Online Approximation of Prediction Intervals Using Artificial Neural Networks

Myrianthi Hadjicharalambous(✉), Marios M. Polycarpou, and Christos G. Panayiotou

KIOS Research and Innovation Center of Excellence,
Department of Electrical and Computer Engineering,
University of Cyprus, Nicosia, Cyprus
{hadjicharalambous.myrianthi,mpolycar,christosp}@ucy.ac.cy

Abstract. Prediction intervals offer a means of assessing the uncertainty of artificial neural networks' point predictions. In this work, we propose a hybrid approach for constructing prediction intervals, combining the Bootstrap method with a direct approximation of lower and upper error bounds. The main objective is to construct high-quality prediction intervals – combining high coverage probability for future observations with small and thus informative interval widths – even when sparse data is available. The approach is extended to adaptive approximation, whereby an online learning scheme is proposed to iteratively update prediction intervals based on recent measurements, requiring a reduced computational cost compared to offline approximation. Our results suggest the potential of the hybrid approach to construct high-coverage prediction intervals, in batch and online approximation, even when data quantity and density are limited. Furthermore, they highlight the need for cautious use and evaluation of the training data to be used for estimating prediction intervals.

Keywords: Prediction intervals · Lower and upper error bounds
Online learning · Adaptive approximation

1 Introduction

The use of Artificial Neural Networks (ANN) in approximating unknown functions has attracted significant research interest over the last decades [1,2], motivated by the universal approximator properties of ANN [2]. However, in practical scenarios where the function to be approximated is unknown, ANN's accuracy relies on the quality and quantity of the available measurements. Noise-corrupted measurements, multi-valued targets along with data uncertainty stemming from variabilities of the physical system, significantly impact ANN's point predictions. The reliability of point predictions is further deteriorated in online approximation scenarios, whereby the training data might be sparse – especially at initial training stages – or might not representatively cover the entire region of interest.

V. Kůrková et al. (Eds.): ICANN 2018, LNCS 11139, pp. 566–576, 2018.
https://doi.org/10.1007/978-3-030-01418-6_56

Such issues will likely force the ANN to extrapolate, limiting its generalisation ability along with the practical utility of point predictions. As an alternative to point predictions, Prediction Intervals (PIs) have been proposed [3–5] which provide lower and upper bounds for a future observation, with a prescribed probability. From a practical point of view, PIs could be preferable to point predictions as they provide an indication of the reliability of the ANN as well as enable practitioners to consider best- and worst-case scenarios. For example, PIs could be particularly useful in control engineering and fault detection applications [6], where uncertainty bounds could help distinguish the healthy operation of the system from faulty behaviour.

A range of methods have been proposed in the literature for constructing PIs and assessing the reliability of ANN. Amongst them, the delta technique [3], the mean variance estimation method and Bootstrap approaches [4] have been used extensively to evaluate PIs on real and synthetic problems. These traditional approaches first generate the point predictions and subsequently compute the PIs following assumptions on error or data distributions, which might be invalid in real world applications. Additionally, as the resulting PIs are not constructed to optimise PI quality, they might suffer from low coverage of the training/test set or might result in wide, over-conservative error bounds.

An alternative approach (Lower Upper Bound Estimation (LUBE)) has been proposed by Khosravi *et al.*, focusing on directly estimating high-quality PIs, while avoiding restrictive assumptions on error distributions [5]. Instead of quantifying the error of point predictions, LUBE uses ANN to directly approximate lower and upper error bounds, by optimising model coefficients to achieve maximum coverage of available measurements, with the minimum PI width [5,7]. Although LUBE has demonstrated significant potential against traditional approaches in terms of accuracy, interval width and computational cost [8,9], it is less reliable when limited or non-uniformly distributed training data are available [10]. In fact, Bootstrap and delta methods produce wider PIs in regions with sparse data, signifying the larger level of uncertainty in ANN approximation; capturing model uncertainty is an important feature of PIs [9,11], lacking in the LUBE approach which mainly accounts for noise variance.

In this work, we propose a combination of the Bootstrap and LUBE methods, which exploits good characteristics from both techniques. The proposed Bootstrap-LUBE Method (BLM) enhances the reliability of the LUBE approach when data is sparse or limited, by augmenting the training set with pseudo-measurements stemming from Bootstrap replications. The pseudo-measurements will present larger variability in regions with sparse data, forcing BLM to produce a wider local PI and thus capture the larger uncertainty in approximation. Following LUBE, BLM constructs PIs by optimising their coverage and width, while at the same time avoiding any assumptions on data/error distributions.

Another important contribution of this work is to extend the proposed hybrid approach to adaptively approximate the PIs during the online operation of the system. In cases where data becomes continuously available in a sequential way, use of the either LUBE or BLM on the entire current dataset would become

infeasible as it would incur a continuously increasing computational cost. At the same time, offline estimation of PIs based on past data would likely be unsuitable as it would be unable to accommodate dynamic changes in data patterns. We propose an online learning scheme for estimating PIs, in which the lower and upper bounds are iteratively updated to also account for recent measurements. At each iteration only recent data are used in PI-optimisation, thus significantly reducing the computational cost and further enhancing the efficiency of BLM.

2 Methods

Throughout this section, we assume that we want to construct a PI for the approximation of an unknown function $f(x)$, $x \in D$, where the region of interest D is a compact subset of $\mathbb{R}$. Available measurements are denoted by $(x_i, Y_i),\ i = 1, \cdots, N$, which are assumed to be corrupted by noise ϵ ($Y_i = f(x_i) + \epsilon_i$). A PI of a predetermined confidence level $(1-a)$ for a future observation Y_{N+1} consists of a lower $L(x_{N+1})$ and upper bound $U(x_{N+1})$, denoting that the future observation will lie within the interval with a probability $1 - a$:

$$P(Y_{N+1} \in [L(x_{N+1}),\ U(x_{N+1})]) = 1 - a. \tag{1}$$

For the Bootstrap method, let us assume that we want to approximate the unknown function $f(x)$ with $\hat{f}(x; \boldsymbol{w}, \boldsymbol{c}, \boldsymbol{\sigma})$, using a Radial Basis Function (RBF) network:

$$\hat{f}(x; \boldsymbol{w}, \boldsymbol{c}, \boldsymbol{\sigma}) = \sum_{h=1}^{H} w_h \phi_h(x; c_h, \sigma_h), \quad \phi_h(x; c_h, \sigma_h) = \exp(\frac{-(x - c_h)^2}{\sigma_h^2}). \tag{2}$$

Here H denotes the number of ANN neurons ($H = 20$ for the tests considered) and w_h are weighting coefficients scaling the RBF ϕ_h. The centres c_h are evenly distributed over the region of interest and the widths σ_h are evaluated using a nearest-neighbour heuristic, leading to a linear-in-parameter approximator $\hat{f}(x; \boldsymbol{w})$. The weight vector $\boldsymbol{w}$ can then be estimated by minimising the error function $\sum_{i=1}^{N} [Y_i - \hat{f}(x_i; \boldsymbol{w})]^2$ using least squares estimation.

2.1 Prediction Interval Estimation Methods

Bootstrap Residual Method. Bootstrap methods rely on multiple pseudo-replications of the training set to approximate unbiased estimates of prediction errors. Here we concentrate on the Bootstrap residual method, whereby model residuals are randomly resampled with replacement. The Bootstrap residual method algorithm described in [4] can be summarised as follows:

- Get an initial estimate $\hat{\boldsymbol{w}}$ from available measurements, compute residuals $r_i = Y_i - \hat{f}(x_i; \hat{\boldsymbol{w}})$ and then compute variance-corrected residuals s_i [4].
- Generate B samples of size N drawn with replacement from residuals $s_1, \cdots, s_N$, denoted by $s_1^b, \cdots, s_N^b$ for the b^{th} sample. For the b^{th} replication:

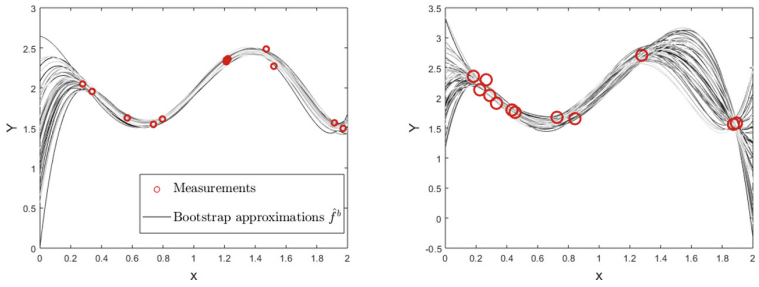

Fig. 1. Function approximations $\hat{f}^b$ at 50 Bootstrap replications (*grey shaded lines*). The variability among approximations from different replications is significantly larger in regions where measurements used for training (*red circles*) are limited. (Color figure online)

- Generate b^{th} replication's "measurements" $Y_i^b = \hat{f}(x_i; \hat{\boldsymbol{w}}) + s_i^b$.
- Estimate $\boldsymbol{w_b}$ by minimising the error $\sum_{i=1}^{N}[Y_i^b - \hat{f}(x_i; \boldsymbol{w})]^2$ and calculate the Bootstrap approximation $\hat{f}^b(x; \boldsymbol{w_b})$.
- Calculate the current estimate for the approximation error ϵ_{N+1}^b.

– Construct PI using percentiles of the error ϵ_{N+1}.

LUBE Method. LUBE's cornerstone is the direct approximation of PIs using ANNs. Instead of the unknown function $f(x)$, LUBE approximates the lower $L(x)$ and upper $U(x)$ bounds using RBFs: $\hat{L}(x; \boldsymbol{w^L}) = \sum_{h=1}^{H} w_h^L \phi_h(x)$, $\hat{U}(x; \boldsymbol{w^U}) = \sum_{h=1}^{H} w_h^U \phi_h(x)$. The main goal is to produce high-quality PIs, where quality is assessed using two indices: (a) PI Coverage Probability (PICP) and (b) Normalised Mean Prediction Interval Width (NMPIW). In particular, PICP is given by:

$$PICP(\boldsymbol{w^L}, \boldsymbol{w^U}) = \frac{1}{N}\sum_{i=1}^{N} C_i, \tag{3}$$

with $C_i = 1$ if $Y_i \in [\hat{L}(x_i; \boldsymbol{w^L}), \quad \hat{U}(x_i; \boldsymbol{w^U})]$ and $C_i = 0$ otherwise. Similarly, for R denoting the range of observations, NMPIW is given by:

$$NMPIW(\boldsymbol{w^L}, \boldsymbol{w^U}) = \frac{1}{N}\sum_{i=1}^{N}[\hat{U}(x_i; \boldsymbol{w^U}) - \hat{L}(x_i; \boldsymbol{w^L})]/R. \tag{4}$$

From a practical point of view it is useful to have narrow PIs (small NMPIW) which offer high coverage of the measurements (large PICP), leading to the following optimisation problem [5,7]:

$$\begin{aligned} Minimise \quad & NMPIW(\boldsymbol{w^L}, \boldsymbol{w^U}) && (5)\\ & 1 - PICP(\boldsymbol{w^L}, \boldsymbol{w^U}) && (6)\\ Subject\ to \quad & NMPIW(\boldsymbol{w^L}, \boldsymbol{w^U}) > 0, && (7)\\ & 1 - PICP(\boldsymbol{w^L}, \boldsymbol{w^U}) \leq a, && (8) \end{aligned}$$

where a is the desired confidence level ($a = 0.05$ for the tests considered). Due to the complexity of the mutli-objective optimisation problem, weights $\boldsymbol{w^L}$ and $\boldsymbol{w^U}$ are estimated using a Non-Dominated Genetic Algorithm II (NSGA-II) [7,12]. Among solutions with PICP$\geq 1-a$, the solution producing the narrowest PI is selected.

Bootstrap-LUBE Method (BLM). BLM is aiming at combining good characteristics from the Bootstrap and LUBE methods. The main objective of BLM is to directly estimate PIs by optimising their quality (similar to LUBE), while at the same time accounting for model uncertainty (similar to Bootstrap).

In fact, Bootstrap produces wider bounds in regions with sparse data, capturing the larger model uncertainty while the LUBE approach which mainly accounts for noise variance lacks this feature (Figs. 1, 2 and 3). Looking closer into Bootstrap (Fig. 1), there is significant variability between the Bootstrap approximations $\hat{f}^b$ from different replications in regions with sparse data, most likely due to extrapolation. In such regions the error at each replication will be large leading to large regional error variance and wide regional error bounds.

The main idea of BLM is to enrich the N available measurements with pseudo-measurements originating from the Bootstrap approximations ($\hat{f}^b$), to force BLM to account for data density. We first define an auxiliary set of points $(x_j^*,\ j = 1, \cdots, N_{aux})$ evenly distributed in the region of interest. We then compute the Bootstrap approximation of each replication for all of the x^* points $(\hat{f}^b(x_j^*),\ b = 1, \cdots, B,\ j = 1, \cdots, N_{aux})$ which will lead to $B \cdot N_{aux}$ pseudo-measurements (light blue dots in Figs. 2 and 3). The multi-objective optimisation problem of LUBE is now augmented to finding $\boldsymbol{w^L}$ and $\boldsymbol{w^U}$ which:

$$\begin{aligned}
Minimise \quad & NMPIW(\boldsymbol{w^L}, \boldsymbol{w^U}) + NMPIW_{pseudo}(\boldsymbol{w^L}, \boldsymbol{w^U}) && (9)\\
& 1 - PICP(\boldsymbol{w^L}, \boldsymbol{w^U}) && (10)\\
Subject\ to \quad & NMPIW(\boldsymbol{w^L}, \boldsymbol{w^U}) > 0, && (11)\\
& 1 - PICP(\boldsymbol{w^L}, \boldsymbol{w^U}) \leq a, && (12)\\
& 1 - PICP_{pseudo}(\boldsymbol{w^L}, \boldsymbol{w^U}) \leq 0.01, && (13)
\end{aligned}$$

where PICP and NMPIW are computed over the N actual measurements, and PICP_{pseudo} and NMPIW_{pseudo} are computed on the pseudo-measurements. With the BLM formulation the PIs will be forced to be wider in regions with sparse data (where pseudo-measurements will present substantial variations), indicating larger model uncertainty. At the same time, regions with dense data will not be affected, as the variation in pseudo-measurements will be small (the Bootstrap approximation in those regions is similar throughout replications (Fig. 1)).

2.2 Online Estimation of Prediction Intervals

During the online operation of a system where data becomes available in a sequential manner, use of either LUBE or BLM on the entire current dataset

would become infeasible. To this end, we propose an online approximation scheme which takes into account past and current data, in a computationally efficient way. Based on a weighted sliding window learning scheme, the lower and upper bounds are iteratively updated at specific time instances.

In particular, the lower and upper bounds' weights (condensed into vector $\boldsymbol{w}$) are first trained on the N_i initial measurements, leading to estimate $\boldsymbol{w_i}$. Assuming a continual and uniform in time inflow of measurements, the bounds are updated at the first sliding window when N_i+N_w measurements are available ($N_w \leq N_i$):

$$\boldsymbol{w}(N_i + N_w) = \boldsymbol{w_i} \frac{N_i}{N_i + N_w} + \boldsymbol{w_w} \frac{N_w}{N_i + N_w}. \tag{14}$$

Here $\boldsymbol{w_w}$ denote the weights of the lower and upper bounds estimated with multi-objective optimisation based only on the most recent N_w measurements of the current window. The contribution of the recent measurements in the current weights' evaluation is determined by the ratio of measurements in the current window (N_w) to the total number of available measurements (N_i+N_w). Similarly, for the k^{th} window, the weights will be iteratively updated to account for past and current measurements with equal contributions:

$$\boldsymbol{w}(N_i + kN_w) = \boldsymbol{w}(N_i + (k-1)N_w) \frac{N_i + (k-1)N_w}{N_i + kN_w} + \boldsymbol{w_w} \frac{N_w}{N_i + kN_w}. \tag{15}$$

For each window only N_w measurements are used in the optimisation, significantly reducing the computational cost of the optimisation problem. Note that when BLM is used, the weights are estimated using the measurements of the current window as well as the auxiliary Bootstrap-based measurements.

3 Results and Discussion

3.1 Comparison of Prediction Interval Estimation Methods

The methods for constructing PIs described in Sect. 2.1 are tested and compared on synthetic tests. Of interest in this work is the quality of the PIs when non-uniformly distributed or sparse data are available. Accordingly, as we are investigating extreme scenarios, the training data are generated from random uniformly distributed data under specific restrictions. In particular, we are replicating two scenarios: (a) the training data do not representatively cover the entire domain, but only regions of it (Fig. 2), (b) very few training data are available over the entire domain (Fig. 3). For both scenarios the test data are uniformly covering the entire domain, to enable reliable assessment of PI accuracy.

Two functions to be approximated are considered ($f_1(x) = 0.5\sin(1.5\pi x + \pi/2)+2$, $f_2(x) = 5\sin(\pi x+\pi/2)+\exp(x)$). Training and test data are generated based on these functions and white Gaussian noise of 10% of the mean function value is added. For both functions we consider the two training scenarios, leading to the following tests: Test1: PI for regional data generated from f_1, Test2: PI for regional data generated from f_2, Test3: PI for sparse data generated

from f_1, Test4: PI for sparse data generated from f_2. For Test1 and Test2, we consider 100 training points, while 15 training points are considered for Test3 and Test4. Every test is repeated 10 times with different randomly generated training data, to enable a more reliable comparison of the methods. Table 1 presents PI quality indices for all methods, averaged over the 10 replications of each test. Representative PIs are demonstrated in Fig. 2 for scenario (a) and Fig. 3 for scenario (b).

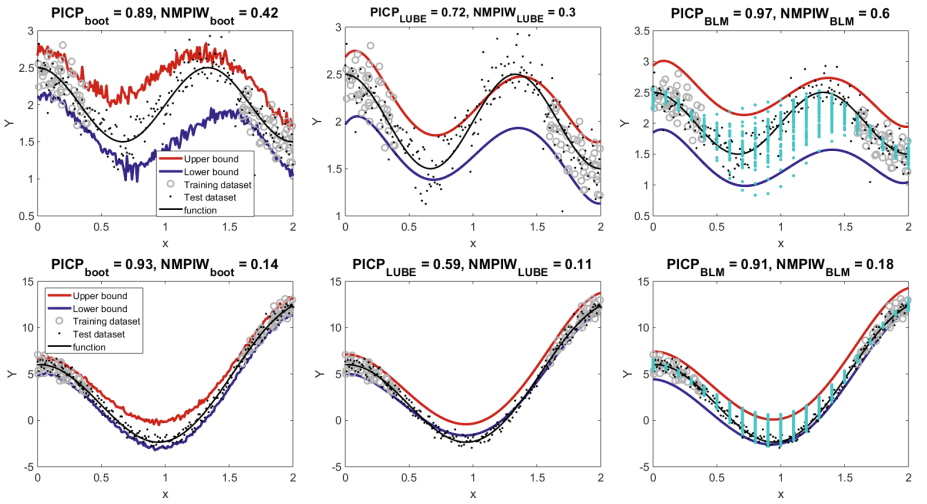

Fig. 2. PIs constructed using the Bootstrap (*left column*), LUBE (*middle column*) and BLM (*right column*) approaches. Data limited to certain regions of the domain following scenario (a), originate from f_1 (Test1, *top row*) and f_2 (Test2, *bottom row*). *Light blue dots* indicate Bootstrap pseudo-measurements used by BLM. PICP and NMPIW are evaluated on the test dataset, uniformly covering the entire domain. (Color figure online)

Across the tests considered BLM clearly outperforms LUBE method in terms of coverage, with an average increase of 15–30% in PICP. By considering Bootstrap pseudo-measurements, BLM is able to produce larger bounds in regions with fewer data, providing an indication of the uncertainty in the estimation. Additionally, due to BLM's optimisation of PI quality, BLM produces a better coverage compared to Bootstrap in the majority of tests. Increased PICP comes at the cost of wider PIs, nevertheless, the fundamental requirement for a PI to reliably include future observations is clearly prioritised over narrow – yet invalid – PIs.

Finally, it is worth noting that BLM is performed on a larger number of training measurements compared to LUBE, without significantly impacting the computational cost. The increased cost in computing PICP and NMPIW over the pseudo-measurements is not substantial (note that B and N_{aux} do not need

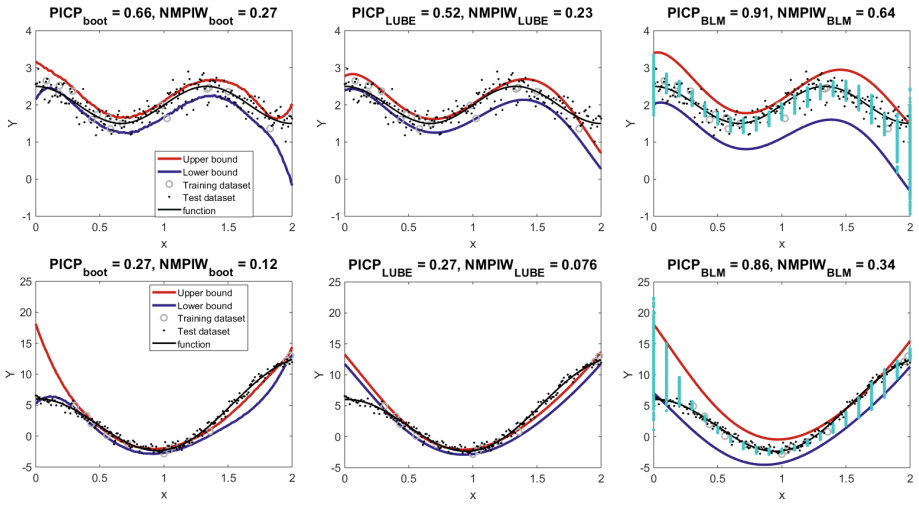

Fig. 3. PIs constructed using the Bootstrap (*left column*), LUBE (*middle column*) and BLM (*right column*) approaches. Sparse data originate from f_1 (Test3, *top row*) and f_2 (Test4, *bottom row*) following scenario (b). *Light blue dots* indicate Bootstrap pseudo-measurements used by BLM. PICP and NMPIW are evaluated on the test dataset, uniformly covering the entire domain. (Color figure online)

to be very large to enable BLM to account for data density), while the dimensions of the parameters ($\boldsymbol{w^L}$ and $\boldsymbol{w^U}$) to be estimated remain the same.

Table 1. Average characteristics of the PIs constructed for four synthetic tests, using the Bootstrap, LUBE and BLM approaches. PICP and NMPIW are evaluated on the test dataset, uniformly covering the entire domain.

Tests	Bootstrap		LUBE		BLM	
	PICP(%)	NMPIW(%)	PICP(%)	NMPIW(%)	PICP(%)	NMPIW(%)
Test1	83.23	43.54	74.33	37.68	89.75	57.63
Test2	65.06	20.74	62.47	18.61	91.76	41.00
Test3	65.27	33.56	66.67	29.72	94.68	62.68
Test4	65.72	10.77	64.97	9.23	90.30	24.93

3.2 Online Estimation of Prediction Intervals with LUBE and BLM

The proposed online learning scheme (Eq. 15) is compared against batch (offline) estimation using both the LUBE and BLM approaches. Initially, LUBE is used with $N_i = 100$ initial training points and $N_w = 10$, subsequently with $N_i = 1000$ and $N_w = 100$ and finally BLM is used with $N_i = 100$ and $N_w = 10$. In all tests $k = 10$ sliding windows are considered, and each of the three cases is repeated

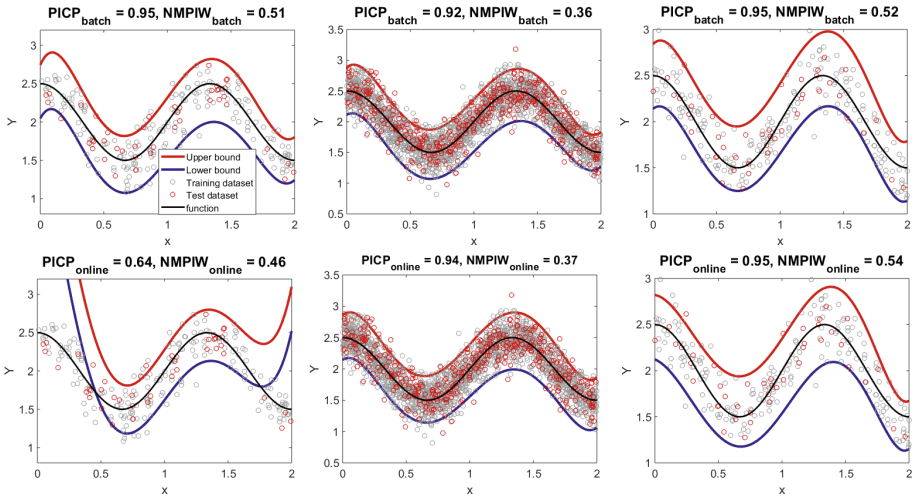

Fig. 4. PIs constructed using batch (*top row*) and online (*bottom row*) estimation. Online PI estimation is tested using LUBE on $N_i = 100$ and $N_w = 10$ training points (*left column*), using LUBE on $N_i = 1000$ and $N_w = 100$ training points (*middle column*) and using BLM on $N_i = 100$ and $N_w = 10$ training points (*right column*).

Table 2. Average characteristics of the PIs constructed using the LUBE and BLM methods, based on batch or online approximation.

	LUBE ($N_w = 10$)		LUBE ($N_w = 100$)		BLM ($N_w = 10$)	
Estimation	PICP(%)	NMPIW(%)	PICP(%)	NMPIW(%)	PICP(%)	NMPIW(%)
Batch	94.95	54.80	94.84	42.64	92.50	46.60
Online	76.14	64.41	95.95	43.94	89.57	43.93

10 times. For batch approximation all $N_i + kN_w$ training points are used for PI optimisation. Representative results are presented in Fig. 4 and average PI indices in Table 2.

When LUBE is used with only $N_w = 10$ training points, online results are suboptimal compared to batch approximation. This is due to the fact that LUBE's accuracy suffers when only sparse data is available (as demonstrated in Fig. 3 and Table 1). This issue can be alleviated by increasing the number of training points ($N_w = 100$), in which case online estimation with LUBE is able to provide very similar PIs to batch estimation, and in a much more efficient way. Alternatively, BLM is able to provide very similar PIs through online and batch estimation without increasing the number of training points as it is designed to provide reliable bounds even when trained on sparse data.

It is worth noting that the proposed learning scheme can easily be adjusted to accommodate the needs of the specific application. For example, the relative contribution of the current sliding window could be increased in cases where recent measurements are considered more critical than past measurements.

4 Conclusions

Combining Bootstrap with LUBE method enables BLM to present improved characteristics in terms of coverage, compared to both Bootstrap and LUBE approaches. In particular, BLM can provide high-coverage PIs even when limited data are available, clearly outperforming the LUBE approach. The results highlight the fact that even commonly used methods such as Bootstrap might provide unreliable PIs when the bounds are based on limited or sparse data, an issue that should be carefully considered by ANN practitioners. Finally, extending BLM to online approximation constitutes a significant improvement, as it enables the efficient and reliable construction of PIs even when approximating dynamically changing processes.

Acknowledgements. This work has been supported by the European Union's Horizon 2020 Research and Innovation Programme under grant agreement No 739551 (KIOS CoE) and from the Republic of Cyprus through the Directorate General for European Programmes, Coordination and Development.

References

1. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Heidelberg (2006)
2. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd edn. Springer, New York (2009). https://doi.org/10.1007/978-0-387-84858-7
3. Hwang, J.T.G., Ding, A.A.: Prediction intervals for artificial neural networks. J. Am. Stat. Assoc. **92**, 748–757 (1997)
4. Davidson, A.C., Hinkley, D.V.: Bootstrap Methods and Their Application. Cambridge University Press, Cambridge (2013)
5. Khosravi, A., Nahavandi, S., Creighton, D., Atiya, A.F.: Lower upper bound estimation method for construction of neural network-based prediction intervals. IEEE Trans. Neural Netw. **22**, 337–346 (2011)
6. Reppa, V., Polycarpou, M.M., Panayiotou, C.G.: Adaptive approximation for multiple sensor fault detection and isolation of nonlinear uncertain systems. IEEE Trans. Neural Netw. Learn. Syst. **25**, 137–153 (2014)
7. Zhang, C., Wei, H., Xie, L., Shen, Y., Zhang, K.: Direct interval forecasting of wind speed using radial basis function neural networks in a multi-objective optimization framework. Neurocomputing **205**, 53–63 (2016)
8. Ye, L., Zhou, J., Gupta, H.V., Zhang, H., Zeng, X., Chen, L.: Efficient estimation of flood forecast prediction intervals via single and multi-objective versions of the LUBE method. Hydrol Process. **30**, 2703–2716 (2016)
9. Pearce, T., Zaki, M., Brintrup, A., Neely, A.: High-quality prediction intervals for deep learning: a distribution-free, ensembled approach. In: 35th International Conference on Machine Learning. arXiv:1802.07167v3 (2018)
10. Khosravi, A., Nahavandi, S., Creighton, D.: Prediction intervals for short-term wind farm power generation forecasts. IEEE Trans. Sustain. Energy **4**, 602–610 (2013)

11. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. In: 31st Conference on Neural Information Processing Systems (2017)
12. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**, 182–197 (2002)