Induced network-based transfer learning in injection molding for process modelling and optimization with artificial neural networks

Yannik Lockner¹ · Christian Hopmann¹

Received: 6 August 2020 / Accepted: 15 December 2020 / Published online: 19 January 2021
© The Author(s) 2021

Abstract

The necessity of an abundance of training data commonly hinders the broad use of machine learning in the plastics processing industry. Induced network-based transfer learning is used to reduce the necessary amount of injection molding process data for the training of an artificial neural network in order to conduct a data-driven machine parameter optimization for injection molding processes. As base learners, source models for the injection molding process of 59 different parts are fitted to process data. A different process for another part is chosen as the target process on which transfer learning is applied. The models learn the relationship between 6 machine setting parameters and the part weight as quality parameter. The considered machine parameters are the injection flow rate, holding pressure time, holding pressure, cooling time, melt temperature, and cavity wall temperature. For the right source domain, only 4 sample points of the new process need to be generated to train a model of the injection molding process with a degree of determination R^2 of 0.9 or and higher. Significant differences in the transferability of the source models can be seen between different part geometries: The source models of injection molding processes for similar parts to the part of the target process achieve the best results. The transfer learning technique has the potential to raise the relevance of AI methods for process optimization in the plastics processing industry significantly.

Keywords Injection molding · Machine learning · Transfer learning · Part similarity

1 Introduction

Customer demands for higher product quality as well as the requirement to flexibly adjust their production are increasing challenges for manufacturing companies. Especially in high-wage countries, companies are aiming for optimized process efficiency with low waste production and fast throughput times of each manufacturing order [1]. One derived focus point of these claims targets fast determination of an optimized operating point in manufacturing processes: Process engineers or operators are required to setup a production process in short time to minimize machine downtime. On the other hand, a minimum of waste production shall be produced

to identify an optimized set of machine parameters for the process. Depending on the complexity of the process and the capability of the engineer or operator, a manual optimization based on expert knowledge or trial-and-error will take up an unpredictable amount of time, generate significant waste, or will not lead to an optimized process at all [2].

In recent years, machine learning (ML) algorithms have proven their feasibility to serve industrial purposes, introducing an objective, data-driven approach to process optimization, scheduling, and failure diagnosis [3–6]. Being fitted to a defined set of process data, ML models serve as a surrogate model of the depicted process with fixed input and output parameters. In opposition to well-established linear modelling technique such as a simple regression, many ML models are able to adapt to non-linear process behavior and relationships between input and output parameters of the models. Especially artificial neural networks (ANNs) have proven to display superior results for non-linear modelling assignments.

The injection molding (IM) process is one of the most important plastics manufacturing processes and an example

✉ Yannik Lockner
yannik.lockner@ikv.rwth-aachen.de

¹ Institute for Plastics Processing in Industry and Craft at RWTH Aachen University, Seffenter Weg 201, 52074 Aachen, Germany

for a highly complex manufacturing process with non-linear process behavior [7]. Typical application fields for the products are automotive, medical, electric, and houseware. IM is a discontinuous manufacturing process during which polymer material is drawn into the barrel of a plasticizing unit by the rotation of a screw. While the material is melted by friction and external heat, introduced by band heaters around the barrel, it accumulates in front of the screw tip, the screw anteroom. During this dosing process, the screw continuously draws back in a translational movement to allow the material accumulation under a certain applied back pressure. Once enough plastic melt has been collected in the screw anteroom, the rotation stops, and the screw injects the plastic melt under high pressure through the nozzle into a tempered mold to achieve a volumetric fill of the cavity. The mold contains one or more cavities which take the melt in, shape and cool the material, and finally eject the solidified part after opening of the mold. The heat exchange is realized by a tempered coolant running in cooling channels in the mold. During the holding pressure phase, the screw keeps pressing melt into the cavity to compensate volumetric shrinkage due to the melt's pV behavior. The plasticizing unit recuperated the injected plastic melt during the cooling and ejection phase, enabling the start of a new production cycle after closure of the mold.

However, as suitable as ML methods may be for the modeling of production processes such as injection molding, a pending hindrance of a broad application is the requirement of an adequate amount of training data. Potential applicants of the technology are repelled by the necessity to generate purposely collected training data for each process that shall be modelled [8]. On the other hand, in times of "plastics industry 4.0," manufacturers can choose from a variety of solutions for a comprehensive data collection on the production floor [9]. Hence, it is likely that manufacturing enterprises will have plenty of existing process data at hand in future. Therefore, it would be desirable to reuse that amount of data from other injection molding processes in order to reduce the necessary amount of training data which needs to be generated in a conventional approach to train an ML model for a new process. In this paper, transfer learning is used to investigate if machine learning is suitable to reduce the amount of training data that is needed to train a ML model for a specific injection molding process, here the target process, when utilizing trained source models from already known processes.

The paper is organized as follows. Chapter 2 gives a rough overview over the state of art of applied ML methods for the optimization of injection molding processes. The idea of transfer learning and its adaption to the use case injection molding process setup is depicted. ANNs are used to model the injection molding processes. Chapter 3 illustrates the used artificial neural network and used hyperparameters, as well as the specimen and the methodology for the transfer learning approaches. Chapter 4 shows the transfer learning results and

analyzes their significance regarding the above described use case. Chapter 5 concludes the proposed work and gives an outlook regarding further research on the topic.

2 Injection molding and transfer learning

2.1 Artificial neural networks in injection molding

In terms of industrial manufacturing processes, the injection molding process is particularly qualified for data-driven optimization because of its high non-linear process behavior and complex relationship between machine, process, and quality parameters [10]. An accurate description of the process behavior based on physical modeling is not possible, e.g., due to the visco-elastic thermoplastic material characteristics [11].

Various researchers have investigated the possibility to implement data-driven methods to model and optimize the process regarding different quality parameters. For example, Sedighi et al. proposed a combination of a radial basis function ANN and a genetic algorithm to optimize the gate location, which served as the input parameter [12]. Therefore, the objective was the reduction of a weld line formation on the considered product before the start of production. Simulations for the injection molding process were conducted by the software Moldflow. The gate location was randomly varied on the bottom of the part to perform an effective system identification.

Bensingh et al., on the other hand, modelled several surface quality parameters such as the surface roughness or waviness for bi-aspheric lenses using ANNs. The input parameters were fill time, fill pressure, holding pressure, melt and mold temperature, as well as cooling time with several quality parameters such as the surface roughness or waviness [13]. The trained ANN was then used as a virtual model for the injection molding process during the optimization of the manufacturing conditions where Bensingh et al. compared a conventional with an adaptive particle swarm optimization (PSO) method and a genetic algorithm to exploit the trained model due to the non-invertibility of feedforward ANNs [14, 15].

Shi et al. developed an offline optimization approach for injection molding machine parameter settings, using ANNs as a process surrogate model and genetic algorithms as well [16]. The mold temperature, melt temperature, injection time, and injection pressure were used as input parameters, while the maximum shear stress served as a quality parameter in the component, as this is a main reason for possible warpage. The training database for the ANN model was formed by simulation data.

Many more researchers proposed ANN-based models of the injection molding process for a subsequent optimization of the product warpage [17–22], mechanical properties [23–25], or even a combination of several quality parameters

together in a single model [26–28]. Each of the above described research works refer to an explicitly generated database, introducing an iterative data generation process and therefore costs into the optimization. The sharing of different aspects of the modelling assignment, e.g., the database or the models itself, could reduce the initial costs for the application of data-based methods in injection molding and lead to a resilient manufacturing considering several processes [29, 30]. Transfer learning is assumed to be a feasible approach to achieve this goal.

2.2 Knowledge transfer in injection molding models

Transfer learning (TL) describes, in terms of machine learning, the transfer of knowledge from one or more source assignments A_S to a target assignment A_T [31]. Similarity between the assignments is assumed to prevent the so-called negative transfer which would ultimately reduce the model quality when performing transfer learning [32]. An assignment A consists of a domain D and a task T . A domain is represented by a defined input parameter space X and a marginal data distribution $P(X)$. A task contains a defined output parameter space Y as well as a model $f(X)$ with $f: X \rightarrow Y$. From a probabilistic point of view, $f(X)$ can also be denoted as the conditional probability $P(Y|X)$. During fitting in supervised learning, a training algorithm utilizes an amount M of labelled data samples (x_i, y_i) to train the model $f(X)$ (see Eqs. 1, 2, and 3).

$$D = \{X, P(X)\} \quad (1)$$

$$T = \{Y, f(X)\} \quad (2)$$

$$M = \{(x_i, y_i) \mid x_i \in \mathbb{R}^a, y_i \in \mathbb{R}^b, a, b, i \in \mathbb{N}\} \quad (3)$$

Torrey and Shavlik define three possible advantages of transfer learning for any new assignment as seen in Fig. 1 [33]. Firstly, in case of a very similar source domain D_S and target domain D_T , transferred models or models trained with substitutionary data from D_S could already be eligible to make

accurate predictions for the target assignment without further training (1). Secondly, few additional target assignment training data samples M_T provoke a fast adaption and fast tuning of the model to the new assignment with regard to a conventional approach without transfer learning (2). Thirdly, the shared knowledge can ultimately lead to a higher generalization ability of the resulting model, e.g., due to the enrichment of the training dataset by the combination of native (M_T) and substitutionary ($M_{S,i}$) data from other data domains $D_{S,i}$ (3).

Besides differentiating transfer learning in model-based and data-based approaches, the three categories inductive, transductive, and unsupervised transfer learning can be used as a classification. Transductive and unsupervised learning are not utilized in this study; further information on these topics can be obtained in the denoted references [34, 35].

Inductive transfer learning assumes that based on the transfer from one assignment A_S to another assignment A_T , the source and target domains D_S and D_T are equal, and the source and target tasks T_S and T_T are different regarding the given definitions. Yosinski et al. describe the fundamental process for inductive transfer in machine learning [36]. In their work, a convolutional ANN is firstly fitted to an adequate dataset size to achieve a good model quality. The model's first n layers are extracted to hybridize a new ANN with partially pretrained and partially newly initialized neuron layers. The results derive that the transfer learning success depends on the similarity of the provided pretrained data or model to the target data: If high similarity can be assumed, only fine tuning to the target task needs to be done as the fundamental feature extraction has been computed in the first n layers of the pretrained model [37]. While Yosinski et al. worked with images, other research was conducted regarding transfer learning, e.g., in the fields of handwriting recognition [38], recommendation systems for online shops [39], or natural language processing [40]. In earlier work, transfer learning has been applied by Tercan et al. for injection molding as a regression Task, improving the model's degree of determination for a small experimental training dataset. A prior training with simulation data of the same process has been conducted in order to let the ANN

Fig. 1 Possible advantages of transfer learning

adapt to fundamental relations between machine setting parameters and quality parameters [41]. For the model, injection time, holding pressure, holding pressure time, mold temperature, cooling time, and melt temperature served as input parameters. The part weight from simulation and the real experiments, respectively, served as the quality parameter. Among other things, Tercan et al. were able to see an accelerated learning adaption of the model to the provided experimental injection molding data as well as a higher generalization capability of the adapted model.

3 Data and methodology

However, the previously described transfer learning approach for injection molding remains iterative for the optimization of several processes: For every new article being produced on a machine, the necessity arises to rerun new simulations and fits a new ANN. Therefore, transfer learning between several geometrically different specimen is conducted and evaluated as a potential approach to reduce the amount of samples that need to be generated.

3.1 Specimen geometries and data generation

Sixty different toy building blocks have been designed with the software Autodesk Inventor Professional 2018 (Autodesk Inc., San Rafael, CA, USA). The geometries can be categorized by size and configuration:

- The size is depicted by the amount of studs in a row and the amount of parallel rows. The illustration of the toy building block in Fig. 2 has namely the size “4 × 2.” Further designed toy building blocks vary in the size by 1, 2, 3, 4, 6, or 8 studs and 1 or 2 rows.
- A total of 5 different configurations of the toy building blocks are generated. Based on a configuration named

Fig. 2 Exemplary dimensions of toy building block “4 × 2 original”

original, all dimensions have either been scaled up or down by the factor 3 ($\times 3$, $/3$), or the shoulder height of the toy building blocks has been doubled or halved (doubled, halved, see Fig. 3).

For each part, injection molding process simulation data is generated with the software Cadmould 3D-F Solver Version 9.0.1.0 (Simcon kunststofftechnische Software GmbH, Würselen, Germany). Seventy-seven different machine settings are sampled per part based on an a priori defined Design of Experiments (DoE). A central composite design is chosen as DoE. It consists of a 2^6 full factorial test plan together with 12 face-centered stars and the center point. In total, 4620 simulations have been conducted for all parts together. A structured DoE for the identification of the process behavior is necessary. This way, the impact of varied machine setting parameters on the chosen quality criteria of the injection molding process can be observed. The use of process data, gathered during series production, usually depicts the process only in a very limited operating point, rendering predictions for different operating points unvalidated [42]. In order to choose a practical approach for the simulations, default meshing is selected when the parts are imported into the Cadmould simulation software. The element length for the meshing for the smallest part is set to 0.376 mm, and for the largest part to 7.225 mm.

Figure 3 shows the exemplary depiction for a 3-dimensional machine parameter search space as well as the absolute values for the chosen machine parameters in the DoE. Six machine parameters are varied for the DoE: injection volume flow, melt temperature, mold temperature, holding pressure, holding pressure time, and cooling time. The part weight is considered an important quality criterion for the injection molding process and, therefore, chosen as such for the modelling assignments [43, 44]. It is determined by parsing the calculation of the part weight of the simulation after the last discrete time step of the holding pressure phase simulation. The 3D-F solver of Cadmould uses a Hele-Shaw approximation of the Navier-Stokes equations to model the flow of the polymer melt [45]. The simulations are conducted with a polypropylene PP 579S from SABIC Deutschland GmbH & Co. KG (Düsseldorf, Germany). The Carreau-WLF law is used to determine the material viscosity [46], and the Renner law serves as a model for the pVT behavior [47]. The parameters for the chosen models can be found in Tables 1 and 2. Further material properties, e.g., the specific heat capacity and thermal conductivity, can be found in Table 3. The simulations for each geometry have been performed with uniform cavity wall temperature.

Each part is molded with two gates on the short wall sides of the toy building blocks. The gates are located on one of the two symmetrical axes of each part, leading to a symmetrical

Fig. 3 DoE for simulation of different part geometries

flow front and temperature distribution during injection and holding pressure phase. Figure 4 shows several filling stages of the surface temperature distribution during the process simulation for the “4 × 2 original” toy building block.

3.2 Transfer learning: categorization and approach

In Chapter 3, inductive transfer learning was described in a detailed manner. The transfer of knowledge of different injection molding setup processes complies with requirements for this respective category regarding the following aspects:

- All data has been generated varying the same machine parameters of the injection molding process: injection volume flow, melt temperature, cavity wall temperature, holding pressure, holding pressure time, and cooling time. Therefore, $X_{S,i} = X_T$.
- The marginal probability of each data sample for every assignment is equal because it has been generated in a controlled DoE. Therefore, $P_{S,i}(X_{S,i}) = P_T(X_T)$.
- The quality parameter for all toy building blocks considered in this study is the part weight, leading to a 1-dimensional quality parameter space. Therefore, $Y_{S,i} = Y_T$.
- Each model $f(X)$ correlates the data of a different injection molding process to the respective part weights. As

different geometries of the produced parts are used, the correlation functions are assumed to vary from process to process. Therefore, $f_i(X_i) \neq f_j(X_j) \neq f_T(X_T)$ with $\{i, j \in \mathbb{N} \mid i, j \leq 60, i \neq j\}$.

One main aspect why ANNs are not widely used for the modelling of injection molding processes is the requirement of an abundant amount of available data for the training process. Torrey and Shavlik found one of the possible three main advantages of transfer learning to be the fast model adaption for a low training effort and superior model determination respective to the conventional training of an ANN [33]. The transfer learning approach for the injection molding process data has been designed accordingly: It shall be investigated how the resulting model determination of the ANN for a transfer learning approach is compared to a conventional training while providing a rising amount of samples for the training of the models.

Table 2 Renner law model parameters

Parameter	Unit	Value
CM1	m³/kg	1,24386E−3
CM2	m³/kg	1,25254E−3
CM3	1/Pa	5,90477E−10
CM4	1/Pa	1,07304E−8
CS1	m³/kg	1,07545E−3
CS2	m³/kg	1,08144E−3
CS3	1/Pa	2,59642E−10
CS4	1/Pa	5E−9
CS5	K	146
CS6	-	15
CS7	-	0
CT1	K	449,873
CT2	K/Pa	7,28147E-8

Table 1 Carreau-WLF model parameters

Parameter	Description	Unit	Value
P1	Zero shear viscosity	Pa*s	271,13478
P2	Reciprocal transition shear rate	s	1,69023E−2
P3	Ascent	-	6,03354E−1
T0	Reference temperature	°C	245
Ts	Standard temperature	°C	15,83185

Table 3 Material properties of SABIC PP 579S

Parameter	Description	Unit	Value
c_p	Specific heat capacity	J/kg°C	2740
λ	Thermal conductivity	W/m°C	0,18
α	Thermal diffusivity	mm ² /s	8,53294E-2
ρ_{Solid}	Solid density	kg/m ³	905
ρ_{Melt}	Melt density	kg/m ³	769,88

The choice of the architecture of the model as well as several hyperparameters for training can significantly influence the success of the model building. In order to ensure comparability between all source and target assignments, the used structure of the ANN was defined and fixed for all experiments. The chosen architecture and hyperparameters are adopted from previously conducted experiments in the style of a random search with the dataset of the “4 × 2 original” toy building block. In the field of injection molding, ANNs with one hidden layer and a single digit number of hidden neurons are frequently used for the modelling of the process and found to be suitable for a high model quality [48, 49]. Therefore, a model structure with one hidden layer and seven neurons in the layer is chosen. The selected hyperparameters in Table 4 resulted in a convergence of the model during training. Early stopping was used to prevent overfitting during training and to keep the training process to a timely minimum [50]. All implementations are done in python (version 3.7.5), utilizing the python modules TensorFlow (version 2.0.0), Keras (version 2.3.1), and Scikit-Learn (version 0.21.3).

It is assumed that for the injection molding assignments $A_{S,i}$, enough data has been sampled as training data for the training of the source models to achieve a good model determination after training. The assignment to learn the correlation between machine parameters and the part weight for the “4 × 2 original” domain is selected and defined as the target assignment A_T . In the experimental setup for source domain training and conventional training,

Table 4 ANN architecture and hyperparameters

Parameter	Value
Number of layers	3
Input layer neurons	6
Input layer activation	Elu
Hidden layer neurons	7
Hidden layer activation	Elu
Output layer neurons	1
Output layer activation	Linear
Regularization	L2-Norm
Initialization	He_Normal [51]
Optimizer	RMSProp [52]
Quality function for training	Mean squared error
Maximal training cycles	2000

- All $f_{S,i}(X_{S,i})$ (source-ANN) are trained with the maximum amount of data without evaluation with a test partition in order to achieve the maximum model quality for the source models.
- $f_T(X_T)$ is trained and evaluated in a conventional approach with defined steps of training data availability in order to determine the respective model quality.

Yosinski et al. tested hybridized models consisting of pretrained and newly initialized layers [36]. Accordingly, in this paper, approaches with varying pretraining degrees are chosen (cf. Fig. 5). Three different intensities of the layer transfer and reassembly as a hybridized transfer learning ANN (TL-ANN) are investigated:

1. Transferring a trained input layer, untrained hidden and output layer (IL)
2. Transferring trained input and hidden layers, untrained output layer (IL_HL)
3. Transferring the complete trained model (CM)

Fig. 4 Surface temperature profile during different stages of filling for part “4 × 2 original”

Fig. 5 Three intensities of network-based knowledge transfer

In order to evaluate the induced transfer learning results for A_T , the TL-ANN model quality is tested against the model quality of a conventional approach for the “4 × 2 original” part without a layer transfer (conv-ANN).

3.3 Data separation and conducted experiments

According to induced transfer learning, the newly created TL-ANN must be adapted to the new task T_T . Therefore, a secondary training step is conducted with some available data from M_T (rf. Eq. 4). In order to determine the model’s generalization quality for few target data samples, artificially reduced datasets DS are provided.

Equation 6 defines the generation of DS for the data. As the cardinal number, the number of samples in the dataset, of M_T is 77, sampling effects are probable to occur, especially when providing a very small amount of training data as an excerpt of M_T . Therefore, the experiments are prepared 10 times with randomly shuffled data by a pseudorandom generator. The resulting data sequences $O_{T, \beta}$ are defined in Eq. 5. Each ordered quantity is indicated by the index β . The data sequences are the discretely sliced into training and validation data $O_{T, \beta; \alpha}$ on one hand and testing data $O_{T, \beta; 1-\alpha}$ on the other hand. Nineteen experiments with a consecutively rising amount of combined training and validation data are conducted to simulate reduced amounts of training and validation data, starting with 4 samples of the available data and increasing in steps of 4 samples. The respective samples which are not used during training and validation are used for testing. Therefore, DS contains 190 different datasets of A_T .

$$M_T = \{(x_i, y_i) \mid x_i \in \mathbb{R}^6, y_i \in \mathbb{R}, i \in \{1, \dots, 77\}\} \quad (4)$$

$$O_{T, \beta} = ((x, y)_j)_{j \in \{1, \dots, 77\}} \quad \forall \beta \in \{1, \dots, 10\} \quad (5)$$

$$DS = \{(O_{T, \beta; \alpha}, O_{T, \beta; 1-\alpha}) \mid \alpha = 0.05 * \delta, \delta \in \{1, 2, \dots, 19\}, \beta \in \{1, \dots, 10\}\} \quad (6)$$

Table 5 displays the overall amount of planned model trainings and evaluations for the investigation in this paper. Within the conventional approach, a total of 1900 $f_{S, i}(X_{S, i})$ are trained as conv-ANN with initially untrained neuron layers. While the conv- and TL-ANNs are trained with a rising amount of training samples, all source models $f_{S, i}(X_{S, i})$ are trained with 80% of the totally available domain data. For the conventional and source domain trainings, as well as the transfer learning approach, each training run is repeated 10 times in order to prevent initialization effects by the *he_normal* initialization of the neuron weights per layer in the results. Also, 10 different data sequences are created by a pseudorandom number generator as explained before for the dataset generation of DS . For the source domain training, the data shuffling is conducted due to the assumption that for the corresponding 64 samples of the injection molding process, sample effects can

Table 5 Description of conducted experiments

	Conventional training	Source domain training	Transfer learning
Target domains	1	59	1
Source models	-	-	5900
Data sequences	10	10	10
Data separation	19	1	19
Initializations	10	10	10
ANN-variations	-	-	3
Σ	1900	5900	33,630,000

cause distorted models and therefore an inferior generalization capability of the ANN which is possibly unfavorable for the subsequent transfer learning. The resulting average of the model qualities is taken into consideration for the calculation and display of the results.

The resulting 5900 source-ANNs serve as base models for the transfer learning approaches IL, IL_HL, and CM. While the conv-ANN experiments are conducted for all elements in DS , the transfer learning experiments are focused on the area between 4 and 24 training data samples for this paper. In previous trials with a significantly smaller amount of experiments, the absolute differences of model determination between conv-ANN and TL-ANN were found to be significant although small and are therefore waived in the investigation here.

Regarding the scaling of all experiments, the provision of few data samples for training of the models bear the risk of distortion because of a scaling method with sample effects. However, in order to experiment as close to real processing conditions, the used *MinMaxScaler* class of the used framework Scikit-Learn for the data from A_T for the adaption training is fitted exclusively to the training data, but also applied to the testing data afterwards. This allows a realistic interpretation of the prediction capabilities of the TL-ANN for reduced data in A_T .

All model qualities are evaluated with the degree of determination R^2 according to Eq. 7:

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y}_i)^2} = \frac{SS_{ex}}{SS_{ov}} \quad (7)$$

R^2 is a frequently used performance indicator for the quality of the predictions in this field [41, 19, 53, 54, 49]. It is defined as the quotient of the explained sum of squares SS_{ex} and the overall sum of squares SS_{ov} . It is computed with the true value of the quality criterion y_i , the average of the true values $\bar{y}_i$, and the predictions $\hat{y}_i$ per sample made by the

model. The possible value range for R^2 in linear models such as linear regressions is usually defined as $R^2 \in [0; 1]$, with 0 indicating the prediction of the dataset mean for all samples and 1 signifying a perfect adaption when tested with an unknown set of samples. Particularly, for non-linear models and datasets, the metric can drop below 0 as an unsuitable model can introduce more variation into the prediction as can be observed in the true values. R^2 is independent of the scaling of the data, in contrast to further performance indicators such as root mean square error (RMSE) or mean square error (MSE).

4 Results

4.1 Comparison of TL intensities

The overall transfer learning result is depicted in Fig. 6. The diagram shows the model's R^2 in relation to the provided cardinal number of training data M_T for adaption which has been taken from the target domain dataset of the “4 × 2 original” toy building block. The display of the R^2 results for the cardinal number 76 is omitted as for only one sample in the test partition; R^2 cannot be computed. The data series (curves) represent the conv-ANN approach without transfer learning and the three transfer intensities IL, IL_HL, and CM. The calculated standard deviations for each approach are shown by the bars with values on the secondary axis. Each conv-ANN series point represents the average model determination of 100 ANNs, whereas each transfer learning series point is averaged by 590.000 ANNs.

Regarding the conv-ANN results, the model architecture and the chosen hyperparameters prove to be suitable for the target assignment A_T . The model achieves a model determination value R^2 of 0.973 for 60 provided training data elements in M_T . With increasing training data, the R^2 value does not fall behind 0.963. This confirms an appropriate choice of

Fig. 6 Results for all domains of all distributions together

hyperparameters for the modelling task. However, for small amounts of training data, the conv-ANN approach produces high standard deviations in the model quality (rf. Table 6): Even with 16 training data elements, the model determination R^2 can still be below 0, indicating an insufficient adaption to the target domain T_T . Therefore, considering the bespoke use case of process setup in injection molding, practitioners of this technique should not employ ANNs as a model building technique if less than 20 data samples for the training are available. The ANN has a chance to perform worse than a linear mean estimator with its degree of determination dropping below 0 (rf. Table 6, Conv, MIN). This emphasizes the necessary effort for data generation when using ANNs in a conventional approach. Interestingly, with the provision of 40 samples for training, the average model quality sinks and the standard deviation has a significant peak. This outlier needs to be compared to the results of the transfer learning approaches once data is available here.

The result is expected as the models retrieve more information about the relationship between machine parameters and the quality parameter part weight once more samples are provided. This implies that information about the injection molding process in general is stored in any of the neuron layers of the models trained on any of the source domains D_S . Equally important to the averaged model quality difference between conventional and all three transfer learning approaches is the stability improvement of the model quality by transfer learning. A practitioner using this modeling technique is therefore less prone to receive a trained model which underperforms significantly regarding the average. Table 3 indicates that for the CM approach, the maximal deviation from the average model quality is 0.0058, for the conventional approach respectively 0.584 for 20 provided training data samples of M_T .

Differences between the TL approaches can be seen in Fig. 6 and Table 6 as well. For the observed cardinal numbers of M_T , the CM approach outperforms the other transfer learning

approaches by the averaged model quality. Especially significant is the quality difference on average for very small training datasets like 4. This indicates that the transfer of the whole model in general is a feasible approach when trying to achieve the highest model quality. However, the CM approach shows the highest standard deviation of the three TL methods. For example, for the provision of 8 training samples, the IL or IL_HL approach could result in a higher model quality for a specific model instance. Interestingly, the averaged model quality sinks again for the conventional, IL, and IL_HL approach for 12 and 16 training data samples, while the CM model quality rises monotonously. This behavior is yet inexplicable and will need careful evaluation which is out of this paper's scope. Regardless of the specific approach, network-based inductive transfer learning appears to be beneficial for the model quality for this use case in injection molding.

4.2 Resulting differences by provided source models

However, as indicated by the standard deviations in Fig. 6, the model qualities vary significantly for the same cardinal number of M_T . One possible explanation lies in the differences of the source models, induced by the different geometrical characteristics of a mold cavity, which influence the process behavior: The longer the maximal flow path, the higher the injection pressure has to be in order to overcome the pressure loss by the mold. This, however, only applies for comparable wall thicknesses. Furthermore, thermal inhomogeneities are often unavoidable as the rising complexity of injection molding parts often requires rapid changes of wall thickness of the part. Due to the variances of injection molding processes introduced by different geometries, the conditional probability $P_{S_i}(Y|X)$ varies from process to process. Therefore, it is assumed that the transfer of a model f_{S_i} to the target task T_T is more successful if the processes are similar to each other, including the cavity geometries.

Table 6 R^2 results for TL intensities and conv-ANN for DS for $\alpha \in [4; 24]$

Cardinal number of M_T	Conv				CM				IL_HL	IL
	AVG	STD	MAX	MIN	AVG	STD	MAX	MIN	AVG	AVG
4	− 1.613	0.701	− 0.425	− 2.587	0.6335	0.3486	0.8755	0.0383	− 0.676	− 0.676
8	0.094	0.797	0.773	− 1.458	0.7458	0.2170	0.8678	0.2595	0.752	0.759
12	− 0.364	1.643	0.839	− 4.649	0.7731	0.2412	0.9044	0.2334	0.498	0.767
16	0.092	0.304	0.378	− 0.580	0.8898	0.0056	0.9007	0.8820	0.380	0.626
20	0.732	0.286	0.896	0.148	0.8948	0.0034	0.8982	0.8891	0.889	0.889
24	0.892	0.002	0.894	0.890	0.9044	0.0098	0.9139	0.8924	0.888	0.889

The results for the transfer learning approaches CM, IL, and IL_HL show an improvement of the model determination on average, based on the testing results for unseen data for a rising amount of training data from A_T for all data series. Up to the cardinal number 20, the IL, IL_HL, and CM approaches result in better degrees of determination than the conventional approach. For 24 provided training data points, the CM model still performs better. Additionally, the standard deviations of all three TL approaches are significantly smaller than the conventional approach up to the provision of 20 training data samples.

Fig. 7 Results of the CM approach for three different source domains

Figure 7 depicts the transfer learning results for the CM approach for three different source domains: “ 4×1 original,” “ $8 \times 2 \times 3$ ” and “ 8×2 original.” While the results for the “ 4×1 original” and “ 8×2 original” source models are at the same level, for the provision of only 4 data samples, the retrained source models of the “ 4×1 original” and “ 8×2 original” show high values for R^2 of 0.89 and 0.92, respectively. Significant differences are visible for the “ $8 \times 2 \times 3$ ” source model: For small training datasets in the transfer learning approach, the model based on this source model performs considerably worse on average. Furthermore, the standard deviation for 4 and 8 provided training data samples appears to be significantly higher than for the other source models.

An initial analysis of the geometrical differences as influence on the process data can be conducted by comparing the geometrical characteristics flow path length and average wall thickness as mentioned above. The flow path for the “ 4×2 original” toy building block is 108.3 mm long; the average wall thickness is 1.97 mm. While the “ 4×1 original” toy building block has a difference of 9.9 mm, the “ 8×2 original” toy building block shows a prolongation of 62.8 mm. This difference, however, is not reflected in the results of the transfer learning approaches. The “ 4×1 original” and “ 8×2 original” toy building block have the same average wall thickness of 1.97 mm as the part of the target domain, while the “ $8 \times 2 \times 3$ ” toy building block shows an average wall thickness of 5.90 mm. In comparison with Fig. 7, the wall thickness appears to have a much higher influence on the transferability of a source model. Regarding the results, the transfer of source models can be especially successful for small amounts of training data samples from the target assignment, if the average wall thickness of the manufactured product in the injection molding process is similar or equal to the one of the source domain. This is an especially encouraging observation as occurring product variations commonly do not change in average wall thickness.

5 Conclusion and outlook

The extensive data need for the modeling of the relationship between machine setting parameters and quality parameters in injection molding is one of the main reasons why artificial neural networks are not yet broadly in use for process optimization. In this paper, different approaches of transfer learning with ANNs between different injection molding processes have been examined with the goal to reduce the necessary amount of data for model training. It has been determined that any applied transfer intensity for network-based induced transfer learning has a beneficial effect on the resulting model’s degree of determination and, hence, should be used to enhance the model quality for small available datasets. The transfer of a complete model with a subsequent retraining has been found to outperform all other approaches, especially no transfer at all as seen for the conventional approach. With 16 training samples from the target domain, an R^2 value of 0.88 could be achieved on average. The conventional approach surpasses these results only with 24 training samples, which indicate a reduction of 33% necessary training data. An even higher reduction of training samples as a result can be seen when choosing a similar source domain: The model determination achieved using the 8×2 original models for a CM approach with 4 training data samples is only surpassed by the conventional training of an ANN when using at least 32 training samples. In comparison, this refers to a reduction of training data of 88%.

However, the similarity of the injection molding processes cannot yet be determined regarding the transfer learning success. Even though the average wall thickness appears to be a relevant factor for the possible transfer learning success, further investigation has to be done: More geometrical parameters of the parts, the cavity and the mold have to be examined for correlations with the transfer learning success in order to receive an estimation if a source model is suitable for a specific target assignment. Other quality criteria such as warpage are expected to behave highly non-linear and will introduce

further complexity into the system. The success of transfer learning between different, practically relevant quality criteria in injection molding has to be compared in additional experiments. Especially for transfer learning between part variations, it needs to be investigated, if the observations for transfer between parts with equal wall thickness can be confirmed. Furthermore, these results based on simulation data have to be validated for real experimental data. The impact of noise in experimental data, introduced by varying environmental conditions, material fluctuations, or inaccurate machine movements, needs to be evaluated regarding the transfer learning results. A real-world case study will be conducted based on the presented results and transfer learning strategy. Finally, material and machine influences could be equally or more relevant factors on the transferability of source models to a target assignment. Experiments need to be designed, in accordance to the test series with geometrically different parts, to identify and measure their impact.

Authors' contributions All authors confirm that the information provided is accurate. In addition, the substantial contribution to the depicted research by all authors is confirmed.

Funding Open Access funding enabled and organized by Projekt DEAL. This research was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under the Germany's Excellence Strategy – EXC-2023 Internet of Production – 390621612. The support and sponsorship is gratefully acknowledged and appreciated.

Data availability Use the following link to get to the repository with public access: https://gitlab.com/yannicklockner/dml_1.

Compliance with ethical standards

Ethical approval Not applicable.

Consent to participate Not applicable.

Consent to publish Not applicable.

Abbreviations ML, Machine learning; ANN, Artificial neural network; IM, Injection molding; conv-ANN, ANN training with the conventional approach; source-ANN, ANN trained in the source domain training; TL-ANN, ANN trained with a transfer learning approach; A_S , Source assignment; $A_{S,i}$, i -th source assignment; A_T , Target assignment; D , Domain; D_S , Source domain; D_T , Target domain; DS , Datasets of M_T for conv-ANN and TL-ANNs; T , Task; T_S , Source task; T_T , Target task; X , Input parameter space; $X_{S,i}$, i -th input parameter space; X_T , Target input parameter space; $P(X)$, Marginal data distribution; $P_{S,i}(X_{S,i})$, Marginal data distribution of the i -th source input parameter space; $P_T(X_T)$, Marginal data distribution of the target input parameter space; Y , Output parameter space; $Y_{S,i}$, Output parameter space of the i -th source domain; Y_T , Output parameter space of the target domain; $P(Y|X)$, Conditional probability of Y under X ; $f(X)$, Model or estimator; $f_i(X_i)$, Model of i -th domain; $f_{S,i}(X_{S,i})$, Model of the i -th source domain; $f_T(X_T)$, Model of the target task; M , Quantity of available labeled samples; $M_{S,i}$, Quantity of labeled samples in source domain i ; M_T , Quantity of labeled samples in the target domain; $O_{T,\beta}$, Tuple of data from M_T with pseudorandom factor β ; $O_{T,\beta,\alpha}$, Sliced tuple with the anterior α percent of the samples; $O_{T,\beta,1-\alpha}$, Sliced tuple

with the posterior $1 - \alpha$ percent of the samples; x_i , Input parameter vector of sample i ; y_i , Output parameter vector of sample i

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Brecher C, Jeschke S, Schuh G, Aghassi S, Amoscht J, Bauhoff F, Fuchs S, Jooß C, Karmann O, Kozielski S, Orilski S, Richert A, Roderburg A, Schiffer M, Schubert J, Stiller S, Tönissen S, Welter F (2011) Integrative Produktionstechnik für Hochlohnländer. In: Brecher C (ed) Integrative Produktionstechnik für Hochlohnländer. Springer Verlag, Berlin
2. Meiabadi MS, Vafaeseefat A, Sharifi F (2013) Optimization of plastic injection molding process by combination of artificial neural network and genetic algorithm. J Optim Ind Eng 6(13):49–54
3. Ademujimi TT, Brundage MP, Prabhu VV (2017) A review of current machine learning techniques used in manufacturing diagnosis. In: Lödding H, Riedel R, Thoben K-D, von Cieminski G, Kiritsis D (eds) Advances in Production Management Systems. The Path to Intelligent, Collaborative and Sustainable Manufacturing. APMS 2017. IFIP Advances in Information and Communication Technology, 513th edn. Springer International Publishing, Cham, pp 407–415
4. Weichert D, Link P, Stoll A, Rüping S, Ihlenfeldt S, Wrobel S (2019) A review of machine learning for the optimization of production processes. Int J Adv Manuf Technol 104:1889–1902. <https://doi.org/10.1007/s00170-019-03988-5>
5. Kim D-H, Kim TJY, Wang X, Kim M, Quan Y-J, Oh JW, Min S-H, Kim H, Bhandari B, Yang I, Ahn S-H (2018) Smart machining process using machine learning: a review and perspective on machining industry. International Journal of Precision Engineering and Manufacturing-Green Technology 5:555–568. <https://doi.org/10.1007/s40684-018-0057-y>
6. Fazel Zarandi MH, Sadat Asl AA, Sotudian S, Castillo O (2018) A state of the art review of intelligent scheduling. Artif Intell Rev 53: 501–593. <https://doi.org/10.1007/s10462-018-9667-6>
7. Shen C, Wang L, Li Q (2007) Optimization of injection molding process parameters using combination of artificial neural network and genetic algorithm method. J Mater Process Technol 183(2-3): 412–418. <https://doi.org/10.1016/j.jmatprotec.2006.10.036>
8. Bourdon R, Hellmann A, Schreckenber J-B, Schwegmann R (2012) Standardisierte Prozess- und Qualitätsoptimierung mit DOE-Methoden - eine Kurzanleitung für die Praxis beim Spritzgießen. Zeitschrift Kunststofftechnik / Journal of Plastics Technology 8:525–549
9. Giordano G (2019) Buying power. Plast Eng 75(1):28–35. <https://doi.org/10.1002/peng.20056>
10. Rosato DV, Rosato MG (2012) Injection Molding Handbook (trans: 10.1007/978-1-4615-4597-2), 2nd edn. Springer US, New York

11. Popov VL, Heß M, Willert E (2019) Viscoelastic materials. In: Handbook of Contact Mechanics. Springer Verlag, Berlin, pp 213–249
12. Sedighi R, Meiabadi MS, Sedighi M (2017) Optimisation of gate location based on weld line in plastic injection moulding using computer-aided engineering, artificial neural network, and genetic algorithm. *Int J Automot Mech Eng* 14(3):4419–4431. <https://doi.org/10.15282/ijame.14.3.2017.3.0350>
13. Bensingh RJ, Machavaram R, Boopathy SR, Jebaraj C (2019) Injection molding process optimization of a bi-aspheric lens using hybrid artificial neural networks (ANNs) and particle swarm optimization (PSO). *Measurement* 134:359–374. <https://doi.org/10.1016/j.measurement.2018.10.066>
14. Ardizzone L, Kruse J, Wirkert S, Rahner D, Pellegrini EW, Klessen RS, Maier-Hein L, Rother C, Köthe U (2019) Analyzing inverse problems with invertible neural networks. In: 7th International Conference on Learning Representations, New Orleans, USA, New Orleans, 6–9 May 2019
15. Osborne CMJ, Armstrong JA, Fletcher L (2019) RADYNVERSION: learning to invert a solar flare atmosphere with invertible neural networks. *Astrophys J* 873(2):128–141. <https://doi.org/10.3847/1538-4357/ab07b4>
16. Shi F, Lou ZL, Lu JG, Zhang YQ (2003) Optimisation of plastic injection moulding process with soft computing. *Int J Adv Manuf Technol* 21:656–661
17. Patel GCM, Krishna P (2012) Prediction and optimization of dimensional shrinkage variations in injection molded parts using forward and reverse mapping of artificial neural networks. *Adv Mater Res* 463–464:674–678. <https://doi.org/10.4028/www.scientific.net/AMR.463-464.674>
18. Iniesta AA, Alcaraz JLG, Borbón MIR (2013) Optimization of injection molding process parameters by a hybrid of artificial neural network and artificial bee colony algorithm. *Revista Facultad de Ingeniería Universidad de Antioquia* 67:43–51
19. Tsai K-M, Luo H-J (2014) An inverse model for injection molding of optical lens using artificial neural network coupled with genetic algorithm. *J Intell Manuf* 28(2):473–487. <https://doi.org/10.1007/s10845-014-0999-z>
20. Zhang J, Wang J, Lin J, Guo Q, Chen K, Ma L (2016) Multiobjective optimization of injection molding process parameters based on Opt LHD, EBFNN, and MOPSO. *Int J Adv Manuf Technol* 85(3):2857–2872. <https://doi.org/10.1007/s00170-015-8100-4>
21. Nagorny P, Pillet M, Pairel E, Le Goff R, Loureaux J, Wali M, Kiener P (2017) Quality prediction in injection molding. In: 2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications, Annecy, Frankreich, Annecy. <https://doi.org/10.1109/CIVEMSA.2017.7995316>
22. Jain ESM, Bhuyan RK (2019) Simulation and optimization of warp of fiber reinforced using human behavior based optimization. *Int J Innov Technol Explor Eng (IJITEE)* 8(10):296–302. <https://doi.org/10.35940/ijitee.I8187.0881019>
23. Kenig S, Ben-David A, Omer M, Sadeh A (2001) Control of properties in injection molding by neural networks. *Eng Appl Artif Intell* 14(6):819–823. [https://doi.org/10.1016/S0952-1976\(02\)00006-4](https://doi.org/10.1016/S0952-1976(02)00006-4)
24. Chen W-C, Wang M-W, Chen C-T, Fu G-L (2009) An integrated parameter optimization system for MISO plastic injection molding. *Int J Adv Manuf Technol* 44:501–511. <https://doi.org/10.1007/s00170-008-1843-4>
25. Trovalusci F, Ucciardello N, Baiocco G, Tagliaferri F (2019) Neural network approach to quality monitoring of injection molding of photoluminescent polymers. *Appl Phys A Mater Sci Process* 125(11):781–787. <https://doi.org/10.1007/s00339-019-3067-x>
26. Yarlagadda PKDV (2001) Prediction of processing parameters for injection moulding by using a hybrid neural network. *Proc Inst Mech Eng* 215(10):1465–1470. <https://doi.org/10.1243/0954405011519097>
27. Spina R (2006) Optimisation of injection moulded parts by using ANN-PSO approach. *J Achiev Mater Manuf* 15(1-2):146–152
28. Lee H, Liao Y, Ryu K (2017) Real-time parameter optimization based on neural network for smart injection molding. *IOP Conf Series: Materials Science and Engineering* 324:012076. <https://doi.org/10.1088/1757-899X/324/1/012076>
29. Han B, Lie CL, Zhang WJ (2016) A method to measure the resilience of algorithm for operation management. *IFAC-PapersOnLine* 49(12):1442–1447. <https://doi.org/10.1016/j.ifacol.2016.07.774>
30. Zhang WJ, van Luttervelt CA (2011) Toward a resilient manufacturing system. *CIRP Ann Manuf Technol* 60:469–472. <https://doi.org/10.1016/j.cirp.2011.03.041>
31. Weiss K, Khoshgoftaar TM, Wang D (2016) A survey of transfer learning. *J Big Data* 3(9):1–40. <https://doi.org/10.1186/s40537-016-0043-6>
32. Rosenstein MT, Marx Z, Kaelbling LP, Dietterich TG (2005) To transfer or not to transfer. In: Inductive Transfer: 10 Years Later - NIPS 2005 Workshop, Whistler, Canada, Whistler, 9 December 2005
33. Torrey L, Shavlik J (2009) Transfer Learning. In: Olivas ES, Guerrero JDM, Sober MM, Benedito J, Magdalena R, Lopez AJ (eds) Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods and Techniques. IGI Global, Hershey, pp 242–264
34. Pan SJ, Yang Q (2010) A survey on transfer learning. *IEEE Trans Knowl Data Eng* 22:1345–1359. <https://doi.org/10.1109/TKDE.2009.191>
35. Zhao P, Hoi SCH, Wang J, Li B (2014) Online transfer learning. *Artif Intell* 216:76–102. <https://doi.org/10.1016/j.artint.2014.06.003>
36. Yosinski J, Clune J, Bengio Y, Lipson H (2014) How transferable are features in deep neural networks? In: NIPS'14 Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, Montreal, Canada, Montreal, pp 3320–3328
37. Bengio Y (2011) Deep learning of representations for unsupervised and transfer learning. *Journal of Machine Learning Research Workshop and Conference Proceedings* 7:1–20
38. Cireşan DC, Meier U, Schmidhuber J (2012) Transfer Learning for Latin and Chinese characters with deep neural networks. In: IEEE World Congress on Computational Intelligence, Brisbane, Australien, Brisbane. <https://doi.org/10.1109/IJCNN.2012.6252544>
39. Li B, Yang Q, Xue X (2009) Transfer learning for collaborative filtering via a rating-matrix generative model. In: 26th International Conference on Machine Learning, Montreal, Kanada, Montreal. <https://doi.org/10.1145/1553374.1553454>
40. Collobert R, Weston JA (2008) Unified architecture for natural language processing: deep neural networks with multitask learning. In: 25th International Conference on Machine Learning, Helsinki, Finland, Helsinki. <https://doi.org/10.1145/1390156.1390177>
41. Tercan H, Guajardo A, Heinisch J, Thiele T, Hopmann C, Meisen T (2018) Transfer-learning: bridging the gap between real and simulation data for machine learning in injection molding. *Procedia CIRP* 72(1):185–190. <https://doi.org/10.1016/j.procir.2018.03.087>
42. Isermann R (1992) Identifikation dynamischer Systeme 1, 2nd edn. Springer-Verlag, Berlin
43. Haman S (2004) Prozessnahes Qualitätsmanagement beim Spritzgießen. Technische Universität Chemnitz
44. Yang Y, Gao F (2006) Injection molding product weight: online prediction and control based on a nonlinear principal component regression model. *Polym Eng Sci* 46(4):540–548. <https://doi.org/10.1002/pen.20522>

45. Simulation of fluid flow and structural analysis within thin walled three dimensional geometries (28.01.2004). European Patent Office Patent, patent no. EP 1 385 103 B1
46. Osswald TA, Rudolph N (2015) Generalized Newtonian Fluid (GNF) Models. In: Osswald TA (ed) Polymer Rheology. Hanser Verlag, München, pp 59–99
47. Cadmould 3D-F User Manual. simcon kunststofftechnische Software GmbH.
48. Hopmann C, Heinisch J, Tercan H (2018) Injection moulding setup by means of machine learning based on simulation and experimental data. In: ANTEC 2018 - The Plastics Technology Conference, Orlando, Florida, USA, Orlando, 7–10 May 2018
49. Hopmann C, Bibow P, Kosthorst T, Lockner Y (2020) Process setup in injection moulding by human-machine-interfaces and AI. In: 30th International Colloquium Plastics Technology 2020, Aachen, Germany, Aachen, 8–11 September 2020
50. Prechelt L (1998) Early stopping - but when? In: Orr G, Müller K-R (eds) Neural Networks: Tricks of the Trade. Springer-Verlag, Berlin, pp 53–67
51. He K, Zhang X, Ren S, Sun J (2015) Delving deep into rectifiers: surpassing human-level performance on ImageNet Classification. In: IEEE International Conference on Computer Vision, Santiago, Chile, Santiago. <https://doi.org/10.1109/ICCV.2015.123>
52. Duchi J, Hazan E, Singer Y (2011) Adaptive subgradient methods for online learning and stochastic optimization. J Mach Learn Res 12:2121–2159. <https://doi.org/10.5555/1953048.2021068>
53. Tsai K-M, Luo H-J (2015) Comparison of injection molding process windows for plastic lens established by artificial neural network and response surface methodology. Int J Adv Manuf Technol 77:1599–1611. <https://doi.org/10.1007/s00170-014-6366-6>
54. Arulsudar N, Subramanian N, Murthy R (2005) Comparison of artificial neural network and multiple linear regression in the optimization of formulation parameters of leuprolide acetate loaded liposomes. J Pharm Sci 8(2):243–258

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.