

A Review of Machine Learning Algorithms for estimating Critical Quality Attributes from Multi-Sensor Data

Niall O' Mahony, Trevor Murphy, Krishna Panduru, Daniel Riordan, Joseph Walsh
IMAR Technology Gateway, Institute of Technology Tralee, Tralee, Ireland

Abstract

Increased globalisation and competition are drivers for process analytical technologies (PAT) that enable seamless process control, greater flexibility and cost efficiency in the process industries. The article will discuss process modelling and control for industrial applications with an emphasis on solutions enabling the real-time data analytics of sensor measurements that PAT demands.

This research aims to introduce an integrated process control approach, embedding novel sensors for monitoring in real time the critical control parameters of key processes in the minerals, ceramics, non-ferrous metals, and chemical process industries.

The article presents a comparison of machine learning algorithms applied to sensor data collected for a polymerisation process. Several machine learning algorithms including Adaptive Neuro-Fuzzy Inference Systems, Neural Networks and Genetic Algorithms were implemented using MATLAB® Software and compared in terms of accuracy (MSE) and robustness in modelling process progression. The results obtained show that machine learning-based approaches produce significantly more accurate and robust process models compared to models developed manually while also being more adaptable to new data. The article presents perspectives on the potential benefits of machine learning algorithms with a view to their future in the industrial process industry.

1. Introduction

This paper expands on the conference paper titled "Machine Learning Algorithms for Process Analytical Technology" published in the conference proceedings of the World Congress on Industrial Control Systems Security 2016 [1].

The goal of this research is to determine the key elements of a control platform for use in industrial processes as part of their implementation of PAT. PAT is concerned with observing the progression of reactions/processes (in batch or semi-continuous processes) as well as controlling process stability (in continuous processes). The adoption of PAT in industry involves the integration a suite of smart sensors to provide real-time, in-line or on-line process measurements providing significant advantages over the offline sampling approaches they replace. These novel sensors integrate into a control platform for data acquisition, data processing

& mining to produce process models by chemometrics. This platform can also provide self-learning and predictive capabilities, i.e. adaptive process control aimed at dramatically reducing overcosts derived from even slight deviations from the optimum process. This article will review state of the art approaches involving machine learning for process modelling and adaptive process control.

A concept closely aligned with PAT is Quality by Design (QbD) by which key strategic process specific attributes are identified to devise a robust control strategy which is monitored and constantly updated for continuous process improvement. Section 2 will mention some chemometric techniques used to develop process models for such a strategy.

The key word in the definition of PAT is 'timely' as PAT is concerned with observing the progression of reactions/processes (in batch or semi-continuous processes) as well as controlling process stability (in continuous processes), which requires real-time, in-line or on-line process measurements and the application of real-time data analytics. This article will focus on the latter topic with Section 3 reviewing some approaches involving machine learning in adaptive process modelling and control. Section 5 will focus on the use of machine learning in multi-sensor fusion (also reviewed previously in [2]). Section 6 will demonstrate an industrial implementation of machine learning and compare the performance of the different methods. Finally, Section 7 will discuss upcoming trends in the industry with regards to machine learning.

2. Process Modelling

2.1. Chemometrics

Chemometrics is an interdisciplinary science encompassing chemistry and computer science, where information is extracted from chemical data by data-driven means [3]. The aim of Process modelling in PAT is to produce an understanding of how the critical control parameters affect product quality.

Much of the work done in the process industry regarding chemometrics involves Near Infrared Spectroscopic (NIRS) sensors for analysing powders and resins. The multivariate data from an NIRS sensor is useless however without supporting reference measurements. Real-time, in-line and cost-

effective sensors for monitoring key strategic product parameters can be used for complementary sensor fusion with the NIR sensors for chemometrics improve process understanding. Multivariate analysis of correlated process parameters is a powerful tool in process control as it can detect deviations in quality that would have been missed in univariate analysis. Process modelling techniques include statistical methods (e.g. Multivariate Statistical Process Control (MSPC) and Partial Least Squares Regression (PLSR)) and those implementing machine learning algorithms. Artificial Intelligence can also be further applied in implementing adaptive control by constantly updating these models autonomously according to incoming sensor data.

3. Predictive/Adaptive Control

Predictive Control involves using process understanding of the effects of adjusting critical control parameters at any given time on future product quality to optimise process development. There is an inherent link between adaptive control and process modelling in the previous section, as this is where the models developed by chemometrics are applied to improve process efficiency. Predictive Control involves monitoring the process to become familiar with its behaviour. This procedure can be done by process experts (through Model Predictive Control (MPC) [4]) or be automated with the use of machine learning techniques.

Industrial MPC applications often use an empirical model obtained through system identification. A MPC controller solves a quadratic programming optimisation problem to determine optimal manipulated variable adjustments to give the desired plant output using predictions based on linear plant, disturbance, and noise models. The current timeslot is optimised, while keeping future timeslots in account by optimising a finite time-horizon or prediction horizon, but only implementing the current timeslot (the control horizon). The application of MPC reduces variation in the output variables which enables more economic operation within constraint boundaries.

Batch to batch variability in batch manufacturing processes may make it difficult to determine defined process or sub-process start point and end points. Therefore, there is a need for dynamic models that are adjusted as the process develops, i.e. Adaptive control.

Adaptive data driven learning approaches follow a different philosophy to MPC and other predictive control methodologies where a dynamic model of the process is established before implementing the controller. On one hand, this is advantageous as a hazard associated with adaptive control is the failure to distinguish between the gradual deterioration of a process over time and natural shifts in process

endpoints which adaptive control was conceived to accommodate. The occurrence of new events a machine learning mechanism has yet to encounter must be overcome by defining a procedure for importing the new data into the learning framework online.

In comparison with conventional statistical methods, machine learning algorithms are easier to use, can be used when the level of prior knowledge about the problem at hand is low, when the problem is highly non-linear or when there are too many variables to be simplified in a model. However, their black-box nature which allows them to be self-adaptive and easy to use also has the drawbacks of being non-parametric meaning the relationship between input and output variables cannot be interpreted or extrapolated, making it difficult to deal with uncertainties and requiring a large amount of data related to the input and output parameters to cover a wide range of possible scenarios. Therefore, machine learning algorithms should not be used when a simple mathematical model can describe a system.

4. Machine Learning Algorithms

Machine Learning Algorithms such as neural networks and genetic algorithms lie in the realm of Connectionist computing. Connectionist computing models incorporate self-organisation, i.e. the ability to spontaneously produce a new organisation according to environmental changes [5]. These models reject the traditional programming paradigm where the given problem must be closely examined, considering every possible case, and represented as an algorithm to be implemented in the computer program. Instead, a more human-like approach to problem solving is used to find solutions: Problem analysis is replaced by a training framework where the system is fed many training patterns (sets of inputs for which the desired outputs are known) which it learns and uses to compute new patterns.

There is no need for a general rule as in classical programming, in fact, they are a by-product of the learning algorithm. Self-organisation and the exploitation of interactions between small units have proven to perform better than central control, particularly for complex non-linear process models in that better fault tolerance and adaptability to new data is achievable [6].

4.1. Artificial Neural Networks

Artificial Neural Networks (ANNs) are a computing paradigm inspired by the functioning of the human brain. Like the human brain, it is composed of many computing cells or 'neurons' that each perform a simple operation and interact with each other to make a decision.

The simplest form of a neuron is a perceptron which multiplies each of its inputs by a weight dependent on the importance of the input to the decision the perceptron is making. The inputs and outputs of perceptrons are binary [7]. To give an example in industry, if we wanted to determine if our process was at a particular stage of its development cycle, we may treat one of the inputs, say whether the viscosity of the product has reached a certain threshold, with a greater weight than another input, e.g. if the moisture content is high, based on the fact that viscosity is a better indicator of process development. The sum of products of the inputs (x_1, x_2, \dots) and their weights (w_1, w_2, \dots), which we will denote as the dot product of vectors w and x :

$$\sum_{n=1}^j (w_n \cdot i_n) = w \cdot x \quad (1)$$

, is then compared with a bias (b) value to determine if the perceptron will return a 1 or 0:

$$\text{output} = \begin{cases} 1, & w \cdot x + b \leq 0 \\ 0, & w \cdot x + b > 0 \end{cases} \quad (2)$$

Adjusting the proportions of the weights with respect to each other or with respect to the bias greatly influences the decision-making process. When many neurons are combined in a network, quite complicated decisions can be made [2]. Neural networks learn using a set of input for which the correct outputs are given, called a training set. The network adjusts the weights and biases of the neurons so that the correct output is produced by the neuron without influencing the decision making of the rest of the network.

As the inputs and outputs of a perceptron are binary, a small change in the weights or bias can greatly affect its decision-making (swaying it between 0 and 1), making it impossible to adjust the perceptron without throwing off the rest of the network. For this reason, Sigmoid neurons are more common in learning algorithms. The inputs and outputs of sigmoid neurons are real numbers which means they can vary between 0 and 1 (i.e. represented in software as a floating-point number) and a transfer function is applied so that the neuron has less sensitivity when the output is greatly negative or greatly positive and greater sensitivity when the output is close to zero. Functions other than the sigmoid function may be applied as the transfer function producing differently shaped curves [2]. Learning can be performed on a pre-collected training set (supervised) or can occur online as a process is running (unsupervised). Supervised algorithms come in two types: regression and

classification. Unsupervised algorithms work off input data alone, i.e. it does not have the answers. Clustering is the most common unsupervised learning technique. It is used for exploratory data analysis to find hidden patterns or groupings in data or to find better representations of input parameters.

In supervised learning, a cost function, otherwise known as a loss or objective function, is used to determine how well the network is learning, e.g. the mean squared error from the correct output. The weights and biases must be adjusted to keep the cost function as low as possible. The cost function is used rather than using the number of correct outputs as a guide because it turns out to be a smooth function where fine adjustment of weight and biases gives an indication of improvement. Since finding the minimum of a function with many variable is not computationally easy using differentiation, a technique known as gradient descent (among other techniques) is used [2]. Gradient descent runs simulations to estimate the minimum cost function. Much research has been done in developing neural networks with shorter training periods that find the minimum of the cost function faster [9]. ANNs are easily implemented in parallel computing architectures, i.e. in multi-core processors, resulting in reduced processing time.

The success of machine learning relies on 'Big Data' (very large data sets) and on the fact that more data beats wiser algorithms. The algorithms produced are tailored to their specific process and focus is taken away from developing the perfect process model meaning less time and labour is spent on manually fine-tuning a model which may have to change anyway. The acquisition and storage of vast amounts of data, the vanishing gradient problem and overfitting (developing over complicated models that fit the training data precisely) are challenges associated with neural networks.

Such is the versatility of Neural Networks that they have been implemented for many sensor applications including metrological performance enhancement (calibration, nonlinearity correction, offset, identification), actuation control, sensor fault detection and classification, process control [8], large sensor arrays and sensor fusion [2]. ANNs are suited to system modelling in manufacturing processes as the problem can be reasonably defined by a limited, well-defined training set. There does exist more powerful algorithms for more general Artificial Intelligence applications. Deep Learning Networks with huge numbers of nodes and layers can achieve a high level of learning with low supervision. Also, kernel-based Support Vector Machines can ensure better generalisation without the need for closed formulation of the problem.

4.2. Adaptive Neuro-Fuzzy Inference Systems

Fuzzy logic is a method by which a human's linguistic interpretation can be translated to and from measurements readable by a microprocessor (in numerical format). For example, we recognise colour as belonging to a subset (a membership set) or even several subsets under which a range of colours can lie, e.g. red, amber, yellow. Fuzzy logic can be used to quantify colour sensor measurements in the same way and use them for industrial control applications [10]. A fuzzy control unit performs three basic processes; fuzzification (translating the sensor readings to the degrees of ownership to each of the controller's membership sets), rule evaluation (the Fuzzy Inference Unit (FIU)), and defuzzification (translating the fuzzy outputs to system outputs for process control). At the core of the controller is the FIU which uses a set of fuzzy rules defined by the user to map fuzzy inputs to fuzzy outputs [2], e.g. in a polymerisation example dealt with in this research, if the colour shade of the resin gets too dark, then stop the process.

Fuzzy control offers the advantage of being able to easily translate operator insight of a process into controller nonlinearities. Thus, it offers a better user interface that enables internal expertise to be exploited. The technology has been applied to improving control in conventional PID controllers in industrial processes, e.g. polymerisation, distillation and manufacturing process control [11]. Fuzzy logic alone does not provide adaptive control, however. ANFIS combine Fuzzy logic and neural networks to provide the advantages of both technologies [12] and mostly start with a fuzzy system to which neural network learning is applied [13]. Combined Fuzzy Logic and Neural Network systems that start with neural networks and apply fuzzy logic are in more recent development and adapt in a more sophisticated manner [13]. ANFIS models have modifiable parameters that can be interpreted by the user but at the cost of increased computational burden in comparison with ANNs.

4.3. Genetic Algorithms

Genetic algorithms (GA), or more broadly, Evolutionary algorithms are global search and optimisation techniques inspired by the principles of natural evolution and genetics. It is a heuristic approach that works well on complicated real-world problems where traditional optimisation methods frequently fail or perform poorly. GAs have received great acclaim from the research community since the 1980s and have been described as a spirit of movement that goes beyond science [6]. Frank describes evolutive learning as the 'ultimate technology' for automatically applying machine learning to any application while still an emerging

technology that has not yet been widely applied in industry [13].

Genetic algorithms work by repeatedly modifying a population of individual solutions. Operations include selection (when selecting parents), cross-over (when mating) and mutation (when applying changes to parents to form children). This process produces children solutions for the next generation [6]. The population evolves towards an optimal solution defined by some fitness function. GA differ from classical Neural Networks in that a population of solutions is produced per iteration, the best one being taken as the result. Also, the solutions have been computed using random number generators rather than deterministic computation as with classical learning algorithms. The operating principle is rooted in chaos theory of highly iterated nonlinear systems where both better and worse solutions are produced per generation but the bad mutations don't matter as they don't survive. What is important is the best solutions which were found by chance resulting in a global optimum being found and overcomes the problem of local minima associated with neural networks for an optimum solution [14]. For this reason, GAs are suited to more complicated problems.

GAs can be used to overcome other problems associated with neural networks: the need for parameters to be set prior to training according to human expertise which can be quite subjective, lacking clear rules and highly deterministic of performance. GAs can be used to set these parameters and determine how a neural network should be represented for any problem [14].

5. Data Fusion

Another application of Machine Learning in the Process Industry is in Sensor fusion, i.e. the analysis of multiple sensor outputs to achieve better process understanding and control than that which would have been possible when considering the sensor outputs independently. For the most part sensor fusion is used to improve accuracy by implementing multiple sensors to achieve a better estimate of some phenomenon. This may be achieved through measurement redundancy with multiple sensors of the same type or with multiple sensors of different types. The latter configuration offers advantages such as a possible increase in range of measurement and an increase in the level of understanding of a phenomenon in relation to surrounding phenomena.

Sensor fusion can be used to create virtual or 'soft sensors'. Soft sensors or virtual sensors can be used to boost process understanding achievable from available hardware. For one reason or another, the sensors used in a process control approach have limitations such as poor accuracy, poor field of measurement, or perhaps the parameter we want to

measure cannot be measured due to limited budget, extreme process conditions or nonexistence of sensor technology [2].

5.1. Kalman Filters

Kalman filters can be used to implement sensorless control of an actuator or to allow less accurate sensors to be used less frequently. The output state of an actuator can be deduced from single/multiple sources using a state estimator/state observer. For example, a GPS system can calculate location with a limited accuracy which can be made more precise with the addition of an accelerometer and tracking the changes in both sensor readings relative to one another over time. The Kalman Filter is a special kind of observer that provides optimal filtering of process and measurement noises if the covariance of this noises are known. The algorithm is recursive meaning new measurements can be processed as they arrive so it can run in real time.

Implementing Kalman Filtering can alleviate the accuracy requirement of sensors to be integrated into a process allowing inexpensive sensors to be implemented more frequently thereby enabling a more complete view of a process. Implementing Kalman filters in a microcontroller digitally is challenging, however, because of the high computational complexity and effect of quantisation error on stability. Technological advancements such as FPGA based System on Chip (SoC) platforms have made it possible and mean that the application of Kalman filters is likely to spread in the near future [15]. Therefore, predictive control is favourable for most applications and is an active area of research as increasing processing capabilities empowers more sophisticated algorithms.

5.2. Large scale data fusion

Larger scale data fusion is often implemented with neural networks. Many fusion algorithms have been developed to perform tasks such as deciding on the weighting applied to sensor outputs based on their reliability/ relevance, handling asynchronous multi-rate multi-sensor conditions [16] and to solve the redundancy allocation problem.

6. Exploration of Machine Learning Algorithms

The following tests aim to demonstrate an industrial implementation of machine learning and compare the performance of the different types of techniques discussed in Section 3. In this example, we want to determine the processes stage within its development cycle based on sensor data.

6.1. Background

A polymerisation process was simulated with smart sensors in place to monitor its critical quality attributes (CQAs). The process in question is of alkenyl succinic anhydride (C16/18 ASA) for paper sizing produced from the high-temperature reaction of isomerised olefins and maleic anhydride. As the reaction progresses, the resin's amber shade gets darker. The current control strategy involves manually obtaining samples periodically to perform a visual test with a Gardner Comparator apparatus (as well as other tests) and stopping the process (cooling the reactor and commencing the stripping process) when the product approaches its final colour as the reaction continues for a time after this point. A sensor that provides continuous, real-time data on the colour of the resin is required to replace the current control strategy and enable a more adaptive, automated control approach.

6.2. Experimental Setup and Input data

Colour was monitored with a multi-spectral colour sensor with a reflectance probe (the selection process for which is detailed in [17]). The sensor data (see Figure 3) consists of 8 variables produced by the colour sensor (of which 2 were considered insignificant and not used in subsequent analysis). A sample was heated in a sand bath for even heating and passed through the stages of the polymerisation reaction (as shown in Figure 1). Process phase was categorised visually using actual process samples as a reference, i.e. the colour of the resin was compared with samples taken at known stages of the process.



Figure 1. Experimental Setup of polymerisation process.



Figure 2. Resin samples at various stages of process progression.

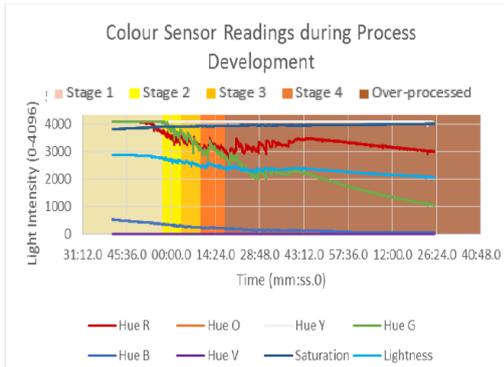


Figure 3. Colour sensor readings vs. process phase.

6.3. Algorithms tested

Regression modelling techniques were primarily investigated as it is desired for process progression to be estimated quantitatively. The samples in Figure 2 were used as markers signifying specific stages in process development. No pre-processing was performed on the input data.

Fuzzy logic was used to categorise process phase according to the colour measurements for implementation in process control. Membership functions were defined manually for each of the inputs (3-5 Mamdani type MFs per input) and the resulting Fuzzy Inference System (FIS) was run on the data in a Simulink® model. Different types of defuzzification methods were investigated, i.e. the output of centroid, bisector, smallest of maximum (SOM), mean of maximum (MOM), and largest of maximum (LOM) defuzzification methods were compared with the desired output (as depicted in Figure 5) with the most appropriate one to the application being selected. ANFIS was then investigated, where a similar MF structure to the Fuzzy Logic Model above (5 Sugeno type MFs per input as shown in Figure 4) was applied using MATLAB® NeuroFuzzyDesigner tool (the generated model is quite complicated in comparison to the fuzzy logic model as can be seen in Figure 6. Both Sub-clustering and grid partitioning for FIS generation were compared in terms of how well the output matches the training data.

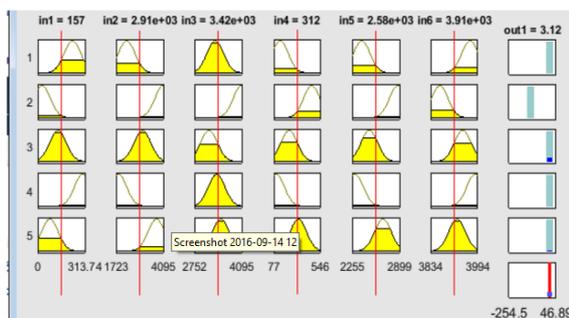


Figure 4. Fuzzy Membership functions.

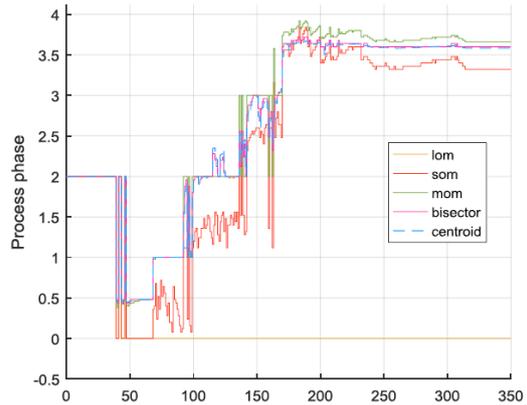


Figure 5. Comparison of defuzzification methods.

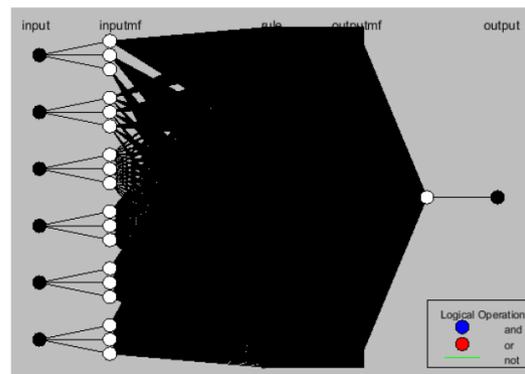


Figure 6. ANFIS Model structure.

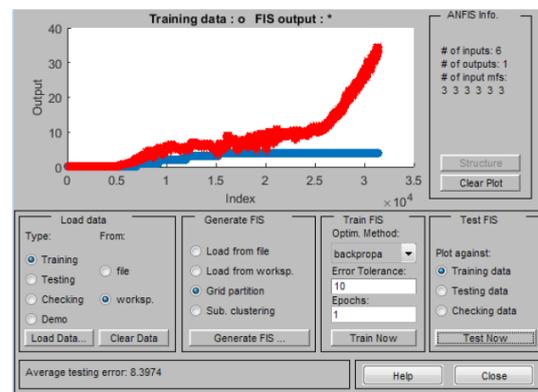


Figure 7. ANFIS results- Grid partitioning.

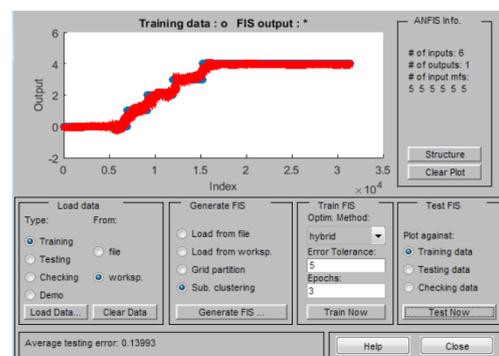


Figure 8. ANFIS results- Sub-clustering.

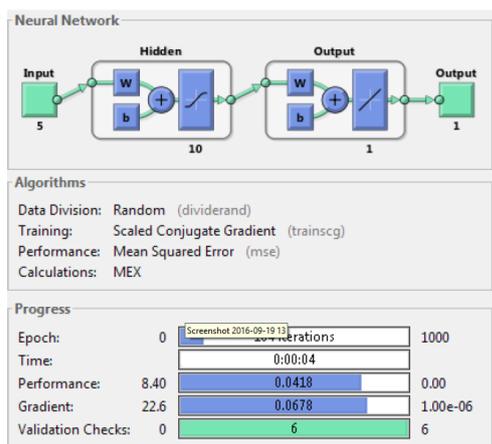


Figure 9. Neural network training results.

Thirdly, 2-layer feedforward Neural networks(see Figure 9 for structure) were trained by propagation to the data. The effects of the number of hidden layers (n) and training function were examined. Fitting neural networks (using ‘nftool’ in MATLAB®) were deemed to be most appropriate for the given application. Finally, a genetic algorithm was used to optimise the weights and biases of an untrained feed forward neural network. A MATLAB® script was written to automatically configure a neural network based on the input training data and optimise the weights and biases of this network using ‘ga’ function with a function calculating Mean Squared Error MSE of the output taken as the fitness function.

6.4. Results

Table 1. Performance comparison of process modelling algorithms tested.

Algorithm	Description	No. of Iterations	MSE	R ² (Goodness of Fit)	Time (mm:ss)
FIS (graph 1)	Manually created Fuzzy model	1	0.8795 RMSE	0.769129	Manual training
ANFIS (graph 2)	Sub-clustering, hybrid optimisation, 5% error tolerance	3	0.13993 avg. error	-	Several minutes
ANFIS	Grid-partitioning, backprop optimisation, 10% error tolerance	2	3.7657 avg. error	-	Approx. 1 hour
ANFIS	Grid-partitioning, backprop optimisation, 5% error tolerance	1	8.2766 avg. error	-	Approx. 1 hour
nftool (graph 4)	Scaled Conjugate Gradient (SCG), n= 10	272	.02955	0.994	Several seconds
nftool	SCG, n= 10	104	.0418	-	00:04
nftool	Levenberg Marquardt (trainlm), n= 10	153	.00825	0.998	01:35
nftool	trainlm, n= 10	192	.0116	-	-
nftool	Bayesian Regularisation (trainbr), n= 10	1000(max)	.0107	-	10:46
nftool	trainbr, n= 10	503	.00667	0.998	05:20
nftool	SCG, n= 20	281	.0267	-	00:15
nftool (graph 3)	trainlm, n= 20	191	.029547	-	02:59
nftool	trainbr, n= 10	1000(max)	.00532	-	00:04
Genetic Algorithm	Feed-forward neural network. 5 layers. gaoptimset function, default parameters.	729 generations	.2534	-	In the order of hours
ga (graph 5)	2 layer ff network	850 (max)	.0667	-	..
ga	1 layer ff network	850 (max)	.0798	-	..

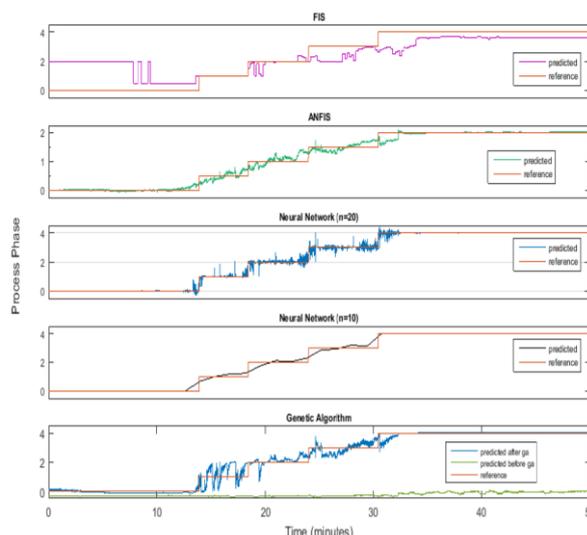


Figure 10. Model output vs. actual process phase for training data.

6.5. Discussion of Results

On comparing defuzzification methods (Figure 5), the output of centroid defuzzification was found to track the desired indication of process phase the most closely. Bisector defuzzification produced almost identical results. The output of MOM defuzzification was also similar for the most part, however, greater instability was observed about process phase 3. SOM classified halfway between process phases which was not desired in this experiment, while the output of LOM remained at zero.

On finding the most appropriate method for generating the FIS during ANFIS implementation (Figure 6 and Figure 7), sub-clustering was found to produce a desirable result (avg. error of 0.14) while also requiring far less computation time (several minutes) as can be seen in Table 1. This is in comparison with the grid partitioning method which produced did not track the desired output effectively and required a computation time in the order of hours.

The computation time, MSE and goodness of fit (R2 value) for the output of each of the algorithms was recorded in Table 1. Note that for consistency in performance evaluation, these parameters are with respect to the training data, although similar results were observed in any validation tests carried out. Accuracy was determined by how correctly process phase was classified with respect to the 4 stages identified. It should be noted that a smooth transition between the 4 phases was observed for some of the outputs produced. This characteristic of these algorithms was viewed to be beneficial in this application as process progression can be predicted with greater resolution using these methods with the phases identified simply being endpoints. Taking this

into consideration, MSE was not taken as the quintessential factor determining performance but the smoothness of the output (observed in Figure 10) was also considered.

Neural networks were found to perform the best giving a smooth output after typically 100-200 epochs and only seconds of training. Overfitting was observed when the number of hidden layers was increased (graph 3 in Table 1). Note that the levels of accuracy recorded would be unusually high and points to too small of a training set to give reliable results or overfitting. The reliability of the model would be determined when the neural network is applied to the actual process. The ANFIS model was found to give significantly better performance than the manually derived FIS while still maintaining transparency and interpretability enabling operators to understand and adjust the reasons it makes predictions. Neural networks are somewhat of a black box in comparison. The genetic algorithms performed poorest with regards to processing time, memory requirements and prediction accuracy, possibly due to the architecture of the underlying neural network, nevertheless, the script offers the advantage of being applicable to a training set of any size.

7. Future Trends

This article has identified some recent research in machine learning for industrial applications to provide an autonomous versatile control platform. Implementing Artificial Intelligence in the form of an autonomous sensor network reduces the need for costly human expert analysis and can achieve better dynamic response to process deviations more consistently. A counter-argument to such unsupervised, decentralised control is that removing human interaction is a major loss. There is no processor more powerful than the human brain and the input from experts on a process and algorithm developers is often very valuable. However, the cost of such expert analysis is very high, unfeasible and unnecessary in the long term. Therefore, autonomous control is favourable for most applications and is an active area of research as increasing processing capabilities empowers more sophisticated algorithms.

The trend of distributed computing sees the movement of sensor processing to the edge closer to the source of the data. Once a machine learning algorithm is trained it can be easily implemented on a microcontroller. A step up from this involves implementing unsupervised learning procedures on embedded systems to adapt to environmental changes in real time, generally by executing the forward propagation phase of a neural network (i.e. using a neural network trained by back propagation

previously on another machine with less processing and memory constraints to classify new samples).

There are many examples in literature of neural network architectures in literature being embedded into a microcontroller including sensor non-linearity correction [18], real-time modelling of phenomena from sensor data [19] and optimisation applications. Advancing microcontroller capabilities have even enabled the use genetic algorithms to train neural network algorithms on embedded systems which opens a whole new spectrum of applications with great potential to benefit from multivariable and multi-objective intelligent control methods [20].

8. Conclusion

This article has reviewed industrial applications of predictive control and data fusion relevant to the implementation of PAT. ANFIS has been identified as a method with wide applications that provides adaptive control while also taking advantage of internal process knowledge. Neural networks have been demonstrated to be remarkably effective in terms of efficiency (training time) and performance ($R^2 > 0.99$). Neural networks still require pre-processing of inputs and adjustment of parameters according to human expertise, however. This limitation can be automated with the hybridisation of genetic algorithms and neural networks.

9. References

- [1] Niall O' Mahony, Trevor Murphy, Krishna Panduru, Daniel Riordan, Joseph Walsh, "Machine Learning Algorithms for Process Analytical Technology," in World Congress on Industrial Control Systems Security, London, UK, 2016, pp. 20–26.
- [2] N. O. Mahony, T. Murphy, K. Panduru, D. Riordan, and J. Walsh, 'Adaptive process control and sensor fusion for process analytical technology', in 2016 27th Irish Signals and Systems Conference (ISSC), 2016, pp. 1–6.
- [3] J. Workman Jr, 'Chemometrics and PAT: What Does it all Mean?', *Role Spectrosc. Process Anal. Technol.*, pp. 18–23, 2005.
- [4] S. J. Qin and T. A. Badgwell, 'A survey of industrial model predictive control technology', *Control Eng. Pract.*, vol. 11, no. 7, pp. 733–764, Jul. 2003.
- [5] A. Smirnov, K. Sandkuhl, N. Shilov, and N. Telsya, 'Context Variation for Service Self-contextualization in Cyber-Physical Systems', in *Business Information Systems*, W. Abramowicz, Ed. Springer International Publishing, 2015, pp. 309–320.
- [6] P. Koehn, 'Combining genetic algorithms and neural networks: The encoding problem', University of Tennessee, Knoxville, 1994.

- [7] "The Nature of Code." [Online]. Available: <http://natureofcode.com/book/chapter-10-neural-networks/>. [Accessed: 09-Mar-2016].
- [8] P. Kadlec, R. Grbić, and B. Gabrys, 'Review of adaptation mechanisms for data-driven soft sensors', *Comput. Chem. Eng.*, vol. 35, no. 1, pp. 1–24, Jan. 2011.
- [9] F. Ortega-Zamorano, J. M. Jerez, J. L. Subirats, I. Molina, and L. Franco, 'Smart sensor/actuator node reprogramming in changing environments using a neural network model', *Eng. Appl. Artif. Intell.*, vol. 30, pp. 179–188, Apr. 2014.
- [10] E. Benoit and L. Foulloy, 'High functionalities for intelligent sensors, application to fuzzy colour sensor', *Measurement*, vol. 30, no. 3, pp. 161–170, 2001.
- [11] D. Filev and F. U. Syed, 'Applied intelligent systems: blending fuzzy logic with conventional control', *Int. J. Gen. Syst.*, vol. 39, no. 4, pp. 395–414, 2010.
- [12] R.-E. Precup and H. Hellendoorn, 'A survey on industrial applications of fuzzy control', *Comput. Ind.*, vol. 62, no. 3, pp. 213–226, Apr. 2011.
- [13] R. Frank, *Understanding Smart Sensors (3rd Edition)*. Norwood, MA, USA: Artech House, 2013.
- [14] R. Mahajan and G. Kaur, 'Neural networks using genetic algorithms', *Int. J. Comput. Appl.*, vol. 77, no. 14, 2013.
- [13] F. Auger, M. Hilaret, J. M. Guerrero, E. Monmasson, T. Orłowska-Kowalska, and S. Katsura, "Industrial Applications of the Kalman Filter: A Review," *IEEE Trans. Ind. Electron.*, vol. 60, no. 12, pp. 5458–5471, Dec. 2013.
- [16] D. Jeon and Y. Eun, 'Distributed asynchronous multiple sensor fusion with nonlinear multiple models', *Aerosp. Sci. Technol.*, no. Journal Article, 2014.
- [17] Niall O' Mahony, Trevor Murphy, Krishna Panduru, Daniel Riordan, Joseph Walsh, 'Fibre-optic sensors for Process Analytical Technology', in *33rd International Manufacturing Conference (IMC33) 2016*, University of Limerick, Ireland, 2016.
- [18] N. J. Cotton and B. M. Wilamowski, 'Compensation of Sensors Nonlinearity with Neural Networks', in *2010 24th IEEE International Conference on Advanced Information Networking and Applications*, 2010, pp. 1210–1217.
- [19] G. M. Lozito, M. Schmid, S. Conforto, F. R. Fulginei, and D. Bibbo, 'A Neural Network Embedded System for Real-time Estimation of Muscle Forces', *Procedia Comput. Sci.*, vol. 51, pp. 60–69, Jan. 2015.
- [20] M. Dendaluce, J. J. Valera, V. Gómez-Garay, E. Irigoyen, and E. Larzabal, 'Microcontroller Implementation of a Multi Objective Genetic Algorithm for Real-Time Intelligent Control', in *International Joint Conference SOCO'13-CISIS'13-ICEUTE'13*, Á. Herrero, B. Baroque, F. Klett, A. Abraham, V. Snášel, A. C. P. L. F. de Carvalho, P. G. Bringas, I. Zelinka, H. Quintián, and E. Corchado, Eds. Springer International Publishing, 2014, pp. 71–80.