases with the size of the training-set. Then, the constructed regression can be used for predicting the different quantities of interest, for evaluating their sensitivity to the parameters, crucial for offline process optimization, and also for real-time process monitoring and control.

Author Contributions: Conceptualization, P.C. and F.C. (Francisco Chinesta); methodology, R.I. and F.C. (Fanny Casteran); software, R.I., C.A., C.G., N.H., and A.A.; validation, F.C. (Fanny Casteran). All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: Authors acknowledge ESI Group by its research chair at Arts et Metiers Institute of Technology, the French ANR through the DataBEST project as well as Nicolas Garois (Hutchinson company) for initiating the experimental part of this project with the support of the Total Research Department (La Defense, Paris). Mathilde Aucler, Aurelie Vanhille and Luisa Barroso-Gago are thanked for their help with the extruder and mechanical experiments.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Machine Learning Techniques

This appendix revisits, with some technical detail, the different machine learning techniques employed in the analyzes addressed in the present work.

Appendix A.1. Support Vector Regression

Finally, Support Vector Regression—SVR—shares some ideas of the so-called Support Vector Machine—SVM—widely used in supervised classification. In SVR, the regression reads

$$y = \beta_0 + \mathbf{W}^T \mathbf{x}, \quad (\text{A1})$$

and the flatness is enforced by minimizing the functional $\mathcal{G}(\mathbf{W})$

$$\mathcal{G}(\mathbf{W}) = \frac{1}{2} \mathbf{W}^T \mathbf{W}, \quad (\text{A2})$$

while enforcing as constraints a regularized form of

$$|y_s - \beta_0 - \mathbf{W}^T \mathbf{x}_s| \leq \epsilon, \quad s = 1, \dots, P, \quad (\text{A3})$$

in particular

$$\mathcal{G}(\mathbf{W}) = \frac{1}{2} \mathbf{W}^T \mathbf{W} + C \sum_{s=1}^P \zeta_s \zeta_s^*, \quad (\text{A4})$$

with $\zeta_s \geq 0$ and $\zeta_s^* \geq 0$, and

$$y_s - (\mathbf{W}^T \mathbf{x}_s + \beta_0) \leq \epsilon + \zeta_s, \quad (\text{A5})$$

and

$$(\mathbf{W}^T \mathbf{x}_s + \beta_0) - y_s \leq \epsilon + \zeta_s^*, \quad (\text{A6})$$

with many other more sophisticated alternatives to extend to the nonlinear case.

Appendix A.2. Code-to-Vector—Code2Vect

Code2Vect maps data, eventually heterogeneous, discrete, categorical, etc. into a vector space equipped of a Euclidean metric allowing computing distances, and in which points with similar output y remain close one to one another as sketched in Figure A1.

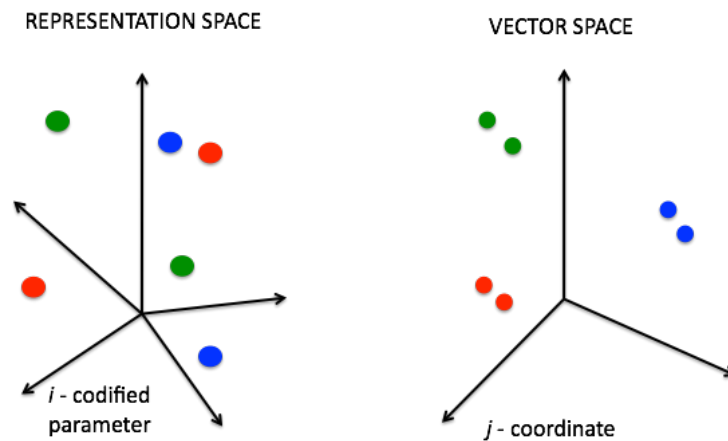


Figure A1. Input space (left) and target vector space (right) [6].

We assume that points in the origin space (space of representation) consist of P arrays composed on D entries, noted by \mathbf{x}_i . Their images in the vector space are noted by $\xi_i \in \mathbb{R}^d$, with $d \ll D$. The mapping is described by the $d \times D$ matrix \mathbf{W} ,

$$\xi = \mathbf{W}\mathbf{x}, \quad (\text{A7})$$

where both the components of \mathbf{W} and the images $\xi_i \in \mathbb{R}^d$, $i = 1, \dots, P$, must be calculated. Each point ξ_i keeps the label (value of the output of interest) associated with its origin point \mathbf{x}_i , denoted by y_i .

We would like to place points ξ_i , such that the Euclidian distance with each other point ξ_j scales with their output difference, i.e.,

$$(\mathbf{W}(\mathbf{x}_i - \mathbf{x}_j)) \cdot (\mathbf{W}(\mathbf{x}_i - \mathbf{x}_j)) = \|\xi_i - \xi_j\|^2 = |y_i - y_j|, \quad (\text{A8})$$

where the coordinates of one of the points can be arbitrarily chosen. Thus, there are $\frac{P^2}{2} - P$ relations to determine the $d \times D + P \times d$ unknowns.

Linear mappings are limited and do not allow with proceeding in nonlinear settings. Thus, a better choice consists of the nonlinear mapping $\mathbf{W}(\mathbf{x})$ [6].

Appendix A.3. Incremental DMD

We reformulate the identification problem in a general multipurpose matrix form

$$\mathbb{W}\mathbf{x} = \mathbf{y}, \quad (\text{A9})$$

where \mathbf{x} and \mathbf{y} represent the input and output vectors, involving variables of different nature, both them accessible from measurements. In what follows, both are assumed D -component arrays.

If we assume both evolving in low-dimensional sub-spaces of dimension d , with $d \ll D$, the rank of \mathbb{K} , the so-called model, is expected reducing to d . The construction of such reduced model was reported in [10], and from the two procedures that were proposed, in what follows, we summarize one of them, the so-called Progressive Greedy Construction.

In this case, we proceed progressively. We consider the first available datum, the pair $(\mathbf{x}_1, \mathbf{y}_1)$. Thus, the first, one-rank, reduced model reads

$$\mathbb{W}_1 = \frac{\mathbf{y}_1 \mathbf{y}_1^T}{\mathbf{y}_1^T \mathbf{x}_1}, \quad (\text{A10})$$

ensuring $\mathbb{W}_1 \mathbf{x}_1 = \mathbf{y}_1$.

Suppose now that a second datum arrives $(\mathbf{x}_2, \mathbf{y}_2)$, from which we can also compute its associated rank-one approximation, and so on, for any new datum $(\mathbf{x}_i, \mathbf{y}_i)$:

$$\mathbb{W}_i = \frac{\mathbf{y}_i \mathbf{y}_i^T}{\mathbf{y}_i^T \mathbf{x}_i}, \quad i = 1, \dots, P. \quad (\text{A11})$$

For any other \mathbf{x} , the model could be interpolated from the just defined rank-one models, \mathbb{W}_i , $i = 1, \dots, P$, according to

$$\mathbb{W}|_{\mathbf{x}} \approx \sum_{i=1}^P \mathbb{W}_i \mathcal{I}_i(\mathbf{x}), \quad (\text{A12})$$

with $\mathcal{I}_i(\mathbf{x})$ the interpolation functions operating in the space of the data \mathbf{x} , functions that in general decrease with the distance between \mathbf{x} and \mathbf{x}_i (e.g., polynomials, radial basis, ...) able to proceed in multidimensional settings.

Appendix A.4. From Polynomial Regression to Sparse PGD-Based Regression

In the regression setting, one could consider a polynomial dependence of the QoI, y , on the parameters x_1, \dots, x_D . The simplest choice, linear regression, reads

$$y(\mathbf{x}) = \beta_0 + \beta_1 x_1 + \dots + \beta_D x_D, \quad (\text{A13})$$

where the $D + 1$ coefficients β_k can be computed from the available data. If $1 + D = P$ data are available, $y_j, j = 1, \dots, 1 + D$, coefficients β_k can be calculated.

Linear regression requires the same amount of data as the number of involved parameters; however, it is usually unable to address nonlinearities.

Nonlinear regressions can be envisaged when the number of parameters remains reduced, due to the fact that the number of terms roughly scales with D to the power of the considered approximation degree.

In this section, we propose a technique able to ensure rich approximations while keeping the sampling richness quite reduced, the so-called multi-local sparse nonlinear PGD-based regression—sPGD. The last reads

$$\int_{\Omega} w(\mathbf{x}) (y(\mathbf{x}) - y) d\mathbf{x} = 0, \quad \forall w(\mathbf{x}) \quad (\text{A14})$$

where Ω is the domain in the parametric space in which the approximation is searched, i.e., $\mathbf{x} \in \Omega$, and $w(\mathbf{x})$ represents the test function whose arbitrariness serves to enforce that the regression $y(\mathbf{x})$ approximates the available data y , and

$$y = \sum_{j=1}^P y \delta(\mathbf{x}_j), \quad (\text{A15})$$

with δ the Dirac mass, to express that data only available at locations \mathbf{x}_j in the parametric space.

Following the Proper Generalized Decomposition (PGD) rationale, the next step is to express the approximated function $y(\mathbf{x})$ in the separated form

$$y(\mathbf{x}) \approx \sum_{i=1}^N F_i^1(x_1) F_i^2(x_2) \cdot \dots, \quad (\text{A16})$$

constructed by using the standard rank-one update [7] that leads to the calculation of the different functions $F_i^j(x_j)$ involved in the separated form (A16).

Appendix A.5. A Simple Neural Network

Deep-learning is mostly based on the use of neural networks, networks composed of components that emulate the neuron functioning that from some incoming data generates an output that, within more complex and large networks, can become the input of other neurons in other layer.

We consider the schema in Figure A2 that illustrates a neuron receiving two input data x_1 and x_2 to produce the output Y . The simplest functioning consists of collecting both data, multiplying each by a weight, W_1 and W_2 and generate the output by adding both contributions according to

$$y = W_1 x_1 + W_2 x_2, \quad (\text{A17})$$

that in the more general case can be written as

$$y = \sum_{i=1}^D W_i x_i \quad (\text{A18})$$

or

$$y = \mathbf{W}^T \mathbf{x}. \quad (\text{A19})$$

The main issue is precisely the determination of vector \mathbf{W} . If an input–output couple is available (\mathbf{x}_1, y_1) , with the input normalized, i.e., $\|\mathbf{x}_1\| = 1$, then the best choice for the searched vector consists of $\mathbf{W} = y_1 \mathbf{x}_1$ that ensures recovering the known output, i.e.,

$$\mathbf{W}^T \mathbf{x}_1 = (y_1 \mathbf{x}_1^T) \mathbf{x}_1 = y_1. \quad (\text{A20})$$

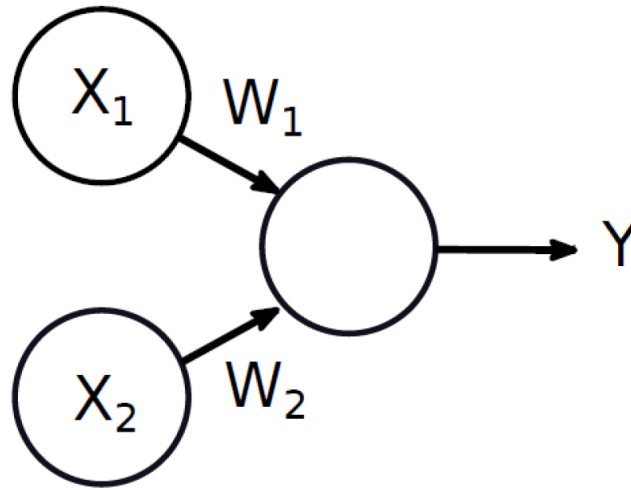


Figure A2. Sketch of a simple neuron.

Imagine now that, instead of a single input–output couple, P couples are available $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_P, y_P)$, the learning can be expressed by minimizing the functional

$$\epsilon(\mathbf{W}) = \sum_{s=1}^P (y_s - \mathbf{W}^T \mathbf{x}_s)^2. \quad (\text{A21})$$

The nonlinear case employs a nonlinear function of the predictor for the neuron activation. When the multicomponent inputs produces multicomponent outputs, \mathbf{W} becomes a matrix instead the vector previously considered. However, the procedures for computing that matrix from the knowledge of P couples $(\mathbf{x}_s, \mathbf{y}_s)$, $s = 1, \dots, P$ remain almost the same as the ones previously discussed. In some circumstances, instead of considering a single layer of neurons, multiple layers perform better.

Appendix A.6. Decision Trees

Decision tree learning is a technique that, given a data set $\mathcal{D} = \{(\mathbf{x}_i, y_i), i = 1, 2, \dots, P\}$, where \mathbf{x}_i are the so-called features and y_i the respective target variable to characterize, predicts an *outcome*, y for a new vector \mathbf{x} that can be a class to which the set \mathbf{x} belongs (in that case, we speak about *classification trees*), or a real number (in which case, we speak about *regression trees*).

In a regression tree, the terminal node j is assumed to have a constant value $y(j)$. The learning procedure consists of the top-down establishment of the variables that best partition the set according to a given criterion. One of the most popular criteria is the so-called Gini impurity criterion. For a set of M classes $m = 1, 2, \dots, M$, and p_m the fraction of elements in \mathcal{D} labeled within class m , the Gini impurity index is defined as

$$I(p) = \sum_{m=1}^M p_m \sum_{n \neq m} p_n$$

and measures the probability of an element to be chosen in its correct set times the probability of a wrong classification. In general, no matter the particular metrics employed, the goal is to classify the set so as to obtain as homogeneous as possible classification at terminal nodes.

Appendix B. Experimental Data

Table A1. Input parameters and training mask.

Training MaskIn	Screw Rotation Speed (rpm) INPUT	Exit Flow-Rate (kg/h) INPUT	Temperature (°C) INPUT	TS Dispersed Phase % INPUT	PP-g-MA/TS Ratio INPUT	Ratio TS INPUT
1	800	3	200	0	0	0.00
1	800	3	200	0	0	0.00
0	300	5	230	0	0	0.00
0	300	5	230	0	0	0.00
1	300	5	230	0	0	0.00
0	300	5	230	0	0	0.00
1	300	5	230	0	0	0.00
1	300	5	230	0	0	0.00
1	300	5	230	2	24.00	1.00
0	300	5	230	6	7.67	1.00
0	300	5	230	27	0.48	0.50
0	300	5	230	6	3.17	1.00
1	300	5	230	12	1.42	0.20
1	300	5	230	7	2.57	0.40
1	300	5	230	15	1.00	0.50
0	300	5	230	11	1.64	0.10
1	300	5	230	4	4.75	1.00
1	300	5	230	31	0.45	0.48
0	300	5	230	4	4.75	1.00
0	300	5	230	31	0.45	0.48
1	300	5	230	4	4.75	1.00
1	300	5	230	31	0.45	0.48
0	300	5	230	4	4.75	1.00
0	300	5	230	31	0.45	0.48
1	300	5	230	4	4.75	1.00
0	600	5	230	30	0.47	0.50
0	450	5	230	4	4.75	1.00
1	300	5	230	19	0.84	0.06
1	450	5	230	19	0.84	0.06
1	600	5	230	19	0.84	0.06
1	300	5	230	18	0.89	0.06
1	300	5	230	18	0.83	0.06
1	300	5	230	18	0	0.06

Table A1. Cont.

Training MaskIn	Screw Rotation Speed (rpm) INPUT	Exit Flow-Rate (kg/h) INPUT	Temperature (°C) INPUT	TS Dispersed Phase % INPUT	PP-g-MA/TS Ratio INPUT	Ratio TS INPUT
1	300	5	230	18	0.89	0.06
0	300	5	230	18	0.83	0.06
1	300	5	230	30	0.43	0.50
1	300	5	230	30	0	0.50
1	300	5	230	30	0.43	0.50
0	800	3	200	26	0.77	1.89
0	800	3	200	26	0.77	1.89
1	800	3	200	26	0.77	1.89
1	800	3	200	12	1.67	2.00
1	800	3	200	12	0.75	2.00
0	300	3	200	10	1.50	0.11
1	300	3	200	10	0.80	0.11
0	800	3	200	10	1.50	0.11
1	800	3	200	26	0.77	1.89
1	800	3	200	12	1.67	2.00
1	800	6	200	5.2	0.96	1.89
1	800	4.5	200	5.2	0.96	1.89
0	800	6	200	5.2	0.96	1.89
1	800	6	200	5	1.00	0.08
0	800	6	200	5	1.00	0.08
1	800	6	200	5	1.00	0.08
1	800	6	200	5.2	0.96	1.89
0	800	6	200	5.2	0.96	1.89
0	800	10	200	3.1	0.97	1.82
0	800	6	200	5	1.00	0.08
0	800	6	200	5	1.00	0.08

Table A2. Output quantities of interest.

Training MaskIn	E (MPa) OUTPUT	Yield (MPa) OUTPUT	σ_{break} (MPa) OUTPUT	ϵ_{break} OUTPUT	Re _{amb} (kJ/m) OUTPUT
1	1638	34	21	58	5
1	1505	37	26	164	5
0	1062	30	10	86	5.3
0	1357	21	21	10	4.9
1	1379	22	21	10	5.0
0	1281	20	13	69	7.6
1	1549	35	21	477	1.9
1	874	19	15	504	13.1
1	1345	29	14	106	5.6
0	1347	29	16	125	6.0
0	1673	30	21	118	7.4
0	1480	25	7	150	5.2
1	1657	28	12	201	6.1
1	1551	29	13	156	6.7
1	1723	30	12	155	5.8
0	1704	28	11	137	5.4
1	1365	21	19	22	4.3
1	1736	27	27	10	5.1
0	1284	22	15	136	5.6
0	1682	26	28	43	6.3
1	1416	34	24	606	2.4
1	1843	39	37	21	3.6
0	921	21	18	483	9.7
0	1371	26	24	49	12
1	1546	37	15	114	4.9
0	1753	39	29	63	5
0	1606	36	19	127	4.9
1	1655	38	16	105	3.8
1	1730	38	19	84	3.9
1	1802	38	21	54	3.3
1	1311	32	20	125	4.5
1	1181	30	18	180	5.8
1	1194	31	19	256	4.1
1	1073	24	14	155	3.7
0	1134	25	12	191	4.5

Table A2. Cont.

Training MaskIn	E (MPa) OUTPUT	Yield (MPa) OUTPUT	σ_{break} (MPa) OUTPUT	ϵ_{break} OUTPUT	Re_{amb} (kJ/m) OUTPUT
1	931	21	19	127	6.8
1	1180	25	16	63	4.4
1	1657	28	12	146	6.1
0	1795	40	37	10	5
0	2077	41	39	7	4.4
1	2273	41	39	10	3.4
1	1976	37	18	47	6.7
1	2021	36	23	44	6
0	1785	35	23	51	10.4
1	1939	37	26	38	10.1
0	1757	35	23	77	9.4
1	2033	38	23	44	5
1	2021	36	23	44	5
1	1943	37	17	46	5
1	2035	36	15	88	5
0	1897	36	26	88	5
1	1754	34	21	48	5
0	1807	35	17	49	5
1	1764	34	20	137	5
1	1067	20	17	215	5
0	2035	36	15	88	5
0	1825	37	16	51	5
0	1024	20	16	98	5
0	1807	35	17	49	5

Appendix C. Coefficients Involved in the Elastic Modulus Regression

Table A3. α_{ki}^1 coefficients.

k\i	0	1	2	3	4	5	6	7	8	9	10
0	1590.2	117.7	1.0	−1.0	−1.0	0.7	0.9	0.4	0.2	0.4	−0.3
1	0.0	171.6	0.0	0.0	0.0	−0.7	0.1	−0.6	1.0	−0.7	0.8
2	0.0	−91.9	0.0	0.0	0.0	−0.2	−0.5	0.7	0.1	0.6	−0.5

Table A4. α_{ki}^2 coefficients.

k\i	0	1	2	3	4	5	6	7	8	9	10
0	1.0	1.0	1.0	0.8	1.0	0.1	0.2	−0.6	0.5	0.0	−0.5
1	0.0	0.0	0.0	0.5	0.0	−0.7	−0.7	−0.1	−0.6	−1.0	0.0
2	0.0	0.0	0.0	0.3	0.0	0.7	0.7	0.8	0.6	0.1	−0.8

Table A5. α_{ki}^3 coefficients.

$k \backslash i$	0	1	2	3	4	5	6	7	8	9	10
0	1.0	1.0	1.0	0.4	0.7	0.6	0.7	1.0	1.0	0.0	1.0
1	0.0	0.0	0.0	0.9	0.8	0.8	0.7	0.0	0.0	1.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table A6. α_{ki}^4 coefficients.

$k \backslash i$	0	1	2	3	4	5	6	7	8	9	10
0	1.0	1.0	−0.2	1.0	0.4	1.0	1.0	1.0	1.0	1.0	1.0
1	0.0	0.0	0.5	0.0	0.9	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	−0.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table A7. α_{ki}^5 coefficients.

$k \backslash i$	0	1	2	3	4	5	6	7	8	9	10
0	1.0	1.0	1.0	1.0	−0.7	−748.1	802.3	−0.6	0.1	1.0	0.5
1	0.0	0.0	0.0	0.0	−0.2	592.3	−159.1	−0.3	0.5	0.0	0.8
2	0.0	0.0	0.0	0.0	0.7	884.7	−917.2	0.7	0.9	0.0	0.4

Table A8. α_{ki}^6 coefficients.

$k \backslash i$	0	1	2	3	4	5	6	7	8	9	10
0	1.0	1.0	152.2	104.3	66.3	1.0	1.0	235.1	−15.6	6.2	3958.4
1	0.0	0.0	57.1	51.5	−389.9	0.0	0.0	384.5	93.3	−10.4	9578.5
2	0.0	0.0	20.7	89.8	243.9	0.0	0.0	263.9	36.5	−11.5	4680.1

References

1. Chinesta, F.; Cueto, E.; Abisset-Chavanne, E.; Duval, J.L.; Khaldi, F.E. Virtual, Digital and Hybrid Twins: A New Paradigm in Data-Based Engineering and Engineered Data. *Arch. Comput. Methods Eng.* **2020**, *27*, 105–134. [\[CrossRef\]](#)
2. Ghnatios, C.; Alfaro, I.; Gonzalez, D.; Chinesta, F.; Cueto, E.E. Data-Driven GENERIC Modeling of Poroviscoelastic Materials. *Entropy* **2019**, *21*, 1165. [\[CrossRef\]](#)
3. González, D.; Chinesta, F.; Cueto, E. Thermodynamically consistent data-driven computational mechanics. *Contin. Mech. Thermodyn.* **2018**. [\[CrossRef\]](#)
4. González, D.; Chinesta, F.; Cueto, E. Learning corrections for hyperelastic models from data. *Frontiers in Materials—section Computational Materials Science. Front. Mater.* **2020**, in press.
5. Moya, B.; Gonzalez, D.; Alfaro, I.; Chinesta, F.; Cueto, E. Learning slosh dynamics by means of data. *Comput. Mech.* **2020**, in press.
6. Argerich, C.; Ibanez, R.; Barasinski, A.; Chinesta, F. Code2vect: An efficient heterogenous data classifier and nonlinear regression technique. *C. R. Mec.* **2019**, *347*, 754–761. [\[CrossRef\]](#)
7. Ibanez, R.; Abisset-Chavanne, E.; Ammar, A.; Gonzalez, D.; Cueto, E.; Huerta, A.; Duval, J.L.; Chinesta, F. A multi-dimensional data-driven sparse identification technique: The sparse Proper Generalized Decomposition. *Complexity* **2018**, 5608286.
8. Schmid, P.J. Dynamic mode decomposition of numerical and experimental data. *J. Fluid Mech.* **2010**, *656*, 5–28. [\[CrossRef\]](#)
9. Williams, M.O.; Kevrekidis, G.; Rowley, C.W. A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition. *J. Nonlinear Sci.* **2015**, *25*, 1307–1346. [\[CrossRef\]](#)
10. Reille, A.; Hascoet, N.; Ghnatios, C.; Ammar, A.; Cueto, E.; Duval, J.L.; Chinesta, F.; Keunings, R. Incremental dynamic mode decomposition: A reduced-model learner operating at the low-data limit. *C. R. Mec.* **2019**, *347*, 780–792. [\[CrossRef\]](#)

11. Aguiar, L.G.; Pessoa-Filho, P.A.; Giudic, R. Modeling of the grafting of maleic anhydride onto poly(propylene): Model considering a heterogeneous medium. *Macromol. Theory Simul.* **2011**, *20*, 837–849. [CrossRef]
12. Beyer, G.; Hopmann, C. *Reactive Extrusion: Principles and Applications*; Wiley-VCH: Hoboken, NJ, USA, 2018.
13. Cassagnau, P.; Bounor-Legaré, V.; Fenouillot, F. Reactive processing of thermoplastic polymers: A review of the fundamental aspects. *Int. Polym. Process.* **2007**, *22*, 218–258. [CrossRef]
14. Cha, J.; White, J.L. Methyl methacrylate modification of polyolefin in a batch mixer and a twin-screw extruder experiment and kinetic model. *Polym. Eng. Sci.* **2003**, *43*, 1830–1840. [CrossRef]
15. Raquez, J.M.; Narayan, R.; Dubois, P. Recent advances in reactive extrusion processing of biodegradable polymer-based compositions. *Macromol. Mater. Eng.* **2008**, *293*, 447–470. [CrossRef]
16. Cassagnau, P.; Bounor-Legaré, V.; Vergnes, B. Experimental and Modeling aspects of the reactive extrusion process. *Mech. Ind.* **2020**, *20*, 820.
17. Hernandez-Ortiz, J.C.; Steenberge, P.H.M.V.; Duchateau, J.N.E.; Toloza, C.; Schreurs, F.; Reyniers, M.F.; Marin, G.B.; Dhooze, D.R. A two-phase stochastic model to describe mass transport and kinetics during reactive processing of polyolefins. *Chem. Eng. J.* **2019**, *377*, 119980. [CrossRef]
18. Vergnes, B.; Valle, G.D.; Delamare, L. A global computer software for polymer flows in corotating twin screw extruders. *Polym. Eng. Sci.* **1998**, *38*, 1781–1792. [CrossRef]
19. Vergnes, B.; Berzin, F. Modeling of reactive systems in twin screw extrusion: Challenges and applications. *C. R. Chim.* **2006**, *9*, 1409–1418. [CrossRef]
20. Choulak, S.; Couenne, F.; Gorrec, Y.L.; Jallut, C.; Michel, P.C.A. Generic dynamic model for simulation and control of reactive extrusion. *Ind. Eng. Chem. Res.* **2004**, *43*, 7373–7382. [CrossRef]
21. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, UK, 2016.
22. Cristianini, N.; Shawe-Taylor, J. *An Introduction to Support Vector Machines: In Addition, Other Kernel-Based Learning Methods*; Cambridge University Press: New York, NY, USA, 2000.
23. Garois, N.; Sonntag, P.; Martin, G.; Vatan, M.; Drouvroy, J. Procédé de Préparation D'une Composition Thermoplastique Renforcée et Réactive, et Cette Composition. 2014. Available online: <https://patents.google.com/patent/EP2415824A1/fr> (accessed on 12 June 2020).



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).