Process Modeling and Parameter Optimization for Machine Calibration in Smart Manufacturing for Laser Seam Welding

João Reis, Gil Gonçalves

Research Center for Systems & Technologies (SYSTEC)
Faculty of Engineering University of Porto
Porto, Portugal
Email: {jpcreis, gil}@fe.up.pt

***Abstract*—One of the main challenges towards a smart factory is the automation of processes and inclusion of personnel experience in those systems. One of these challenges is related to advances in artificial intelligence that have already been proven to be effective in solving real world problems in the last decade. The problem addressed in this paper is finding the most suitable machine parameters of a laser seam welding process. Once new quality requirements are defined by the customer, normally, a machine calibration phase is required in order to find the proper parameters that yield the desired quality of the product. To address this problem, first a modeling phase was performed to create a suitable model using Artificial Neural Networks (ANNs) that map process parameters onto the observed product quality, and second, the Basin-Hopping search algorithm was used to find the machine parameters needed to achieve a target quality. In order to demonstrate the robustness of the presented approach, three datasets were used that represent three different pairs of materials used for welding in the same machine. The results demonstrate that ANNs are a flexible and robust technique to be used in industry for process modeling and the calibration phase can be minimized.**

Keywords–Process Modeling; Process Parameter Optimization; Artificial Neural Networks; Smart Manufacturing; Machine Learning.

I. INTRODUCTION

The increasing number of product variations as a result of Mass Production to Mass Customization paradigm shift [1] has been leading to the necessity of knowing in detail the machine process dynamics. This is due to the quick change between product variations being produced in a small-lot fashion, or to the introduction of new machines in the shop-floor. This happens mainly because manufacturing companies are getting closer and closer to the end-customer, allowing for customized products composed of multiple options and combinations, and consequently leading to a high number of product variations. This forces the manufacturing companies to be much more responsive to the market needs as a way to increase their market share and create new competitive advantages. However, in order to achieve this level of competitiveness, smarter and innovative ways to explore equipment capabilities and reconfiguration are required. Given the machine operations heterogeneity and shorter production cycles, there is a demand for new techniques that intelligently can operate machinery according to new and diverse product requirements, and rapidly respond and react to these requirement changes, ultimately leading to the automation of the manufacturing process.

Normally, the operation of a certain machine is guided by a set of process parameters that influence process quality that dictate the final result of a certain product. In order to achieve that, the correct process parameters need to be chosen that would yield the correct process quality subject to a set of process conditions. Hence, there is an implicit relation between the influence of machine parameters in the final quality of the product. This way, a good understanding of how process parameters influence the process quality is peremptory for process automation. Normally, the exploration of these relations is made by a set of experiments by performing a Design of Experiment (DoE) - Full Factorial Design or Fractional Factorial Design - to know how of the process parameters map into the process quality. From these experimental findings, normally a dataset is built and machine learning techniques can be used to build process models, which is a simplified version of the real world dynamics - also known as surrogate model. However, as referred before, for the selection of the most suitable process parameters according to certain process quality, this model is necessary but not sufficient. Additionally, an optimization problem is normally formulated to explore the machine parameter feature space that minimizes the distance between the desired process quality and the ones yielded by the process model.

Such an approach is being widely used as a way to perform process optimization as presented in several works reported in the literature. Some examples of such works are [2] and [3] where they use an ANN to model the process using experimental data, and use the concept of Inverse ANN to optimize, using Nelder-Mead algorithm, the process parameters for COD removal in the aqueous treatment of alazine and for energy processes, correspondingly. Another example is presented in [4] where the authors used an ANN to model a thermoplastic joining process and use Genetic Algorithms to find the most suitable process parameters for joining. Moreover, [5] compared Symbolic Regression via Genetic Algorithms with ANN on the modeling and optimization of a controlled drug release of pharmaceutical formulation. For a more thorough understanding of the subject, [6] presents a good review of the High-Dimensional, Expensive (computationally) and Black-box (HEB) problems, presenting multiple examples on a variety of disciplines.

The rest of the paper is organized as follows. Section II details the laser seam welding manufacturing process where this research is focused. Sections III and IV explain how the process modeling and process parameter optimization was performed in this context, leading to Section V where the main

ISBN: 978-1-61208-646-0

TABLE I. PROCESS CONDITIONS, PROCESS PARAMETERS, PROCESS QUALITY AND NUMBER OF EXPERIMENTS

	PROCESS 1	PROCESS 2	PROCESS 3
UPPER THICKNESS	1.5	0.6	1.2
LOWER THICKNESS	1.5	1.2	1.5
P (KW)	4676.2±666.8	4408.3±742.1	4594.6±702.6
F (MM)	-0.6±12.6	-0.3±14.0	-0.1±12.4
V (MM/S)	104.8±26.5	154.1±32.0	120.9±28.8
D (MM)	0.6±0.3	0.6±0.2	0.6±0.3
W (MM)	0.9±0.1	1.0±0.1	0.9±0.1
EXPERIMENTS	188	260	220

results are depicted and discussed. Finally, Section VI draws some conclusions about the performed work.

II. LASER SEAM WELDING SCENARIO

To better understand the presented scenario of process parameter optimization, a description of the process will be given. The laser seam welding process is composed of laser head mounted in a robotic arm with the goal of welding two metal sheets by issuing radiation from the laser head to a local area where the materials need to be joined. Thus, it creates a melting zone around the laser focus in both sheets, which solidifies once the the laser beam is moved through the desired welding area. This produces a continuous welding seam while the beam is moved along the overlapping sheets at a controlled speed. In this particular scenario, the process parameters that can be changed are described by 3 independent variables: Laser Head Power (P); Focal Distance (F) from the surface and Robotic Arm Velocity (V). The observed process quality is described by the Weld Width (W) and Penetration Depth (D) of the welded area. For this work, 3 different datasets are used representing 3 different welding processes in the same machine, where different pairs of materials with different properties and thicknesses were used. These pairs are namely DC04-HC380LA (Process 1), HC260LA-HC420LA (Process 2) and HC420LA-HC380LA (Process 3). Table I presents a summary of the 3 datasets used. If the influence of process parameters over process objectives is explicit in a dataset, machine learning techniques can be used to model this relation, building up a process model.

On top of this information, the process conditions define in which context the process model is valid. For example, if the a process model is trained using the process parameters and quality of two metal sheets, both with 1.5mm of thickness as in Process 1, such a process model becomes obsolete if these thicknesses change, mainly because the relation between process parameters and quality also change. In this context, different thicknesses represent different product variations. As a consequence of that, if a new product variation is introduced in the manufacturing process, this process parameter and quality relation needs to be discovered and detailed as a dataset, so the proper techniques can be used for Process Parameter Optimization. If one wants to explore the relation between already known processes and the conditions that describe the new unseen processes, different machine learning techniques must be applied. Transfer Learning is an emerging research area that is yielding good results in multiple domains, and can be applied to solve the presented problem of learning a new process of a new product variation. In the recent years, the Hyper-Model approach is also being applied to the manufacturing context, which is named as Hyper-Process Modeling [7], to deal with such an issue. However, the details of how these techniques operate are out of the scope of this paper. In the next sections we will present the approach for modeling and optimization in the presented scenario of laser seam welding.

III. PROCESS MODELING

Since we are modeling a predictor for continuous variables, the presented problem is classified as regression. Hence, the well known Multi-Layer Perceptron (MLP) was used to model an ANN to map machine parameters onto the observed quality data for the laser seam welding process. The concept of artificial neuron is a generalization and simplification of the biological neuron, which is nothing more than a mathematical representation of information processing [8]. This way, the same principle observed in biological systems is then used in the concept of ANNs, where multiple layers of neurons are stacked and connected to perform pattern recognition and predictions. This results in feedforward ANN that proved already of great practical value in solving difficult and specific real life problems.

As its name indicates, for the MLP there are multiple layers of fully connected neurons, meaning that all the neurons of a layer are connected to each neuron of the subsequent layer. These connections are often called weights and dictate how much significance a neuron has to one another. The first layer is called the input layer, the last layer is called output layer, and the remaining in between are called hidden layers. This means that we should have at least three layers to have an MLP, and multiple topologies since these networks can grow by number of hidden layers and number of neurons by hidden layer. Normally, the input and output layers are fixed and correspond to the number of features used for the prediction (independent variables) and number of features that compose the prediction (dependent variables). Based on this, the input of each neuron is composed by the sum of the output of M neurons from the precedent layer and the corresponding weight, and is represented as follows:

$$a_j = \sum_{i=0}^{D} w_{ji}^{(n)} x_i \tag{1}$$

where j is the corresponding layer, D is the number neurons connected to the subsequent layer j plus 1 considering the bias, w is the weight of the corresponding neuron, n is the current layer and finally x is the output of the corresponding neuron. The values of the variable w are called the model parameters. Based on this, the input of a neuron in a subsequent layer can be calculated based on each neuron output (x) of the current and its influence (w). However, this is simply a linear transformation of data, and no nonlinear dynamics of the system can be grasped. Hence, the calculated input normally is transformed using a nonlinear activation function $h(.)$. This dictates the final form of a neuron output based on the neurons in the previous layer:

$$z_j = h(a_j) \tag{2}$$

Normally, the chosen nonlinear functions are sigmoid or hyperbolic tangent.

Based on this, we have trained our ANN with Adaptive Subgradient Methods for weight optimization [9]. P, F and V specify the inputs feature space X and D and W define the output feature space Y, leading to 3 neurons for the input layer and 3 neurons for the output layer. All the neurons from both input and output layers have a linear activation function, while in the hidden layers the sigmoid activation function was used. The number of hidden layers and neurons was obtained experimentally through a trial and error process of all combination of number of layers $L = \{2, 3\}$ and number of neurons per layer $M = \{4, 6, 8, 10\}$. An adaptive learning rate was used starting at 0.5 and decreased once two consecutive epochs fail to decrease the training loss by at least 1e-8, or fail to increase validation score by the same value. For the purposes of training, the input values were normalized between 0 and 1. All the network topologies assessed are depicted in Table II together with the MSE and R^2 to evaluate, which one should be used in order to minimize the overfitting effect. As for the training process, a 5-fold Cross-Validation was used for each topology, meaning that 80% of the data was used for the training set, and the remaining was kept aside to assess the generalization capability of the networks. In the training routine of the ANN, the number of epochs was set to 30000, and 10% of the data was used as a validation set during training. After the training process, the network was evaluated in the test set.

Instead of the usual Early Stopping where the training is stopped when the error of the validation set stops decreasing representing overfitting and loss of generality, a Model Checkpoint technique was used. The reason behind not using the Early Stopping lies in the difficulty of specifying a reasonable patience value - number of epochs that the method should wait to stop training once the validation error stops decreasing [10]. On one hand, if the value of patience is set too low, the training might stop before the network converges to a suitable parameter solution, and on the other hand, if the patience is too high, the validation error might increase quickly and model generalization is lost. Both cases depict a situation that we consider not fair to compare networks in terms of performance. The Model Checkpoint just keeps track of the best parameter set regarding the validation error and once the network is trained, the best parameters are returned. This way, we consider this approach to be the most fair for network comparability. However, the main drawback of such an approach is longer periods for training the network due to constant storage and comparability of the best parameters regarding the current parameter set of the ANN. If the cost per minute for training is not a constraint, we strongly encourage to use such an approach.

As main training results, and as can be seen from Table II, for process DC04-HC380LA the best topology regarding the minimization of MSE is 6-6-6, not considering the network input and output layers, where the lower MSE is 0.0086 and R^2 of 0.918. This means that the network will have a total of 5 layers, being 2 the input and output layers, together with these 3 hidden layers. As for the HC260LA-HC420LA process, the best topology is 10-10 where the lowest MSE is 0.0063 and a R^2 of 0.926. Finally, for the last process HC420LA-HC380LA the minimum MSE found was 0.0084 for a topology of 10-10-10, leading to a R^2 of 0.916. Once found these topologies, we need to finalize the models so they can be ready for the following optimization step. For this case, the whole dataset was used to train a ANN with the topology that minimizes the MSE on the test set on 5-fold cross validation, and therefore is the topology that maximizes the generalization of the ANN.

TABLE II. ANN TOPOLOGY ASSESSMENT IN ORDER TO FIND THE MOST SUITABLE MODEL FOR EACH PROCESS.

Process	ANN Topology	MSE	R^2
DC04 - HC380LA	[4,4]	0.0120	0.883
	[6,6]	0.0091	0.913
	[8,8]	0.0100	0.898
	[10,10]	0.0110	0.897
	[4,4,4]	0.0100	0.904
	[6,6,6]	**0.0086**	**0.918**
	[8,8,8]	0.0097	0.908
	[10,10,10]	0.0093	0.910
HC260LA - HC420LA	[4,4]	0.0065	0.922
	[6,6]	0.0065	0.923
	[8,8]	0.0063	0.926
	[10,10]	**0.0063**	**0.926**
	[4,4,4]	0.0070	0.916
	[6,6,6]	0.0070	0.919
	[8,8,8]	0.0065	0.924
	[10,10,10]	0.0065	0.924
HC420LA - HC380LA	[4,4]	0.0087	0.912
	[6,6]	0.0088	0.912
	[8,8]	0.0088	0.913
	[10,10]	0.0092	0.909
	[4,4,4]	0.0085	0.914
	[6,6,6]	0.0085	0.915
	[8,8,8]	0.0086	0.916
	[10,10,10]	**0.0084**	**0.916**

In order to better evaluate the generalization of the trained ANNs, Figure 1 presents the MSE prediction histograms for all the presented welding processes. As can be seen, most of the samples are between the range of 0 and 0.02, being the most of them around 0. Thus, this supports the presented results in Table II where a good performance was achieved with the ANN training using the real datasets provided.

IV. PROCESS OPTIMIZATION

As the main purpose of training such models is to perform process parameter optimization, we will now assess the performance of the model by providing a set of process quality values from the dataset, and by using optimization algorithms, the best process parameters should be found. This optimization process simulates what could happen in a real scenario when a shop-floor operator needs to know the most suitable machine parameterization in order to meet the customer specifications. In this context, the process quality parameters defined by the customer are the weld width and depth yielding more robust or fragile welds in the final product. Different customers might have different requirements depending on the product application. One might only want to join metal sheets for aesthetics, where not a strong joining is required when compared with a car chassis that should be as strong and robust as possible in the automotive industry. Therefore, based on these quality values, the process parameter optimization should return the parameters to be used in the machine.

More concretely, the process models provide a prediction from a certain x (process parameters) finding the most suitable $\hat{y}$ (process quality). Contrary to this, in the process parameter optimization, the idea is to provide the desired process quality y in order to find the best process parameterization $\hat{x}$. This means that we can specify the customer requirements and

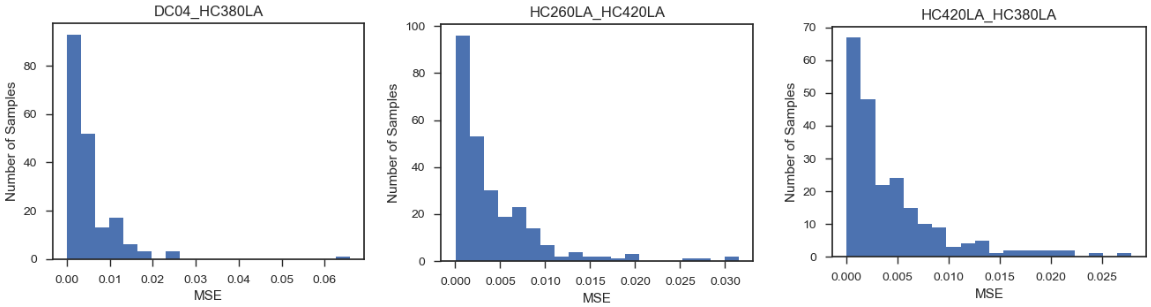

Figure 1. MSE Prediction histograms for the three ANNs trained for each of the processes.

obtain the optimal, or near optimal, machine parameterization. Based on this, a set of optimization routines was made using the trained models to assess how close the parameterization found is from the ground truth. For this test, the whole dataset was used to assess the robustness of the model in the wider range of shop-floor conditions.

For the process parameter optimization the Basin-Hopping algorithm [11] was used to find the most suitable machine parameters by minimizing the difference between the target y and optimized process quality $\hat{y}$. The Basin-Hopping (BH) algorithm was first introduced by Wales and Doye in 1997 to study the lowest-energy structures of Lennard–Jones clusters consisting of up to 110 atoms, and is based on the Monte-Carlo algorithm and gradient-based local search. It is therefore a stochastic algorithm aiming to find the global minimum of a certain function (in this case a loss function) and is mainly based on the following steps: 1) Random perturbation of the coordinates to be tested in the provided function; 2) Step towards the local minimization of the solution; 3) Reject or Accept the proposed coordinates based on the minimization step. As for the acceptance test, the Metropolis criterion is used from the Monte-Carlo application. For this algorithm an initial Temperature of 20 was set to cause large jumps in the loss function value, a number of 20000 iterations for the optimization process and stop after 1000 iterations of no solution improvement. As for the optimization process, the process models are used to iteratively assess a set of process quality values according to a certain process parameters produced by the optimization algorithm. Since these process models have used a normalized dataset between 0 and 1, we have constrained the solution search by the algorithm also between 0 and 1. As an initial guess of a solution, we have set the value to 0 for each of the parameters to be optimized.

Regarding the problem formulation, we aim to minimize the difference between the real process quality (here called target) and the solution generated by the algorithm. For that purpose, the loss function used was simply the MSE to assess these differences. Therefore, 3 defines the minimization problem:

$$\begin{aligned}\hat{x} &= \arg\min_{x} L(\hat{y}, y) \\ &= \arg\min_{x} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2\end{aligned} \tag{3}$$

where $\hat{x}$ is the machine parameterization, y is the target process quality to be achieved, x is the tested input and $\hat{y}$ is the process quality predicted by the process model trained with experimental data.

V. Results

In the present section, the best optimizations will be depicted as a main result of this paper. However, we must first clarify what is a good or bad optimization process in this context. Intuitively, one might think that a good optimization process is just to find a certain process parameterization that yields the closest process quality considering a defined target. The objective is to minimize a loss function that calculates the distance between what the model produces and the provided target. Hence, as this distance is close to 0, the best. However, in practice, this might not be useful if the difference from the ground truth of process parameters x is too far from the solution found from the algorithm $\hat{x}$.

Hence, Table III depicts the best solutions that minimize the distance between the target quality and the optimized one for all three processes, along with the process parameters to be suggested to the operator in a real application. Additionally, both MSE for process parameters found and resulting process quality are depicted. As can be seen, the obtained MSE for the process quality is very low, meaning that the algorithm used for the optimization process is very effective in finding the global optimum solution. Complementarily, Figure 2 presents the histogram for each process with number of samples in relation to the MSE between target and optimized quality. It can be seen that most of the MSE samples are near the value 0 regarding the total samples present in each dataset of the 3 processes.

However, as previously discussed, this is not very useful if the solution minimizes the distance from the target but the real parameterization is not close to the real application, or if it is even out of the parameterization bounds. This can be observed in Table III in some parameterizations suggested on Process 2 (Opt.), which are not very close to the real parameterization used. Thus, we need to ensure that this is an exception and not the rule.

In order to correctly evaluate the process parameter optimization using the trained process models, not only this distance from the target should be considered, but also the difference between process parameters and the ground truth. Therefore, Table IV presents the 3 best solutions that minimize

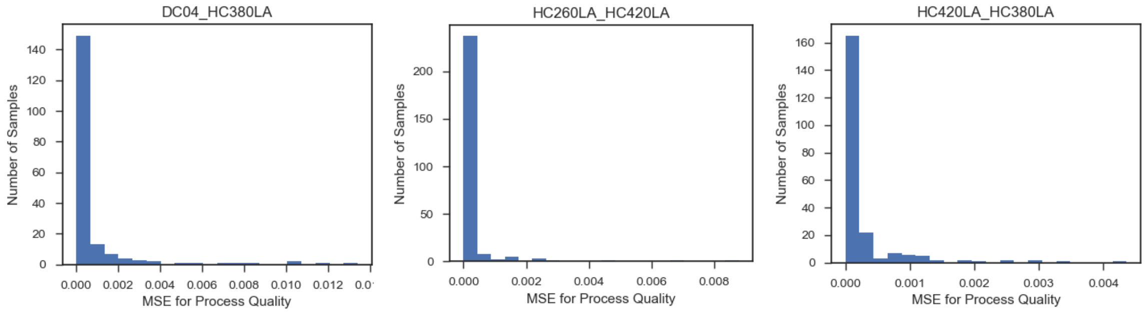

Figure 2. Histogram depicting the number of samples in relation to the MSE for the target and optimized quality.

TABLE III. PROCESS PARAMETER OPTIMIZATION FOR THE BEST 3 SOLUTIONS IN THE TEST SET THAT MINIMIZES THE DISTANCE BETWEEN OPTIMIZED AND REAL QUALITY.

Process	Case	Parameterization			Quality		MSE	MSE
		P	F	V	Depth	Width	Param.	Quality
Process 1	Real	3500	-10	100	0.14	0.67	0.362	1.828e-7
	Opt.	5500	-20	127.18	0.14	0.67		
	Real	5000	-10	80	1.1	1.05	0.079	5.99e-7
	Opt.	5100	8.97	61.72	1.09	1.048		
	Real	5000	5	160	0.15	0.72	0.021	6.003e-7
	Opt.	5385	11.45	153.46	0.149	0.718		
Process 2	Real	3500	-20	80	0.67	1.25	0.0711	3.848e-10
	Opt.	4264	-19.96	124.13	0.67	1.25		
	Real	4500	-20	130	0.65	1.2	0.016	3.911e-10
	Opt.	4934	-20	138.86	0.65	1.2		
	Real	4000	15	170	0.21	0.71	0.414	1.495e-9
	Opt.	5333	-18.79	220	0.21	0.709		
Process 3	Real	4000	-15	90	0.87	1.07	0.023	2.11e-9
	Opt.	3500	-15.41	75.85	0.869	1.069		
	Real	4500	15	90	0.97	1.2	0.345	1.058e-7
	Opt.	3500	-20	65.13	0.97	1.199		
	Real	4500	15	120	0.47	1.12	0.281	1.968e-7
	Opt.	5500	-15.38	142.98	0.469	1.12		

TABLE IV. PROCESS PARAMETER OPTIMIZATION FOR THE BEST 3 SOLUTIONS IN THE TEST SET THAT MINIMIZES THE DISTANCE BETWEEN OPTIMIZED AND REAL PARAMETERIZATION.

Process	Case	Parameterization			Quality		MSE	MSE
		P	F	V	Depth	Width	Param.	Quality
Process 1	Real	3500	-10	80	0.69	1.01	9.688e-5	1.349e-5
	Opt.	3500	-9.77	82.73	0.689	1.004		
	Real	5500	20	120	0.12	0.73	1.655e-4	2.405e-4
	Opt.	5500	20	123.78	0.138	0.718		
	Real	4500	5	120	0.44	0.94	4.657e-4	1.613e-4
	Opt.	4437	4.36	122.28	0.422	0.937		
Process 2	Real	3500	-20	100	0.57	1.2	3.285e-5	1.743e-4
	Opt.	3500	-19.99	101.68	0.573	1.181		
	Real	5500	10	220	0.48	0.86	4.501e-5	4.66e-4
	Opt.	5500	10.46	220	0.46	0.883		
	Real	3500	-20	120	0.41	1.07	8.348e-5	3.267e-4
	Opt.	3500	-20	122.69	0.43	1.05		
Process 3	Real	3500	20	60	0.93	1.21	1.55e-6	2.095e-5
	Opt.	3500	20	59.63	0.934	1.215		
	Real	3500	-20	70	0.76	1.15	3.778e-6	3.86e-6
	Opt.	3500	-20	70.56	0.762	1.148		
	Real	3500	20	90	0.23	0.64	2.075e-5	3.883e-5
	Opt.	3500	20	91.34	0.222	0.645		

the MSE for process parameterization, where a more balanced trade-off between MSEs is achieved. We can see that the presented solutions are near in both process parameters and process quality, being the ideal case in a practical application where a shop-floor operator can truly rely on what the system advises him to do. Hence, in order to understand if these results are consistent throughout the entire dataset, Figure 3 depicts the histogram for each process with the MSE between desired x and optimized process parameters $\hat{x}$. It can be seen that the majority of the samples are around 0, meaning that the process model, together with the optimization technique, are capable of indicating a suitable machine parameterization according to a given process quality. Although, there are some samples with higher errors, also revealing that the process model, for a very small amount of data points is not capable of providing a good indication of machine parameters.

VI. CONCLUSION

As discussed in the present paper, the process automation is one of the key challenges to be addressed in this fourth industrial revolution, and can be tackled using machine learning. Hence, we will conclude this paper by wrapping up with the pros and cons related with the approach of process parameter optimization and also some future work and research directions.

As for the pros, the first and most obvious is the automation of finding the most suitable machine parameters of a certain process model, or at least give a good initial guess for the machine calibration phase. Moreover, we must also highlight the suitability of ANNs in the context of process modeling, referring its flexibility, robustness and versatility when compared to the difficult process of analytical modeling by experts defining a set of equations that define the process dynamics. Additionally, we should also refer that search algorithms for global optimum are good candidates to address the problem of process parameter optimization and quickly find a close parameterization to the one used in the machine. All together, these factors are of great importance for manufacturing companies that are willing to explore the benefits of key enabling technologies associated with Industry 4.0.

Regarding the cons of such approach, we should refer to the constraint associated with most machine learning techniques of data availability. In order to train a model that should perform well in real world applications, a fair amount of data is required, which is often very difficult in manufacturing systems since these data come from machine experiments and require high material and personnel costs. Moreover, a good understanding of the machine learning algorithms to be used is also required to achieve fair results, otherwise results might not be the most satisfactory for real world scenarios and or even incorrect. Related with this topic, we should highlight approaches to address overfitting, where k-fold cross validation is one of the most widely used approaches when finding hyperparameters for the model, where a wide range

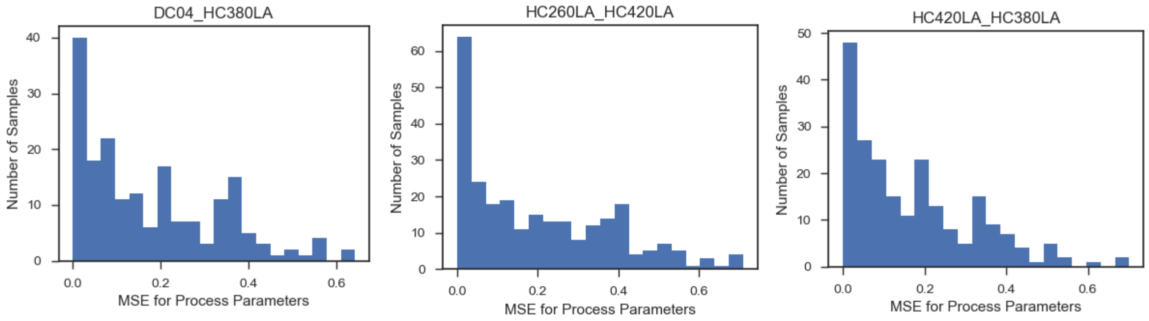

Figure 3. Histogram depicting the number of samples in relation to the MSE for the desired and optimized parameters.

of values need to be tested. Also, one should be aware of model finalization where the model with the parameters that maximize generalization should be trained with the whole dataset, and not only with training set. This is one of the most critical points that should be understood once machine learning models are used in real life applications and not only in scientific papers. Ultimately, as these techniques might tend to increase the complexity once optimizing all its parameters, it is also very important to have experience dealing with such techniques.

As for the future work, there are at least two challenges that we should discuss in the context of manufacturing systems. One of them is the topic of Transfer Learning in manufacturing systems [12] [13]. As one of the presented cons is the amount of data required for modeling, this issue can be tackled with Transfer Learning where the main goal is to improve the learning process of a new task using little amount of data, based on already existing models. In the context of manufacturing systems, this could represent training a process model with a small amount of experiments of a new machine or a new process in an already deployed machine.

Last, but not least, is the topic of Adaptive Learning where the process model is updated during time. It is known that unforeseen events and the inherent degradation of machine components forces to maintenance activities and replacement for new parts that are no longer the same as the initial state of the machine. Complementary Learning System (CLS) theory [14] has brought new promising methods that address the update of a machine learning algorithm as a stream of data is available. The CLS proposed the organization of a learning system in two different parts: 1) Hippocampus as a quick learner of new information with volatile properties and seen as short term memory, and 2) Neocortex, as a high level structural learner with a long term memory [15]. This architecture, which has its roots in neuroscience, have inspired a set of new works that recently tackle the problem of adaptive learning or continuous learning for machine learning systems.

As a conclusion, there are very interesting opportunities for machine learning to enter into manufacturing systems, and help to improve the efficiency and effectiveness of processes through the use of techniques like the ones presented in this work, and many others that still lack the validation in industry.

REFERENCES

[1] S. Wang, J. Wan, D. Li, and C. Zhang, “Implementing smart factory of industrie 4.0: an outlook,” International Journal of Distributed Sensor Networks, 2016.

[2] Y. E. Hamzaoui et al., “Optimal performance of {COD} removal during aqueous treatment of alazine and gesaprim commercial herbicides by direct and inverse neural network,” Desalination, vol. 277, no. 1–3, 2011, pp. 325 – 337.

[3] J. A. Hernández et al., “Inverse neural network for optimal performance in polygeneration systems,” Applied Thermal Engineering, vol. 50, no. 2, 2013, pp. 1399–1406.

[4] X. Wang et al., “Modeling and optimization of joint quality for laser transmission joint of thermoplastic using an artificial neural network and a genetic algorithm,” Optics and Lasers in Engineering, vol. 50, no. 11, 2012, pp. 1522–1532.

[5] P. Barmpalexis, K. Kachrimanis, A. Tsakonas, and E. Georgarakis, “Symbolic regression via genetic programming in the optimization of a controlled release pharmaceutical formulation,” Chemometrics and Intelligent Laboratory Systems, vol. 107, no. 1, 2011, pp. 75–82.

[6] S. Shan and G. G. Wang, “Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions,” Structural and Multidisciplinary Optimization, vol. 41, no. 2, 2010, pp. 219–241.

[7] J. Reis, G. Gonçalves, and N. Link, “Meta-process modeling methodology for process model generation in intelligent manufacturing,” in IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society, Oct 2017, pp. 3396–3402.

[8] F. Rosenblatt, “Principles of neurodynamics: Perceptrons and the theory of brain mechanisms.” Washington: Spartan Books, 1962.

[9] J. Duchi, E. Hazan, and Y. Singer, “Adaptive subgradient methods for online learning and stochastic optimization,” Journal of Machine Learning Research, vol. 12, no. Jul, 2011, pp. 2121–2159.

[10] L. Prechelt, “Early stopping—but when?” in Neural networks: tricks of the trade. Springer, 2012, pp. 53–67.

[11] D. J. Wales and J. P. Doye, “Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms,” The Journal of Physical Chemistry A, vol. 101, no. 28, 1997, pp. 5111–5116.

[12] S. J. Pan and Q. Yang, “A survey on transfer learning,” IEEE Transactions on knowledge and data engineering, vol. 22, no. 10, 2010, pp. 1345–1359.

[13] K. Weiss, T. M. Khoshgoftaar, and D. Wang, “A survey of transfer learning,” Journal of Big Data, vol. 3, no. 1, 2016, pp. 1–40.

[14] J. L. McClelland, B. L. McNaughton, and R. C. O’reilly, “Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory.” Psychological review, vol. 102, no. 3, 1995, pp. 419–457.

[15] R. C. O’Reilly, R. Bhattacharyya, M. D. Howard, and N. Ketz, “Complementary learning systems,” Cognitive science, vol. 38, no. 6, 2014, pp. 1229–1248.