Constructing Optimal Prediction Intervals by Using Neural Networks and Bootstrap Method

Abbas Khosravi, *Member, IEEE*, Saeid Nahavandi, *Senior Member, IEEE*, Dipti Srinivasan, *Senior Member, IEEE*, and Rihanna Khosravi

***Abstract*— This brief proposes an efficient technique for the construction of optimized prediction intervals (PIs) by using the bootstrap technique. The method employs an innovative PI-based cost function in the training of neural networks (NNs) used for estimation of the target variance in the bootstrap method. An optimization algorithm is developed for minimization of the cost function and adjustment of NN parameters. The performance of the optimized bootstrap method is examined for seven synthetic and real-world case studies. It is shown that application of the proposed method improves the quality of constructed PIs by more than 28% over the existing technique, leading to narrower PIs with a coverage probability greater than the nominal confidence level.**

***Index Terms*— Bootstrap, uncertainty quantification.**

I. Introduction

Prediction intervals (PIs) are a promising tool for quantifying effects of uncertainties on predicted values. Two features make PIs richly informative and useful for analysis and decision making. First, relatively wide PIs mean that the predicted values are less reliable and should be used carefully. There is a high level of uncertainty in data in these cases that its effects cannot be eliminated from prediction process. Second, PIs have an indication of their accuracy called the confidence level. A $(1-\alpha)\%$ confidence level means that in an infinite run, properly constructed PIs will cover the targets in $(1-\alpha)\%$ of cases. Availability of such a measure provides decision-makers and analysts with an indicative measure related to the quality of PIs, while the predicted values lack it.

Consider finite pairs of inputs and corresponding output are given in the form of $\{(\mathbf{x}_i, t_i)\}_{i=1}^{N}$, where $\mathbf{x}_i \in R^m$ is a random input vector with m real-number components and $t_i \in R$ is a random target vector with one component. Mathematically, a PI with a $100(1-\alpha)\%$ confidence level constructed for the ith target is a random interval written as $\hat{\mathbf{I}}_\alpha(\mathbf{x}_i) = [\hat{q}_{\alpha/2}(\mathbf{x}_i), \hat{q}_{1-\alpha/2}(\mathbf{x}_i)]$. The lower $(\hat{q}_{\alpha/2}(\mathbf{x}_i))$ and upper $(\hat{q}_{1-\alpha/2}(\mathbf{x}_i))$ points are predictive quantiles at level $\alpha/2$ and $1-\alpha/2$.

Considering a stochastic process, the ith measured target t_i can be represented as

$$t_i = y(\mathbf{x}_i) + \epsilon(\mathbf{x}_i) = f(\mathbf{x}_i, \Theta) + \epsilon(\mathbf{x}_i) \quad (1)$$

where $y(\mathbf{x}_i)$ is the true regression mean and $\epsilon(\mathbf{x}_i)$ is the additive noise (random variable) with a zero expectation. $f(\mathbf{x}_i, \Theta)$ is a mapping between the input variables $\mathbf{x}_i$ and the true regression mean $y(\mathbf{x}_i)$. According to [1], a trained neural network (NN) model can capture the characteristics of the conditional expected value of targets. Therefore, the output of the NN model $\hat{y}(\mathbf{x}_i) = f(\mathbf{x}_i, \hat{\Theta})$ is an estimate of the true regression mean $y(\mathbf{x}_i)$

$$\hat{y}(\mathbf{x}_i) = E(t_i|\mathbf{x}_i). \quad (2)$$

Consequently, the prediction error can be written as

$$t_i - \hat{y}(\mathbf{x}_i) = [y(\mathbf{x}_i) - \hat{y}(\mathbf{x}_i)] + \epsilon(\mathbf{x}_i). \quad (3)$$

PIs quantify the uncertainty associated with the difference between the measured value, t_i, and the predicted value, $\hat{y}(\mathbf{x}_i)$. Because of the statistical independence of two terms in (3), the total variance associated with the model outcome $\sigma_t^2(\mathbf{x}_i)$ can be expressed as

$$\sigma_t^2(\mathbf{x}_i) = \sigma_{\hat{y}}^2(\mathbf{x}_i) + \sigma_\epsilon^2(\mathbf{x}_i) \quad (4)$$

where $\sigma_{\hat{y}}^2(\mathbf{x}_i)$ is the variance of the model mis-specification uncertainty. This includes errors related to model structure selection and parameter estimation. $\sigma_\epsilon^2(\mathbf{x}_i)$ is also the measure of noise variance. PIs considers both the uncertainty in model structure and noise in data. Upon proper estimation of these values, PIs can be constructed for the outcomes of NN models.

A variety of techniques have been proposed in literature for construction of PIs for predicted values by NNs: the delta technique [2], the optimized delta technique [3], the Bayesian method [1], the mean-variance estimation method [4], [5], the bootstrap method [6], and the lower upper bound estimation method [7]–[9].

The focus of this brief is on the bootstrap technique for construction of PIs for outcomes of NN models. The research objective is to improve the quality of constructed PIs in terms of their width and coverage probability. In fact, this brief aims to make bootstrap-based PIs narrower than traditional PIs [6] without compromising their coverage probability. These optimized PIs will be more informative and useful for decision making than intact PIs. This will be achieved through an innovative new model development technique applied to NNs used for estimation of the target variance. This is the key contribution of this brief and its main difference with existing literature in this field [4], [6]. Performance of the proposed method will be examined for different case studies. Quantitative measures will be used for assessing the quality of optimized bootstrap-based PIs.

The rest of this brief is organized as follows. Section II describes the bootstrap method for construction of PIs. PI assessment measures are briefly discussed in Section III. The proposed method for optimizing the quality of bootstrap-based PIs is introduced in Section IV. Simulation results are represented in Section V. Some guidelines for future work are discussed in Section VI. Section VII concludes this brief with a conclusion and some remarks.

II. Bootstrap Method for PI Construction

The bootstrap is a data resampling technique that aims to approximate an unknown distribution by an empirical distribution [10]. Taking advantage of computational resources, the bootstrap method has emerged as a powerful tool for supporting decision making and inferential processes. In the paired bootstrap method, B training data sets are uniformly resampled from the original data set

Manuscript received March 19, 2014; revised August 21, 2014; accepted August 27, 2014. Date of publication September 10, 2014; date of current version July 15, 2015.

A. Khosravi, S. Nahavandi, and R. Khosravi are with the Centre for Intelligent Systems Research, Deakin University, Geelong 3216, Australia (e-mail: abbas.khosravi@deakin.edu.au; saeid.nahavandi@deakin.edu.au; rihanna.khosravi@deakin.edu.au).

D. Srinivasan is with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 119077 (e-mail: dipti@nus.edu.sg).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2014.2354418

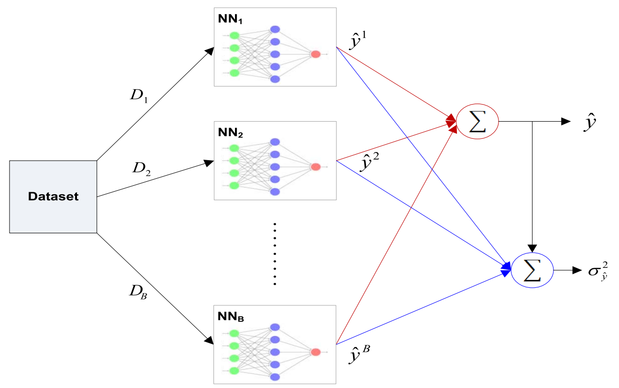

Fig. 1. Ensemble of B NN models used by the bootstrap method.

with replacement, $D_b = \{(\mathbf{x}_i^*, t_i^*)\}_{i=1}^{N^*}$, $b = 1, \ldots, B$. The method estimates the variance caused by model misspecification, $\sigma^2_{\hat{y}}(\mathbf{x}_i)$, by building B NN_y models (Fig. 1) [6]. The true regression is estimated by averaging the point forecasts of B models

$$\hat{y}(\mathbf{x}_i) = \frac{1}{B}\sum_{b=1}^{B} \hat{y}_b(\mathbf{x}_i). \tag{5}$$

where $\hat{y}_b(\mathbf{x}_i)$ is the prediction of the ith sample generated by the bth bootstrap model. Then, model misspecification variance is estimated using the variance of B model outcomes

$$\sigma^2_{\hat{y}}(\mathbf{x}_i) = \frac{1}{B-1}\sum_{b=1}^{B}(\hat{y}_b(\mathbf{x}_i) - \hat{y}(\mathbf{x}_i))^2. \tag{6}$$

This variance is mainly due to the random initialization of parameters and using different data sets for training NNs.

To construct PIs, we also need to estimate the variance of errors, $\sigma^2_{\epsilon}(\mathbf{x}_i)$. The idea is to develop a separate individual NN model, called NN$_\epsilon$ to provide an estimate of $\sigma^2_{\epsilon}(\mathbf{x}_i)$ when presented with an input vector. The transfer function of the output unit of NN$_\epsilon$ is assumed to be exponential instead of a linear transfer function to ensure that the variance is always positive.

The target variance conditioned on input vector can be expressed as

$$\sigma_t^2(\mathbf{x}_i) = E[(t_i - E[t_i \mid \mathbf{x}_i])^2 \mid \mathbf{x}_i]. \tag{7}$$

According to (2), we can rewrite $\sigma_t^2(\mathbf{x}_i)$ as

$$\sigma_t^2(\mathbf{x}_i) = E[(t_i - \hat{y}(\mathbf{x}_i))^2 \mid \mathbf{x}_i] \tag{8}$$

and using (4) we have

$$\sigma_\epsilon^2(\mathbf{x}_i) = E[(t_i - \hat{y}(\mathbf{x}_i))^2 \mid \mathbf{x}_i] - \sigma^2_{\hat{y}}(\mathbf{x}_i). \tag{9}$$

Recall that the output of a trained NN is an estimate of the expected value of target values conditioned on input variables. Therefore, a NN$_\epsilon$ can be trained in a supervised manner to estimate $\sigma_\epsilon^2(\mathbf{x}_i)$ in (9) by using $\mathbf{x}_i$ as its input.

According to (9), a set of variance squared residuals is developed

$$r^2(\mathbf{x}_i) = \max\big((t_i - \hat{y}(\mathbf{x}_i))^2 - \sigma^2_{\hat{y}}(\mathbf{x}_i), 0\big) \tag{10}$$

where $\hat{y}(\mathbf{x}_i)$ and $\sigma^2_{\hat{y}}(\mathbf{x}_i)$ are obtained from (5) and (6). These residuals are linked by the set of corresponding inputs to form a new data set

$$D_{r^2} = \{(\mathbf{x}_i, r^2(\mathbf{x}_i)\}_{i=1}^{n}. \tag{11}$$

A new NN model can be indirectly trained to estimate the unknown values of $\sigma_\epsilon^2(\mathbf{x}_i)$, to maximize the probability of observing samples in D_{r^2}. The model can be developed using the maximum likelihood as the cost function [6]

$$C_{\text{BS}} = \frac{1}{2}\sum_{i=1}^{n}\left(\ln\big(\sigma_\epsilon^2(\mathbf{x}_i)\big) + \frac{r^2(\mathbf{x}_i)}{\sigma_\epsilon^2(\mathbf{x}_i)}\right). \tag{12}$$

Using this cost function, an indirect two-phase training technique can be used [4] for adjusting parameters of bootstrap NNs and NN$_\epsilon$. The algorithm needs two data sets, namely D^1_{train} and D^2_{train}, for training B bootstrap NN models, NN$_y$, and one noise variance estimation NN model, NN$_\epsilon$. In Phase I of the training algorithm, bootstrap NN models are trained to estimate t_i. In Phase II, bootstrap NN models are kept unchanged, and D^2_{train} is used for adjusting parameters of NN$_\epsilon$. Adjusting parameters of NN$_\epsilon$ is achieved through minimizing the cost function defined in (12).

Parameters of NN$_\epsilon$ can be updated using the traditional gradient descent-based methods or stochastic-based optimization techniques such as genetic algorithm and simulated annealing.

Once both $\sigma^2_{\hat{y}}(\mathbf{x}_i)$ and $\sigma^2_\epsilon(\mathbf{x}_i)$ are known, the ith PI with a confidence level of $(1-\alpha)\%$ can be constructed [6]

$$\hat{y}(\mathbf{x}_i) \pm t_{1-\frac{\alpha}{2},df}\sqrt{\sigma^2_{\hat{y}}(\mathbf{x}_i) + \sigma^2_\epsilon(\mathbf{x}_i)} \tag{13}$$

where $t_{1-\alpha/2,df}$ is the $1-\alpha/2$ quantile of a cumulative t-distribution function with df degrees of freedom. df is defined as the difference between the number of training samples and the number of parameters of NN models.

The key advantage of bootstrap technique compared with other PI construction techniques is its simplicity. In contrast to the delta and Bayesian techniques, this method does not require calculation of complex matrices, such as Jacobian and Hessian matrices. This makes the method stable and free from singularity problems. Furthermore, the online computational burden of the method is very limited. It only includes point prediction by $B+1$ NNs and calculation of (5) and (6).

III. PI Assessment

The most important characteristic of PIs is their coverage probability. PI coverage probability (PICP) is measured by counting the number of target values covered by the constructed PIs

$$\text{PICP} = \frac{1}{n}\sum_{i=1}^{n}\xi_i(\hat{I}_\alpha(\mathbf{x}_i), t_i) \tag{14}$$

where

$$\xi_i(\hat{I}_\alpha(\mathbf{x}_i), t_i) = \begin{cases} 1 & t_i \in \hat{I}_\alpha(\mathbf{x}_i) \\ \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

where n is the number of samples and $I_\alpha(\mathbf{x}_i)$ is the $100(1-\alpha)\%$ PI.

Although wide PIs have a satisfactory coverage probability greater than the nominal confidence level, they are poorly informative. Therefore, it is essential to assess PIs on the basis of their width. PI-normalized averaged width (PINAW) quantifies the wide constructed PIs

$$\text{PINAW} = \frac{1}{nR}\sum_{i=1}^{n}\left(\hat{q}_{1-\frac{\alpha}{2}}(\mathbf{x}_i) - \hat{q}_{\frac{\alpha}{2}}(\mathbf{x}_i)\right) \tag{16}$$

where R is the target range in observed samples. $\hat{q}_{\alpha/2}(\mathbf{x}_i)$ and $\hat{q}_{1-\alpha/2}(\mathbf{x}_i)$ are, respectively, lower and upper bounds of the ith PI.

Both PICP and PINAW are 1-D measures evaluating the quality of PIs from one aspect. The coverage width-based criterion (CWC) simultaneously evaluates PIs from both coverage probability and width perspectives

$$\text{CWC} = \text{PINAW} + \gamma\,(\text{PICP})\; e^{\eta(\mu-\text{PICP})} \tag{17}$$

where $\gamma\,(\text{PICP})$ is given by

$$\gamma\,(\text{PICP}) = \begin{cases} 0 & \text{PICP} \geq \mu \\ \\ 1 & \text{PICP} < \mu \end{cases} \tag{18}$$

η and μ in (17) are two hyperparameters controlling the location and amount of CWC jump. The CWC provides an effective compromise between informativeness and correctness of PIs. Note that CWC is a negatively oriented unique skill score. The smaller, the better.

IV. Proposed Method

The focus of the proposed method for construction of optimized bootstrap PIs is on training of NN_ϵ for more accurate estimation of $\sigma_\epsilon^2(\mathbf{x}_i)$. The key idea here is to train the NN_ϵ through minimization of a PI-based cost function, rather than the one defined in (12). The ultimate purpose of NN development is construction of PIs. So, it is more reasonable to train NNs for improving the quality of constructed PIs. The CWC measure, as defined in (17), is considered as the cost function for training of NN_ϵ. It is expected that adjustment of parameters of NN_ϵ through minimization of CWC will greatly improve the quality of PIs constructed using the bootstrap technique.

CWC is highly nonlinear, nondifferentiable, and sensitive to small changes in NN parameters. Therefore, we use evolutionary optimization algorithms to minimize it and adjust parameters of $\text{NN}_\epsilon^{\text{opt}}$. Compared with traditional mathematical optimization techniques, evolutionary algorithms offer a number of advantages such as being derivative-free and less-likely to be trapped in local minima. The detailed discussion of the proposed technique is as follows.

1) *Traditional Bootstrap-Based PIs*: PIs are constructed using the traditional bootstrap method described in Section II. Hereafter, these traditional PIs and their corresponding NN parameters are indicated by sub/superscript trad.
2) *Initialization*: After construction of PI_{trad}, parameters of $\text{NN}_\epsilon^{\text{trad}}$ are used as the initial set for $\text{NN}_\epsilon^{\text{opt}}$, that is, $\Theta_\epsilon^{\text{opt}} = \Theta_\epsilon^{\text{trad}}$. Also, optimization algorithm parameters are initialized to reasonable values.
3) *Optimization Process*: Parameters of NN_ϵ are optimally adjusted using the optimization algorithm. Mathematically, $\Theta_\epsilon^{\text{opt}} = \arg\min_\Theta \text{CWC}$. In each iteration of the optimization process, PIs are constructed and assessed using CWC for samples in D_{train}^2. Optimization continues until one of the following termination criteria is satisfied: reaching the maximum number of iterations, no further improvement for a specific number of consecutive iterations, or having a very small CWC.
4) *Examination*: Upon termination of the optimization algorithm, parameters of $\text{NN}_\epsilon^{\text{opt}}$ are set to $\Theta_\epsilon^{\text{opt}}$. Then PIs are constructed for test samples, where $\text{NN}_\epsilon^{\text{opt}}$ is used for more accurate estimation of $\sigma_\epsilon^2(\mathbf{x}_i)$.

A variety of evolutionary optimization algorithms such as genetic algorithm and particle swarm optimization can be applied here for minimization of CWC and optimal training of $\text{NN}_\epsilon^{\text{opt}}$.

It is important to note that proposed method here can also be applied to specialized bootstrapping procedures such as moving block bootstrap [11]–[13]. The concept of optimization based on CWC minimization and its implementation remain the same.

V. Simulation Results

Seven synthetic and real-world case studies are used for examining performance of the proposed method for PI construction (Table I). Case study 1 is a 5-D synthetic function with a highly nonlinear behavior. The 1-D mathematical function in case studies 2–4 is the same: $y = g(x) + \epsilon$, where $g(x) = x^2 + \sin(x) + 2$. x are randomly generated between values -10 and 10. ϵ follows a Gaussian distribution with a zero mean and variance $g(x)/\tau$, where $\tau = 1, 5, 10$. The smaller the τ, the stronger the noise. In contrast to case study 1 where the additive noise is homogenous, it is heterogeneous (heteroscedastic) for case studies 2–4. The other three case studies are from a real-world industrial dryer (dry bulb temperature), power systems (one day ahead load demand forecasting), and a baggage handling system (time required for processing 70% of a flight bags). The level of uncertainty in operation of all these systems is very high because of noise, unknown relationships, and occurrence of probabilistic events.

TABLE I

Summary of Case Studies

Case Study	Samples	Attributes	Reference
#1	500	5	[14]
#2	500	1	[15]
#3	500	1	[15]
#4	500	1	[15]
#5	867	3	[16]
#6	716	5	[17]
#7	272	3	[7] [18] [19]

Available samples are split into three subsets: first and second training sets (D_{train}^1 and D_{train}^2) each account for 40% of samples (totally 80% for training). The test set (D_{test}) consists of 20% of the samples. All variables are preprocessed to have zero mean and unit variance. Single layer NNs are considered for developing B NN_y models and an individual NN_ϵ model. The number of neurons is set to ten for all NN models. The more the attributes, the lower the degrees of freedom. All PIs are constructed with a confidence level of 90%. Values for η and μ are set to 50 and 0.9, respectively, as recommended in [7]. 100 bootstrap NN models are considered for prediction of the ith target and estimation of $\sigma_{\hat{y}}^2(\mathbf{x}_i)$ in (6). We originally set the number of bootstrap models to large values in the order of 100–1000. Soon, we realized that these figures either lead to poor quality PIs or have no effect on the quality of PIs. That is why we conducted some trials and errors to determine an optimal number of models which are suitable for all case studies here. Of course, this approach is arguable as it may not lead to the best possible quality PIs. The values used here for the percentage of samples in D_{train}^1 and D_{train}^2, the number of neurons in the hidden layer of NNs, and the quantity of bootstrap models are indicative. Their best values can be determined through trial and error or using an optimization algorithm. Experiments for each case study are repeated ten times and all results are reported to avoid any misleading judgment. In each replicate, training and test data sets are randomly regenerated and used for construction of PIs.

The modified firefly algorithm is implemented here for minimization of CWC [20], [21]. As an evolutionary optimization method, firefly algorithm has proven to be quite efficient and effective in finding the global minimum. Its superiority over other optimization algorithms including genetic algorithm has been already reported in [22]. The number of fireflies is set to ten for the purpose of this brief.

Table II summarizes the statistics of PICP, PINAW, and CWC measures computed for ten sets of traditional and optimized bootstrap PIs. The key point here is that $\text{PICP}_{\text{trad}}$ and PICP_{opt} are greater than the nominal confidence level, 90%, in 140 replicates of seven case studies (with the exception of a few replicates). The maximum coverage bias is less than 2% for those cases with a PICP smaller than 90%. Also note that mean and median values

TABLE II
STATICS OF PICP, PINAW, AND CWC FOR TEST SAMPLES OF SEVEN CASE STUDIES

Case Study	Statistics	Traditional Bootstrap			Optimized Bootstrap		
		PICP	PINAW	CWC	PICP	PINAW	CWC
#1	Mean	98.8	33.9	33.9	97.6	21.8	21.8
	Median	99.0	29.9	29.9	98.0	21.2	21.2
	Std Dev	1.0	12.2	12.2	1.2	2.3	2.3
	IQR	1.8	7.0	7.0	1.0	2.8	2.8
#2	Mean	97.0	28.4	28.7	93.9	10.2	10.4
	Median	98.5	22.1	22.1	92.5	9.5	9.5
	Std Dev	4.2	16.8	17.1	4.3	3.6	3.4
	IQR	3.0	21.3	21.3	7.3	4.6	4.1
#3	Mean	97.2	40.1	40.1	94.8	25.1	25.1
	Median	98.5	35.6	35.6	94.5	17.8	17.8
	Std Dev	3.2	27.3	27.3	2.7	16.6	16.6
	IQR	5.0	18.8	18.8	2.5	13.0	13.0
#4	Mean	95.8	61.9	61.9	94.3	40.4	40.4
	Median	95.5	52.6	52.6	94.5	33.1	33.1
	Std Dev	3.2	34.0	34.0	3.2	25.0	25.0
	IQR	2.8	40.7	40.7	4.3	12.5	12.5
#5	Mean	93.3	43.5	43.7	90.4	36.4	38.1
	Median	93.4	39.3	39.9	90.8	34.5	37.4
	Std Dev	3.2	12.2	12.0	2.9	4.1	4.1
	IQR	3.9	6.9	6.9	3.6	4.8	7.0
#6	Mean	95.0	32.1	32.5	92.6	22.8	23.4
	Median	95.8	26.4	26.4	92.7	21.8	22.0
	Std Dev	4.2	14.9	14.9	3.1	3.3	4.2
	IQR	4.5	9.1	10.3	3.5	2.2	2.6
#7	Mean	98.3	86.8	86.8	96.3	73.8	74.0
	Median	98.1	82.0	82.0	98.1	70.8	71.7
	Std Dev	1.4	17.0	17.0	3.1	17.4	17.3
	IQR	1.4	13.2	13.2	1.9	9.7	9.7

for $\text{PICP}_{\text{trad}}$ and PICP_{opt} are all well greater than 90% and thus all constructed PIs are valid. This issue, PICP $\geq$ 90%, is practically important, because it makes constructed PIs reliable and trustworthy. In fact, both traditional and optimized bootstrap methods show an acceptable performance from this perspective.

While both constructed PIs are theoretically valid, optimized PIs are more informative. Indeed, PI_{trad} are significantly wider than PI_{opt}. This is well reflected in the statics of $\text{PINAW}_{\text{trad}}$ and PICP_{opt} reported in Table II. Excessive wideness of PI_{trad} compared with PI_{opt} can be observed for case studies 1–4. In all these cases, PICP_{opt} and $\text{PICP}_{\text{trad}}$ are greater than 90%, but the mean and median values of $\text{PINAW}_{\text{opt}}$ are much smaller than $\text{PINAW}_{\text{trad}}$. This is due to the fact that NN_{ϵ} trained using the traditional bootstrap method overestimates $\sigma_{\epsilon}^{2}(\mathbf{x}_i)$, resulting in excessively wide PI_{trad}. In contrast, $\text{NN}_{\epsilon}^{\text{opt}}$ trained using the proposed method more precisely estimates $\sigma_{\epsilon}^{2}(\mathbf{x}_i)$ leading to superior quality PIs compared with traditional PIs. According to the results for the 70 experiments summarized in Tables II, PI_{opt} are much narrower than PI_{trad} with a PICP greater than 90%. Therefore, they carry more precise and reliable information about variation of targets and are practically more useful. We can accordingly conclude that the proposed method more efficiently handles effects of uncertainties on predicted values. This is a result of directly training NN models due to characteristics of PIs.

The other important issue is the consistency of the quality of PI_{opt} compared with PI_{trad}. The standard deviation and interval quantile range of $\text{PINAW}_{\text{opt}}$ and CWC_{opt} are always smaller than their corresponding values for $\text{PINAW}_{\text{trad}}$ and CWC_{trad}. The robustness in minimizing effects of uncertainties and the consistency in generating quality PIs are due to the PI-based training process used for construction of optimized bootstrap PIs.

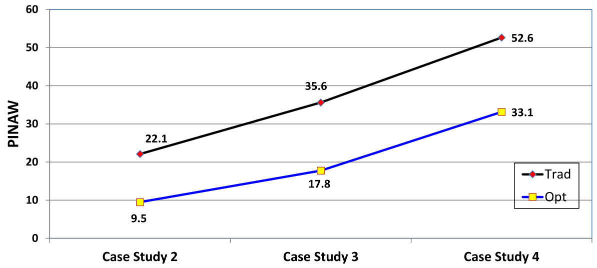

Fig. 2. Averaged values of PINAW for case studies 2–4 with an increasing heterogenous noise.

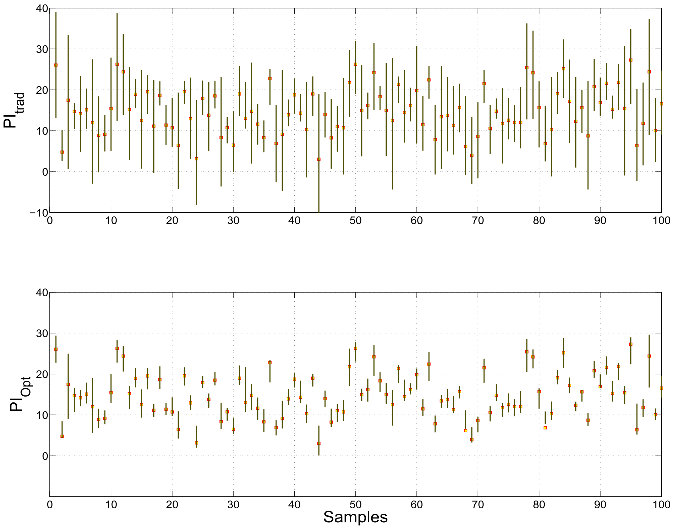

Fig. 3. $\text{PICP}_{\text{trad}}$ and PICP_{opt} for case study 1 in its eighth replicate.

Great results are achieved for case studies 2–4 with the heteroscedastic noise. The method performs so well in the case of heteroscedastic data due to the proper selection of the cost function (CWC) and its minimization procedure (FA). The method considers both quality aspects of PIs for developing variance estimation models: width and coverage probability. As the noise variance is not fixed, PICP is very sensitive to any small change in parameters of the NNs. During the training of NN parameters, parameter values that lead to a low PICP under presence of variable noise are discarded. This makes the trained NN robust against nonconstant variance noise.

It is also important to investigate the effects of increasing the heterogeneous noise on the width of PI_{trad} and PI_{opt} in case studies 2–4. Fig. 2 shows averaged values of $\text{PINAW}_{\text{trad}}$ and $\text{PINAW}_{\text{opt}}$ for these three case studies in their ten replicates. As the noise level (uncertainty in data) goes up from case studies 2–4, the width of intervals is increased. This phenomenon is reasonable as wider intervals have a better coverage probability. However, the width of PI_{opt} is much less increased in comparison with the width of PI_{trad}. The lower increment rate of optimized interval widths indicates that the proposed method much better responds to the heterogeneous noise and more efficiently handles effects of uncertainties.

Fig. 3 shows PI_{trad} and PI_{opt} for case study 1 in its eighth replicate. While PICP_{opt} and $\text{PICP}_{\text{trad}}$ are greater than the nominal confidence level, PI_{opt} are significantly narrower than PI_{trad}. Therefore, these valid PIs are much more useful for decision-making.

Traditional and optimized PIs are shown in the form of two dark and light tubes in Fig. 4 for case study 2 in its replicate three. This figure also shows the actual targets as green spots. Target values

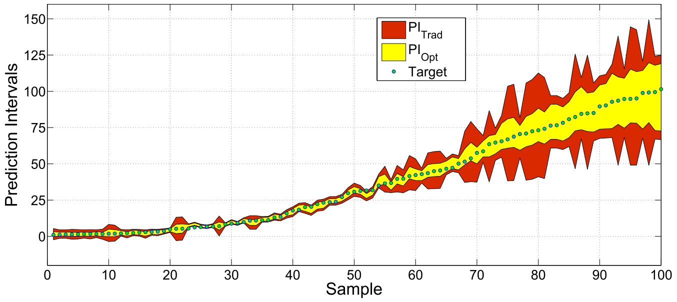

Fig. 4. PI_{trad}, PI_{opt}, and actual targets for case study 2 in its replicate three.

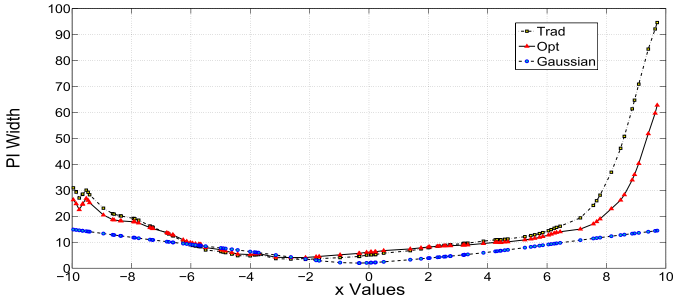

Fig. 5. Width of PI_{trad}, PI_{opt}, and $PI_{Gaussin}$ for case study 3 in its tenth replicate.

are here sorted in an ascending manner. According to the results shown in Table II, both $PICP_{trad}$ and $PICP_{opt}$ are greater than the nominal confidence level, 90%. However, the tube formed by PI_{opt} is encompassed by the tube formed by PI_{trad} for all samples. This visualization clearly shows that PI_{opt} are theoretically valid and practically more informative than PI_{trad}.

The other thing shown in Fig. 4 is that the width of intervals gradually increases as the target values become larger and larger. According to the definition of case study 2, the noise and uncertainty level have a direct relationship with the target values. This relationship is well reflected in the width of intervals where wider intervals are obtained for larger targets. Also note that the growth in the width of intervals is much more consistent for PI_{opt} compared with PI_{trad}.

Fig. 5 shows the width of PI_{trad} and PI_{opt} for case study 2 in its tenth replicate. Also, it displays the width of theoretical Gaussian PIs, called $PI_{Gaussian}$. Assuming σ is known, $PI_{Gaussian}$ are the shortest valid PIs that can be constructed. The horizontal axis indicates the x-values in the range of -10 to 10. When x values are close to zero, the level of uncertainty is low, so constructed PIs should be narrow. When x approaches its extreme values, the level of uncertainty increases, so constructed PIs should become wider than the previous case. Such a pattern is observed for both PI_{trad} and PI_{opt}, in particular to the right side of x-axis. It is also important to note that the width of PI_{opt} is smaller than the width of PI_{trad} for the majority of samples.

The box plot of percentage of improvement, obtained through dividing the difference between CWC_{trad} and CWC_{opt} by CWC_{trad}, is shown in Fig. 6 for ten replicates of seven case studies. It shows the values for the upper quartile (75th percentile), the median, the lower quartile (25th percentile), the mean (indicated by a diamond), and the minimum and maximum values. Mean and median values are close for the majority of case studies. The minimum improvement is obtained for case study 5 where the median value is 7.4%. The maximum improvement is achieved for case study 2 where the median value is 62.5%. The total averaged improvement value for 70 conducted experiments is 28.2%.

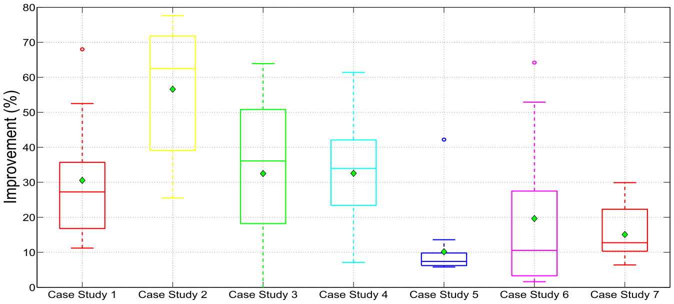

Fig. 6. Improvement values for ten replicates of seven case studies.

VI. Future Work

Performance of the proposed method in this brief for construction of PIs can be further improved. Here are some guidelines for future studies in this field.

1) The method proposed in this brief for construction of optimized PIs is a heuristic one. Further studies are required for investigating its statistical characteristics including its asymptotic behavior. One of the main objectives of those studies will be to establish fundamental properties of the proposed method contributing to its asymptotic behavior. Accomplishment of such studies is a prerequisite for the application of the proposed method for rigorous analysis and decision making.
2) The number of NN models in the bootstrap method (parameter B) has a direct effect on the quality of PIs. A validation set may be considered for systematically analyzing its effects and determining its optimal value.
3) Other evolutionary optimization algorithms can be applied for minimization of the cost function instead of firefly algorithm.

VII. Conclusion

The performance of the traditional NN bootstrap technique for construction of PIs is improved in this brief. In the proposed method, NNs for estimation of the target variance are trained using an innovative PI-based cost function. The cost function covers two important quality aspects of PIs: width and coverage probability. The quality of PIs is improved on average by 28% in 70 experiments with synthetic and real case studies. This achievement indicates the strength and suitability of the proposed method in generating excellent quality PIs.

References

[1] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford, U.K.: Oxford Univ. Press, 1995.
[2] J. T. G. Hwang and A. A. Ding, "Prediction intervals for artificial neural networks," *J. Amer. Statist. Assoc.*, vol. 92, no. 438, pp. 748–757, 1997.
[3] A. Khosravi, S. Nahavandi, and D. Creighton, "Construction of optimal prediction intervals for load forecasting problems," *IEEE Trans. Power Syst.*, vol. 25, no. 3, pp. 1496–1503, Aug. 2010.
[4] D. A. Nix and A. S. Weigend, "Learning local error bars for nonlinear regression," in *Advances in Neural Information Processing Systems*, vol. 7, G. Tesauro, D. Touretzky, and T. Leen, Eds. Cambridge, MA, USA: MIT Press, 1995, pp. 489–496.
[5] A. Khosravi and S. Nahavandi, "An optimized mean variance estimation method for uncertainty quantification of wind power forecasts," *Int. J. Elect. Power Energy Syst.*, vol. 61, pp. 446–454, Oct. 2014.
[6] T. Heskes, "Practical confidence and prediction intervals," in *Neural Information Processing Systems*, vol. 9, T. P. M. Mozer and M. Jordan, Eds. Cambridge, MA, USA: MIT Press, 1997, pp. 176–182.
[7] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya, "Lower upper bound estimation method for construction of neural network-based prediction intervals," *IEEE Trans. Neural Netw.*, vol. 22, no. 3, pp. 337–346, Mar. 2011.

[8] H. Quan, D. Srinivasan, and A. Khosravi, "Particle swarm optimization for construction of neural network-based prediction intervals," *Neurocomputing*, vol. 127, pp. 172–180, Mar. 2014.

[9] H. Quan, D. Srinivasan, and A. Khosravi, "Short-term load and wind power forecasting using neural network-based prediction intervals," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 2, pp. 303–315, Feb. 2014.

[10] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*. New York, NY, USA: Chapman & Hall, 1993.

[11] H. R. Künsch, "The jackknife and the bootstrap for general stationary observations," *Ann. Statist.*, vol. 17, no. 3, pp. 1217–1241, 1989.

[12] A. C. Davison and D. V. Hinkley, *Bootstrap Methods and Their Application*. Cambridge, U.K.: Cambridge Univ. Press, 1997.

[13] P. Bühlmann and H. R. Künsch, "Block length selection in the bootstrap for time series," *Comput. Statist. Data Anal.*, vol. 31, no. 3, pp. 295–310, 1999.

[14] L. Ma and K. Khorasani, "New training strategies for constructive neural networks with application to regression problems," *Neural Netw.*, vol. 17, no. 4, pp. 589–609, May 2004.

[15] A. A. Ding and X. He, "Backpropagation of pseudo-errors: Neural networks that are adaptive to heterogeneous noise," *IEEE Trans. Neural Netw.*, vol. 14, no. 2, pp. 253–262, Mar. 2003.

[16] B. De Moor. DaISy: Database for the identification of systems. Dept. Elect. Eng., ESAT/SISTA, Katholieke Univ. Leuven, Leuven, Belgium. [Online]. Available: http://homes.esat.kuleuven.be/~smc/daisy/, accessed Aug. 2013.

[17] A. Khosravi and S. Nahavandi, "Load forecasting using interval type-2 fuzzy logic systems: Optimal type reduction," *IEEE Trans Ind. Informat.*, vol. 10, no. 2, pp. 1055–1063, May 2014.

[18] A. Khosravi, S. Nahavandi, and D. Creighton, "A prediction interval-based approach to determine optimal structures of neural network metamodels," *Expert Syst. Appl.*, vol. 37, no. 3, pp. 2377–2387, 2010.

[19] A. Khosravi, S. Nahavandi, and D. Creighton, "Prediction interval construction and optimization for adaptive neurofuzzy inference systems," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 5, pp. 983–988, Oct. 2011.

[20] X. S. Yang, *Nature-Inspired Metaheuristic Algorithms*. Frome, U.K.: Luniver Press, 2008.

[21] T. Niknam and A. Kavousifard, "Impact of thermal recovery and hydrogen production of fuel cell power plants on distribution feeder reconfiguration," *IET Generat., Transmiss., Distrib.*, vol. 6, no. 9, pp. 831–843, Sep. 2012.

[22] X.-S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," *Int. J. Bio-Inspired Comput.*, vol. 2, no. 2, pp. 78–84, 2010.