Comprehensive Review of Neural Network-Based Prediction Intervals and New Advances

Abbas Khosravi, *Member, IEEE*, Saeid Nahavandi, *Senior Member, IEEE*, Doug Creighton, *Member, IEEE*, and Amir F. Atiya, *Senior Member, IEEE*

***Abstract*— This paper evaluates the four leading techniques proposed in the literature for construction of prediction intervals (PIs) for neural network point forecasts. The delta, Bayesian, bootstrap, and mean-variance estimation (MVE) methods are reviewed and their performance for generating high-quality PIs is compared. PI-based measures are proposed and applied for the objective and quantitative assessment of each method's performance. A selection of 12 synthetic and real-world case studies is used to examine each method's performance for PI construction. The comparison is performed on the basis of the quality of generated PIs, the repeatability of the results, the computational requirements and the PIs variability with regard to the data uncertainty. The obtained results in this paper indicate that: 1) the delta and Bayesian methods are the best in terms of quality and repeatability, and 2) the MVE and bootstrap methods are the best in terms of low computational load and the width variability of PIs. This paper also introduces the concept of *combinations* of PIs, and proposes a new method for generating combined PIs using the traditional PIs. Genetic algorithm is applied for adjusting the combiner parameters through minimization of a PI-based cost function subject to two sets of restrictions. It is shown that the quality of PIs produced by the combiners is dramatically better than the quality of PIs obtained from each individual method.**

***Index Terms*— Bayesian, bootstrap, delta, mean-variance estimation, neural network, prediction interval.**

I. Introduction

As a BIOLOGICALLY inspired analytical technique, neural networks (NNs) have the capacity to learn and model complex nonlinear relationships. Theoretically, multilayered feedforward NNs are universal approximators and, as such, have an excellent ability of approximating any nonlinear mapping to any degree of accuracy [1]. They do not require *a priori* model to be assumed or *a priori* assumptions to be made on the properties of data [2]. They have been widely employed for modeling, prediction, classification, optimization, and control purposes [3]–[5]. Paliwal *et al.* [6] comprehensively reviewed comparative studies on applications of NNs in accounting and finance, health and medicine, engineering, manufacturing, and marketing. After reviewing over 100 comparative studies, they concluded that NN models outperform their traditional rivals in the majority of cases, no matter the source or type of application.

NNs suffer from two basic limitations despite their popularity. The first problem is the unsatisfactorily low prediction performance when there exists uncertainty in the data. The reliability of point forecasts significantly drops as a result of the prevalence of uncertainty in operation of the system. Machine breakdowns on the shopfloor, unexpected passenger demand in public transportation systems, or abrupt changes in weather conditions in the national energy market may have direct impacts on the throughput, performance, or reliability of the underlying systems. As none of these events can be properly predicted in advance, the accuracy of point forecasts is in doubt and questionable. Even if these are known or predictable, the targets will be multivalued, making predictions prone to error. This weakness is due to the theoretical point that NNs generate averaged values of targets conditioned on inputs. Such a reduction cannot be mitigated through changing the model structure or repeating the training process. Liu [7] describes this problem for a NN application in the semiconductor industry where there are large errors in forecasts of industry growth. Similar stories have been reported in other fields, including, but not limited to, the surface mount manufacturing [8], electricity load forecasting [9], [10], fatigue lifetime prediction [11], financial services [12], hydrologic case studies [13], transportation systems [14]–[16], and baggage handling systems [17].

The second problem of NNs is that they only provide point predictions without any indication of their accuracy. Point predictions are less reliable and accurate if the training data is sparse, if targets are multivalued, or if targets are affected by probabilistic events. To improve the decision making and operational planning, the modeler should be aware of uncertainties associated with the point forecasts. It is important to know how well the predictions generated by NN models match the real targets and how large the risk of un-matching is. Unfortunately, point forecasts do not provide any information about associated uncertainties and carry no indication of their reliability.

To effectively cope with these two fundamental problems, several researchers have studied the development of prediction intervals (PIs) for NN forecasts. A PI is comprised

Manuscript received December 5, 2010; revised April 9, 2011; accepted July 9, 2011. Date of publication July 29, 2011; date of current version August 31, 2011. This work was fully supported by the Centre for Intelligent Systems Research at Deakin University.

A. Khosravi, S. Nahavandi, and D. Creighton are with the Centre for Intelligent Systems Research, Deakin University, Geelong, Vic 3117, Australia (e-mail: abbas.khosravi@deakin.edu.au; saeid.nahavandi@deakin.edu.au; douglas.creighton@deakin.edu.au).

A. F. Atiya is with the Department of Computer Engineering, Cairo University, Cairo 12613, Egypt (e-mail: amir@alumni.caltech.edu).

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/TNN.2011.2162110

of upper and lower bounds that bracket a future unknown value with a prescribed probability called a confidence level $[(1-\alpha)\%]$. The main motivation for the construction of PIs is to quantify the likely uncertainty in the point forecasts. Availability of PIs allows the decision makers and operational planners to efficiently quantify the level of uncertainty associated with the point forecasts and to consider a multiple of solutions/scenarios for the best and worst conditions. Wide PIs are an indication of presence of a high level of uncertainty in the operation of the underlying system. This information can guide the decision makers to avoid the selection of risky actions under uncertain conditions. In contrast, narrow PIs mean that decisions can be made more confidently with less chance of confronting an unexpected condition in the future.

The need to construct PIs for unseen targets is not new, yet the need has intensified dramatically in the last two decades. The corresponding number of papers reporting applications of PIs has also increased in recent years. This is primarily due to the increasing complexity of man-made systems. Examples of such systems are manufacturing enterprises, industrial plants, transportation systems, and communication networks, to name a few. More complexity contributes to high levels of uncertainty in the operation of large systems. Operational planning and scheduling in large systems is often performed on the basis of the point forecasts of the system's future (unseen targets). As discussed above, the reliability of these point forecasts is low and there is no indication of their accuracy.

Delta [18], [19], Bayesian [2], [20], mean-variance estimation (MVE) [21], and bootstrap [22], [23] techniques have been proposed in the literature for construction of PIs. Studies recommending methods for construction and use of PIs have been completed in a variety of disciplines, including transportation [14]–[16], energy market [9], [10], manufacturing [24], and financial services [12]. Although past studies have clarified the need for PI [and confidence interval (CI)] construction, there has been no effort to quantitatively evaluate the performance of the different methods together. The literature predominantly deals with the individual implementation of these methods, but comprehensive review studies are rare. Furthermore, the existing comparative studies are often represented subjectively rather than objectively [24], [25]. Often the coverage probability index, which is the percentage of target values covered by PIs, is used for assessment of the PI quality, and discussion about the width of PIs is either ignored or vaguely presented [8], [9], [11], [14], [18], [21], [26], [27]. As discussed later, this may lead to a misleading judgment about the quality of PIs and selection of wide PIs.

The purpose of this paper is to comparatively examine the performance of the four frequently used methods for construction of PIs. Instead of subjective and imprecise discussions, quantitative measures are proposed for the accurate evaluation of the PI quality. Unlike other studies in this field, the proposed quantitative measures simultaneously evaluate PIs from two perspectives, width and coverage probability. A number of synthetic and real-world case studies are implemented to check the performance of each method. The used datasets feature a different number of attributes, training samples, and data distribution. The methods are judged on the basis of the quality of constructed PIs, repeatability of results, computational requirements, and the variability of PIs against the data uncertainty.

As a major contribution, this paper also proposes a new method for constructing combined PIs using traditionally built PIs. As far as we know, this is the first study that uses the concept of PI combination. A genetic algorithm (GA)-based optimization method is developed for adjusting the parameters of linear combiners. A unique aspect of the combining method is the cost function used for its training. While traditional cost functions are often based on the errors, the proposed one here is a PI-based type. Two sets of restrictions are applied to the combiner parameters to make them theoretically meaningful. It is shown that the proposed combiners outperform the traditional technique for construction of PIs in the majority of case studies.

This paper is structured as follows. In Section II, we review the theoretical backgrounds of the delta, Bayesian, MVE, and bootstrap methods for PI construction. Section III describes quantitative measures for assessment of the PI quality. Simulation results are discussed in Section IV for the 12 case studies. Section V introduces the new method for constructing optimal combined PIs. The effectiveness of the proposed combiners is comparatively examined in Section VI for different case studies. Section VII concludes this paper with a summary of results.

II. Literature Review of PI Construction Methods

It is often assumed that targets can be modeled by

$$t_i = y_i + \epsilon_i \tag{1}$$

where t_i is the ith measured target (totally n targets). ϵ_i is the noise, also called error, with a zero expectation. The error term moves the target away from its true regression mean y_i toward the measured value t_i. In all PI construction methods discussed here, it is assumed that errors are independently and identically distributed. In practice, an estimate of the true regression mean is obtained using a model $\hat{y}_i$. According to this, we have

$$t_i - \hat{y}_i = [y_i - \hat{y}_i] + \epsilon_i. \tag{2}$$

CIs deal with the variance of the first term in the right-hand side of (2). They quantify the uncertainty between the prediction $\hat{y}_i$ and the true regression y_i. CIs are based on the estimation of characteristics of the probability distribution $P(y_i \mid \hat{y}_i)$. In contrast, PIs try to quantify the uncertainty associated with the difference between the measured values t_i and the predicted values $\hat{y}_i$. This relates to the probability distribution $P(t_i \mid \hat{y}_i)$. Accordingly, PIs will be wider than CIs and will enclose them.

If the two terms in (2) are statistically independent, the total variance associated to the model outcome will become

$$\sigma_i^2 = \sigma_{\hat{y}_i}^2 + \sigma_{\hat{\epsilon}_i}^2. \tag{3}$$

The term $\sigma_{\hat{y}_i}^2$ originates from model misspecification and parameter estimation errors, and $\sigma_{\hat{\epsilon}_i}^2$ is the measure of noise variance. Upon proper estimation of these values, PIs can be

constructed for the outcomes of NN models. In the following sections, four traditional methods for approximating these values and construction of PIs are discussed.

A. Delta Method

The delta method has its roots in theories of nonlinear regression [28]. The method interprets an NN as a nonlinear regression model that allows us to apply asymptotic theories for PI construction. Consider that w^* is the set of optimal NN parameters that approximates the true regression function, i.e., $y_i = f(x_i, w^*)$. In a small neighborhood of this set, the NN model can be linearized as

$$\hat{y_0} = f(x_0, w^*) + g_0^T(\hat{w} - w^*). \tag{4}$$

g_0^T is the NN output gradient against the network parameters w^*, and

$$g_0^T = \left[\frac{\partial f(x_0, w^*)}{\partial w_1^*} \quad \frac{\partial f(x_0, w^*)}{\partial w_2^*} \quad \cdots \frac{\partial f(x_0, w^*)}{\partial w_p^*}\right] \tag{5}$$

where p indicates the number of NN parameters. In practice, the NN parameters, i.e., $\hat{w}$, are adjusted through minimization of the sum of squared error (SSE) cost function. Under certain regularity conditions, it can be shown that $\hat{w}$ is very close to w^*. Accordingly, we have

$$\begin{aligned} t_0 - \hat{y}_0 &\simeq [y_0 + \epsilon_0] - [f(x_0, w^*) + g_0^T(\hat{w} - w^*)] \\ &= \epsilon_0 + g_0^T(\hat{w} - w^*) \end{aligned} \tag{6}$$

and so

$$var(t_0 - \hat{y}_0) = var(\epsilon_0) + var(g_0^T(\hat{w} - w^*)). \tag{7}$$

Assuming that the error terms are normally distributed ($\epsilon \approx N(0, \sigma_\epsilon^2)$), the second term in the right-hand side of (7) can be expressed as

$$\sigma_{\hat{y}_0}^2 = \sigma_\epsilon^2 \ g_0^T(F^T \ F)^{-1} g_0. \tag{8}$$

F in (8) is the Jacobian matrix of the NN model with respect to its parameters computed for the training samples

$$F = \begin{bmatrix} \frac{\partial f(x_1,\hat{w})}{\partial \hat{w}_1} & \frac{\partial f(x_1,\hat{w})}{\partial \hat{w}_2} & \cdots & \frac{\partial f(x_1,\hat{w})}{\partial \hat{w}_p} \\ \frac{\partial f(x_2,\hat{w})}{\partial \hat{w}_1} & \frac{\partial f(x_2,\hat{w})}{\partial \hat{w}_2} & \cdots & \frac{\partial f(x_2,\hat{w})}{\partial \hat{w}_p} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f(x_n,\hat{w})}{\partial \hat{w}_1} & \frac{\partial f(x_n,\hat{w})}{\partial \hat{w}_2} & \cdots & \frac{\partial f(x_n,\hat{w})}{\partial \hat{w}_p} \end{bmatrix}. \tag{9}$$

By replacing (8) in (7), the total variance can be expressed as

$$\sigma_0^2 = \sigma_\epsilon^2(1 \ + \ g_0^T(F^T \ F)^{-1} g_0). \tag{10}$$

An unbiased estimate of σ_ϵ^2 can be obtained from

$$s_\epsilon^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(t_i - \hat{y}_i\right)^2. \tag{11}$$

According to this, the $(1-\alpha)\%$ PI for $\hat{y}_i$ is computed as detailed in [18]

$$\hat{y}_0 \pm t_{n-p}^{1-\frac{\alpha}{2}} s_\epsilon \sqrt{1 + g_0^T(F^T F)^{-1} g_0} \tag{12}$$

where $t_{n-p}^{1-\frac{\alpha}{2}}$ is the $(\alpha/2)$ quantile of a cumulative t-distribution function with $n-p$ degrees of freedom.

A weight decay cost function (WDCF) can be used instead of the SSE cost function to minimize the overfitting problem and to improve the NN generalization power. The WDCF tries to keep the magnitude of the NN parameters as small as possible

$$WDCF = SSE \ + \ \lambda \ w^T \ w. \tag{13}$$

De Veaux *et al.* [19] derived the following formula for PI construction for the case that NNs are trained using the WDCF:

$$\hat{y}_0 \pm t_{n-p}^{1-\frac{\alpha}{2}} s_\epsilon \sqrt{1 + g_0^T(F^T F + \lambda I)^{-1}(F^T F)(F^T F + \lambda I)^{-1} g_0}. \tag{14}$$

Inclusion of λ in (14) improves the reliability and quality of PIs, particularly for cases that $F^T F$ is nearly singular. We will return to the singularity problem again in the simulation result section.

Computationally, the delta technique is more demanding in its development stage than its application stage. Both the Jacobian matrix (F) and s_ϵ^2 should be calculated and estimated offline. For PI construction for a new sample, we need to calculate g_0^T and replace it in (12) or (14). With the exception of this, other calculations are virtually very simple.

The estimation of $\sigma_{\hat{y}_0}^2$, and the calculation of the gradient and Jacobian matrices can be potential sources of error in the construction of PIs using (12) or (14) [29]. Also, the literature does not discuss how λ affects the quality of PIs and how its optimal value can be determined. The delta method assumes that s_ϵ^2 is constant for all samples (noise homogeneity). However, there are cases in practice in which the level of noise is systematically correlated by the target magnitude or the set of NN inputs. Therefore, it is not unexpected that the delta method will generate low-quality PIs for these cases.

B. Bayesian Method

In the Bayesian training framework, NNs are trained on the basis of a regularized cost function

$$E(w) = \rho E_w + \beta E_D \tag{15}$$

where E_D is SSE and E_w is the sum of squares of the network weights ($w^T w$). ρ and β are two hyperparameters of the cost function determining the training purpose. The method assumes that the set of NN parameters w is a random set of variables with assumed *a priori* distributions. Upon availability of a training dataset and an NN model, the density function of the weights can be updated using the Bayes' rule

$$P(w|D, \rho, \beta, M) = \frac{P(D|w, \beta, M)P(w|\rho, M)}{P(D|\rho, \beta, M)} \tag{16}$$

where M and D are the NN model and the training dataset. $P(D|w, \beta, M)$ and $P(w|\rho, M)$ are the likelihood function of data occurrence and the prior density of parameters, respectively. Representing our knowledge, $P(D|\rho, \beta, M)$ is a normalization factor enforcing that total probability is 1.

Assuming that ϵ_i are normally distributed and $P(D|w,\beta,M)$ and $P(w|\rho,M)$ have normal distributions, we can write

$$P(D|w,\beta,M)=\frac{1}{Z_D(\beta)}e^{-\beta E_D} \tag{17}$$

and

$$P(w|\rho,M)=\frac{1}{Z_w(\rho)}e^{-\rho E_w} \tag{18}$$

where $Z_D(\beta)=(\pi/\beta)^{(n/2)}$ and $Z_w(\rho)=(\pi/\rho)^{(p/2)}$. n and p are the number of training samples and NN parameters, respectively. By substituting (17) and (18) into (16), we have

$$P(w|D,\rho,\beta,M)=\frac{1}{Z_F(\beta,\rho)}e^{-(\rho E_w+\beta E_D)}. \tag{19}$$

The purpose of NN training is to maximize the posterior probability $P(w|D,\rho,\beta,M)$. This maximization corresponds to the minimization of (15), and that makes the connection between the Bayesian methodology and regularized NNs. By taking derivatives with respect to the logarithm of (19) and setting it equal to zero, the optimal values for β and ρ are obtained [2], [20]

$$\beta^{MP}=\frac{\gamma}{E_D\left(w^{MP}\right)} \tag{20}$$

$$\rho^{MP}=\frac{n-\gamma}{E_w\left(w^{MP}\right)} \tag{21}$$

where $\gamma=p-2\rho^{MP}tr(H^{MP})^{-1}$ is the so-called effective number of NN parameters, and p is the total number of NN model parameters. w^{MP} is the most probable value of the NN parameters. H^{MP} is the hessian matrix of $E(w)$

$$H^{MP}=\rho\nabla^2E_w+\beta\nabla^2E_D. \tag{22}$$

Usually, the Levenberg–Marquardt optimization algorithm is applied to approximate the Hessian matrix [30]. Application of this technique for training results in NNs with a variance in their prediction of

$$\begin{aligned}\sigma_i^2&=\sigma_D^2+\sigma_{w^{MP}}^2\\&=\frac{1}{\beta}+\nabla_{w^{MP}}^T\hat{y}_i\ \ (H^{MP})^{-1}\ \ \nabla_{w^{MP}}\hat{y}_i.\end{aligned} \tag{23}$$

While the first term in the right-hand side of (23) quantifies the amount of uncertainty in the training data (the intrinsic noise), the second term corresponds to the misspecification of NN parameters and their contribution to the variance of predictions. These terms are $\sigma_{\hat{\epsilon}_i}^2$ and $\sigma_{\hat{y}_i}^2$ in (3), respectively. As the total variance of the ith future sample is known, a $(1-\alpha)\%$ PI can be constructed

$$\hat{y}_i\pm z^{1-\frac{\alpha}{2}}\left(\frac{1}{\beta}+\nabla_{w^{MP}}^T\hat{y}_i\ \ (H^{MP})^{-1}\ \ \nabla_{w^{MP}}\hat{y}_i\right)^{\frac{1}{2}} \tag{24}$$

where $z^{1-(\alpha/2)}$ is the $1-(\alpha/2)$ quantile of a normal distribution function with zero mean and unit variance. Also $\nabla_{w^{MP}}^T\hat{y}_i$ is the gradient of the NN output with respect to its parameters w^{MP}.

The Bayesian method for PI construction has a strong mathematical foundation. NNs trained using the Bayesian learning technique typically have a better generalization power than other networks. This minimizes the effects of $\sigma_{w^{MP}}^2$ in (23) on the width of PIs. Furthermore, it eliminates the hassle for optimal determination of the regularizing parameters. The Bayesian method is computationally demanding in the development stage, similar to the delta technique. It requires calculation of the Hessian matrix in (22), which is time consuming and cumbersome for large NNs and datasets. However, the computational load is decreased in the PI construction stage as we only need to calculate the gradient of NN output.

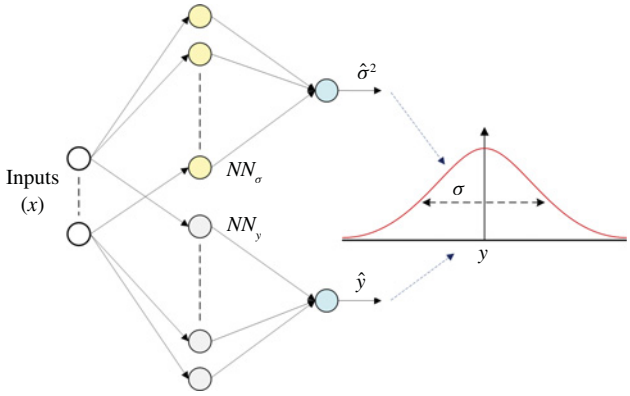

Fig. 1. Schematic of the MVE method for construction of PIs.

C. MVE Method

The MVE method was originally proposed by Nix and Weigend [21] for construction of PIs. This method also assumes that errors are normally distributed around the true mean of targets, $y(x)$. Therefore, PIs can easily be constructed if the parameters of this distribution (mean and variance) are known. Both delta and Bayesian techniques use a fixed target variance for PI construction. In contrast to these techniques, the MVE method estimates the target variance using a dedicated NN. This consideration provides enough flexibility for estimating the heteroscedastic variation of the targets. The dependence of the target variance on the set of inputs is the fundamental assumption of this method for PI construction.

Fig. 1 shows a schematic representation of the MVE method. The set of inputs to the NN models can be identical or different. There is no limitation on the size and structure of the two networks. Consideration of an exponential activation function for the unit corresponding to $\hat{\sigma}^2$ guarantees strictly positive estimates of variance. Assuming that NN_y accurately estimates $y(x)$, the approximate PIs with a $(1-\alpha)\%$ confidence level can be constructed as follows:

$$\hat{y}(x,w_y)\pm z_{1-\frac{\alpha}{2}}\sqrt{\hat{\sigma}^2(x,w_\sigma)} \tag{25}$$

where w_y and w_σ are parameters of NN models for estimation of $\hat{y}$ and σ^2, respectively. The target variance values σ_i are not known *a priori*. This excludes the application of the error-based minimization techniques for training of NN_σ. Instead, a maximum likelihood estimation approach can be applied for training these NNs. Based on the assumption of normally distributed errors around y_i, the data conditional distribution

will be

$$P(t_i \mid x_i, NN_y, NN_\sigma) = \frac{1}{\sqrt{2\pi\hat{\sigma}_i^2}} \; e^{-\frac{(t_i-\hat{y}_i)^2}{2\hat{\sigma}_i^2}}. \tag{26}$$

Taking the natural log of this distribution and ignoring the constant terms result in the following cost function, which will be minimized for all samples:

$$C_{MVE} = \frac{1}{2}\sum_{i=1}^{n}\left[ln(\hat{\sigma}_i^2) + \frac{\left(t_i-\hat{y}_i\right)^2}{\hat{\sigma}_i^2}\right]. \tag{27}$$

Using this cost function, an indirect three-phase training technique was proposed in [21] for simultaneously adjusting w_y and w_σ. The proposed algorithm needs two datasets, namely, D_1 and D_2, for training NN_y and NN_σ. In Phase I of the training algorithm, NN_y is trained to estimate y_i. Training is performed through minimization of an error-based cost function for the first dataset D_1. To avoid the overfitting problem, D_2 can be used as the validation set and for terminating the training algorithm. Nothing is done with NN_σ in this phase. In Phase II, w_y are fixed, and D_2 is used for adjusting parameters of NN_σ. Adjusting of w_σ is achieved through minimizing the cost function defined in (27). NN_y and NN_σ are used to approximate y_i and σ^2 for each sample, respectively. The cost function is then evaluated for the current set of NN_σ weights w_σ. These weights then are updated using the traditional gradient-descent-based methods. D_1 can also be applied as the validation set to limit the overfitting effects. In Phase III, two new training sets are resampled and applied for simultaneous adjustment of both network parameters. The retraining of NN_y and NN_σ is again carried out through minimization of (27). As before, one of the sets is used as the validation set.

The main advantage of this method is its simplicity, and that there is no need to calculate complex derivatives and the inversion of the Hessian matrix. Nonstationary variances can be approximated by employing more complex structures for NN_σ or through proper selection of the set of inputs.

The main drawback of the MVE method is that it assumes NN_y precisely estimates the true mean of the targets, i.e., y_i. This assumption can be violated in practice because of a variety of reasons, including the existence of a bias in fitting the data due to a possible underspecification of the NN model or due to omission of important attributes affecting the target behavior. In these cases, the NN generalization power is weak, resulting in accumulation of uncertainty in the estimation of y_i. Therefore, the constructed PIs using (25) will underestimate (or overestimate) the actual $(1-\alpha)\%$ PIs, leading to a low coverage probability.

Assuming $\hat{y}_i \simeq y_i$ implies that the MVE method only considers one portion of the total uncertainty for construction of PIs. The considered variance is only due to errors, not due to misspecification of model parameters (either w_y or w_σ). This can result in misleadingly narrow PIs with a low coverage probability. This critical drawback has been theoretically identified and practically demonstrated in the literature [31].

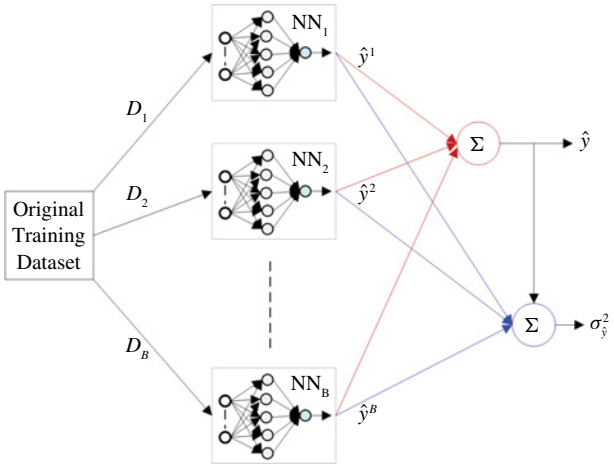

Fig. 2. Ensemble of B NN models used by the bootstrap method.

D. Bootstrap Method

Bootstrap is by far the most common technique documented in the literature for the construction of CIs and PIs. The method assumes that an ensemble of NN models will produce a less biased estimate of the true regression of the targets [23]. As generalization errors of NN models are made on different subsets of the parameter space, the collective decision produced by the ensemble of NNs is less likely to be in error than the decision made by any of the individual NN models. B training datasets are resampled from the original dataset with replacement, $\{D\}_{b=1}^{B}$. The method estimates the variance due to model misspecification, i.e., $\sigma_{\hat{y}}^2$, by building B NN models (Fig. 2). According to this assumption, the true regression is estimated by averaging the point forecasts of B models

$$\hat{y}_i = \frac{1}{B}\sum_{b=1}^{B}\hat{y}_i^b \tag{28}$$

where $\hat{y}_i^b$ is the prediction of the ith sample generated by the bth bootstrap model. Assuming that NN models are unbiased, the model misspecification variance can be estimated using the variance of B model outcomes

$$\sigma_{\hat{y}_i}^2 = \frac{1}{B-1}\sum_{b=1}^{B}\left(\hat{y}_i^b - \hat{y}_i\right)^2. \tag{29}$$

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

CIs can be constructed using the approximation of $\sigma_{\hat{y}_i}^2$ in (29). To construct PIs, we need to estimate the variance of errors, i.e., $\sigma_{\hat{\epsilon}_i}^2$. From (3), $\sigma_{\hat{\epsilon}}^2$ can be calculated as follows:

$$\sigma_{\hat{\epsilon}}^2 \simeq E\left\{\left(t-\hat{y}\right)^2\right\} - \sigma_{\hat{y}}^2. \tag{30}$$

According to (30), a set of variance squared residuals

$$r_i^2 = \max\left((t_i-\hat{y}_i)^2 - \sigma_{\hat{y}_i}^2, 0\right) \tag{31}$$

is developed, where $\hat{y}_i$ and $\sigma_{\hat{y}_i^2}$ are obtained from (28) and (29). These residuals are linked by the set of corresponding inputs to form a new dataset

$$D_{r^2} = \left\{(x_i, r_i^2)\right\}_{i=1}^{n}. \tag{32}$$

A new NN model can be indirectly trained to estimate the unknown values of $\sigma^2_{\hat{\epsilon}_i}$, so as to maximize the probability of observing the samples in D_{r^2}. The procedure for indirect training of this new NN is very similar to the steps of the MVE method described in Section II-C. The training cost function is defined as

$$C_{BS} = \frac{1}{2}\sum_{i=1}^{n}\left[ln(\sigma^2_{\hat{\epsilon}_i}) + \frac{r_i^2}{\sigma^2_{\hat{\epsilon}_i}}\right]. \tag{33}$$

As noted before, the NN output node activation function is selected to be exponential, enforcing a positive value for $\sigma^2_{\hat{\epsilon}_i}$. The minimization of C_{BS} can be done using a variety of methods, including traditional gradient descent methods.

The described bootstrap method is traditionally called the *bootstrap pairs*. There exists another bootstrap method, called *bootstrap residuals*, which resamples the prediction residuals. Further information on this method can be found in [29].

For construction of PIs using the bootstrap method, $B+1$ NN models are required in total. B NN models (assumed to be unbiased) are used for estimation of $\sigma^2_{\hat{y}_i}$ and one model is used for estimation of $\sigma^2_{\hat{\epsilon}_i}$. Therefore, this method is computationally more demanding than other methods in its development stage ($B+1$ times more). However, once the models are trained offline, the online computational load for PI construction is only limited to $B+1$ NN point forecasts. This is in contrast with the claim in the literature that bootstrap PIs are computationally more intensive than other methods [27]. This claim will be precisely verified in the later sections. Simplicity is another advantage of using the Bootstrap method for PI construction. There is no need to calculate complex matrices and derivatives, as required by the delta and Bayesian techniques.

The main disadvantage of the bootstrap technique is its dependence on B NN models. Frequently some of these models are biased, leading to an inaccurate estimation of $\sigma^2_{\hat{y}_i}$ in (29). Therefore, the total variance will be underestimated resulting in narrow PIs with a low coverage probability.

III. PI Assessment Measures

Discussion in the literature on the quality of constructed PIs is often vague and incomplete [8], [9], [11], [14], [18], [21], [26], [27]. Frequently, PIs are assessed from their coverage probability perspective without any discussion about how wide they are. As discussed in [10] and [32], such an assessment is subjective and can lead to misleading results. Here we briefly discuss two indices for quantitative and comprehensive assessment of PIs.

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

$$PICP = \frac{1}{n_{test}}\sum_{i=1}^{n_{test}} c_i \tag{34}$$

where

$$c_i = \begin{cases} 1, & t_i \in [L_i, U_i] \\ 0, & t_i \notin [L_i, U_i] \end{cases} \tag{35}$$

where n_{test} is the number of samples in the test set, and L_i and U_i are lower and upper bounds of the ith PI, respectively. Ideally, PICP should be very close or larger than the nominal confidence level associated to the PIs.

PICP has a direct relationship with the width of PIs. A satisfactorily large PICP can be easily achieved by widening PIs from either side. However, such PIs are too conservative and less useful in practice, as they do not show the variation of the targets. Therefore, a measure is required to check how wide the PIs are. Mean PI width (MPIW) quantifies this aspect of PIs [10]

$$MPIW = \frac{1}{n_{test}}\sum_{i=1}^{n_{test}}(U_i - L_i). \tag{36}$$

MPIW shows the average width of PIs. Normalizing MPIW by the range R of the underlying target allows us to compare PIs constructed for different datasets respectively (the new measure is called NMPIW)

$$NMPIW = \frac{MPIW}{R}. \tag{37}$$

Both PICP and NMPIW evaluate the quality of PIs from one aspect. A combined index is required for the comprehensive assessment of PIs from both coverage probability and width perspectives. The new measure should give a higher priority to PICP, as it is the key feature of PIs determining whether constructed PIs are theoretically correct or not. The coverage width-based criterion (CWC) evaluates PIs from both coverage probability and width perspectives

$$CWC = NMPIW\left(1 + \gamma\,(PICP)\,e^{-\eta(PICP-\mu)}\right) \tag{38}$$

where $\gamma\,(PICP)$ is given by

$$\gamma = \begin{cases} 0, & PICP \geq \mu \\ 1, & PICP < \mu. \end{cases} \tag{39}$$

η and μ in (38) are two hyperparameters controlling the location and amount of CWC jump. These measures can be easily determined on the basis of the level of confidence associated with PIs. μ corresponds to the nominal confidence level associated with PIs and can be set to $1-\alpha$. The design of CWC is based on two principles: 1) if PICP is less than the nominal confidence level $(1-\alpha)\%$, CWC should be large regardless of the width of PIs (measured by NMPIW), and 2) if PICP is greater than or equal to its corresponding confidence level, then NMPIW should be the influential factor. $\gamma\,(PICP)$, as defined in (39), eliminates the exponential term of CWC when PICP is greater or equal to the nominal confidence level.

In the CWC measure, the exponential term penalizes the violation of the coverage probabilities. This is, however, a smooth penalty rather than a hard one, for the following reasons. It is appropriate to penalize the degree of violation, rather than just an abrupt binary penalty. Also, it allows for statistical errors due to the finite samples.

The CWC measure tries to compromise between informativeness and correctness of PIs. As indicated by Halberg *et al.* [33], precision is much more important than ambiguity. Therefore, PIs should be as narrow as possible from the

TABLE I
CASE STUDY DATASETS USED IN THE EXPERIMENTS

Case Study	Target	Samples	Attributes	Reference
1	Five-dimensional function	300	5	[34], [35]
2	Percentage of body fat	252	13	[36]
3	T70	272	2	[17], [37]
4	T90	272	2	[17], [37]
5	Air pollution (NO_2)	500	7	[36]
6	Concrete compressive strength	1030	9	[38]
7	1-D function with heterogenous noise	500	1	[25]
8	Dry-bulb temperature	876	3	[39]
9	Wet-bulb temperature	876	3	[39]
10	Moisture content of raw material	876	3	[39]
11	T_{12}	2500	5	[39]
12	T_{23}	2500	5	[39]

informativeness perspective. However, as discussed above, the narrowness may lead to not bracketing some targets and result in a low coverage probability (low-quality PIs). CWC evaluates PIs from the two conflicting perspectives: informativeness (being narrow) and correctness (having an acceptable coverage probability).

IV. QUANTITATIVE ASSESSMENT AND SIMULATION RESULTS

A. Case Studies

The performance of the delta, Bayesian, MVE, and bootstrap techniques for construction of PIs is compared in this section. Twelve case studies are defined in the proceeding list and used to evaluate the effectiveness of each method. More information about datasets can be found in Table I and cited references. All datasets are available from the authors on request.

1) Data in the first case study comes from a synthetic mathematical function $g(x_1, x_2, x_3, x_4, x_5) = 0.0647\ (12 + 3x_1 - 3.5x_2^2 + 7.2x_3^3)\ (1 + \cos(4\pi x_4))\ (1 + 0.8\ \sin(3\pi x_5))$. A normally distributed noise is added to samples as data uncertainty.
2) This case study includes a dependent variable (the estimated percentage of body fat) and 13 continuous independent variables in 252 men. The aim is to predict body fat percentage using the independent variables.
3) The third case study considers a real-world baggage handling system. The goal is to estimate the time required to process 70% of each flight bags (T70). The level of uncertainty in this system is high due to the frequent occurrence of probabilistic events.
4) Similar to the previous item, this case study attempts to estimate the required time for processing 90% of each flight bags (T90). In practice, prediction of T90 is more difficult than T70 due to a high level of uncertainty affecting it.
5) Case study 5 relates air pollution to the traffic volume and meteorological variables.
6) The relationship between the concrete compressive strength and selected attributes is studied in case study 6.
7) In case study 7, target values are generated through the following model: $y = g(x) + \epsilon$, where $g(x) = x^2 + \sin(x) + 2$. x are randomly generated between values -10 and 10. ϵ follows a Gaussian distribution with mean zero and variance $(g(x)/\tau)$, where $\tau = 1, 5, 10$. The smaller the τ, the stronger the noise. While the additive noise in case study 1 is homogeneous, it is heterogeneous in this case study. As indicated in [25], heterogeneity of the noise ruins the point prediction performance of regression models.
8) Data in this case study comes from a real industrial dryer sampled at ten second intervals. The purpose is to model relationship between the dry bulb temperature and three independent attributes.
9) As for previous case study, the wet bulb temperature is approximated using three inputs.
10) This case study is again related to the case study 8. The modeling goal is to estimate the moisture content of raw materials based on independent inputs.
11) The data in this case study is generated from an industrial winding process. NN models are trained to approximate the tension in the web between reel 1 and 2 (T_{12}) and four inputs.
12) Similar to the previous case study, the goal in this case study is to model the relationship between the tension in the web between reel 2 and 3 (T_{23}) using NN models.

B. Experimental Procedure

Fig. 3 shows the procedure for performing experiments with the 12 case studies in this paper. The data is randomly split into the first training set, D_1, (40%), the second training set, D_2, (40%), and the test set (20%). D_2 is required for the MVE and bootstrap methods for PI construction. The optimal NN structure for each dataset is determined using a fivefold cross-validation technique. Mean absolute percentage errors (MAPEs) are calculated and compared for single- and two-layer NNs to determine the optimal structure.

After determining the NN structure, delta, Bayesian, MVE, and bootstrap methods are used for PI construction for test samples. PIs are constructed with an associated 90% confi-

TABLE II
STATISTICAL CHARACTERISTICS OF CWC FOR PI_{Delta}, PI_{Bays}, PI_{MVE}, AND $PI_{Bootstrap}$ FOR THE 12 CASE STUDIES

Case	Delta			Bayesian			MVE			Bootstrap		
Study	CWC_{Best}	CWC_{Median}	CWC_{SD}	CWC_{Best}	CWC_{Median}	CWC_{SD}	CWC_{Best}	CWC_{Median}	CWC_{SD}	CWC_{Best}	CWC_{Median}	CWC_{SD}
1	82.04	106.55	3645	88.97	120.84	21.30	100.49	123	49 119	73.99	114.44	1638
2	42.64	55.93	32650	42.45	59.52	30.39	64.20	136.64	60.66	64.15	142.98	107.50
3	33.59	64.01	418	29.57	1318	1.2×10^{8}	81.31	110.34	22.23	47.39	103.82	30.24
4	49.73	60.00	102	47.99	82.00	313.43	83.96	131.69	50.22	59.03	116.95	71.58
5	47.23	163.75	640	44.63	60.04	945.39	52	112	55	44.77	94.95	50.38
6	29.98	112.98	114.80	68.01	353.57	10 099	68.63	120.94	34.20	34.19	69.74	41.03
7	6.15	20.49	68.76	7.99	8.47	42.43	48.12	107.59	26.29	33.56	90.13	35.64
8	28.73	100.34	371.40	31.85	36.64	2.31	75.25	107.04	32.91	32.83	44.39	37.87
9	22.24	50.39	58.09	26.59	28.83	76.24	58.95	89.60	36.14	49.06	93.01	36.62
10	55.71	122.48	416.46	72.08	92.12	14.93	84.81	125.12	6×10^{6}	121.02	144.75	99.13
11	51.17	88.39	58.83	56.37	58.81	1.74	65.07	88.54	43166	73.51	98.40	20.57
12	38.50	106.84	63.33	33.04	55.77	11.82	56.51	110.62	63.49	66.86	115.36	51.92

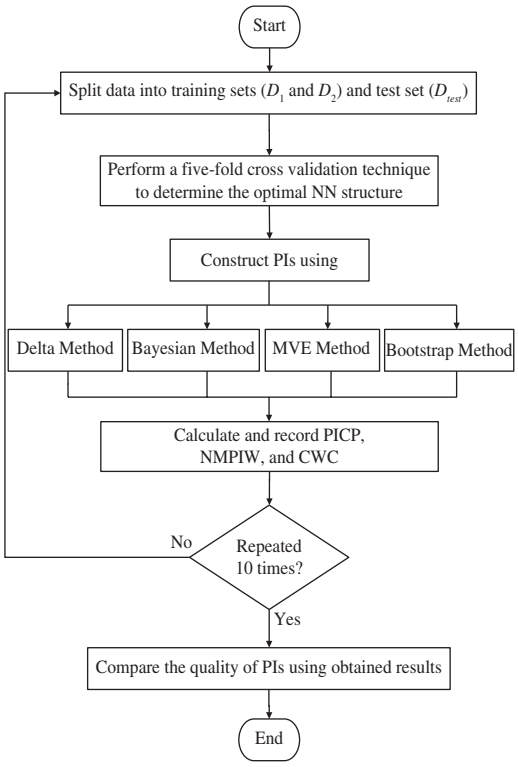

Fig. 3. Experiment procedure for construction and evaluation of PIs.

dence level (α equal to 0.1). η and μ are set to 50 and 0.9. This greatly penalizes PIs with a PICP lower than 90%. PICP, NMPIW, and CWC are computed for the obtained PIs and are recorded for later analysis. PI construction is repeated 10 times using the randomly split datasets and redeveloped NNs. Upon the termination of this loop, performance of the four methods is judged by calculating the statistical characteristics of PICP, NMPIW, and CWC.

PI_{Delta} were first constructed using (12). We later decided to use (14) for PI construction due to the singularity problem and the low quality of generated PIs using (12). λ is equal to 0.9 in all experiments for constructing PI_{Delta}.

A simulated annealing (SA) method [40] is applied for the minimization of cost functions (27) and (33) in the MVE and bootstrap methods. Traditionally, these cost functions have been optimized using gradient-descent-based methods. However, such methods are likely to be trapped in local minima and, therefore, may not find the optimal set of NN parameters. In contrast, SA has been shown to have an excellent efficiency in finding the optimal solution for complex optimization problems [10]. Therefore, it is applied here for minimization of the cost functions and adjustment of the corresponding NN parameters. A geometric cooling schedule with a cooling factor of 0.9 is applied for guiding the optimization process.

C. Results and Discussion

Table II shows the summary of all obtained results for test samples of the 12 case studies using the four PI construction methods. CWC and its statistics are computed for the quantitative assessment and comparison of different methods' performance. Hereafter and for ease in reference, *Delta*, *Bays*, *MVE*, and *Bootstrap* subscripts are used for indicating the delta, Bayesian, MVE, and bootstrap methods for construction of PIs. Also *Best* subscript corresponds to the highest quality PIs constructed using a method in 10 replicates. For median and standard deviation, *Median* and *SD* subscripts are used, respectively.

As per the results in Table II, the delta technique generates the highest quality PIs in 6 out of 12 case studies. CWC_{Best} of the delta technique is either the smallest or second in the rank for 11 out of 12 case studies. PI_{Median} of the delta technique are also smallest for the majority of case studies (either rank 1 or 2 in 10 out of 12 case studies). This indicates that the frequency of generating high-quality PIs using the delta technique is high. However, the method has a large standard deviation, meaning that it may generate low-quality PIs in some cases. These large values are an indication of unreliability of PIs in specific cases. Unreliability of PIs can be due to violation of fundamental assumptions of the delta technique, for instance, noise distribution. An overfitted NN often has a low generalization ability, leading to some constructed PIs not bracketing the corresponding targets. This results in a low PICP and a large CWC. Alternatively, an improperly trained NN has a large prediction error. Therefore,

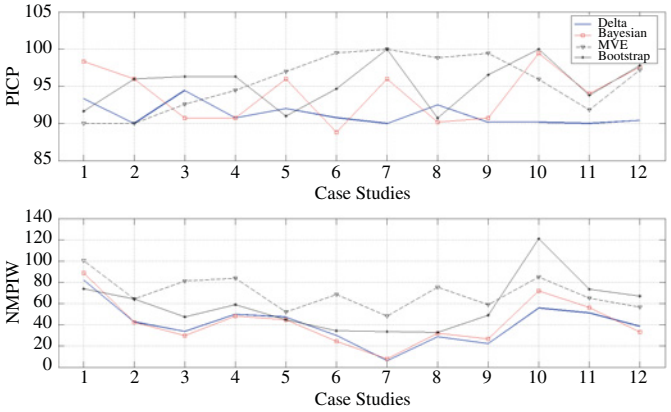

Fig. 4. PICP (top) and NMPIW (bottom) for the best PIs generated by delta, Bayesian, MVE, and bootstrap methods.

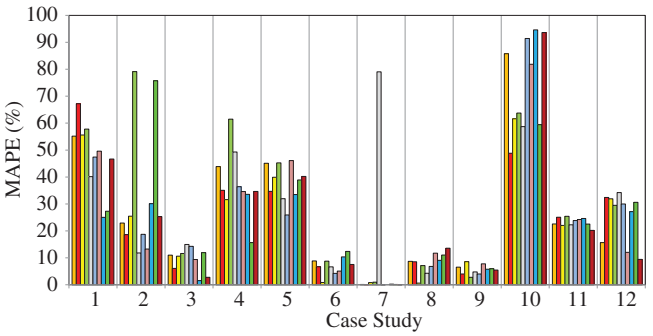

Fig. 5. MAPE for 10 replicates of NN_{Bays} experiments for 12 case studies.

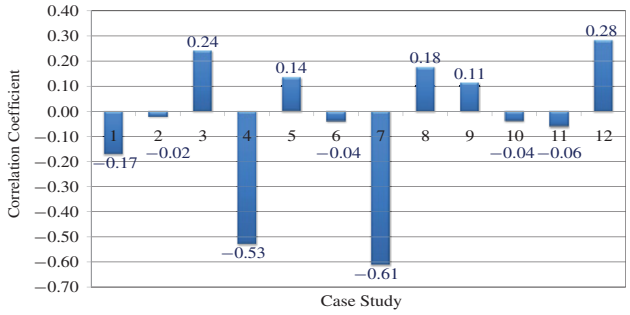

Fig. 6. Correlation coefficient between the number of bootstrap models and the quality of constructed PIs measured by CWC for 12 case studies.

s_ϵ in (14) will be large, leading to unnecessarily wide PIs. In either case, the quality of constructed PIs will not be the best.

The medians and best of PI_{Bays} are close for the majority of case studies. This indicates that the Bayesian method generates consistent results in different replicates of an experiment. Also, the obtained CWCs show that the constructed PIs effectively bracket the targets. However, the method tends to build wide PIs to achieve a satisfactorily high PICP. Fig. 4 shows PICPs and NMPIWs for PI_{Best} for all case studies. Comparing the best PI_{Delta} and PI_{Bays} reveals that $PICP_{Bays}$ are greater than or equal to $PICP_{Delta}$ in 9 out of 12 case studies. In contrast, best PI_{Delta} are, on average, narrower than PI_{Bays}. As $PICP_{Delta}$ in all cases is greater than the prescribed confidence level (90%), one may conclude that the delta technique generates higher quality PIs. However, both median and standard deviation of CWC_{Bays} are smaller than CWC_{Delta} in 7 out of 12 case studies. Increased repeatability of the results highlights the strength of the Bayesian method for regenerating high-quality PIs, although they are wider than others.

An important feature of the Bayesian method for PI construction is the consistency of the obtained results. For PI_{Bays}, CWC_{SD} is small for 7 out of 12 case studies. Also the median of CWC_{SD} is 30.39, which is the smallest amongst the four methods. This small value is due to the nature of the Bayesian regularization technique for training of NNs. MAPE for 10 replicates of an experiment is shown in Fig. 5 for 12 case studies. With the exception of a few cases, MAPE remains almost the same for 10 replicates of an experiment. As per these results, performance of NN models trained using the Bayesian technique is less affected by the random initialization of NN parameters or the random partition into training/test examples.

Bootstrap is also a stable method for construction of PIs. $CWC_{Bootstrap}$ does not rapidly fluctuate and remains approximately the same for different replicates of an experiment. The smoothness of CWC is mainly due to incorporating the variability caused by random initialization of NN parameters (using B models instead of a single model). Traditionally, it is proven that model aggregation is effective in improving the generalization ability [41], [42]. Training multiple NNs from various random initial points provides a better coverage of the parameter space. The other methods miss properly capturing the variability due to the random choice of initial parameters.

Compared to other methods, specially the delta method, $PI_{Bootstrap}$ are wider. This extra width is due to overestimation of $\sigma^2_{\hat{y}_i}$ in (29) and $\sigma^2_{\hat{\epsilon}}$ in (30). Apart from NN model selection and training process, such an overestimation might be caused by the small number of bootstrap models. In our experiments, we trained and used 10 bootstrap models for PI construction. However, Efron and Tibshirani [43] and Davis and Hinkley [44] have suggested that B to be considered greater than at least 250 (and in some cases 1000) for estimating $\sigma^2_{\hat{y}_i}$ in practice. Assuming this claim is true, there should be a negative coefficient of correlation between B and CWC as the measure of the PI quality. The effectiveness of this claim is examined for all case studies. The number B of bootstrap models is changed from 100 to 1000 with an increment of 100. Then correlation coefficients between CWC and B are calculated for the 12 case studies. These correlation coefficients are shown in Fig. 6. Although coefficients are negative for 7 out of 12 case studies, the relationships are not strong. Besides, the quality of PIs has decreased in 5 out of 12 case studies. The mean of correlation coefficients for 12 case studies is -0.04. According to the obtained results, we may conclude that a strong inverse relationship (a large negative correlation coefficient) between CWC_{Best} and B does not exist. Greatly increasing the number of bootstrap models does not always improve the quality of $PI_{Bootstrap}$.

PI_{MVE} have the worst quality among the constructed PIs. This is mainly due to the unsatisfactorily low PICPs, which

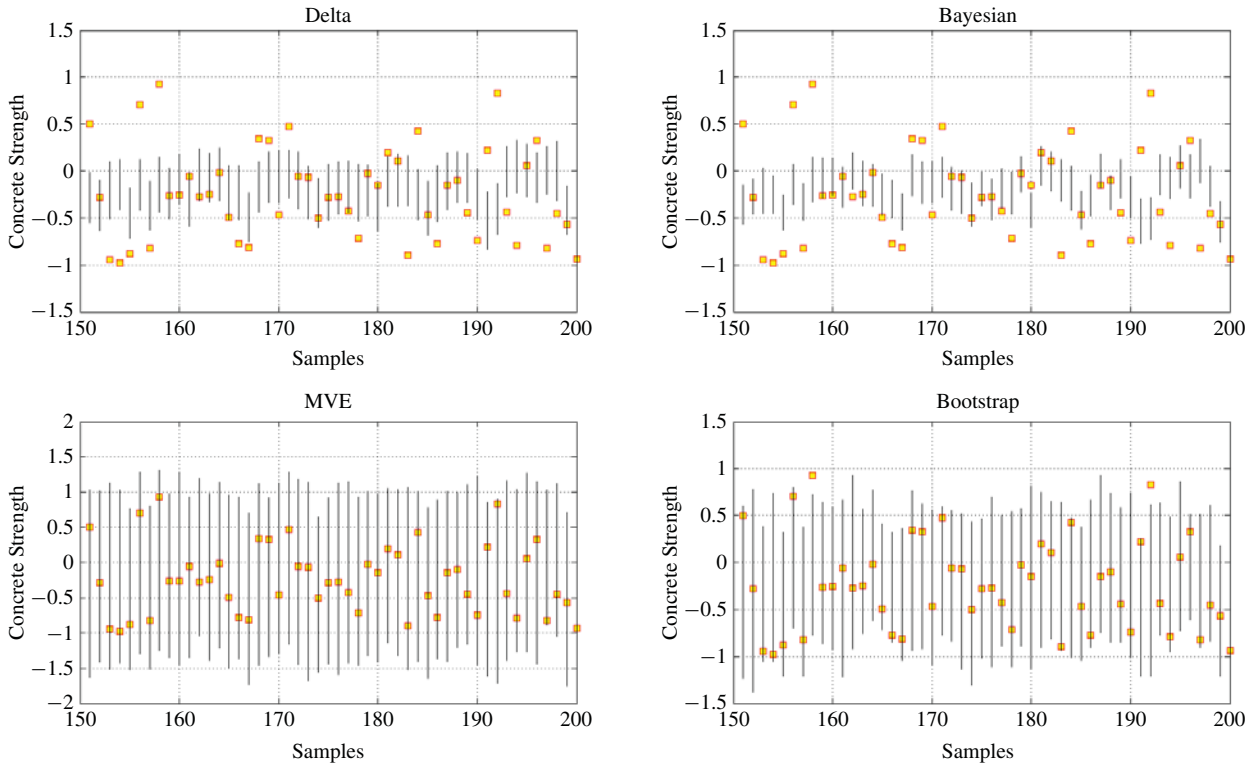

Fig. 7. PI_{Median} constructed for case study 6 using the delta method (top left), the Bayesian method (top right), the MVE method (bottom left), and the bootstrap method (bottom right).

stems from the improper estimation of the target variance using the NN models. It is highly likely that the target variance has little systematic relationship with the inputs. Even if there exists a relationship, some important variables may be missing. Another source of problem can be the invalidity of the fundamental assumption of the MVE method ($\hat{y}_i \simeq y_i$). This assumption can be violated for many reasons, specially for practical cases. Misspecification of NN model parameters, non-optimal selection of the NN architecture, and improper training of the NN model are among the potential causes.

It is also important to observe how the widths of PIs change for different observations of a target. The variability of the widths of PIs is an indication of how well they respond to the level of uncertainty in the datasets. Practically, we expect wider PIs for cases in which there is more uncertainty in datasets (e.g., having multivalued targets for approximately equal conditions or a high level of noise affecting the targets). Fig. 7 shows PI_{Median} for the delta, Bayesian, MVE, and bootstrap methods calculated for case study 6.[1] Comparing PI_{Delta} with others points out that PI_{Delta} are approximately constant in width and only their centers go up and down (which are in fact the point forecasts generated by the NN model). From the mathematical point of view, it means $g_0^T(F^TF + \lambda I)^{-1}(F^TF)(F^TF + \lambda I)^{-1}g_0 \ll 1$, and therefore the width of PI_{Delta} is mainly affected by s_ϵ. A similar story happens for the Bayesian method, the width of PI_{Bays} is dominated by $(1/\beta)$ in (24) ($(1/\beta) \gg \nabla_{w^{MP}}^T \hat{y}_i \quad (H^{MP})^{-1} \quad \nabla_{w^{MP}} \hat{y}_i$). The MVE and bootstrap methods show a better performance and their PIs have more variable widths. These variable widths imply that the estimation of variances using NNs enables the methods to appropriately respond to the level of uncertainty in the data.

[1]For better visualization, only PIs for samples 150 to 200 are shown.

TABLE III
COV FOR PI_{Median} CONSTRUCTED USING THE DELTA, BAYESIAN, MVE, AND BOOTSTRAP METHODS

Case Study	COV			
	Delta	Bayesian	MVE	Bootstrap
1	2.40	2.75	7.77	8.45
2	7.38	8.27	8.95	10.94
3	8.41	15.02	12.88	16.77
4	5.06	7.29	12.65	20.25
5	6.82	6.49	12.46	13.80
6	6.04	7.35	8.22	14.46
7	42.75	36.39	14.17	18.59
8	9.61	4.20	11.32	19.83
9	8.10	3.36	10.63	11.57
10	3.12	1.75	12.11	8.44
11	1.75	4.50	11.08	12.17
12	9.17	8.70	13.03	12.72

The coefficient of variations (COVs) (ratio of the standard deviation to the mean) for the width of PI_{Median} constructed using the delta, Bayesian, MVE, and bootstrap are shown in Table III. According to these statistics, $PI_{Bootstrap}$ have the largest variability in the width. The MVE method also shows an acceptable performance in terms of variable widths of PIs. PI_{Delta} and PI_{Bays} have the lowest COVs, which are

TABLE IV
STATISTICAL CHARACTERISTICS OF FOR CWC_{Delta}, CWC_{Bays}, CWC_{MVE}, AND $CWC_{Bootstrap}$ FOR CASE STUDY 7 WITH DIFFERENT VALUES OF τ

Method	Index	$\tau = 10$	$\tau = 5$	$\tau = 1$
Delta	CWC_{Best}	6.15	9.45	18.84
	CWC_{Median}	20.49	24.33	104.71
	CWC_{SD}	68.76	197.14	109.06
Bayesian	CWC_{Best}	7.99	10.92	23.09
	CWC_{Median}	8.47	12.65	25.31
	CWC_{SD}	42.43	4.48	31.35
MVE	CWC_{Best}	48.12	80.37	70.40
	CWC_{Median}	107.59	102.64	103.04
	CWC_{SD}	26.29	15.52	21.34
Bootstrap	CWC_{Best}	33.56	42.00	16.90
	CWC_{Median}	90.13	91.49	76.61
	CWC_{SD}	35.64	39.64	46.66

TABLE V
REQUIRED TIME FOR PI CONSTRUCTION USING THE DELTA, BAYESIAN, MVE, AND BOOTSTRAP METHODS FOR TEST SAMPLES

Case Study	Time (s)			
	Delta	Bayesian	MVE	Bootstrap
1	1.24	2.77	0.03	0.06
2	0.94	2.19	0.03	0.07
3	0.98	2.37	0.01	0.05
4	0.97	2.34	0.01	0.05
5	2.34	4.48	0.01	0.05
6	4.91	9.32	0.01	0.06
7	1.46	4.11	0.02	0.06
8	3.04	7.41	0.01	0.05
9	3.03	7.44	0.01	0.06
10	3.04	7.42	0.01	0.06
11	54.30	21.74	0.01	0.06
12	53.91	21.74	0.01	0.06

on average 9.21% and 8.84%, respectively. These are lower than COVs for the MVE (11.27%) and bootstrap (14.00%) methods.

To check the effects of noise heterogeneity on the quality of PIs, we change the amount of noise affecting the targets in case study 7. PIs are constructed for three different values of τ: 10, 5, and 1. As per the model of case study 7, $\tau = 1$ means the additive noise has the largest variance (more uncertainty in data). Table IV shows the characteristics of CWC_{Delta}, CWC_{Bays}, CWC_{MVE}, and $CWC_{Bootstrap}$ for this experiment. As the noise level (variance) increases, PIs become wider to keep the coverage probability of PIs satisfactorily high (at least 90%). The delta and Bayesian techniques show the best performance for three cases. PI_{Delta} and $PI_{Bootstrap}$ are the narrowest among the four methods with a small median and standard deviation. It is important to mention that PI_{Bays} are of better quality compared to others in the three cases. The median is very close to the best PI_{Bays}. The MVE method is the worst and its performance is highly affected by the level of noise in data.

Increase in the width of PIs reflects the most important advantage of PIs against point predictions, how they respond to the uncertainty in the data. While the forecasted points carry no indication of presence of uncertainty in data, variable width of PIs is an informative index about how uncertain the data is or what the risks of decision makings are.

The computational requirements of the four methods for PI constructions are different. It is important to note that NN-based PI construction methods have two types of computational loads: offline and online. The offline requirements include the elapsed time for training NN models and calculation of some matrices, such as the Jacobian or Hessian. The online computational load is the amount of time spent for the construction of PIs for a new sample. From a practical viewpoint, the offline computational load is less important. Models can be trained, fine-tuned, and saved for later use in a convenient time. Therefore, it is not reasonable to compare PI construction methods for their offline computational requirement. The online computational requirement for PI construction for a new sample is a more important and critical issue. As PIs may be used for optimization purposes and conducting "what-if" analysis in real-world cases, their low construction cost is a real advantage for the underlying method. In the experiments performed here, PI construction methods are reviewed and compared from this perspective.

Table V summarizes the elapsed times for the construction of PIs for test samples of the 12 case studies. The tabulated times are only for a single replicate. According to the obtained results, the Bayesian and delta techniques are computationally much more expensive than the MVE and bootstrap methods for PI construction. Both MVE and bootstrap methods are hundreds of times faster than the other two methods. The MVE method is the least expensive method for PI construction, as its online computational requirement is negligible. The elapsed time for construction of $PI_{Bootstrap}$ is also very small. This is in contrast with the frequently stated claim in the literature that the bootstrap method is computationally highly demanding. The delta method, in 10 out of 12 case studies, has a less computational load than the Bayesian method. However, PI_{Delta} are computationally more expensive than PI_{Bays} for cases 11 and 12. According to the dataset information shown in Table I, the number of samples for these two cases is greater than the other cases. This implies that the delta technique has a larger computational burden than the Bayesian technique for large datasets.

The presented results in Table V are an indication of the online computational requirements of the four methods. Consider the case in which we need to construct PIs for travel times of bags for case study 4, T90. At least 1000 what-if scenarios are required for optimal operational planning and scheduling. According to Table V, the elapsed times for conducting these experiments using the four methods are $T_{Delta} = 2345$ s, $T_{Bays} = 4479$ s, $T_{MVE} = 12$ s, and $T_{Bootstrap} = 55$ s. Real-time operational planning and optimization using the MVE and bootstrap methods is possible,

as both T_{MVE} and $T_{Bootstrap}$ are less than 1 min. The delta and Bayesian methods do not finish their online computation in a practical time, and are therefore not suitable for real-time operational planning in a baggage handling system.

Decision regarding the suitability of a PI construction method for optimization and decision-making purposes depends on the size of dataset and constraints of the underlying system. For instance, while the delta and Bayesian methods are not the best option for case studies 3 and 4, they can be easily applied to case studies 8–10. The PI construction methods have enough time to compute the required calculations, as the rate of completion of the tasks and operations for this case study is slow. Therefore, the operational planners and schedulers can enjoy the excellent quality of PI_{Delta} and PI_{Bays} without considering the computational load.

D. Summary

To quantify the performance of the PI construction methods, we rank each method from 1 to 4 depending on the constructed PIs for test samples. The lower the rank, the better the method. The ranking scores are given in four categories as per the following.

1) Quality: CWC_{Median} is used as an index to quantify the performance of each method for producing high quality PIs. For each case study, CWC_{Median} in Table II are sorted increasingly and scored between 1 (the lowest) and 4 (the greatest) for the four PI construction methods. The scores are then averaged to generate a total score of the method's performance.
2) Repeatability: This is measured by calculating the 70th percentile of CWC (i.e., seventh best replicate). As a conservative measure, this provides information about how each method will do in worst cases, which methods are prone to fail, and which methods do well even in their bad runs. Similar to the quality metric, the 70th percentiles are sorted, scored between 1 and 4, and then averaged for twelve case studies.
3) Computational load: The same scoring method is applied to the elapsed times as shown in Table V.
4) Variability: This metric relates to the response of PIs to the level of uncertainty associated with data. To measure this, we first obtain the width of PI_{Best} for the delta, Bayesian, MVE, and bootstrap techniques. Then, the COV is calculated for this set as an indication of its variation. The method with the greatest COV is scored 1, the method corresponding to the second greatest COV is scored 2, and so on.

It is important to observe that these metrics have unequal importance in practice. While the PI quality is the most important metric for some decision makers, the computational load can be the key factor in optimization problems.

Table VI presents these four metrics for the delta, Bayesian, MVE, and bootstrap techniques. According to this table and results represented in the previous section, we can make the following conclusions.

1) PI_{Delta} have the highest quality of the four methods. They are the narrowest with an acceptable coverage probability above the prescribed confidence level. However, the repeatability of the results is not good and the method may generate low-quality PIs. The computational requirements of the method are also large and it constructs PIs with almost a fixed width.
2) PI_{Bays} are second in terms of quality, and their reproducibility is the best. Similar results are generated in different replicates of the method. The method is the worst in terms of the computational requirements amongst the four investigated methods. Last but not least, PI_{Bays} have one of the most fixed widths (scored 3.25 out of 4).
3) PI_{MVE} are the least computationally expensive to construct. This is because only two NNs are used in the process of PI construction. Therefore, the method's computational requirements are almost negligible compared to other methods, in particular the delta and Bayesian techniques. In some replicates of this method, high-quality PIs are constructed. However, the quality and repeatability metrics of PI_{MVE} are the worst, making PI_{MVE} unreliable for real-world applications.
4) The bootstrap method does not generate high-quality PIs compared to the delta and Bayesian methods. The method tends to overestimate the variance of the targets, resulting in wider PIs compared to PI_{Delta}. Increasing the number of bootstrap models does not guarantee an improvement of the PI quality. In terms of the variability, $PI_{Boostrap}$ are by far the best among the four methods. Also, the method is ranked second in terms of online computational load.

TABLE VI
SUMMARY OF RESULTS BASED ON THE AVERAGED RANK OF EACH METHOD FOR 12 CASE STUDIES

Assessment Metric	Delta	Bayesian	MVE	Bootstrap
Quality	1.58	1.75	3.75	2.92
Repeatability	2.75	1.67	2.92	2.67
Computational load	3.25	3.75	1.00	2.00
Variability	3.33	3.25	2.08	1.33

V. COMBINED PIS

Results in Table VI show that there is no method that is dominating in all performance criteria. Each method has its own theoretical and practical limitations, dependent on the application area or due to its overall effectiveness. Moreover, the best method in quality was not the top method for every single problem. Given this uncertainty as to which method will obtain best results on a given problem, we propose an ensemble approach for PI construction. This means that we apply the four available methods and combine their PIs in some way. The rationale for considering a combination of methods is similar to that of ensemble NNs [45], [46]. It is more robust and mitigates the effect of one method giving bad results and ruining performance.

It is also important to note that the quality of PIs constructed using a method significantly varies in different replicates of a method (different results for different NNs). As per the results

demonstrated in Table II, there is often a large discrepancy between the best result and median of the results. The standard practice dictates that we trust PIs constructed using the NN that generates the highest quality PIs for the validation set. However, it is well known that best results are not guaranteed for another set. Therefore, it is preferable to keep a set of NNs for each method (an ensemble of models) and run them all with an appropriate collective decision strategy to get high quality PIs.

Simple averaging, weighted averaging, ranking, and nonlinear mapping can be applied as the combination strategy in the ensemble. The greatest advantage of simple averaging is its simplicity. However, the main drawback of the method is that it treats all individuals equally, though they are not equally good. For the purpose of this paper, we consider a linear combination of the PIs generated using the delta, Bayesian, MVE, and bootstrap techniques. As per this mechanism, the lower and upper bounds of the combined PI will be equal to the summation of the weighted lower and upper bounds of four PIs

$$PI_{comb} = \theta \begin{bmatrix} PI_{Delta} \\ PI_{Bays} \\ PI_{MVE} \\ PI_{Bootstrap} \end{bmatrix} \tag{40}$$

where $\theta = [\theta_1, \theta_2, \theta_3, \theta_4]$ is the vector of combiner parameters. Two sets of constraints can be considered for the combiner parameters.

1) Restriction 1: They are positive and less than 1: $0 \leq \theta_i \leq 1, \ i = 1, 2, 3, 4$. This means that PIs of four methods positively contribute to the construction of the new combined PIs. This idea has been advocated in the literature [47], [48].
2) Restriction 2: The parameters are restricted to sum to 1: $\sum_{i=1}^{4} \theta_i = 1$. This restriction makes the new combined PI a weighted average, which is a flexible form of the simple average.

The key question is how the combiner parameters in (40) can be obtained. Traditionally, these parameters are determined through minimization of an error-based cost function, such as SSE. In those cases, the purpose of combination is to improve the generalization ability and achieving smoother results over the error space. However, the purpose of combiner in our case is to improve the quality of combined PIs. Therefore, it is more meaningful to adjust the combiner parameters in (40) through minimization of a PI-based cost function.

Another issue in adjusting the combiner parameters relates to the unavailability of target PIs. The ground truth of the upper and lower bounds of the desired PIs is not *a priori* known, and cannot be used during the training stage of the combiner. Therefore, a method should be developed that indirectly adjusts the combiner parameters leading to the highest quality PIs.

The quality of PIs in this paper is assessed using the CWC measure defined in (38). As CWC covers both key features of PIs (width and coverage probability), it can be used as the objective function in the problem of enhancing the quality of PIs using the combiner. The combiner parameters can be optimally determined through minimization of CWC as the cost function. In fact, these parameters are indirectly fine-tuned to generate high quality combined PIs. This approach eliminates the need for knowing the desired values of PIs for adjusting the combiner parameters.

As per restrictions described above, two optimization problems can be defined.

Option A

$$\begin{aligned} \theta_A &= \arg \min_{\theta} CWC \\ &\text{s.t. } 0 \leq \theta \leq 1. \end{aligned} \tag{41}$$

Option B

$$\begin{aligned} \theta_B &= \arg \min_{\theta} CWC \\ &\text{s.t. } 0 \leq \theta \leq 1 \\ &\sum_{i=1}^{4} \theta_i = 1. \end{aligned} \tag{42}$$

Option A has the advantage compared to option B that it can lower or raise the absolute level of the PIs (parameters are not restricted to sum to one), thereby correcting any general bias in the PIs.

During the training of the combiner parameters using CWC as the cost function, we set $\gamma(PICP) = 1$. CWC formulation with γ equal to 1 has the advantage of leaving some slack in the training, in order to avoid the serious downside of violating the PIs' constraint for the test set (i.e., that $PICP \geq 90\%$). This conservative approach is applied to avoid excessively narrowing PIs during the training stage, which may result in a low PICP for test samples. After the training stage, all PIs are assessed using CWC with $\gamma(PICP)$ as defined in (39).

CWC, as the cost function, is nonlinear and nondifferentiable with many local minima. Therefore, descent-based optimization methods, such as those used in traditional NN training, cannot be applied for its minimization. Here, we use GA [49]–[51] for minimization of the cost function in the optimization stage.

First, the available data is divided into two training sets (D_1 and D_2) and the test set (D_{test}). Similar to the experiments performed in the previous section, a cross-validation technique is applied to determine the optimal NN structure. The optimal NN is trained 10 times using samples of D_1. PI_{D_2} are then constructed using the delta, Bayesian, MVE, and bootstrap methods for each set of NN models (totally 10 sets of PIs for each method). The combiner parameters θ are initialized to values between 0 and 1. The optimization algorithm is then applied for finding the optimal values of the combiner parameter. In each iteration of the optimization algorithm, PI_{comb} are constructed using the new set of combiner parameters and PI_{D_2} from the four methods.

In addition to the PI combination across the four methods, we have another layer of combination over the 10 runs. This also improves the robustness of the combination approach, as it averages over different network initializations and different train/validation partitions. So, the median of PI_{comb} over the 10 runs, called PI_{comb}^{median}, is computed, and this gives the final PIs. PI_{comb}^{median} are evaluated using the cost function,

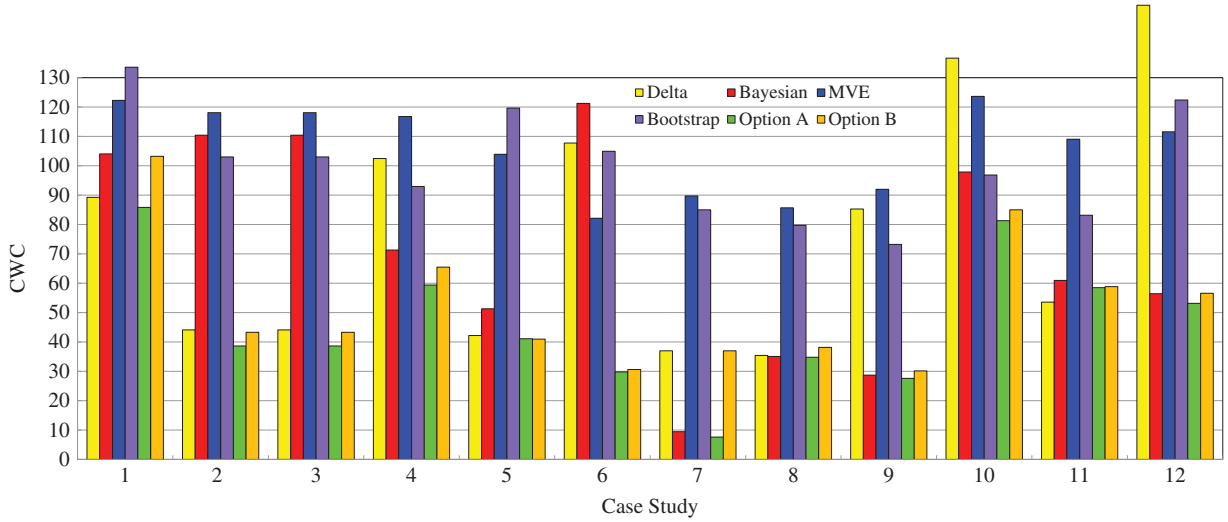

Fig. 8. Performance of two combiners for generating high-quality PIs compared to the four traditional methods.

TABLE VII
PERCENTAGE DIFFERENCE BETWEEN THE BEST METHOD FOR PI CONSTRUCTION AND OTHER METHODS

Case Study	Delta	Bayesian	MVE	Bootstrap	Option A	Option B
1	3.95	21.21	42.41	55.55	0.00	20.26
2	14.21	185.71	205.53	166.63	0.00	12.00
3	14.21	185.71	205.53	166.63	0.00	12.00
4	72.56	20.06	96.59	56.53	0.00	10.29
5	2.88	25.13	153.35	191.83	0.30	0.00
6	261.23	306.49	175.26	251.91	0.00	2.68
7	387.01	24.79	1080	1018	0.00	386.29
8	1.76	0.78	145.94	128.77	0.00	9.49
9	208.45	3.90	232.58	164.83	0.00	9.17
10	68.09	20.46	52.14	19.19	0.00	4.58
11	0.00	13.80	103.41	55.16	9.16	9.74
12	191.01	6.28	110.00	130.25	0.00	6.48

CWC with $\gamma(PICP) = 1$, and their appropriateness is comparatively checked. Upon completion of the optimization stage, the combiner with its set of optimal parameters is used for construction of PI_{comb}^{median} for test samples (D_{test}).

The computation burden of the optimization algorithm in each iteration is limited to generation of a new population of the combiner parameters, constructing PI_{comb} for the new population, calculation of median of PI_{comb} (PI_{comb}^{median}), and evaluation of PI_{comb}^{median} using CWC. As these tasks do not require complex calculation, the optimization algorithm is computationally inexpensive.

In summary, the proposed method here uses two types of ensembles for achieving high quality PIs: 1) an ensemble of NN models in each method for constructing PIs, and 2) an ensemble of four different methods for construction of combined PI_{comb}^{median} using PIs of individual methods. This two-stage mechanism maximizes the diversity (using different models and methods) for constructing high-quality PIs.

VI. NUMERICAL RESULTS FOR COMBINED PIS

The performance of the proposed method in previous section for construction of combined PI, i.e., PI_{comb}, is here examined for 12 case studies. The GA program is run with a crossover fraction of 0.8, a stall generation limit of 1000, and a population of 100 individuals. Again, we randomly split data into D_1, D_2, and D_{test} sets. The NN structure is determined through a fivefold cross-validation technique. PIs for D_2 are constructed using the delta, Bayesian, MVE, and bootstrap methods. Then the proposed method in the previous section is applied for adjusting the combiner parameters as per (41) and (42).

Fig. 8 shows CWCs for PI_{comb} constructed using the combiner trained based on option A and Option B. Hereafter and for ease in reference, we refer to these PIs as PI_{comb-A} and PI_{comb-B}. For the purpose of comparison, CWC medians for other methods are also shown in this figure. As per these results, PI_{comb-A} and PI_{comb-B} are the best for 10 and 1 out of 12 case studies, respectively (totally for 11 out of 12 case studies). It is only for case 11 that the proposed methods do not generate the best results. Ranks of option A and B for this case study are 2 and 3, respectively. With the exception of this case, both methods, and in particular option A, outperform any individual method in terms of the quality of constructed PIs.

The amount of difference between the best method for PI construction and the other methods is demonstrated in Table VII. The percentage difference is the ratio of difference between the CWCs and the minimum of CWCs normalized

by the minimum of CWCs

$$Difference = \frac{CWC - CWC_{min}}{CWC_{min}} \tag{43}$$

where CWC_{min} is the minimum of CWCs for each case study shown in Fig. 8. A zero difference for a method means that it has generated the highest quality PIs in the conducted experiments. As per the computed difference, the proposed combining methods, option A and B, significantly improve the quality of PIs. The median values of differences for the six methods are 41.1%, 20.8%, 149.6%, 147.5%, 0.0%, and 9.6%, respectively. As per these values, it is obvious that using the proposed combiners for PI construction significantly improves the quality of constructed PIs.

VII. Conclusion

In this paper, we comprehensively reviewed and examined the performance of four frequently cited methods for construction of PIs using NNs. The theoretical background of the delta, Bayesian, MVE, and bootstrap techniques was first studied to find the advantages and disadvantages of each method. Twelve synthetic and real-world case studies were implemented to assess the performance of each method for generating high-quality PIs. Effects of homogeneous and heterogeneous noise on the quality of PIs were investigated. Quantitative and comprehensive assessments were performed by using a hybrid measure related to the width and coverage probability of PIs. According to the obtained results, the delta technique generates the highest quality PIs, the Bayesian method is the most reliable for reproducing quality PIs, and the MVE method is the least computationally expensive method. The bootstrap-based PIs have the most variable widths and appropriately respond to the level of uncertainty in data. Results indicate that there is no best method for all cases. Therefore, selection and application of a PI construction method will depend on the purpose of analysis, the computational constraints, and which aspect of the PI is more important.

This paper also proposed a new method for construction of PIs through combination of traditionally built PIs. The proposed method uses an ensemble of NNs for each traditional method to construct PIs, and an ensemble of four methods to build combined PIs based on the medians of PIs from each ensemble. The combiner parameters were indirectly adjusted through minimization of a PI-based cost function. GA was applied for minimization of the nonlinear nondifferentiable cost function. It was shown that the proposed combiner methods outperform any individual method in terms of generating higher quality PIs.

References

[1] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, 1989.

[2] C. M. Bishop, *Neural Networks for Pattern Recognition*. London, U.K.: Oxford Univ. Press, 1995.

[3] M. A. Hussain, "Review of the applications of neural networks in chemical process control — simulation and online implementation," *Artif. Intell. Eng.*, vol. 13, no. 1, pp. 55–68, Jan. 1999.

[4] J. G. De Gooijer and R. J. Hyndman, "25 years of time series forecasting," *Int. J. Forecast.*, vol. 22, no. 3, pp. 443–473, 2006.

[5] B. K. Bose, "Neural network applications in power electronics and motor drives–an introduction and perspective," *IEEE Trans. Ind. Electron.*, vol. 54, no. 1, pp. 14–33, Feb. 2007.

[6] M. Paliwal and U. A. Kumar, "Review: Neural networks and statistical techniques: A review of applications," *Expert Syst. Appl.*, vol. 36, no. 1, pp. 2–17, Jan. 2009.

[7] W.-H. Liu, "Forecasting the semiconductor industry cycles by bootstrap prediction intervals," *Appl. Econ.*, vol. 39, no. 13, pp. 1731–1742, 2007.

[8] S. Ho, M. Xie, L. Tang, K. Xu, and T. Goh, "Neural network modeling with confidence bounds: A case study on the solder paste deposition process," *IEEE Trans. Electron. Packag. Manuf.*, vol. 24, no. 4, pp. 323–332, Oct. 2001.

[9] J. H. Zhao, Z. Y. Dong, Z. Xu, and K. P. Wong, "A statistical approach for interval forecasting of the electricity price," *IEEE Trans. Power Syst.*, vol. 23, no. 2, pp. 267–276, May 2008.

[10] 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.

[11] S. G. Pierce, K. Worden, and A. Bezazi, "Uncertainty analysis of a neural network used for fatigue lifetime prediction," *Mech. Syst. Signal Process.*, vol. 22, no. 6, pp. 1395–1411, Aug. 2008.

[12] D. F. Benoit and D. Van den Poel, "Benefits of quantile regression for the analysis of customer lifetime value in a contractual setting: An application in financial services," *Expert Syst. Appl.*, vol. 36, no. 7, pp. 10475–10484, Sep. 2009.

[13] D. L. Shrestha and D. P. Solomatine, "Machine learning approaches for estimation of prediction interval for the model output," *Neural Netw.*, vol. 19, no. 2, pp. 225–235, Mar. 2006.

[14] C. van Hinsbergen, J. van Lint, and H. van Zuylen, "Bayesian committee of neural networks to predict travel times with confidence intervals," *Transport. Res. Part C: Emerg. Technol.*, vol. 17, no. 5, pp. 498–509, Oct. 2009.

[15] A. Khosravi, E. Mazloumi, S. Nahavandi, D. Creighton, and J. W. C. van Lint, "Prediction intervals to account for uncertainties in travel time prediction," *IEEE Trans. Intell. Transport. Syst.*, vol. 12, no. 2, pp. 537–547, Jun. 2011.

[16] A. Khosravi, E. Mazloumi, S. Nahavandi, D. Creighton, and J. W. C. van Lint, "A genetic algorithm-based method for improving quality of travel time prediction interval," *Transport. Res. Part C: Emerg. Technol.*, Jun. 2011, to be published.

[17] 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, Mar. 2010.

[18] J. T. G. Hwang and A. A. Ding, "Prediction intervals for artificial neural networks," *J. Amer. Stat. Assoc.*, vol. 92, no. 438, pp. 748–757, Jun. 1997.

[19] R. D. De Veaux, J. Schumi, J. Schweinsberg, and L. H. Ungar, "Prediction intervals for neural networks via nonlinear regression," *Technometrics*, vol. 40, no. 4, pp. 273–282, Nov. 1998.

[20] D. J. C. MacKay, "The evidence framework applied to classification networks," *Neural Comput.*, vol. 4, no. 5, pp. 720–736, Sep. 1992.

[21] D. A. Nix and A. S. Weigend, "Estimating the mean and variance of the target probability distribution," in *Proc. IEEE Int. Conf. Neural Netw.*, vol. 1. Orlando, FL, Jun.–Jul. 1994, pp. 55–60.

[22] B. Efron, "Bootstrap methods: Another look at the jackknife," *Ann. Stat.*, vol. 7, no. 1, pp. 1–26, Jan. 1979.

[23] T. Heskes, "Practical confidence and prediction intervals," in *Neural Information Processing Systems*, vol. 9, T. P. M. Mozer and M. Jordan, Eds. Cambridge, MA: MIT Press, 1997, pp. 176–182.

[24] G. Papadopoulos, P. J. Edwards, and A. F. Murray, "Confidence estimation methods for neural networks: A practical comparison," *IEEE Trans. Neural Netw.*, vol. 12, no. 6, pp. 1278–1287, Nov. 2001.

[25] 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.

[26] F. Giordano, M. La Rocca, and C. Perna, "Forecasting nonlinear time series with neural network sieve bootstrap," *Comput. Stat. Data Anal.*, vol. 51, no. 8, pp. 3871–3884, May 2007.

[27] I. Rivals and L. Personnaz, "Construction of confidence intervals for neural networks based on least squares estimation," *Neural Netw.*, vol. 13, nos. 4–5, pp. 463–484, Jun. 2000.

[28] C. J. Wild and G. A. F. Seber, *Nonlinear Regression*. New York: Wiley, 1989.

[29] R. Tibshirani, "A comparison of some error estimates for neural network models," *Neural Comput.*, vol. 8, no. 1, pp. 152–163, Jan. 1996.

[30] M. Hagan and M. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Trans. Neural Netw.*, vol. 5, no. 6, pp. 989–993, Nov. 1994.
[31] R. Dybowski and S. Roberts, "Confidence intervals and prediction intervals for feed-forward neural networks," in *Clinical Applications of Artificial Neural Networks*, R. Dybowski and V. Gant, Eds. Cambridge, U.K.: Cambridge Univ. Press, 2000.
[32] 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.
[33] A.-M. Halberg, K. H. Teigen, and K. I. Fostervold, "Maximum versus minimum values: Preferences of speakers and listeners for upper and lower limit estimates," *Acta Psychol.*, vol. 132, no. 3, pp. 228–239, Nov. 2009.
[34] S. Hashem, "Optimal linear combinations of neural networks," *Neural Netw.*, vol. 10, no. 4, pp. 599–614, Jun. 1997.
[35] 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.
[36] P. Vlachos. (2010, Jan.). *StatLib Datasets Archive* [Online]. Available: http://lib.stat.cmu.edu/datasets
[37] A. Khosravi, S. Nahavandi, and D. Creighton, "Constructing prediction intervals for neural network metamodels of complex systems," in *Proc. Int. Joint Conf. Neural Netw.*, Atlanta, GA, Jun. 2009, pp. 1576–1582.
[38] A. Asuncion and D. J. Newman. (2010, Jan.). *UCI Machine Learning Repository*. School Inf. Comput. Sci., Univ. California, Irvine [Online]. Available: http://www.ics.uci.edu/∼mlearn/MLRepository.html
[39] B. De Moor. (2010, Jan.). *DaISy: Database for the Identification of Systems*. Dept. Electr. Eng., ESAT/SISTA, K.U.Leuven, Leuven, Belgium [Online]. Available: http://homes.esat.kuleuven.be/∼smc/daisy/
[40] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, May 1983.
[41] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.
[42] M. Islam, X. Yao, and K. Murase, "A constructive algorithm for training cooperative neural network ensembles," *IEEE Trans. Neural Netw.*, vol. 14, no. 4, pp. 820–834, Jul. 2003.
[43] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*. New York: Chapman & Hall, 1993.
[44] A. C. Davison and D. V. Hinkley, *Bootstrap Methods and Their Application*. Cambridge, U.K.: Cambridge Univ. Press, 1997.
[45] L. Hansen and P. Salamon, "Neural network ensembles," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 10, pp. 993–1001, Oct. 1990.
[46] X. Yao and M. Islam, "Evolving artificial neural network ensembles," *IEEE Comput. Intell. Mag.*, vol. 3, no. 1, pp. 31–42, Feb. 2008.
[47] S. I. Gunter, "Nonnegativity restricted least squares combinations," *Int. J. Forecast.*, vol. 8, no. 1, pp. 45–59, Jun. 1992.
[48] J. W. Taylor and S. Majithia, "Using combined forecasts with changing weights for electricity demand profiling," *J. Oper. Res. Soc.*, vol. 51, no. 1, pp. 72–82, 2000.
[49] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley, 1989.
[50] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA: MIT Press, 1996.
[51] C. R. Reeves and J. E. Rowe, *Genetic Algorithms: Principles and Perspectives: A Guide to GA Theory*. Norwell, MA: Kluwer, 2003.

Abbas Khosravi (M'07) received the B.Sc. degree in electrical engineering from the Sharif University of Technology, Tehran, Iran, in 2002, the M.Sc. degree in electrical engineering from the Amirkabir University of Technology, Tehran, in 2005, and the Ph.D. degree from Deakin University, Geelong, Australia, in 2010.

He joined the eXiT Group, University of Girona, Girona, Spain, from 2006 to 2007, conducting research in the field of artificial intelligence. Currently he is a Research Fellow at the Center for Intelligent Systems Research, Deakin University. His current research interests include theory and the application of neural networks and fuzzy logic systems for modeling, analysis, control, and optimization of operations within complex systems.

Dr. Khosravi is a recipient of the Alfred Deakin Post-Doctoral Research Fellowship in 2011.

Saeid Nahavandi (SM'07) received the B.Sc. (Hons.), M.Sc., and Ph.D. degrees in automation and control from Durham University, Durham, U.K.

He is currently the Alfred Deakin Professor, Chair of Engineering, and the Director for the Center for Intelligent Systems Research, Deakin University, Geelong, Australia. He has published over 350 peer-reviewed papers in various international journals and conferences. He designed the world's first 3-D interactive surface/motion controller. His current research interests include modeling of complex systems, simulation-based optimization, robotics, haptics, and augmented reality.

Dr. Nahavandi was a recipient of the Young Engineer of the Year Title in 1996 and six international awards in engineering. He is an Associate Editor of the IEEE SYSTEMS JOURNAL, an Editorial Consultant Board Member for the *International Journal of Advanced Robotic Systems*, and an Editor (South Pacific Region) of the *International Journal of Intelligent Automation and Soft Computing*. He is a fellow of Engineers Australia and Institution of Engineering and Technology.

Doug Creighton (M'10) received the B.Eng. (Hons.) degree in systems engineering and the B.Sc. degree in physics from the Australian National University, Canberra, Australia, in 1997, where he attended as a National Undergraduate Scholar. He received the Ph.D. degree in simulation-based optimization from Deakin University, Geelong, Australia, in 2004.

He has spent several years as a Software Consultant. He is currently a Research Academic and Stream Leader at the Center for Intelligent Systems Research, Deakin University. He has developed algorithms to allow the application of learning agents to industrial-scale systems for use in optimization, dynamic control, and scheduling. His current research interests include modeling, discrete event simulation, intelligent agent technologies, human machine interface and visualization, and simulation-based optimization research.

Amir F. Atiya (S'86–M'90–SM'97) received the B.S. degree from Cairo University, Cairo, Egypt, in 1982, and the M.S. and Ph.D. degrees from California Institute of Technology (Caltech), Pasadena, in 1986 and 1991, respectively, all in electrical engineering.

He is currently a Professor with the Department of Computer Engineering, Cairo University. He has held several visiting appointments, such as in Caltech and in Chonbuk National University, Jeonju, South Korea. His current research interests include neural networks, machine learning, theory of forecasting, computational finance, and Monte Carlo methods.

Dr. Atiya was a recipient of several awards, including the Kuwait Prize in 2005. He was an Associate Editor for the IEEE TRANSACTIONS ON NEURAL NETWORKS from 1998 to 2008.