# Improved Machine Learning using Confidence

Nima TaheriNejad and Axel Jantsch
Institute of Computer Technology
TU Wien, Vienna, Austria
Email: {nima.taherinejad, axel.jantsch}@tuwien.ac.at

***Abstract*—Wearable gadgets are in for an exponential rise thanks to the improvements in the silicon scaling and ubiquity of Internet as well as battery technology and sensor amelioration. However, despite these advances, wearable gadgets remain resource constrained devices requiring further improvements in all those areas. Self-awareness enables a system to adjust its behaviors to enhance the operations of the system and meet its goals. In this paper, we review one of the self-awareness techniques used in wearable devices and machine learning, namely confidence, which leads to their improvements. In particular, we focus on how confidence helps to maintain or enhance performance of machine learning techniques while reducing the complexity of the processes and required resources for running them on resource constrained devices. We look into three examples, epilepsy monitoring, iris flower detection, and image classification.**

## I. Introduction

The progress in Machine Learning (ML), Wearable Healthcare Systems (WHS), and cloud computing promised to bring health-care from the exclusive realm of clinics to our daily lives. This may be seen as just another Internet of Things (IoT) commodity, however, its impacts are beyond the luxury of IoT. It can have a significant impact for disabled and elderly people to enable health-care professionals to assist them at their own home [1], an issue that is gaining more and more attention with the aging population [2]. WHS can also reduce hospitalizations [3, 4, 1] by enabling the health-care professionals to perform certain check-ups and follow-ups after releasing the patients. Even clinics benefit from the combined power of these technologies [1] since they enable 24/7 continuous monitoring of patients which has been hardly possible even inside clinics [3]. They also allow more objectivity in areas of health such as mental health and emotion recognition [5, 6], which traditionally have been relying on self-reports of patients which is largely influenced by the subjective experience of the patient.

Another benefit for the clinics is the improved diagnosis enabled by ML. There have already been cases where ML algorithms have outperformed highly trained physicians [1]. For instance, last year CheXNet, a 121-layer Convolutional Neural Network (CNN), detected and analyzed pneumonia better than the average of four radiologist [7]. In another example, Esteva et al. [8] showed that their deep CNN can perform on par with 21 board-certified dermatologist in classification of skin cancer. The importance of these achievements can be better understood when we consider that 251'000 of deaths in the US during 2013 was caused by preventable medical errors [9], third only to heart disease and cancer. Nevertheless, it is important to keep in mind that despite all the progress, ML still remains limited in the scope and clinical [1]. So far, the larger focus of the literature has been on development and hence some of the fundamental challenges have been less addressed [1]. Some of these challenges include computational costs, power consumption, accuracy, real-time requirements, and reliability. Self-awareness is a method that can tackle these challenges in an efficient manner.

Self-awareness has been well received and continues to attract more attention from scientists and engineers in various disciplines such as artificial intelligence [10, 11], embedded systems [12, 13, 14], industrial production [15, 16, 17, 18], health-care [19, 4, 3, 11], and control of camera networks [20, 21], to name a few. The amazing efficiency and resilience it brings for biological systems is among the main aspiration for the self-awareness community [22]. *A self-aware system observes its own state, performance, and goals, as well as the state and behavior of the environment [23],* which is often followed by a reaction in order to achieve or approach its goals. Even though self-awareness starts with observation, it has received but little attention till recently [10]. In 2016, Taherinejad et al. [24] published a study on comprehensive observation, its various aspects, and its potential role in self-aware system. Since then, there has been work on some of these aspects, such as data reliability [19, 4], attention [4, 25], history [16, 18], and confidence [10, 11, 26]. In this paper, we focus on the latter. First, we show how confidence can improve the performance of ML in terms of energy efficiency, accuracy, real-time requirements, and reliability all in one shot. Second, given its importance and helpfulness, we delve deeper into the fundamentals of confidence and discuss its potential definition. This helps engineers and system designers by facilitating the selection of right parameters and definition for confidence in their system. It will thus enable them to implement and optimally benefit from the advantages of confidence.

## II. Self-aware Machine Learning

ML requires significant amount of resources, including processing power and available memory and energy for running it [26]. Wearable devices at the edge of the cloud suffer from the shortage of all those elements. Therefore, only simpler techniques and algorithms can be used on wearable devices. However, even for those algorithms available processing power may be very limited (possibly insufficient) and hard to provide for. In such a constraint situation, self-awareness enables the

system to improve the overall performance of the system by a better use of available resources. On that account, we review three of the works which have used self-awareness to improve the ML algorithms for resource limited devices. All of these works have used the concept of "confidence" to improve the performance of the used ML, particularly the computational cost and energy.

### A. Epileptic Seizure Detection

In [11], the authors use single-lead Electrocardiography (ECG) signals to monitor and detect epilepsy seizures. They use the Lausanne University Hospital database which contains 141 hours of data, including 34 seizures, collected by the SmartCardia INYU wearable sensor. The top level block diagram of their system, with and without self-aware unit, is shown in Figure 1, where the ML technique used is Support Vector Machine (SVM) [11]. The ML model is developed in two modes, *simple* which consumes less energy and uses less features for detection, and *complex* or *full* which consumes more energy and uses more features for detection. During the training phase, after training both the simple and full models, the simple model is used on the training data set. The predictions of the simple model is then compared to the ground truth in order to develop the confidence model. The confidence model is another SVM classifier which provides information about the confidence of the simple model in its prediction. In the test (detection) phase, the data is first fed to the confidence model. If its predicted confidence is high enough, the simple model is used for the classification, otherwise the full model is be used. The second self-aware technique that they use is progressive learning, which enhances the quality of their classification. In this method, after the initial training is done, during the progressive learning phase, if a prediction is confident enough (using the confidence model/classifier), that data will be used to train the SVM models further. To test their system, they ran it on a low power 32-bit micro-controller, namely ARM Cortex-M3 (STM32L151RDT6), with 48kB Random Access Memory (RAM) and 384kB flash storage, running at a maximum frequency of 32MHz. The reported results show a 36% improvement in the classification time (from 840ms to 538ms) while obtaining a 0.7% improvement in terms of geometrical mean ($\sqrt{\text{Specificity} \cdot \text{Sensitivity}}$) of the detection, where

$$\text{Specificity} = \frac{TN}{FP+TN}, \quad \text{Sensitivity} = \frac{TP}{TP+FN} \tag{1}$$

in which $TP, TN, FP$, and $FN$ denote True Positive, True Negative, False Positive, and False Negative, respectively. The second method led to 10% absolute and 15% relative improvement in the geometrical mean, which was largely thanks to the improvements of specificity [11].

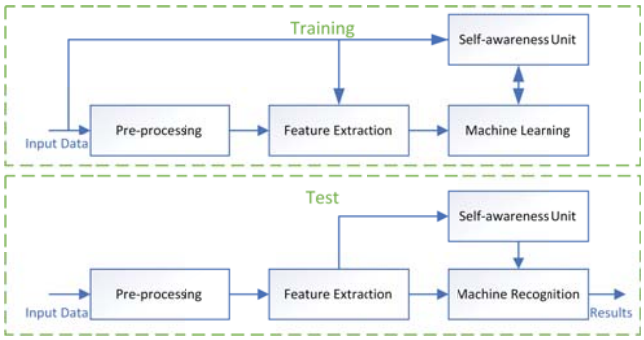


Fig. 1. Block diagram of a traditional (in black) epileptic seizure monitoring system for wearable devices, and the additional self-aware unit (in red) [11].

### B. Iris Flower Detection

In [10] the authors consider an MCS for Iris flower detection. It is assumed that each algorithm has a certain performance for various classes, and it can provide a probability value as to its confidence regarding each classification it does. Therefore, awareness of the overall system regarding these two factors can help it to use the right algorithm at the right time, as opposed to the common MCS practice of running all of the classifiers (here, Neural Network (NN), Naive Bayesian (NB), and SVM). After each training, during the cross validation phase, each algorithm is ranked based on its success rate in each class and its overall average success rate. In the test phase, classification is done based on confidence, following the algorithm shown in Figure 2. That is, if the default algorithm (the algorithm with the best overall success rate) is confident about its classification, the predicted class is sent to the output. However, if it is not confident enough, then the best (or next best) algorithm which had a good track record of classifying the unconfidently predicted class will be invoked to classify

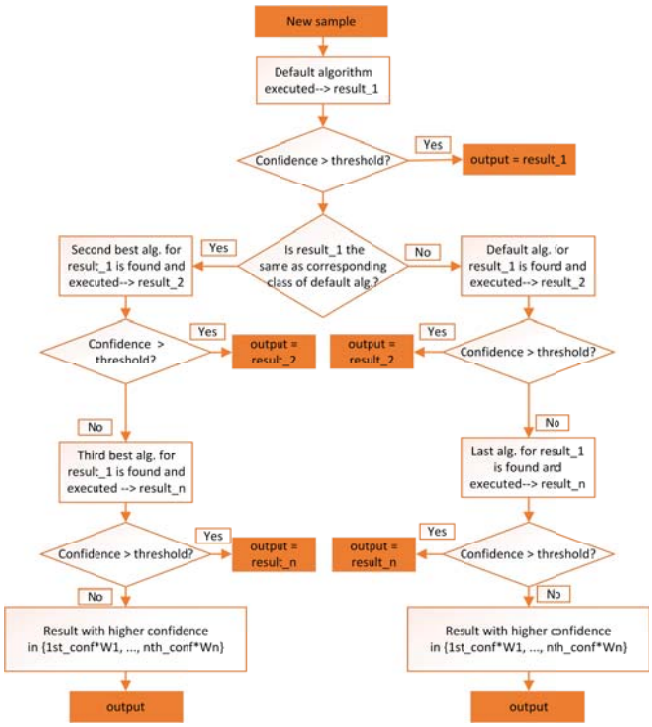


Fig. 2. The flowchart of the self-aware, confidence-based MCS [10].

the input data. This will continue till one of the classifiers is confident enough about its prediction, or all classifiers have been invoked. In the former case, the confidently predicted class will be chosen as the answer. In the latter case, the result will be decided using a weighted maximum confidence function, where the weights are proportional to the rank (the overall success rate) of the algorithms. Their experiments on the iris flower detection, ran on the data set from the UCI Machine Learning Repository [27], supports the hypothesis on lower complexity by having an average of 1.27 algorithm runs per sample, as opposed to the 3 algorithms which would be traditionally run in their MCS setup. The algorithm showed a particular superiority in classifying small data sets with up to 17% higher success rate than NN and 7.6× smaller standard deviation which shows its robustness. Although in terms of success rate it was in some cases on par or only marginally better than individual classification algorithms, its standard deviation was consistently better than others and the distribution of the predictions were more concentrated than others. Both of which speak of its better reliability.

### C. Iterative Convolutional Neural Network (ICNN)

In [26], the authors try to decrease the computation load of AlexNet [28], which requires a massive 0.7-1.0G Floating Point Operations (FLOPs) per classification. The proposed algorithm is a 1000-class image classifier for ImageNet dataset, which includes 1.2 Million labeled images. Their approach is to break the CNN of AlexNet to a set of much smaller Micro CNNs ($\mu$CNNs) which are fed by various sub-band inputs sampled from the original input image using Discrete Wavelet. Then, as shown in 3, the algorithm starts running in iterations, where each iteration has a number of $\mu$CNNs. At the end of each iteration, if the confidence of the predicted class is larger than a threshold, the algorithm is stopped and the predicted class is taken as the result. If the confidence is extremely low (lower than another threshold), the algorithm is prematurely terminated since the chances of correct classification is too low. This save a considerable amount of computations which would be otherwise spent on a most-likely misclassification. If the confidence is within these two thresholds, next iteration is run which uses more input samples and has a better accuracy (which of course comes at the cost of additional computations). Again the same procedure is repeated to decide whether the predicted results are good enough or more iterations should run. The idea is that even if all iterations are run, the computational load does not exceed the original load, however, be able to stop computations whenever the results are good enough. That is when the system is confident enough about its prediction. In their implementation the first six iterations required 12×, 6×, 3×, 2.3×, and 1.8× less FLOPs, whereas the seventh had 30% less computations load than the original AlexNet. The accuracy of the algorithm was within ∼25%-2% of AlexNet, respectively for first to seventh iteration. Therefore, using the concept of confidence, ICNN can massively reduce the computational load at a comparatively much smaller cost in accuracy.

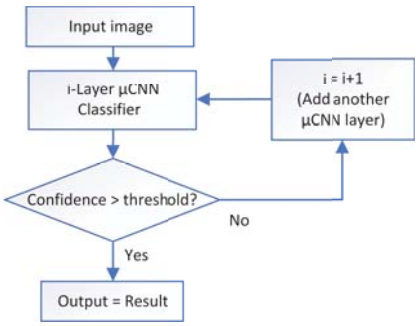


Fig. 3. The flow chart of the confidence-based ICNN [26].

TABLE I
SUMMARY OF CONFIDENCE-BASED ML WORKS PRESENTED HERE.

| Work | Approach | Comments |
|---|---|---|
| [11] | SVM Model | Improved detection, less power consumption |
| [10] | Probability | Improved reliability, better/similar detection |
| [26] | Probability | Computation load reduction, similar detection |
| [29] | Distance | Improved detection, increased robustness |

## III. FUNDAMENTALS OF CONFIDENCE

Although the benefits of using confidence has been shown in the above applications, summarized in Table I, its fundamentals are far from fully defined. For instance, in [11], the authors don't define explicitly what confidence is and use a black box approach (SVM) to model it. We contend that a better understanding of the fundamentals of confidence can help in a better application-tailored modeling and ultimately lead to a more efficient usage.

### A. Definitions

In [24], confidence is defined as "the extent to which a procedure may yield the same results on repeated trials." That distinguishes the confidence as a property, in particular reliability, of "procedures", that is algorithms, rather than data. However, repeated results do not speak of its correctness. Therefore, in [10], the authors discuss this issue with regard to learning specifically. There, they claim that, if $T_{x_i}^{k_l}$ is the Ground Truth (we know that sample $x_i$ belongs to class $k_l$) and $E_{x_i}^{A_j}$ is the class that the algorithm $A$ estimates for $x_i$, then, the confidence of $A$ is,

$$c\Big(A(x_i, k_l)\Big) = p\Big(E_{x_i}^{A} == T_{x_i}^{kl}\Big), \tag{2}$$

that is, the probability of $E_{x_i}^{A_j}$ being equal to $T_{x_i}^{k_l}$. The overall confidence of algorithm $A$ for class is then determined by averaging its confidence over all the samples it was cross-validated on, i.e.,

$$C_A = \frac{1}{m}\sum_{k_l=0}^{m} c\big(A^{k_l}\big), \tag{3}$$

in which $m$ is the total number of classes and

$$c\Big(A^{k_l}\Big) = \frac{1}{n}\sum_{i=0}^{n} c\big(A(x_i, k_l)\big), \tag{4}$$

where $n$ is the total number of samples classified as belonging to each class during the cross-validation. In [26], the confidence is not explicitly defined but it is repetitively referred to as a probability. Therefore, especially considering the common practice in ML, it is safe to assume that they consider a similar definition for it. However, in other systems, especially those outside the realm of traditional ML, the problem becomes fuzzier and more complicated. An example of which is [29].

In [29], the authors have a different take on what confidence is. They assume that, if $f$ present an ideal function or algorithm defined over a sample set, $x_i$, and $g$ an unideal function or algorithm at hand, then a function like $\Delta$ can be defined that captures the "distance" of $f(x_i)$ and $g(x_i)$. They contend that this "distance" can have any dimension, however, the emphasis is on confidence being a distance. Moreover, since often $f(x_i)$ is not available (and thus we resort to unideal $g(x_i)$), $\Delta$ cannot be calculated either and can be only estimated by $\Delta'$. In their specific application, they enhance context-aware monitoring [16, 18] by defining a piecewise linear confidence function. This function determines the confidence with which the system considers a new sample belonging to existing sample set. There, $\Delta'$ is proportional to the difference in the value of samples and confidence inversely proportional. That is the confidence of sample $j$ belonging to the same subset as sample $i$ is

$$c_{i,j} = \frac{1}{\Delta'_{i,j}} = \begin{cases} \propto \frac{1}{x_i - x_j} & \text{if } Condition\ 1 \\ K_c & \text{if } Condition\ c \end{cases} \tag{5}$$

and the overall confidence is calculated using an averaging equation similar to Equation 3 with $m$ being the sample set over which $x$ is defined. It is tempting to say that the difference between the values as a measure of confidence present the probability of one sample belonging to the same subset as the other one. However, that requires statistical data and a large number of samples to extract a probability function, which has not been done in [29] with no apparent negative effect.

### B. Distance or Probability

As discussed above, so far in the literature the main contenders for the nature of confidence are distance and probability. In other words, how far the results are from the ground truth, or how likely it is that the results are true. There are arguments for and against each of those. If there is a one or multi-dimensional numerical parameter space, an appropriate distance metric is often easy to establish. For example, if a heartbeat extraction algorithm calculates a value of $y$ and the real value is $x$, then the distance $|x-y|$ is an appealing quality metric for the confidence of its calculation For categorization tasks, such as identifying pears and apples in images, distance metrics are not as straight forward. Breaking down the categorization task into elementary measured parameters such as color, size, and shape, may help. However, a good categorizer such as an NN integrates these elementary data with a non-linear, non-obvious weight function into a categorization, a process which may be hard to mimic with an appropriate distance metric. Another aspect, which pragmatically may not be paramount but conceptually is important, is the meaning of this definition in the context. What is the meaning of the distance between two points in a multi-dimensional space with different units on each axis?

On the other hand, NN and other categorizers inherently provide an assessment of their results which can, with some justification, be interpreted as probabilities, as has been done in [10] and [26]. The outputs of the NN in these examples sums up to "1" and give a relative assessment of the different categories. However, in some cases with small data sets, like in [10], it is somewhat of a stretch to take the produced values of the NN as probabilities of correct classification (due to lack of very large data sets and repeated experiments enough for being statistically meaningful). In the case of distance (value difference) as in [29] (and applications like [16, 18]) or decision-tree methods in general, the distance could be outside the range of $[0, 1]$. Since probability cannot be outside this range, that undermines the candidacy of probability as the nature of the confidence, although, normalization could address this issue. Another point is the complication of defining probability in methods such as maximum likelihood which uses a probability density function for classification. There, defining a distance seems relatively more straight forward.

However, there is a third possibility to consider. That is, confidence as the probability of distances. That is, how likely it is that the result has a distance lower than a specific value with the ground truth. This creates a space that can be shaped, with different assumptions, to both of above cases. If distance cannot be defined (like in categorization tasks), or zero distance is intended, confidence would be a simple probability value. On the other hand, if we assume that all distances have (almost) similar probability, confidence would be simplified to a distance (like the cases in [29]). Using this definition more complex cases could be tackled, for example, by using the probability distribution to form the confidence function which maps distance to confidence. Currently, it seems that regardless of what the true nature of confidence is, choosing the mathematical formulation of confidence depending on the nature of the problem is the best existing strategy.

## IV. Conclusion

Self-awareness can play an important role in improving the performance of resource constraint devices such IoT and wearable devices. In this paper, we focused on "confidence" as one of the primitives of self-awareness. We first showed how it has been used to improve ML algorithms in terms of success rate and required computational resources and energy, as well as the reliability of the use ML techniques. Then we had a closer look on the existing definitions and models of confidence and analyzed them. "What is the nature of confidence?" remains still an open question to be studied further. We claim that answering that question can have an

important impact on the usage of this concept in various systems. Nevertheless, as shown in the examples, that does not undermine the benefit of using confidence, even with an ad-hoc definition. To this end, we provided the most recent insight on the state-of-the-art and on which of the existing definitions may be more useful for the specific application the reader has in mind.

## REFERENCES


[1] H. Yin et al. Smart healthcare. *Foundations and Trends® in Electronic Design Automation*, 2(14):401–466, 2018.

[2] A. M. Nia et al. Energy-efficient long-term continuous personal health monitoring. *IEEE Transactions on Multi-Scale Computing Systems*, 1(2):85–98, April 2015.

[3] M. Götzinger et al. Enhancing the self-aware early warning score system through fuzzified data reliability assessment. In *International Conference on Wireless Mobile Communication and Healthcare*. Springer, 2017.

[4] A. Anzanpour et al. Self-awareness in remote health monitoring systems using wearable electronics. In *Design and Test Europe Conference (DATE)*, 2017.

[5] N. TaheriNejad and D. Pollreisz. Assessment of physiological signals during happiness, sadness, pain or anger. In *6th MobiHealth*, 2016.

[6] D. Pollreisz and N. TaheriNejad. A simple algorithm for emotion recognition, using physiological signals of a smart watch. In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2353–2356, July 2017.

[7] P. Rajpurkar et al. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *CoRR*, abs/1711.05225, 2017.

[8] A. Esteva et al. Dermatologist-level classification of skin cancer with deep neural networks. 542, 01 2017.

[9] M. A Makary and M. Daniel. Medical error—the third leading cause of death in the us. *BMJ*, 353, 2016.

[10] H. A. Kholerdi et al. Enhancement of classification of small data sets using self-awareness; an iris flower case-study. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, May 2018.

[11] F. Forooghifar et al. Self-aware wearable systems in epileptic seizure detection. In *Euromicro Conference on Digital System Design (DSD)*, 2018.

[12] A. Bouajila et al. Organic computing at the system on chip level. In *2006 IFIP International Conference on Very Large Scale Integration*, pages 338–341, Oct 2006.

[13] H. Hoffmann et al. A generalized software framework for accurate and efficient management of performance goals. In *Proceedings of the International Conference on Embedded Software*, pages 1–10, Sept 2013.

[14] S. Sarma et al. CyberPhysical-System-On-Chip (CPSoC): A self-aware MPSoC paradigm with cross-layer virtual sensing and actuation. In *2015 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 625–628, March 2015.

[15] L. C. Siafara et al. Samba: A self-aware health monitoring architecture for distributed industrial systems. In *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, pages 3512–3517, 2017.

[16] M. Götzinger et al. On the design of context-aware health monitoring without a priori knowledge; an AC-motor case-study. In *IEEE Canadian Conference on Electrical and Computer Engineering*, pages 1–5, 2017.

[17] L. Siafara et al. Samba – an architecture for adaptive cognitive control of distributed cyber-physical production systems based on its self-awareness. *e & i Elektrotechnik und Informationstechnik*, 135(3):270–277, 2018.

[18] M. Götzinger et al. Applicability of context-aware health monitoring to hydraulic circuits. In *44th Annual Conference of the IEEE Industrial Electronics Society (IECON)*. IEEE, 2018.

[19] M. Götzinger et al. Enhancing the early warning score system using data confidence. In *International Conference on Wireless Mobile Communication and Healthcare*, pages 91–99. Springer, 2016.

[20] B.d Rinner et al. Self-aware and self-expressive camera networks. *Computer*, 48(7):21–28, 2015.

[21] P. R. Lewis et al. Architectural aspects of self-aware and self-expressive computing systems: From psychology to engineering. *Computer*, 48(8):62–70, Aug 2015.

[22] N. Dutt and N. TaheriNejad. Self-awareness in cyber-physical systems. In *2016 29th International Conference on VLSI Design and 2016 15th International Conference on Embedded Systems (VLSID)*, pages 5–6, Jan 2016.

[23] N. Dutt et al. Toward smart embedded systems: A self-aware system-on-chip (SoC) perspective. *ACM Trans. Embed. Comput. Syst.*, 15(2):22:1–22:27, 2016.

[24] N. TaheriNejad et al. Comprehensive observation and its role in self-awareness; an emotion recognition system example. In *the Federated Conference on Computer Science and Information Systems (FedCSIS)*, 2016.

[25] N. TaheriNejad et al. Self-aware sensing and attention-based data collection in multi-processor system-on-chips. In *2017 15th IEEE International New Circuits and Systems Conference (NEWCAS)*, pages 81–84, 2017.

[26] K. Neshatpour et al. ICNN: An iterative implementation of convolutional neural networks to enable energy and computational complexity aware dynamic approximation. In *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 551–556, March 2018.

[27] PM Murphy and DW Aha. UCI repository of machine learning databases, university of california, department of information and computer science, irvine, ca, 1994.

[28] A. Krizhevsky et al. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA, 2012.

[29] Maximilian Götzinger et al. Model-free monitoring with confidence. In *International Journal of Computer Integrated Manufacturing*, pages 1–25. Accepted, 2019.