Modélisation à base de réseaux de neurones dédiés à la prédiction sous incertitudes appliqué aux systèmes energétiques

Ronay Ak

N° d'ordre: 2014-15-TH

SUPELEC

ECOLE DOCTORALE STITS

« Sciences et Technologies de l'Information des Télécommunications et des Systèmes »

THÈSE DE DOCTORAT

DOMAINE : SPI

SPECIALITE : ENERGIE

Soutenue le 2 Juillet 2014

Présenté par:

RONAY AK

Titre de la thèse:

Neural Network Modeling for Prediction under Uncertainty in Energy System Applications

<u>Composition du jury</u> :

Enrico ZIO	SUPELEC, France	*Directeur de thèse*
Pierre PINSON	DTU, Denmark	*Rapporteur*
Bertrand IOOSS	EDF - R&D, France	*Rapporteur*
Philippe DESSANTE	SUPELEC, France	*Examinateur*
Yann Le BIHAN	Université Paris Sud, France	*Examinateur*
Paul ULMEANU	Uni. Politehnica of Bucharest, Romania	*Examinateur*

To my mother…

ACKNOWLEDGEMENTS

I would like to thank my thesis supervisor, Professor Enrico Zio, for his continuous support, guidance and encouragement throughout the three years of my Ph. D. work. It was a pleasure and a great honor to pursue a Ph. D. degree under his supervision.

I would like to express my sincere appreciation and gratitude to my co-advisors, Assistant Professor Yan-Fu Li and Dr. Valeria Vitelli, for their support, time, and patience. This research work would not have been possible without their guidance.

My deepest appreciation goes to all the jury members that agreed to be part of the committee: Professor Pierre Pinson, Senior Research Engineer (HDR) Bertrand Iooss, Professor Philippe Dessante, Professor Yann Le Bihan, and Professor Paul Ulmeanu. In particular, I am grateful to reviewers, Pierre Pinson and Bertrand Iooss, for their careful reading of the manuscript and all the constructive and helpful remarks, which helped improve the quality of my thesis work.

I would like to thank the European Foundation for New Energy of Electricité de France (EDF) for their support allowing me to conduct this three-year Ph. D. work under the Chair on Systems Science and the Energy Challenge (SSEC).

I would like to offer my special thanks to my family for their love, support, and constant encouragement. In particular, I would like to thank my mother and my brother. They are the meaning of my life, and I undoubtedly could not have done this without them.

I would like to thank all my friends/colleagues in EDF Chair and/or at Laboratory LGI, ECP. It has been a pleasure to work with them during the last three years. I want to give them my great thanks for everything they have done, for their encouragement and their disposition to offer me a hand whenever I needed help. In addition, great thanks to my office mates, Jie Liu and Yanhui Lin, for being always friendly and nice to me.

Last but not least, special thanks to the Director of the Department of Power & Energy Systems, Supélec, Professor Jean-Claude Vannier, and to the Director of Laboratory LGI, ECP, Professor Jean-Claude Bocquet, for a three-year hosting.

ABSTRACT

This Ph. D. work addresses the problem of prediction within energy systems design and operation problems, and particularly the adequacy assessment of renewable power generation systems. The general aim is to develop an empirical modeling framework for providing predictions with the associated uncertainties. Along this research direction, a non-parametric, empirical approach to estimate neural network (NN)-based prediction intervals (PIs) has been developed, accounting for the uncertainty in the predictions due to the variability in the input data and the system behavior (e.g. due to the stochastic behavior of the renewable sources and of the energy demand by the loads), and to model approximation errors. A novel multi-objective framework for estimating NN-based PIs, optimal in terms of both accuracy (coverage probability) and informativeness (interval width) is proposed. Ensemble of individual NNs via two novel approaches is proposed as a way to increase the performance of the models. Applications on real case studies demonstrate the power of the proposed framework.

Keywords: Prediction intervals, multi-layer perceptron neural networks, multi-objective genetic algorithm, adequacy assessment, short-term wind speed prediction, wind power production, load forecasting, uncertainty.

RÉSUMÉ ÉTENDU

La production et la fourniture d'énergie posent des enjeux aussi bien économiques, environnementaux, sociaux que politiques. Dans une optique de développement durable, il s'agit de tendre vers un meilleur équilibre dans la prise en compte de ces différents enjeux. La demande mondiale d'énergie va continuer de croître mais devra être satisfaite de manière efficiente et efficace. L'accroissement de la mobilité, de l'urbanisation et de l'industrialisation, en particulier dans les pays en développement, et l'intégration toujours plus importante de l'économie mondiale vont accélérer l'augmentation de la consommation mondiale d'énergie et la dépendance énergétique de nos sociétés. Aujourd'hui, le principal moyen de répondre à la demande d'énergie électrique est la combustion de combustibles fossiles (par exemple du pétrole, du charbon ou du gaz naturel). Bien que les carburants fossiles puissent produire une quantité importante d'énergie, les principaux inconvénients de ces sources d'énergie sont leurs réserves limitées et leurs effets négatifs sur l'environnement. En particulier, les changements climatiques induits constituent un défi supplémentaire pour les fournisseurs d'énergie, les consommateurs et les opérateurs du marché. Pour lutter contre le changement climatique, à savoir réduire les effets négatifs des sources d'énergie traditionnelles (par exemple les émissions de gaz à effet de serre), les sources d'énergie fossiles doivent être remplacées par des sources d'énergie alternatives, c'est à dire les sources d'énergie renouvelables, plus propres et moins dommageables pour les individus et l'environnement.

D'un point de vue général, l'objectif principal des systèmes de production d'énergie devrait être de répondre à tout moment à la demande d'énergie tout en minimisant les impacts environnementaux associés. Ceci nécessite de développer des formes propres d'énergie, tout en s'assurant de la cohérence et de la fiabilité de leur approvisionnement et de leur utilisation. Cependant, les acteurs du marché de l'énergie (investisseurs, producteurs d'électricité, les gestionnaires de réseau, les consommateurs, etc.) font face à des défis potentiels [1]-[5]:

- la demande croissante d'énergie
- de nouveaux défis dans les modes de consommation d'énergie
- l'intégration des sources d'énergie intermittentes (stochastiques) renouvelables dans les réseaux électriques
- l'extraction de combustibles fossiles dans des conditions extrêmes d'une manière plus économique et plus sûre

- le renforcement des capacités de raffinage
- la réduction des déchets nucléaires
- améliorer la fiabilité du transport de l'électricité

Les énergies renouvelables sont produites à partir de ressources naturelles renouvelables comme le soleil, le vent, la pluie, les marées, la biomasse et la chaleur géothermique, c'est-à-dire à partir de ressources qui se reconstituent naturellement avec l'écoulement du temps, soit par la reproduction biologique, soit par d'autres processus naturels récurrents [4], [6]. En tant que source d'énergie renouvelable non polluante et beaucoup moins chère que l'énergie nucléaire, l'énergie éolienne, parmi les différents candidats, a rencontré un succès grandissant partout dans le monde. L'utilisation de l'énergie éolienne a ainsi augmenté de façon spectaculaire au cours de la dernière décennie. Selon le rapport annuel de 2013 publié par l'Association Mondiale de l'Energie Eolienne (WWEA) [2], la capacité éolienne mondiale a atteint 296 GW à la fin du mois de juin 2013, dont 13980 MW de nouvelles capacités installées au cours du premier semestre 2013. Selon le rapport annuel de 2012 de l'Association Européenne de l'Energie Eolienne (EWEA) [7], la capacité installée dans l'Union européenne (UE) a augmenté d'environ 13 GW en 2000 mais de 107 GW en 2012. Cela répond aux besoins de puissance électrique de 57 millions de foyers et est équivalent à la fermeture de 39 centrales nucléaires [7]. Cette croissance continue et rapide indique que l'énergie éolienne représente une solution populaire, respectueuse de l'environnement et durable pour répondre au besoin croissant d'électricité. A titre d'exemple, la région occidentale du Danemark a l'un des taux de pénétration de l'énergie éolienne les plus élevés dans le monde, qui est resté stabilisé entre 25 et 30 % au cours des dernières années [5]. Selon le rapport statistique annuel de BP sur la consommation mondiale d'énergie, le pétrole reste la source d'énergie la plus utilisée dans le monde et représente 33,1% de la consommation mondiale d'énergie en 2012. Cependant, à l'échelle mondiale, il a eu le taux de croissance le plus faible parmi les combustibles fossiles, ce pour la troisième année consécutive [3].

L'évolution des réseaux électriques classiques vers des réseaux intégrant des sources distribuées d'énergies renouvelables induit des incertitudes supplémentaires concernant leur fonctionnement. En effet, les défis qui posent un fonctionnement fiable et sécurisé des systèmes électriques augmentent avec la proportion d'énergies renouvelables intermittentes (par exemple, éolienne, solaire, etc.) introduites dans les réseaux électriques. Du côté des fournisseurs, en particulier, l'intégration des sources d'énergie renouvelables (par exemple,

éolien et solaire) dans le réseau imposent des défis techniques et économiques, en raison de la difficulté de contrôle et de distribution de ces sources d'énergie due à leurs caractéristiques intermittentes.

Il convient de souligner que le recours à l'énergie éolienne va continuer à augmenter: l'Association Mondiale de l'Energie Eolienne (WWEA) a prédit une capacité éolienne mondiale potentielle de plus de 700 000 MW en 2020 [8]. A titre d'exemple à l'échelle d'un pays, le Danemark a pour objectif de produire 50% de son électricité à partir de sources d'énergies renouvelables en 2020. Ces objectifs montent ensuite à 100% de la production d'électricité et de chaleur en 2035, et 100% de la consommation d'énergie dans les transports en 2050. L'Ecosse a également pour objectif d'atteindre 100% d'énergies renouvelables dans sa production d'électricité à l'horizon 2020 [9]. Ces projections mettent en évidence l'importance de la maîtrise de l'intégration d'une grande quantité d'énergie éolienne au réseau électrique, c'est-à-dire en préservant la fiabilité du réseau.

Dans le Tableau 1, nous présentons les principaux avantages et inconvénients de l'exploitation de l'énergie éoliene. La vitesse du vent est une variable météorologique très irrégulière, avec des variations instantanées, horaires, journalières et saisonnières qui induisent une production d'électricité volatile. La nature volatile du vent pose un problème de prévisibilité du fonctionnement des éoliennes et de la gestion du réseau électrique. Il est ainsi nécessaire d'utiliser un modèle de prédiction, qui doit être également capable de fournir des informations sur l'incertitude de cette prédiction, afin de prendre les meilleures décisions en connaissance de cause.

Au long de cette thèse, nous fournissons donc un cadre utile pour la prédiction d'énergie éolienne qui permet, en outre, de quantifier l'incertitude de cette prédiction de manière pertinente.

Tableau 1. Les principaux avantages et inconvénients de l'exploitation de l'énergie éolienne.

Avantages	Inconvénients
• Propre (impacts environnementaux réduits). • Source gratuite et illimitée. • Peut être facilement utilisée par les ménages individuels dans les petites villes et les villages. • Economique (une des technologies les moins chères d'énergies renouvelables aujourd'hui disponibles). • Des éoliennes peuvent être construites dans les fermes ou ranchs, ce qui permet de générer des bénéfices pour l'économie locale de zones rurales. • Technologie relativement simple.	• Le vent est une source intermittente, la production d'énergie est donc variable. • Prévisibilité limitée en raison de ce caractère intermittent, variabilité et incertitude inhérentes au vent. • Les installations requièrent des investissements initiaux considérables. • La fabrication de turbines provoque des impacts environnementaux. • Les éoliennes sont bruyantes.

L' évaluation de l'adéquation d'un réseau électrique à la production d' énergie renouvelable distribuée est difficile en raison de nombreuses incertitudes, comme les fluctuations de la demande d'énergie, la prévision des conditions météorologiques (par exemple la vitesse du vent, le rayonnement solaire, etc.), l'indisponibilité de certains équipements (par exemple, les générateurs, les lignes, etc.), les défaillances dans les transactions d'énergie électrique, des erreurs de fonctionnement (erreurs de manipulation, dysfonctionnements des répartiteurs et des relais), etc. En particulier, la variabilité inhérente aux sources d'énergie renouvelables et les incertitudes associées peuvent avoir un impact significatif sur le réseau électrique, et des prédictions précises et fiables de la puissance de sortie obtenue à partir de ces sources sont nécessaires sur différentes échelles de temps. Ainsi, la prédiction de la production d'énergie à partir des sources d'énergie renouvelables est un point critique si l'on veut les intégrer efficacement au réseau électrique. D'autre part, une prédiction précise de la demande d'électricité est également capitale pour l'évaluation de la pertinence du système : elle permet aux opérateurs de réseaux et aux fournisseurs de services énergétiques de planifier l'allocation des ressources, et de mettre en place des stratégies optimisées de contrôle (par exemple, le pilotage de certains équipements en fonction de la demande, la révision des tarifs de l'électricité, etc.) pour assurer l'équilibre entre l'offre et la demande d'électricité. Par

conséquent, la résolution de problèmes de prédiction dans ce contexte d'évolution et d'adaptation progressive des réseaux électriques a généré une quantité considérable de travaux de recherche depuis plusieurs décennies.

En particulier, un axe de recherche important a consisté à développer des méthodes de prédiction précises et fiables de l'énergie éolienne et de la demande d'énergie électrique, et de nombreux systèmes de prédiction reposant sur des approches différentes ont été proposées. Une vue d'ensemble des méthodes de prédiction existantes qui ont été prises en compte dans cette thèse est donnée à la section 2.2.

Motivation et Objectifs

La prédiction joue un rôle clé dans de nombreux processus de décision et l'incertitude devrait être systématiquement prise en compte dans les résultats obtenus. L'incertitude de prédiction peut être due à des erreurs de mesure, à un manque de connaissances des données d'entrée, ou encore à des erreurs liées aux approximations faites pour établir le modèle de prédiction (imperfections dans la formulation du modèle, processus d'estimation, etc.).

Concernant les systèmes énergétiques, l'incertitude de prédiction des facteurs clés, due à la fois au caractère stochastique des données et aux approximations des modèles de prédiction, peut entraîner des coûts élevés pour les acteurs du marché (producteurs, clients, etc.) lorsqu'elle n'est pas correctement prise en compte. En particulier, dans les réseaux électriques intégrant de l'énergie éolienne, l'impact d'une telle source d'énergie très variable sur la fiabilité du système est un aspect important qui doit être évalué si le taux de pénétration de l'énergie éolienne (c'est-à-dire, le poids de l'énergie éolienne dans la réponse à la demande d'énergie électrique) est important. Par conséquent, compte tenu du fort taux de pénétration des sources d'énergie éolienne dans les nouveaux systèmes électriques en concurrence, les méthodes de prédiction fiables des vitesses de vent et de la puissance éolienne sont progressivement devenues des outils importants pour un management efficace et durable du marché de l'énergie: la combinaison des prédictions court-terme précises de la vitesse du vent et de la demande d'énergie permet aux opérateurs de réseaux d'ajuster, pour le jour suivant, l'étagement des différents moyens de production disponibles. Ceci permet de répondre à la demande en optimisant les coûts de fonctionnement du réseau tout en s'assurant d'une fourniture fiable et sécurisée.

La grande majorité des études existantes sur la prédiction de la vitesse du vent / puissance et la demande d'énergie ne fournissent que des prédictions ponctuelles, i.e. une valeur unique à chaque pas de temps considéré, sans tenir compte des incertitudes dans la structure du modèle et des données d'entrée.

Les intervalles de prédiction (PIs) constituent un moyen simple de communiquer une mesure de l'incertitude dans les prédictions. Un intervalle de prédiction est caractérisé par ses deux bornes dans lesquelles tombera vraisemblablement une nouvelle observation de y si elle fait partie de la même population statistique que l'échantillon. Plus de détails sur les PIs et leur application aux systèmes énergétiques sont donnés à la Section 2.3. Ici, il suffit de mentionner que deux éléments caractérisant les PIs sont leur probabilité de couverture (PICP) et leur largeur (PIW), qui doivent être respectivement maximisé et minimisé.

Dans ce travail, nous appliquons donc les PIs à l'analyse de l'adéquation des réseaux électriques à l'énergie éolienne. Nous nous intéressons à des prédictions court-terme de la vitesse du vent, de la puissance éolienne produite et de la demande d'énergie, car le fonctionnement des systèmes électriques est étroitement lié à ces variations court-terme (gestion opérationnelle des unités de puissance, etc.). Plus précisément, nous nous intéressons à l'estimation les PIs via des réseaux de neurones artificiels (NNs), en améliorant une méthode existante dite "*Lower Upper Bound Estimation Method for Construction of NN-based PIs* (LUBE)". Nous nous plaçons dans un cadre d'optimalité multi-objectifs de Pareto qui tient compte à la fois du PICP et du PIW. Ils s'expriment comme suit [36]:

$$PICP = \frac{1}{n_p}\sum_{i=1}^{n_p} c_i \tag{1}$$

où $c_i = 1$ si $y_i \in [L(x_i), U(x_i)]$, et $c_i = 0$ sinon.

$$NMPIW = \frac{1}{n_p}\sum_{i=1}^{n_p} \frac{(U(x_i)-L(x_i))}{y_{max}-y_{min}} \tag{2}$$

où $NMPIW$ est la largeur de l'intervalle de prédiction par Moyenne Normalisée , et y_{min} et y_{max} représentent respectivement les valeurs-cibles minimale et maximale dans le jeu d'apprentissage. $L(x_i)$ et $U(x_i)$ sont respectivement les limites inférieure et supérieure de la PI estimées pour la sortie $y(x_i)$ qui correspond à l'entrée x_i.

Concernant les questions présentées ci-dessus, les objectifs suivants ont guidé le travail effectué au cours de cette activité de recherche doctorale:

1. Développer une méthode de prédiction capable de tenir compte de l'incertitude dans les paramètres du modèle qui affectent la prédiction;
2. Représenter l'incertitude des données d'entrée et la propager par l'intermédiaire du modèle de prédiction afin d'en observer l'effet sur les résultats du modèle;
3. Améliorer la performance d'une méthode de prédiction non-paramétrique basée sur des ensembles de réseaux de neurones
4. Tester le modèle proposé sur des études de cas réels dans le cadre d'applications à des systèmes énergétiques (en particulier en ce qui concerne l'évaluation de l'adéquation des réseaux électrique).

Organisation

La thèse se compose de deux parties: la Partie I, composée de huit Chapitres, présente les enjeux et les défis d'importance pour les systèmes de production décentralisés. Elle décrit les objectifs de recherche entrepris, illustre les méthodes mises au point et appliquées dans ce travail de thèse, discute quelques-uns des résultats obtenus dans les études de cas réalisées et fournit des conclusions générales et des perspectives pour des travaux futurs. La deuxième partie est constituée d'un ensemble de sept articles sélectionnés, qui détaillent les travaux scientifiques effectués au cours de cette thèse et leurs résultats.

Dans l'article I [38], nous avons implémenté NSGA-II pour l'apprentissage un perceptron multicouche (MLP NN) afin de fournir les PIs de dépôt de calcaire sur les équipements de pétrole et de gaz, dans un cadre multi-objectif visant à minimiser le PIW et à maximiser en même temps le PICP des intervalles de prédiction (PIs) estimés. Nous avons effectué *k*-validation croisée (CV) pour guider le choix de la structure NN (i.e. le nombre de neurones dans la couche cachée) avec une bonne performance de généralisation. Nous avons utilisé un indicateur métrique *hypervolume* pour comparer les fronts de Pareto obtenus dans chaque échantillon de CV. Les expériences ont été faites avec des entrées à valeur unique.

L'article II présente une comparaison de l'algorithme génétique mono-objectif (SOGA) qui a été présenté dans l'article original LUBE avec l'algorithme du recuit simulé (SOSA) et la Moyenne Mobile Intégrée Autorégressive (ARIMA) pour la prédiction de la vitesse du

vent à court terme (l'heure suivante) sur une étude de cas réel qui se compose de quatre différents profils de jeux de données historiques.

De manière similaire à l'article II, nous avons effectué dans l'article III une comparaison entre MOGA-NN méthode et l'Extreme Learning Machines (ELM) combiné avec la méthode du plus proche voisinage pour l'estimation des PIs. Les algorithmes sont utilisés pour la prédiction de la vitesse du vent à court terme (l'heure suivante) en utilisant une étude de cas réel qui se compose de trois différents profils de jeu de données historiques.

L'article IV propose une approche basée sur l'analyse d'intervalle et généralise le cadre de l'estimation multi-objectif PI basée sur RN à la prévision des séries chronologiques en se basant sur la théorie de l'analyse d'intervalle, i.e. avec intervalles de données. Dans cet article, nous cherchons à quantifier l'incertitude dans la prédiction en combinant les incertitudes associées à la fois aux données d'entrée et au modèle de prédiction. La démonstration de la méthode proposée est faite sur deux études de cas: (*i*) une étude de cas synthétique, avec 5 minutes de données simulées; (*ii*) une étude de cas réel, impliquant des mesures de la vitesse du vent horaire. Dans les deux cas, la prévision à court terme (à 1 heure et à un jour, respectivement) est effectuée en prenant en compte à la fois de l'incertitude dans la structure du modèle, et la variabilité (intra-heure et intra-jour, respectivement) dans les données.

L'article V présente un cadre de modélisation et de simulation pour la conduite de l'évaluation de l'adéquation d'un système de puissance intégrée à éolienne tenant compte des incertitudes associées. Notre approche de l'évaluation de l'adéquation permet l'évaluation de l'Expected Energy not Supplied (EENS) en considérant des données de la vitesse du vent et de la demande d'énergie sous forme d'intervalle. L'originalité du travail réside dans la proposition non seulement d'un indice de fiabilité d'une valeur unique, i.e. point, mais également de résultats de valeurs d'EENS sous forme d'intervalle permettant d'informer les décideurs (DMs) sur l'incertitude dans les prédictions.

Dans l'article VI, une nouvelle approche pour la prévision de l'énergie éolienne, i.e. de la production de puissance électrique d'origine éolienne, avec la quantification des incertitudes est décrite. Cette approche peut être schématisée en deux étapes: d'abord, l'estimation des PIs de la vitesse du vent à court terme est réalisée dans le cadre de l'optimisation multi-objectif élaboré par NSGA-II. Ensuite, l'incertitude de la vitesse du vent et celle dans la courbe de

puissance sont combinées par une technique bootstrap, obtenant ainsi les PIs de l'énergie éolienne avec la même couverture que les PIs de la vitesse du vent.

L'article VII présente une méthode d'ensemble de NNs pour estimation des intervalles de prédiction de la vitesse du vent et vise à proposer une version améliorée de la méthode MOGA-NN non-paramétrique proposée dans ce travail de thèse (voir les Chapitres 4 et 5, et les articles I-III). Nous proposons deux méthodes d'ensemble de NNs, différant par la séparation ou non de jeux de données, et intégrant l'algorithme des *k*-plus proches voisins (*k*-nn) à la phase d'agrégation pour l'identification des voisins d'une entrée dans un jeu de donné test. Sur les données réelles considérées comme étude de cas, les deux méthodes ont obtenu des résultats supérieurs à ceux donnés par les réseaux individuels sélectionnés choisis dans les ensembles respectifs.

Estimation Intervalles de Prédiction (PIs) via Réseaux de Neurones (NNs)

Les principales techniques utilisées pour estimer les PIs pour les sorties du modèle NN sont les méthodes Delta, Bayesienne, et de Bootstrap. Les méthodes de Bayes et Delta sont fondées sur des bases mathématiques solides. Une comparaison de ces trois méthodes a été effectuée par Khosravi et al. sur différentes études de cas dans [35], et par (l'auteur) dans [94] dans lequel l'auteur considère le problème de prédiction des temps de trajet de bus par autoroute.

La méthode Delta est basée sur un développement de Taylor de la fonction de régression non linéaire [95]. Cette méthode nous permet de générer des PI de haute qualité. Cependant, le calcul d'une matrice Jacobienne et l'estimation de la variance du bruit non biaisée nécessaires à l'application de cette méthode requière un grand temps de calcul lors de sa phase de développement.

L'approche bayésienne utilise les méthodes statistiques Bayésiennes pour exprimer l'incertitude des paramètres du réseau de neurones en tant que distributions de probabilité, avant de les intégrer afin d'obtenir la distribution de probabilité a posteriori de la cible conditionnelle sur l'ensemble de la formation observée [58], [96].

Les fondements mathématiques axiomatiques forts rendent cette méthode robuste et répétable. En fin de compte, NNs formées par une technique d'apprentissage en bayésienne ont le

pouvoir de généralisation supérieur [35]. Ainsi, les méthodes NNs utilisant une technique d'apprentissage bayésienne ont un meilleur pouvoir de généralisation. Le temps de calcul requis ici est également élevé, en raison du calcul d'une matrice Hessienne dans l'étape de développement.

La méthode Bootstrap est fréquemment grâce à sa simplicité d'utilisation comparativement aux précédentes méthodes [35]. Il s'agit d'une technique de ré-échantillonnage qui permet d'assigner des mesures de précision des estimations statistiques sans nécessiter le calcul des matrices et des dérivés de [97] complexes, [98]. Le but du ré-échantillonnage est de produire des estimations moins biaisées de la véritable régression des objectifs, et d'améliorer les performances de généralisation du modèle [35]. Les principaux inconvénients de cette méthode sont les suivants: i) grand temps de calcul lorsque les ensembles de formation et des réseaux de neurones sont grandes; ii) avec un petit nombre de modèles d'entrée, les réseaux de neurones individuels ont tendance à être trop formés, conduisant à une performance de généralisation pauvre [35], [99].

La caractéristique commune des méthodes d'estimation PI mentionnés ci-dessus, est qu'ils ne prennent pas en compte les largeurs des intervalles dans le processus d'estimation [35]. En ce qui concerne ce point, Khosravi et al. [36] ont proposé le LUBE, dans lequel ils obtiennent des intervalles de prédiction basées sur NN en tenant compte à la fois CP et PIW dans la phase de construction de PI. Ces deux mesures quantitatives déterminent la qualité des PIs estimées. Le PICP représente la probabilité que le jeu des PIs estimées contiendra les vraies valeurs de sortie, calculée à travers la proportion de vraies valeurs de sortie se trouvant dans les PIs estimées; PIW mesure simplement la prolongation de l'intervalle comme la différence entre les valeurs liées limite inférieure et supérieure estimées. Ce sont des mesures contradictoires générales (intervalles plus larges donnent la plus grande couverture), et dans la pratique, il est important d'avoir des PIs étroites à haute probabilité de couverture. La définition mathématique des mesures PICP et PIW sont définies dans (1) et (2).

Dans cette thèse, notre méthode basée sur les réseaux de neurones pour estimer les PIs ne nécessite aucune hypothèse sur la distribution de probabilités qui engendre les données. La seule hypothèse que nous faisons est que les données sont indépendantes et identiquement distribuées (i.i.d.), ainsi nous proposons une approche empirique et non-paramétrique pour l'estimation des PIs. Dans ce travail, nous étendons la méthode LUBE [36] pour la

formulation multi-objectif du problème d'estimation des PIs. Plus précisément, nous estimons les PIs en exploitant la propriété qu'ils constituent des optima de Pareto, ainsi nous utilisons Non-dominated Sorting Genetic Algorithm (NSGA-II) [44] pour optimiser le paramètre d'une MLP NN en minimisant la PIW et maximisant la PICP simultanément. Notez que l'approche proposée dans ce travail de recherche intègre l'estimation de la PI dans sa procédure d'apprentissage, tandis que toutes les autres méthodes (décrites plus haut) à l'exception de LUBE construisent les PIs en deux étapes (la première estimant le point prédiction et la seconde construisant le PI). Sur la Figure 1, la structure typique à trois couches (d'entrée, cachée et de sortie) MLP utilisée dans cette thèse pour construire les PIs est illustrée : le neurone de sortie ci-dessus donne la limite supérieure du PI et celui ci-dessous fournit la limite inférieure. Par ces deux neurones de sortie, la MLP génère un intervalle de prédiction pour chaque d'entrée.

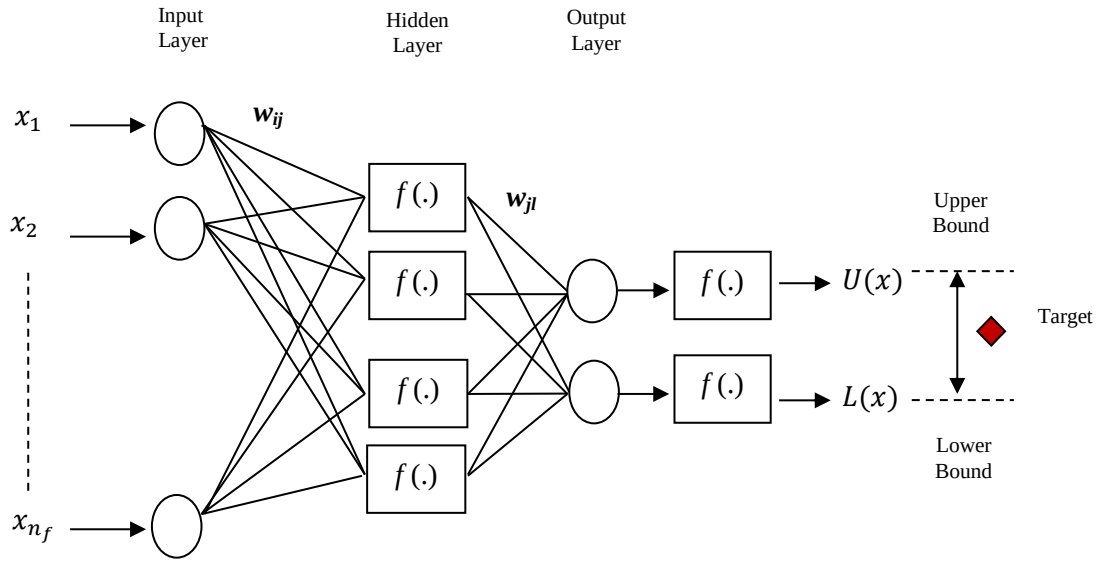


Figure 1. La structure le MLP utilisée dans cette thèse pour construire les PIs.

Conclusions Générales et Perspectives

L'objectif principal de toute installation d'électricité est de répondre à la demande d'énergie à tout moment au plus bas coût possible pour les clients tout en maintenant un niveau satisfaisant de qualité de service. Pour cet objectif, il est crucial de disposer d'une bonne information sur la production et la demande futures. Les prévisions de la demande d'énergie et de la production peuvent être incluses directement dans la gestion opérationnelle des unités de

puissance utilisée pour veiller à ce que suffisamment d'électricité soit générée pour répondre à la demande, et/ou elles constituent une information à court-terme aux opérateurs du réseau électrique. Pour ce faire, les méthodes et les modèles de prévision doivent fournir un moyen de mesurer le risque liés à l'utilisation des résultats estimés dans le processus de décision en raison des incertitudes associées.

C'est dans ce contexte que, dans cette thèse de doctorat, nous avons développé une approche non paramétrique, empirique pour générer des PIs basés sur les réseaux de neurones pour tenir compte de l'incertitude dans la prédiction en raison de la variabilité des données d'entrée et des approximations du modèle. Comme application, nous avons considéré le problème général de l'évaluation de l'adéquation des réseaux électriques éoliens intégrés. Dans les études de cas, nous nous sommes concentrés en particulier sur les prévisions de la vitesse/puissance du vent et de la demande d'énergie à court terme (à des horizons variant d'une heure à un jour), pour leur importance pour le fonctionnement du système à la fois du point de vue de l'engagement de l'unité et des phases d'expédition économiques.

Les principales contributions de la thèse dans le domaine de l'évaluation de l'adéquation des réseaux d'énergie, en particulier dans les systèmes intégrés éoliens de production distribuée, présentées dans les articles I à VII de la Partie II sont formalisées comme des réponses aux objectifs de cette recherche.

Les contributions au titre de l'objectif 1 :

Objectif 1: Développer une méthode de prédiction capable de tenir compte de l'incertitude dans les paramètres du modèle qui affectent la prédiction.

- Nous présentons un cadre d'analyse multi-objectif pour estimer des intervalles de prédiction, qui seront optimaux en termes de précision et de largeur d'intervalle. Plus précisément, nous proposons une méthode multi-objectif d'estimation des intervalles de prédiction basés sur les réseaux de neurones capable de quantifier les incertitudes associées à la prédiction. Avec cette méthode, nous sommes capables de générer la frontière de Pareto des solutions non-dominées. Chaque solution correspond à un réseau de neurones. Les décideurs peuvent choisir une solution sur cette frontière de Pareto selon leurs préférences (i.e. en termes d'arbitrage entre un PICP élevé et un NMPIW faible).

- La connaissance des intervalles de prédiction permet aux décideurs et aux opérateurs de réseaux de quantifier le niveau d'incertitude associé aux prédictions et d'envisager une variété de solutions/scénarios pour les meilleures et pires conditions

- Nous utilisons NSGA-II qui est l'un des MOEAs les plus puissants, pour l'apprentissage du NN. Une comparaison avec un autre algorithme d'optimisation multi-objectif puissant, MO-CMA-ES, a été effectuée. Les résultats de la comparaison ont montré que les PIs produits par NSGA-II sont supérieurs à ceux obtenus avec MO-CMA-ES, et satisfaisants à la fois en termes de couverture élevée et de largeur faible. Il convient de souligner que c'est la première fois que NSGA-II est utilisé pour résoudre le problème de la détermination des bornes optimales (inférieure et supérieure) des PIs.

- Afin de montrer la supériorité de la méthode multi-objectif proposée sur les méthodes mono-objectif, en particulier sur la méthode originale de LUBE [36], dans l'article II, nous avons effectué des comparaisons sur différents jeux de données. En outre, dans l'article II, nous avons également effectué une comparaison avec une méthode de régression des séries temporelles classique (i.e. ARIMA). Les résultats confirment la supériorité de notre approche MOGA-NN.

Les contributions au titre de l'objectif 2:

Objectif : Représenter l'incertitude des données d'entrée et la propager par l'intermédiaire du modèle de prédiction afin d'en observer l'effet sur les résultats du modèle;

- Afin de représenter l'incertitude dans les données d'entrée et la propager à travers le modèle de prédiction sur ses résultats, nous présentons un modèle de prédiction des séries temporelles en formes d'intervalles, basé sur les NNs. Par la représentation par intervalles, nous pouvons traduire la variabilité dans les données d'entrée (ex: vitesses du vent extrêmes dans une zone donnée, pic de demande d'électricité durant le jour, températures minimale et maximale, etc.), ou les incertitudes qui leur sont associées (ex: distributions de la vitesse du vent fortement asymétriques, profil non-stationnaire de la demande, etc.).

- Nous avons présenté deux approches qui peuvent être utilisées pour l'apprentissage des NNs par des entrées en formes d'intervalles, qui visent à fournir une quantification plus précise de l'incertitude d'entrée dans le problème de la prédiction. Les résultats de l'expérience montrent que l'approche par les données d'entrée sous forme d'intervalles est

capable de capturer la variabilité dans les données d'entrée avec une couverture satisfaisante. Les résultats nous permettent de planifier des stratégies différentes en fonction de l'éventail des résultats possibles au sein des intervalles de prédiction.

- En ce qui concerne les résultats de la comparaison d'études de cas, nous pouvons conclure que notre méthode de prédiction de la vitesse du vent pour le jour à venir par intervalle est plus performante que la méthode des intrants à valeur unique.

- De plus, les résultats de la comparaison effectuée entre les deux algorithmes d'apprentissage, i.e. SOSA et NSGA-II, montrent la supériorité de ce dernier dans l'apprentissage du MLP NN dans notre problème spécifique.

Les contributions au titre de l'objectif 3:

Objectif 3: Améliorer la performance d'une méthode de prédiction non-paramétrique basée sur des ensembles de réseaux de neurones

- Cet objectif est abordé à travers deux méthodes pour estimer des PIs de la vitesse du vent à court terme dans le cadre d'une nouvelle approche de l'ensemble des NNs. Dans la phase d'agrégation des résultats NN individuels sélectionnés, nous avons utilisé l'approche de *k*-nn pour déterminer les échantillons-points similaires entre les données d'apprentissage et les données de tests. Cela nous permet d'obtenir des résultats plus précis également sur le jeu de test en utilisant les informations locales provenant des échantillons les plus proches des jeux d'apprentissage.

- Les deux méthodes montrent des résultats cohérents et de haute précision par rapport aux NN individuels dans l'ensemble et aux méthodes conceptuellement similaires vues dans la littérature [150].
- Nous pouvons conclure que l'approche par l'ensemble des NNs proposée dans l'article VII peut apporter une amélioration significative de la qualité de la prédiction de la vitesse du vent à court-terme.

Les contributions au titre de l'objectif 4:

Objective 4: Tester le modèle proposé sur des études de cas réels dans le cadre d'applications à des systèmes énergétiques (en particulier en ce qui concerne l'évaluation de l'adéquation des réseaux électriques).

- Dans l’article I, les données ont été obtenues à partir d'expériences visant à observer le processus de dépôt de calcaire dans [117], [119].

- Dans les articles II et VII, le test de l'approche MOGA-NN proposé se fait sur plusieurs bases de données différentes concernant la prévision de la vitesse du vent à court terme et de la demande. Les jeux de données de vitesse de vent montrent des profils de vitesse du vent différents selon la saisonnalité, mesurée pour la région de Regina, en Saskatchewan, au Canada.

- Les fluctuations de la demande horaire sont modélisées à l'aide de la courbe de charge annuelle chronologique de l’IEEE Reliability Test System (RTS) [10], avec le pic annuel de demande mesuré.

- Les études de cas sur différents jeux de données nous ont permis de tester la performance de la méthode MOGA-NN sur divers jeux de données présentant divers degrés de variabilité.

Comme toutes les méthodes d'apprentissage, les RN présentent certaines limites outre leurs avantages. En général, les RN ont une performance satisfaisante en prévision. Leur capacité à apprendre la relation non linéaire entre l'entrée et la sortie, ainsi que leur faculté à utiliser des fonctions arbitraires les rendent aptes et prometteur pour les tâches de prévision. D'autre part, les NNs sont guidés par les données et dépendent fortement de la représentativité du jeu de données d’apprentissage, i.e. les méthodes de prédiction par les données sont sujettes à donner des résultats moins précis lorsque la variabilité est forte dans le jeu de données test considéré. Par conséquent, la précision de la prédiction peut diminuer lorsque le jeu de données de test présente une grande variabilité par rapport au jeu de données d’apprentissage. En d'autres termes, la différence entre les jeux de données de test et d’apprentissage joue un rôle important dans le pouvoir de généralisation du modèle. Par conséquent, les méthodes de prédiction fondées sur les données ne garantissent pas toujours des prévisions de haute qualité sur les données invisibles. En outre, un modèle NN peut exiger une procédure intensive pour l’apprentissage, ce qui requiert de grands temps de calcul. La plupart du temps, le temps de calcul est en corrélation avec la taille du réseau, i.e. le nombre de paramètres qui doivent être optimisés et le nombre d’échantillons dans le jeu de données.

Il convient de souligner qu'aborder un problème de régression nécessite également prétraitement approprié des données d'entrée. La façon de sélectionner les variables d'entrée pertinentes pour la (les) variable(s) de sortie, i.e. la sélection des fonctionnalités, pour inclusion dans un modèle est un facteur important qui affecte à la fois la précision de la prédiction et le coût de calcul du modèle sous-jacent. Pour la prévision des séries chronologiques, il faut déterminer convenablement le nombre de retards à introduire. Dans cette thèse, nous abordons également ces questions dans nos études de cas, avec des techniques classiques.

Perspectives

Le travail entrepris dans cette thèse peut être développé selon plusieurs directions :

- Combiner avec des méthodes de prévision différentes pour réduire l'erreur de prédiction.
- Implémenter des algorithmes d'apprentissage en ligne en mesure d'ajuster leurs paramètres aux nouveaux échantillons sans réapprendre. L'utilisation complémentaire des données de mesure des vents en temps réel au potentiel d'améliorer les prévisions en particulier lorsque l'ensemble de données disponibles est trop courte pour couvrir tous les motifs possibles ou lorsque les conditions environnementales ou opérationnelles changent.
- Les sous-ensembles flous de type 2 peuvent être intégrés dans le modèle proposé comme une alternative pour représenter l'incertitude d'entrée.
- Une nouvelle formulation, en particulier pour la prévision de l'énergie éolienne offshore, peut être fournie en exploitant la corrélation spatio-temporelle.
- La méthode proposée peut être intégrée dans un modèle de coûts pour estimer l'incertitude sur les prix de l'électricité.
- Les domaines d'application peuvent être élargis. Par exemple, pour la prévision de la demande d'énergie, la consommation d'énergie dans les bâtiments peut être considérée comme une étude de cas prenant en compte les incertitudes potentielles pour la gestion de l'énergie.

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

PART I

PART II

Paper I R. Ak, Y. F. Li, V. Vitelli, E. Zio, E. López Droguett and C. Magno Couto Jacinto. "NSGA-II-trained neural network approach to the estimation of prediction intervals of scale deposition rate in oil & gas equipment," *Expert Systems with Applications*, vol. 40, no. 4, pp. 1205-1212, March 2013.

Paper II R. Ak, Y. F. Li, V. Vitelli and E. Zio. (2014). "Multi-objective Genetic Algorithm Optimization of a Neural Network for Estimating Wind Speed Prediction Intervals," submitted to *Applied Soft Computing* (under review).

Paper III R. Ak, O. Fink and E. Zio. (2014). "Two Machine Learning Approaches for Short-Term Wind Speed Time Series Prediction," submitted to *Special Issue on Neural Networks and Learning Systems Applications in Smart Grid* (under review).

Paper IV R. Ak, V. Vitelli and E. Zio. (2014). "An Interval-Valued Neural Network Approach for Prediction Uncertainty Quantification," submitted to *IEEE Transactions on Neural Networks and Learning Systems* (under review).

Paper V R. Ak, Y. F. Li, V. Vitelli and E. Zio. (2014). "Adequacy Assessment of a Wind-integrated Power System using Neural Network-based Interval Predictions of Wind Power Generation and Load," submitted to *International Journal of Electrical Power & Energy Systems* (under review).

Paper VI R. Ak, V. Vitelli and E. Zio. "Uncertainty Modeling in Wind Power Generation Prediction by Neural Networks and Bootstrapping," *In Proc Esrel 2013 Conference*, 29 Sept. – 2 Oct. 2013, Amsterdam.

Paper VII R. Ak, V. Vitelli and E. Zio. (2014). "Ensemble of Neural Networks for Estimating Short-term Prediction Intervals of Wind Speed", (under preparation).

ACRONYMS

ARIMA	Autoregressive Integrated Moving Average Model
ARMA	Autoregressive Moving Average Model
CV	Cross-validation
CWC	Coverage Width-based Criterion
DE	Differential Evolution
DER	Distributed Energy Resources
DG	Distributed Generation
EENS	Expected Energy Not Supplied
ELM	Extreme Learning Machines
ES	Exponential Smoothing
FNNs	Feed-forward neural networks
GA	Genetic Algorithm
LUBE	Lower Upper Bound Estimation Method for Construction of NN-based PIs
MA	Moving Average
MC	Monte Carlo
MLP	Multi-layer Perceptron
MOGA	Multi-objective Genetic Algorithm
MOP	Multi-objective Optimization Problem
NMPIW	Normalized Mean Prediction Interval Width
NNs	Artificial Neural Networks
NSGA-II	Non-dominated Sorting Genetic Algorithm-II
NWP	Numerical Weather Prediction
PICP	Prediction Interval Coverage Probability
PIs	Prediction Intervals
PIW	Prediction Interval Width
RBF NNs	Radial Basis Function Neural Networks
RNNs	Recurrent Neural Networks
RSS	Residual Sum of Squares
RTS	IEEE Reliability Test System
SA	Simulated Annealing
SMCS	Sequential Monte Carlo Simulation
SOGA	Single-objective Genetic Algorithm
SOSA	Single-objective Simulated Annealing
STLF	Short-term Load Forecasting
SVM	Support Vector Machine
WTG	Wind Turbine Generator

1. INTRODUCTION

The research presented in this Ph. D. concerns the development of regression and prediction methods for application to energy systems, and particularly the adequacy assessment of renewable power generation systems. Specific attention is given to uncertainties, in the system behavior (due to renewable sources stochasticty and system component failures) and in the prediction results themselves. The present introductory chapter of the thesis is structured as follows. Section 1.1 discusses some current main issues, challenges and needs in the energy sector. Section 1.2 describes an example of distributed power generation system with renewables and introduces the related adequacy assessment problem. Section 1.3 reviews the prediction problem in general terms and its specific role in the adequacy assessment of distributed power generation systems. Section 1.4 provides a statement of the research motivations and objectives pertinent to this applicative context, and introduces the methods proposed in this thesis work. Finally, Section 1.5 presents the structure of the thesis.

1.1 Energy Sector: issues, challenges and needs

Energy is an economic, environmental, social and political issue, and a balance between energy management, economic interests, and the environment care is needed for sustainable development. The world energy demand continues to grow and must be satisfied in an efficient and effective way. The tendency towards increased mobility, urbanization and industrialization, especially in developing countries, and an integrated global economy will further accelerate the worldwide energy consumption and dependence. Today, the primary means to meet the electrical energy demand is combustion of the fossil fuels (e.g. oil, coal, and natural gas). Figure 1 indicates the world total primary energy supply by source [1]. Although fossil fuels can produce a significant amount of energy, the main drawbacks of fossil fuel energy sources are their limited reserves and their negative effects on the environment. The climate changes induced by the impact on the environment, in particular, pose an additional challenge for energy suppliers, consumers and market operators. To fight climate change, i.e. to reduce the negative effects (e.g. greenhouse gases) of the traditional energy sources, the usage of the fossil energy sources should be replaced with alternative energy sources, i.e. renewable energy sources, which are cleaner and less harmful to the people and the environment.

In general, the main goal of the energy systems should be to meet the energy demand at any time by reducing the environmental impacts, by developing consistent, reliable and clean forms of energy. However, energy market participants (investors, power producers, system operators, consumers, etc.) face some potential challenges [1]-[5]:

- growing energy demand
- new challenges in the energy consumption patterns
- integration of intermittent (stochastic) renewable energy sources into the electricity grids
- extracting fossil fuels under extreme conditions in a more economical and safer manner
- building up refining capacity
- reducing nuclear waste
- ensuring more reliable means for transporting electricity

Renewable energy is generated from natural renewable resources such as sunlight, wind, rain, tides, biomass and geothermal heat, i.e. naturally replenished with the passage of time, either through biological reproduction or other naturally recurring processes [4], [6]. As a non-polluting renewable energy source considerably cheaper than nuclear energy, wind energy, among the various candidates, has received fast growing attention throughout the world, and the utilization of wind power has increased dramatically over the past decade. According to the Half-Year Report 2013 released by The World Wind Energy Association (WWEA) [2], the worldwide wind capacity reached 296 GW by the end of June 2013, out of which 13980 MW were added in the first half of 2013. According to the Annual Report 2012 of the European Wind Energy Association (EWEA) [7], the installed capacity in the European Union (EU) has increased from around 13 GW in 2000 to 107 GW in 2012. This meets the power needs of 57 million households, and it is equivalent to the output of 39 nuclear power plants [7]. This continuous and rapid growth indicates that wind energy represents a popular, respectful of the environment and sustainable solution for meeting the increasing need of electricity. For exemplification, the Western Denmark region has one of the highest wind power penetrations in the world, consistently between 25 and 30% over the last few years [5].

According to the BP Statistical Review of World Energy, oil remains the world's leading fuel source representing 33.1% of global energy consumption by 2012. However, it had the weakest global growth rate among fossil fuels for the third consecutive year [3].

The evolution from conventional power grids towards grids with integration of distributed renewable energy sources leads to additional uncertainty in the system. Indeed, the challenge of reliably and safely operating power systems increases with the proportion of intermittent renewable energy (e.g. wind, solar, etc.) which is fed into the energy grids. From the supplier side, particularly, the integration of renewable energy sources (e.g. wind and solar) into the grid imposes an engineering and economic challenge, because of the limited ability to control and dispatch these energy sources due to their intermittent characteristics.

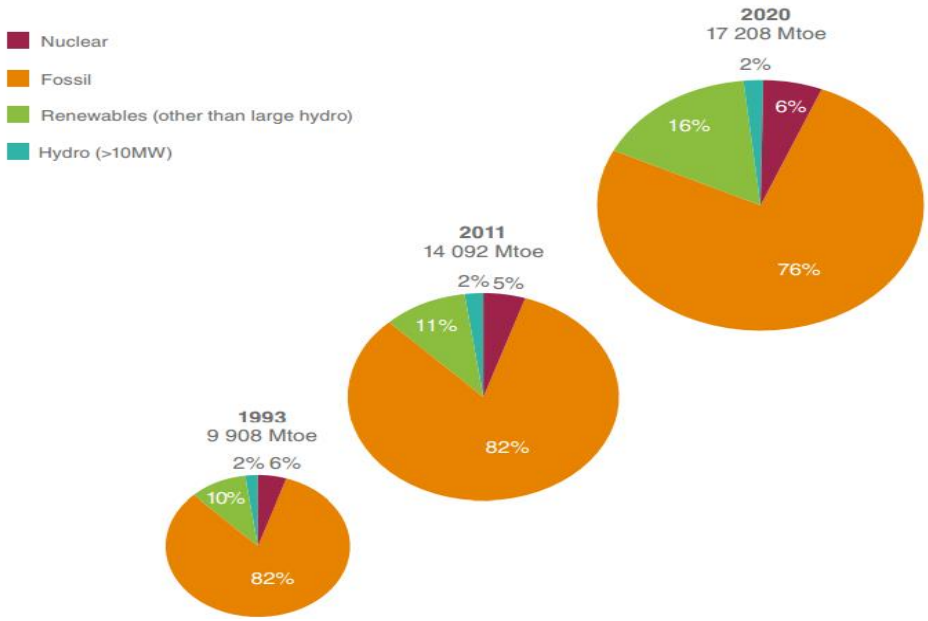


Figure 1. Total primary energy supply by source (source: World Energy Resources Survey 2013) [1].

It is worth pointing out that the use of wind energy will continue to increase: the World Wind Energy Association (WWEA) has predicted a possible wind capacity of more than 700000 MW by 2020 [8]. As an exemplification to the country level, Denmark proposes to

meet more than 50% of its electricity supply with renewables by 2020, 100% of electricity and heat by 2035, and 100% of transport by 2050, whereas Scotland has a mandate to achieve 100% renewable power supply by 2020 [9]. These projections enhance the importance of the reliable integration of large amount of wind energy to the power grid without harming its reliability.

In Table 1, we report the major advantages and drawbacks/challenges in the exploitation of wind energy [5], [10]. Wind speed is a highly variable meteorological variable with instantaneous, hourly, diurnal and seasonal variations, thus producing volatile power delivery. The volatile nature of wind poses a problem of predictability for wind turbine operation and energy system management. Then, a prediction model must be introduced, capable of providing also information on the uncertainty of the prediction, for informed decision-making.

Along these lines, in this Ph. D. work, we provide a useful framework for prediction and pertinent uncertainty quantification.

Table 1. Some advantages and challenges of wind energy.

Advantages	Challenges
• Clean (no impact on the environment) • Free and unlimited source • Can be easily used by single households in small towns and villages • Economical (one of the lowest-priced renewable energy technologies available today) • Can be built on farms or ranches, thus benefiting the economy in rural areas • Relatively simple technology	• Dependent on the availability of wind • Limited predictability owing to intermittent character, inherent variability and uncertainty of the wind • Requires high initial investment • Pollution deriving from turbines manufacturing • Wind turbines are noisy

1.2 The Adequacy Assessment of Distributed Power Generation Systems

Distributed generation (DG), also called decentralized generation, consists of a set of electric power units connected to the distribution network/grid [11], [12]. As DG power plants refer to a variety of small-scale power generators, DG plants produce power in capacities that

range from a fraction of a kilowatt (KW) to about 100 megawatts (MW) [13], [14]. Being "distributed", they can be opportunistically placed at or near the points of energy consumption to reduce losses and to better meet the consumers demands, unlike traditional "centralized" systems, whereby electricity is generated at a remotely located, large-scale power plant and, then, transmitted down power lines to the consumer. The most common DG technologies include Combined Heat and Power (CHP) generators, micro-turbine gas generators, solar photovoltaic generators, wind generators, fuel cells, battery storages and micro-hydro schemes [15].

Several benefits can be obtained when Distributed Energy Resources (DER) are correctly integrated to the power grid. First, the reliability of electric power systems can be increased by distributed generation [16]. A power system based on a large number of reliable small generators can operate with the same reliability and a lower capacity margin than a system of equally reliable large generators [17]. Moreover, DER are more economical: transmission costs can be reduced by allocating the generation closer to the load, and construction time and investment costs are lower for smaller power generation plants than for larger central power plants [18]. However, the main drawback of DER is that the uncertainties involved in system planning and operation become larger especially with the increasing use of renewable energy sources.

The assessment of the reliability of these systems can be performed by considering two main aspects: adequacy and security [19]. Power system adequacy is an indicator of the availability of a sufficient power capacity installed within the system to satisfy the consumer load demand at any time without violating the system operational constraints. This requires the necessary facilities to generate sufficient energy, and the associated transmission and distribution facilities required to transport the energy to consumer load points [19], [20]. Then, the basic elements of generating capacity adequacy assessment are the system generating facilities and the system load demands.

Considerable work has been done to develop mathematical models and techniques for the reliability evaluation of power systems, including wind energy integrated power grids. In this regard, some of the existing works have been solely devoted to adequacy evaluation of wind farms [21], whereas some handle the adequacy assessment problem of the power grids integrated to a renewable energy source (e.g. wind, solar, etc.) [22], [23]. In general, reliability indices calculation may be performed considering deterministic and probabilistic

approaches [24]. The main drawback of deterministic techniques is that they do not take into account the probabilistic or stochastic nature of system behavior (e.g. component failures, uncertainties of renewable energy sources and customer energy demands) [25]. The probabilistic approaches used in power system reliability evaluation can be classified into two basic categories: analytical and Monte Carlo (MC) simulation methods [25], [26]. Sequential Monte Carlo Simulation (SMCS) is ideally suited to the analysis of intermittent generating sources such as wind power. An important advantage of using SMCS in bulk electric system reliability evaluation is its ability to incorporate the chronological characteristics of wind speed (diurnal and seasonal), the load profiles, and the chronological transition states of all the components within a system. Sequential simulation can, thus, provide realistic and more accurate results than other traditional methods when considering wind power [25]-[27].

Most existing adequacy assessment methods provide a point estimate of the reliability indices such as loss of load probability (LOLP), loss of load expectation (LOLE), loss of energy expectation (LOEE), Expected energy not supplied (EENS), frequency of loss of load (FLOL), etc. [21], [25], [28], [29]. In the context of this Ph. D. work, we propose an innovative approach to provide not only point-valued reliability indices, but also interval-valued results to inform the decision makers (DMs) on the uncertainty in the predictions. Herein, the uncertainties considered in the system are due to load fluctuations, wind variability, and component failures.

1.3 The Prediction Problem and Its Role in the Adequacy Assessment of Distributed Power Generation Systems

The adequacy assessment of a power system with distributed renewable power generation is challenging due to the many uncertainties, like fluctuations in energy demand, future weather conditions (e.g. wind speed, solar irradiation, etc.), possible equipment (e.g. generators, lines, etc.) unavailability, failures in electric power transactions, errors in operation (operator errors, dispatcher and relay malfunctions) and others. In particular, the inherent variability and uncertainty affecting the renewable energy sources can have a significant impact on power supply, and accurate and reliable predictions of the power output obtainable from these sources are needed on different time scales. Thus, predicting the output of renewable energy sources is critical for integrating them efficiently in the

power grid. On the other hand, accurate electricity demand forecasting is also critical for the system adequacy as it enables system operators and utility providers to plan resource allocation and take control actions (e.g. switching on/off demand response appliances, revising electricity tariffs, etc.) to ensure the balance between supply and demand of electricity. Therefore, solving prediction problems in the context of power system adequacy has been receiving considerable attention for several decades [5], [30]-[32].

In particular, a specific research has been directed towards the development of accurate and reliable wind power and load forecasts, and many different forecasting systems with different approaches have been proposed. An overview of the existing prediction methods considered along this research line is given in Section 2.2.

1.4 The Research Problem and Motivation

Prediction plays a key role in many decision-making processes and should take into the uncertainty in its outcome. Prediction uncertainty can arise due to measurement errors, lack of knowledge in input data, and model approximation errors (e.g. due to imperfections in the model formulation, to the estimation process, etc.) [33], [34].

In energy systems, uncertainty in the prediction of the key factors, due both to the stochasticity in the data and the approximation of the prediction models, can cause high costs to the market participants (generators, customers, etc.) when not properly accounted for. Particularly, in wind-integrated power systems, the impact of such a highly variable energy source onto system reliability is an important aspect that must be assessed when the wind power penetration (that is, the share of wind power in meeting the electric energy demand) is significant. Therefore, considering the high penetration of wind power sources in the new competitive power systems, the necessity of having access to reliable prediction methods of wind speed/power predictions has become more evident for the sustainability and efficient management of the energy market: combining accurate short-term wind and load forecasts enables operators to commit the balance of the generation fleet to economically and securely serve load on the next day.

The vast majority of the existing studies on wind speed/power and load predictions only provide point predictions, without considering the uncertainties in the network structure and input data [35]-[37].

Prediction Intervals (PIs) are a simple way to communicate a measure of the uncertainty in the predictions. Further details about PIs and their use in energy system applications are given in Section 2.3. Here, let it suffice to mention that two elements characterizing PIs are their coverage probability (PICP) and width (PIW), that the objective of their estimation is to maximize the former and minimize the latter.

In this work, we consider PIs estimation within the adequacy assessment of wind-integrated power systems. We consider short-term wind speed/power and load forecasting because they are closely related to power system operations (unit commitment, real power scheduling, economic dispatch, etc.) More specifically, we consider PIs estimation via Neural Networks (NNs) [36], [38]-[43] extending a method proposed in the literature called "Lower Upper Bound Estimation Method for Construction of NN-based PIs (LUBE)" [36] within a multi-objective Pareto optimality framework that consider both PICP and PIW.

In Chapter 2, a review of existing methods and techniques for short-term wind speed/power prediction and load prediction is given.

With reference to the issues presented above, the following objectives have guided the work performed during this Ph. D. research activity:

1. to develop a prediction method capable of considering the uncertainty in the model parameters affecting the prediction;
2. to represent the uncertainty in input data and propagate it through the prediction model onto its results;
3. to enhance the performance of a NN-based, non-parametric prediction method by an ensemble approach;
4. to test the proposed model on real case studies in the context of energy system applications (in particular adequacy assessment).

In Chapters 4-7, all the above objectives are discussed, together with their relevance to each case study described in the papers (see Part II).

1.5 The Structure of the Thesis

The thesis is composed of two parts: Part I, subdivided in eight Chapters, introduces the current issues and challenges pertinent to distributed power generation systems, describes the research objectives undertaken, illustrates the methods developed and applied in this Ph. D. work, discusses some of the results obtained in the case studies and provides general conclusions and some future work perspectives. Part II is a collection of seven selected papers, scientifically reporting on the outcomes of the research work performed during the thesis, to which the readers are referred for further details. Tables 2 and 3 summarize the thesis structure with respect to the topics considered in Part I and to the case studies considered during the Ph. D., respectively.

For what concerns Part I, the Introduction Chapter presents the details of the background and motivation of our research work, its objectives and the organization of the thesis manuscript. Chapter 2 is devoted to the description of the prediction problem, and the traditional and recent methods in the context of wind speed/power and energy demand prediction. Chapter 3 focuses on the description of NNs and on the specific method of literature considered NN-based PIs estimation [35], [36] (objectives 1-3). Chapter 4 describes the multi-objective optimization problem (MOP) of PIs estimation and gives the details of the training of a NN by a multi-objective genetic algorithm (MOGA), i.e. non-dominated sorting genetic algorithm (NSGA-II) [44], for the prediction intervals estimation. In Chapter 5, applications of the method to provide estimated PIs for the scale deposition rate in oil & gas production equipment and for the short-term (1-h-ahead) wind speed prediction are detailed. In Chapter 6, relevant methods for the treatment of uncertainty in the input data are described, and the estimation of interval-inputs-based PIs and its applications in power system adequacy are explained (objective 2). Chapter 7 provides the details on existing NN ensemble methods, it presents the methods for ensemble-based NN PIs estimation in proposed this Ph. D. work (objective 3), and two applications of it.

Part II includes the papers collection. In Paper I [38], we have implemented the NSGA-II to train a multi-layer perceptron neural network (MLP NN) to provide the PIs of the scale deposition rate in oil & gas equipment, within multi-objective framework aimed at concurrently minimizing the PIW and maximizing the PICP of the estimated PIs. We have performed *k*-fold cross-validation (CV) to guide the choice of the NN structure (i.e. the number of hidden neurons) with good generalization performance. We have used a

hypervolume indicator metric to compare the Pareto fronts obtained in each CV fold. The experiments have been done with single-valued inputs.

Papers II [39] presents a comparison of the NN-based multi-objective approach for PI estimation to single-objective (SO) genetic algorithm (SOGA) and simulated annealing (SOSA) methods, and to a baseline Autoregressive Integrated Moving Average (ARIMA) method, for short-term wind speed prediction (1-h ahead). SOSA has been proposed in support of the original LUBE method in [36] with a case study concerning the analysis of four different wind speed datasets involving different wind speed profiles with seasonality.

Similar to Paper II, in Paper III [40] we have proposed and compared two machine-learning approaches, MOGA-NN and Extreme Learning Machines (ELM) combined with the nearest neighbors approach, for estimating PIs. The algorithms have been applied on a case study of short-term wind speed prediction using a real dataset of hourly wind speed measurements.

Paper IV [41] proposes an approach, based on interval analysis and generalizes the multi-objective NN-based PI estimation framework to interval-valued time series prediction. In this paper, we aim at quantifying the uncertainty in the prediction by combining uncertainties arising from both the input data and the prediction model. Demonstration of the proposed method is given on two case studies: (*i*) a synthetic case study, with 5-minutes simulated data; (*ii*) a real case study, involving hourly wind speed measurements. In both cases, short-term prediction (1-hour and day-ahead, respectively) is performed taking into account both the uncertainty in the model structure, and the variability (within-hour and within-day, respectively) in the inputs.

Paper V [42] presents a modeling and simulation framework for conducting the adequacy assessment of a wind-integrated power system accounting for the associated uncertainties. Our adequacy assessment framework leads to the evaluation of the EENS based on interval-valued load and wind power input data. The originality of the work lies in proposing not only a point-valued reliability index, but also interval-valued EENS results to inform the decision makers (DMs) on the uncertainty in the predictions.

In Paper VI [43], a novel approach for wind power forecasting with uncertainty quantification is described. The approach can be schematized in two steps: first, short-term estimation of wind speed PIs is performed within a multi-objective optimization framework worked out by NSGA-II. Then, the uncertainty in wind speed and the uncertainty in the

power curve are combined via a bootstrap sampling technique, thus obtaining wind power PIs with the same coverage as the wind speed PIs.

Paper VII presents a NN ensemble framework for wind speed PIs estimation and aims at proposing an enhanced version of the non-parametric MOGA-NN method proposed in this thesis work (see Chapters 4 and 5, and Papers I-III). We propose two NN ensemble methods, differing in the partitioning or not of the training dataset, and embedding the *k*-nearest neighbors (*k*-nn) approach in the aggregation phase for the identification of the neighborhoods of a test pattern. On the real data considered as case study, both methods have obtained superior results compared to those yielded from the selected individual networks selected in the respective ensembles.

Figure 2 illustrates a pictorial view of the flow of the Ph. D. thesis: the research motivation, the focus and the methodological approaches considered in the present work on the prediction problem in the context of the adequacy assessment of distributed power generation systems.

Table 2. Structure of the work with respect to the methodological topics considered.

	PART I	**PART II**
Topic	**Section**	**Paper**
The Prediction Problem and Its Role in the Adequacy Assessment of Distributed Power Generation Systems	1, 2	I-VII
NNs for Prediction	3	I-VII
The Multi-objective Optimization Problem of Training NN for Prediction Intervals Estimation	4	I-VII
Uncertainty Treatment: Interval-based Estimation of Prediction Intervals	6	IV, V
Uncertainty Treatment: NN Ensembles	7	VI, VII

Table 3. Structure of the work with respect to the case studies considered.

	PART I	**PART II**
Case study	**Section**	**Paper**
PIs Estimation for the Scale Deposition Rate in Oil & Gas Equipment	5	I
PIs Estimation for Short-Term Wind Speed Prediction	5-7	II-VII
Estimation of a Point-valued and Interval-valued EENS Index for a Wind-Integrated Power System	6	V
Estimation of PIs for Short-Term Wind Power Prediction with Interval Wind Speed Inputs and with a Stochastic Wind Power Curve	7	VI

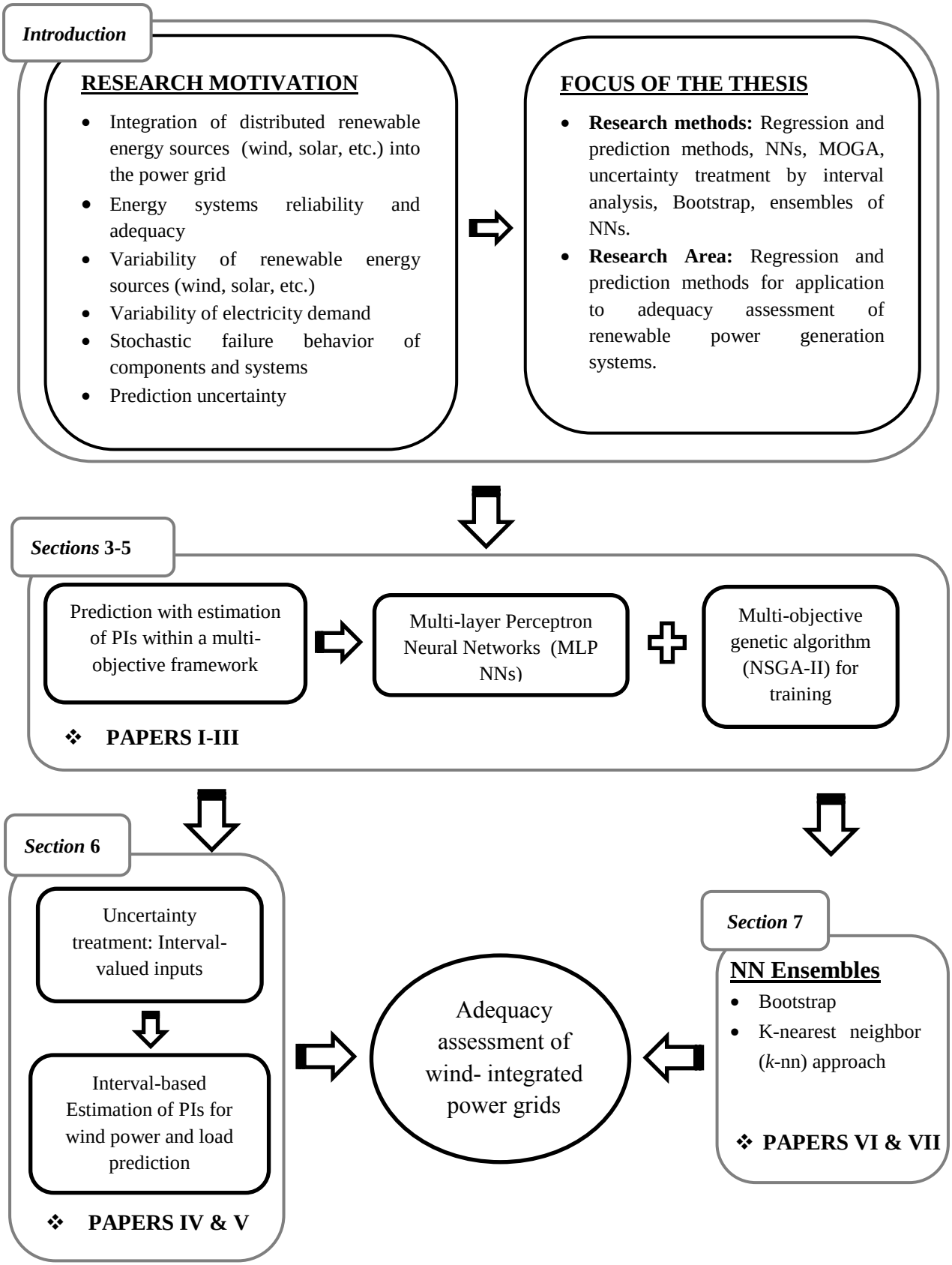


Figure 2. Pictorial view of the flow (motivation, focus and methods) of the thesis work on the prediction problem in the context of adequacy assessment of wind-integrated distributed generation systems.

2. THE PREDICTION PROBLEM

2.1 Problem Statement

The goal of prediction is to predict an output variable (e.g. load, wind speed/power, etc.) from some known inputs. In other words, the aim is to predict a quantity of interest, the response variable, Y given a set of explanatory (input) variables X and a sample of observations, i.e., input-target (output) pairs, $D = \{(x_i, y_i),\ i = 1,2, \dots, n_p\}$, where x_i is a scalar or vector of observations and n_p indicates the number of samples in the input dataset.

The existing prediction methods may be broadly classified into qualitative and quantitative techniques. The quantitative techniques, which are based on mathematical or statistical models, include regression methods (e.g. linear and nonlinear regressions), time series forecasting methods, machine learning algorithms (e.g. NNs, support vector machines (SVMs), model trees, etc.).

A prediction (forecasting) model can be linear or nonlinear depending on the relationship between input-output variables being linear or not. We can assume that the target vector Y is related to the input vector X by an unknown deterministic function, i.e., [45]:

$$Y = f(X; w) + \varepsilon(X), \quad \varepsilon(X) \sim N\left(0, \sigma_\varepsilon^2(X)\right) \tag{1}$$

where w represents the vector of values of the parameters of the model function f. The term $\varepsilon(X)$ is the error associated with f, assumed normally distributed with zero mean and with the variance $\sigma_\varepsilon^2(X)$. For the simplicity of illustration, in the following we assume Y one-dimensional. An estimate $\widehat{w}$ of w can be obtained by minimizing the quadratic error (cost) function [45] on the set of input/output values D.

$$E(w) = \sum_{i=1}^{n_p} (\hat{y}_i - y_i)^2 \tag{2}$$

where $\hat{y}_i = f(x_i; \widehat{w})$ represents the output estimated by the underlying model in correspondence to the input x_i. Note that a discrepancy between the model output $\hat{y}_i$ and the target y_i will be always present and it is due to various reasons including the presence of noise in the data, the limited number of data samples, the imperfect knowledge of the non-linear relationship between the dependent and independent variables, and the errors in

estimating the model parameters [45]. This discrepancy between the model output $\hat{y}_i$ and the real target y_i is called prediction error.

The goal of all prediction methods is to reduce the prediction error, i.e. to find the underlying unknown function $f(x_i; \widehat{w})$, which describes the relationship between inputs and outputs with the smallest prediction errors. It is of prime interest for any type of prediction problem to avoid model over-fitting, i.e. to secure that the model will be able to perform well on unseen data, which have not been used in the process of constructing (training) the model. That is, the method should be able to generalize the new samples from the same data domain, hence, the generalization power of a model is a critical issue that should be taken into account.

2.2 Methods

2.2.1 Regressions

As mentioned in the previous Section, there are many different statistical techniques proposed in the literature both for linear and nonlinear regressions. Linear regression is a model aim at determining a line that best fits the set of data points D.

Given a vector of inputs $X^T = (X_1, X_2, \cdots, X_{n_f})$ in n_f-dimensional input space, the single output Y is thus predicted as follows [46]:

$$\hat{Y} = \widehat{W}_0 + \sum_{j=1}^{n_f} X_j \widehat{W}_j \tag{3}$$

where the term $\widehat{W}_0$ is the intercept, also known as the bias in machine learning. We can write (3) in vector form as an inner product by including the constant variable 1 in X, and $\widehat{W}_0$ in the vector of coefficients $\widehat{W} = (\widehat{W}_0, \widehat{W}_1, \dots, \widehat{W}_{n_f})^T$ as follows:

$$\hat{Y} = X^T \widehat{W} \tag{4}$$

In order to estimate the unknown coefficients over the n_f-dimensional input space, we use the least square method which is the most popular estimation approach in linear regression. In this approach, we select the coefficients W to minimize the residual sum of squares (RSS), a quadratic function of the parameters [46]:

$$RSS\ (W) = \sum_{i=1}^{n_p} (x_i^T W - y_i)^2, \qquad \widehat{W} = \operatorname{argmin}_{W \in \mathbb{R}^{n_f+1}} RSS(W) \tag{5}$$

2.2.2 Time-series forecasting

In addition, there exist several methods dedicated to time series forecasting. It is the use of a model to predict future values of a variable based on a previously observed series of values of the same variable [47], [48]. Analysis and forecasting of time series is of fundamental importance in many practical domains. Examples can be found in very different fields of application: the sales of a particular product in successive months, wind power generation and electricity consumption in a particular location for successive 1-hour periods, hourly observations made on the yield of a chemical process, etc. [49].

A time series is a sequential set of data points, measured typically over successive times. It is mathematically defined as a set of vectors $x(t)$, t = 0, 1, 2,... where t represents the time elapsed [48]. $x(t)$ is a random vector and the measurements in a time series are arranged in a proper chronological order. The historical observations are carefully studied to build up a proper model that is then used to forecast unseen future values.

Over the years, various stationary and non-stationary models have been developed and used in the literature for time series forecasting [47], [48], [50]. Traditional statistical models including exponential smoothing (ES), Moving Average (MA), Autoregressive Moving Average (ARMA) and ARIMA are defined as linear regression methods where the future values are constrained to be linear function of past observations [51]. ARMA models are successfully used to represent the behavior of stationary time series. However, for non-stationary time series, differencing is necessary to resort to stationarity. To this aim, an ARIMA (p, d, q) model can be used, where parameters p, d, q are non-negative integers that refer to the order of the autoregressive, integrated and moving average parts of the model, respectively. More precisely, p is the order of the autoregressive process (highest number of significant lags); d is the order of differencing that is required to make the series stationary and q is the order of the moving average process [50], [52]. If the series is stationary, then d is equal to 0 and the ARIMA (p, 0, q) is equivalent to an ARMA (p, q) model. The interested readers can found a more extensive review of time series methods in [47], [48].

2.2.3 Machine learning methods

Data-driven machine learning methods such as NNs [53], [54], SVM [55] and Extreme Learning Machines (ELM) [56], [57], have been successfully used in various prediction

problems including time series forecasting recently. The support vector machine method has been developed by Vapnik [55] and has gained popularity due to its many attractive analytic and computational features, and to the promising performances. SVM has its motivation in the geometric interpretation of maximizing the margin of discrimination, and it is characterized by the use of a kernel function. NNs have attracted increasing attention in the domain of forecasting, pattern recognition, clustering, diagnosis, etc., as they offer a very powerful and very general framework for representing non-linear mappings from multiple input variables to multiple output variables, where the form of the mapping is governed by a number of adjustable parameters [58]. NNs do not require any assumption about the statistical distribution followed by the observations. The appropriate model is adaptively formed based on the given data.

2.2.4 Methods for wind speed/power and load forecasting

Much research has been carried out on the modeling and forecasting of wind speed/power based on different time scales and horizons, e.g. very short-term (seconds to minutes), short-term (hours up to two days), medium-term (days up to one week) and long-term (weeks to months or more ahead) [31]. General overviews of existing methodologies can be found in [31], [59]-[61]. The methods used in the literature can be classified as [31], [59]: *i*) physical approaches, e.g. numerical weather prediction (NWP); *ii*) statistical approaches, e.g. time series models such as ES, ARIMA; *iii*) artificial intelligence methods (heuristics), e.g. NNs, fuzzy logic systems, expert systems; *iv*) hybrid approaches, which combine physical and statistical methods, in particular using weather forecasts and time series analysis.

For what concerns the problem of load forecasting, it has attracted the attention of researchers since 1990's. Likewise, in wind speed/power forecasting problems, the various approaches involve different time scales and horizons (short, medium and long-term). It is critical to develop accurate short-term load forecasting (STLF) methods, since forecasted load values are used by market operators to determine day-ahead market prices, and by market participants to prepare bids. In addition, the accurate estimated loads are necessary information for the electric power price forecast on the electric power markets. There exist some works which make use of meteorological variables (e.g. temperature, humidity, cloud coverage, etc.) to forecast load, whereas some others treat the load pattern as a time series signal and predict the future load by using various time series analysis techniques [62]. In

addition, recent models consider also socio-demographic and economic characteristics of consumers (occupants), which substantially influence the energy consumption particularly in residential buildings [63], [64]. Likewise the case of wind speed/power forecast, artificial intelligence methods [65], time series models [66] and hybrid models combining both techniques have been also extensively used [67] for load forecasting.

A drawback of traditional data-driven machine learning methods is that they do not associate their predictions with confidence information, but they only output simple (point) predictions. These methods providing only point predictions cannot properly handle both the uncertainty in the model parameters and the noise in the input data. To quantify potential uncertainties associated with forecasts, in recent years, several researches have been conducted to estimate the PIs for the target of interest. Among them, NN-based PI construction approaches have become popular, and this area of research has been established and well accepted due to the superiority of these approaches on classical regression models for complex prediction problems [35], [36], [38], [39], [68]. In addition to NN-based PI models, there exist probabilistic approaches (both parametric and non-parametric) based on quantile regression, which can perform forecasting taking into account the associated uncertainty [37], [69], [70]. It is worth mentioning that probabilistic forecasting is more informative and useful than point forecasting.

In this thesis, a NN-based regression model for the construction of PIs is considered. Thorough details about NN regression models are not reported here for the sake of brevity: the interested readers may refer to the cited references, to the copious literature in the field and to Chapter 3 of this thesis (which is dedicated to the methodology of the multi-perceptron neural networks). Techniques for estimating PIs for NN model outputs are mentioned in Section 3.4 and Paper II of Part II. Moreover, the details of the comparison with some existing algorithms and methods are also given in Part II.

2.3 PIs Definition

An interval forecast is comprised of the upper and lower limits between which a future unknown value of the target, $y(x)$, is expected to lie with a prescribed probability, called confidence level and in general indicated with $1-\alpha$. These limits are called prediction limits or bounds, while the interval is called the PI (see Figure 3).

PIs can be used to provide information on the confidence in the predictions accounting for both the uncertainty in the model parameters and the noise in the input data [35].

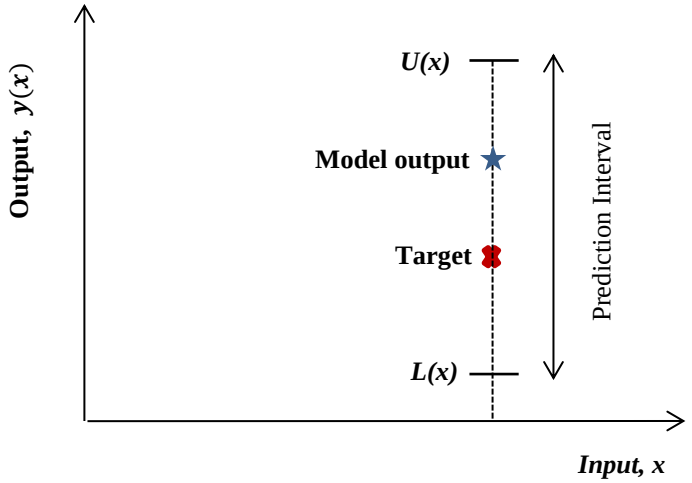


Figure 3. Exemplification of the terminology and concept of a prediction interval [45].

A PI should be distinguished from a Confidence Interval (CI). A PI corresponds to an interval estimation regarding the value of a new observation (unseen data); in other words, the PI deals with the accuracy of our estimate for a new observation. In contrast, the CI quantifies the accuracy of our estimate of the true regression, or in other words it is an interval estimate of the expected value of the output [71]. It should be noted that the PI is wider than the CI [72], and it encloses the corresponding CI because the CI takes into account only the uncertainty in the model, while the PI accounts also the variability in the data. For a mathematical formulation of the CI and the PI estimation problem in statistical inference, we refer the readers to [71].

In real world applications, PIs are of more practical use than CIs because a PI is concerned with the accuracy with which we can predict the observed target value itself, and not just the accuracy of our estimate of the true regression [45], [72]. Although the point predictions are relatively easy to compute and easy to understand, the main motivation for the construction of PIs is to quantify the associated uncertainty in the point forecasts. Availability of PIs allows the DMs 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. Several applications of PIs are found in a number of areas including engineering problems, health care, finance, business, etc.

3. ARTIFICIAL NEURAL NETWORKS (NNs) FOR PREDICTION

3.1 Basics of NNs modeling

Originally inspired by the function of the nerve cells in the brain, NNs have been widely used for decades to solve a variety of problems in pattern recognition, prediction, optimization, associative memory, and control [53]. Various applications of NNs have been studied in physics, biology, psychology, engineering, and mathematics. NNs are generally used as a non-linear regression model capable of learning complex systems with incomplete or corrupted data.

NNs are composed of computing units (called neurons or nodes) operating in parallel. These units are arranged in different layers and interconnected by weighted connections (called synapses). Here a layer refers to the usual term for a vertical row of neurons (see Figure 4). Each of these computing units performs a few simple operations and communicates the results to its neighboring units (see Figures 4 and 6). From a mathematical viewpoint, NNs consist of a set of nonlinear (e.g. sigmoidal) basis functions with free parameters *w* that are adjusted in a training process (on many different input/output data samples), by minimizing the error associated to regression in an iterative process.

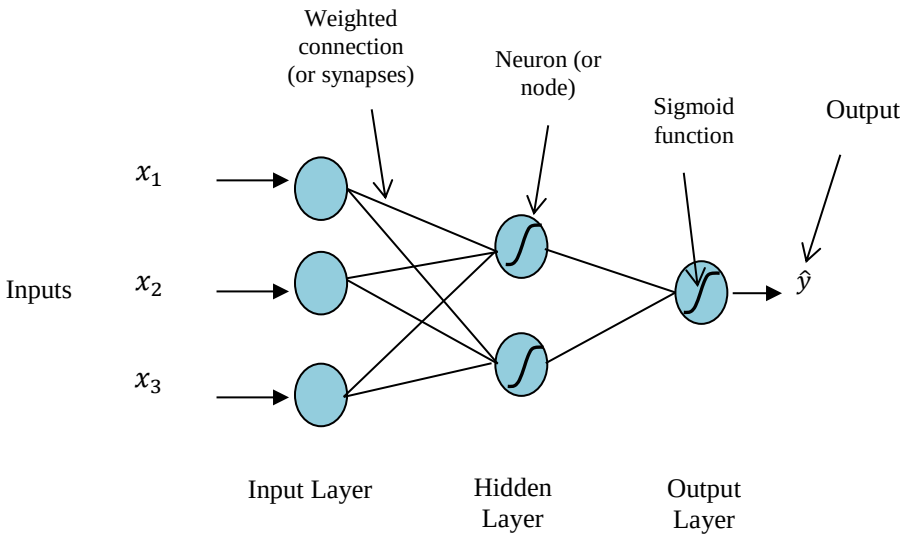


Figure 4. Sketch of a three-layered feed-forward NN architecture with $n_f = 3$ neurons (or nodes) in the input layer (i), $h = 2$ neurons in the hidden layer (h) and $n_o = 1$ neuron in the output layer (o).

As activation function, i.e. the function used to convert the net input value to the neuron's output value (signal) (see (7) and (8)), the sigmoid function is by far the most frequently used in NN. The standard sigmoid function is the logistic function, i.e. the logarithmic sigmoid function, and it ranges from 0 to 1. It is a convenient differentiable non-linear activation function $s_c(x)$: $\mathbb{R} \rightarrow (0,1)$ defined by

$$s_c(x) = \frac{1}{(1+exp(-x))} \tag{6}$$

In Figure 5, a taxonomy of NN types is illustrated [73]. Feed-forward neural networks (FNNs) and recurrent neural networks (RNNs) are the two main types. A RNN has neurons that transport a signal back through the network, whereas FNNs feed outputs from individual neurons forward to one or more neurons or layers in the network [54]. MLP NNs and Radial Basis Function Neural Networks (RBF NNs) are two of the most common types of FNNs used as empirical regression models especially for nonlinear regression. RBF networks use a radial basis function, i.e. a Gaussian kernel, as activation function. RBFs networks have similar universal approximation capabilities as MLP networks. For the theory and application of the RBF networks, we refer the readers to [74].

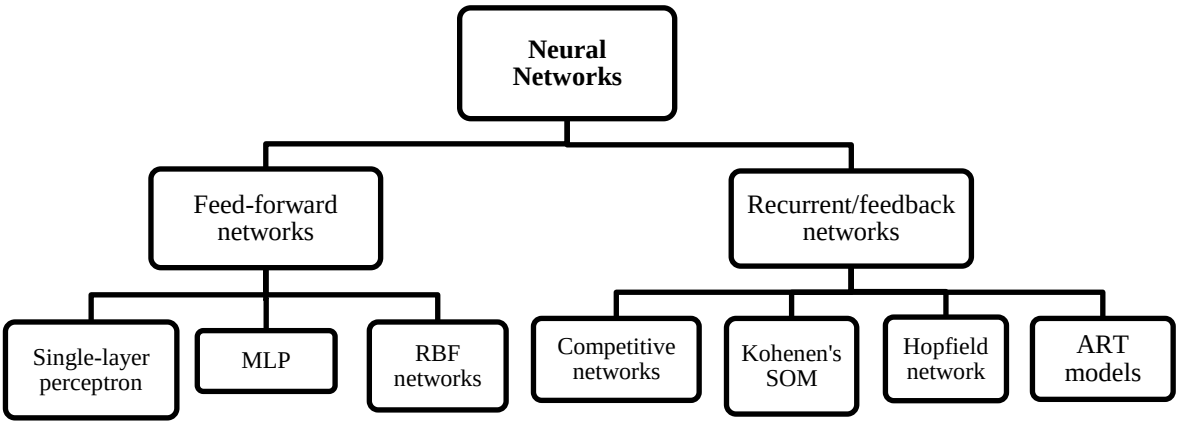


Figure 5. A taxonomy of neural network architectures [75].

MLP is a class of universal approximators [75], which are flexible statistical models used to model high dimensional and non-linear data. A MLP consists of multiple layers (an input and an output layer with one or more hidden layers) of nonlinearly-activating nodes in a directed graph, with each layer fully connected to the next one. Each node in one layer connects with a

certain weight w_{ij} to every node in the following layer (see Figures 4 and 6). A three-layer MLP has been found in practice to generalize well, i.e. when trained on a relatively sparse set of data points, it will often provide a good estimate of the output for an input not in the training set. In other words, these networks have been shown to approximate any continuous function to any desired accuracy [75].

An illustration of two multiple-input neurons, and the information processing through them to generate an output, is shown in Figure 6. Multiple signals $x_1, x_2, \dots, x_{n_f}$ are weighed and fed onto a non-linear sigmoid transfer (activation) function. The multi-layer structure of such neurons (nodes) defines the structure and functioning of the NN: input signals from a previous layer produce output signals that are distributed to the neurons of the subsequent layer.

Precisely, the input (net input) and output signals H_j of node j of the hidden layer are given, respectively, by

$$net_j = \sum_{k=0}^{n_f} w_{kj} x^k \tag{7}$$

$$H_j = f_h\left(\sum_{k=0}^{n_f} w_{kj} x^k\right) \qquad j = 1, \dots, h \tag{8}$$

where h is the number of hidden neurons, n_f is the number of input neurons (equal to the dimension of the input features) and n_p the total number of training samples, $f_h()$ is the activation function used in the hidden layer, $x^0 = 1$ is a bias factor, and for $k = 1,2, \dots, n_f$, x^k is the *k-th* input vector, $x^k = (x_1^k, x_2^k, \dots, x_{n_p}^k)$, and w_{kj} is the synaptic weight. After each hidden neuron output has been computed, the signal is sent to each of the neurons o_l in the output layer. Each output neuron o_l computes its output signal O_l to form the response of the network [16], [19]:

$$O_l = f_o\left(\sum_{j=0}^{h} w_{jl} H_j\right) \quad l = 1,2, \dots, n_o\ ,\ H_0 = 1 \tag{9}$$

where $H_0 = 1$ is a bias factor in the hidden layer, $\sum_{j=0}^{h} w_{jl} H_j$ is the net input (see (7)) to the output neuron o_l, n_o is the number of output neurons and $f_o()$ indicates the activation function used in the output layer.

In this thesis work, we consider the most widely used architecture for prediction and classification: a MLP NN with a single hidden layer. In particular, we target a three-layer

(input, hidden and output) feed-forward network, with the hyperbolic tangent function in the hidden layer and the logarithmic sigmoid function in the output layer.

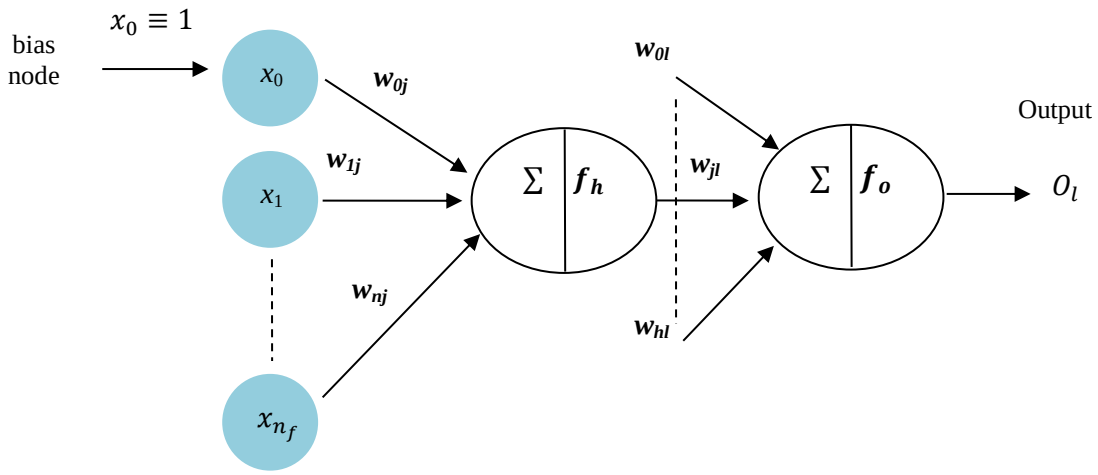


Figure 6. Scheme of artificial neurons with synaptic weights and corresponding transfer functions [56].

3.2 Training of NNs

There are three major learning paradigms which apply in general to statistical data-based methodologies, each corresponding to a particular learning task: supervised, reinforced and unsupervised learning. In the context of this thesis work, we focus on the supervised learning paradigm, i.e. learning from a set of observed cases (a training set of data). Supervised learning is analogous to a situation in which a student is guided by a teacher: the presence of the output (target) variable guides the learning process. Thus, the important principle is that supervised learning requires a training set of data in which the inputs and the corresponding outputs are both observed [46]. In supervised learning theory, the experiments can be roughly divided into two steps: training and testing. The training step has the aim of properly defining and estimating the model components: the training process thus continues until the model achieves a desired level of accuracy on the training data. In the testing (predicting) step, the derived model is applied on the testing set to predict the target values with regard to new unseen samples/patterns. Thus, this step has the scope of assessing the model correct functioning, predictive and generalization power. It is worth mentioning that the input dataset can also be split into three parts as training, validation, and testing, and, then, validation set can be used to avoid over-fitting (see Section 3.3).

In the context of machine learning strategies such as NNs, the process of learning through the training set, statistically regarded as an estimation process, is best viewed as an optimization process. During the training, the weights (parameters) of the network are gradually adjusted to reduce the prediction error between the network output and the corresponding target pattern [76]. The values of the weight vector *w* characterizing the network is, thus, optimized during the training. More precisely, the training procedure aims at minimizing the quadratic error function (see (10)) on a training set of input/output values $D = \{(x_i, y_i),\ i = 1,2, \dots, n_p\}$ by adjusting the values of the connections (weights) w between elements [77]. Therefore, the training of the neural network can be formulated as a non-linear unconstrained optimization problem as follows:

$min_w\, Error(D, w)$

where (10)

$Error(D, w) = E(D, w) = \frac{1}{2}\sum_{i=1}^{n_p}(O(x_i) - y_i)^2$

where x and y are the input and target vectors respectively, $O(x_i)$ is the output value estimated by the network for the *i*-th input sample x_i.

The back-propagation, first introduced by Rumelhart et al. in 1985 [77], is a widely used method for performing supervised learning tasks, i.e. the training of FNNs. By means of this procedure, the network can learn to map a set of inputs to a set of outputs, by minimizing the criterion in (10). Indeed, the back-propagation algorithm looks for the minimum of the error function in weight space using the gradient descent method [78]. The combination of weights which minimizes the error function is considered to be a solution of the learning problem. Note that, since gradient descent tries to minimize the sum-squared error between the network output values and the given target values by computing the gradient of the error function (see (10)) at each iteration step, the error function in use must be continuous and differentiable [79]. The algorithm can be decomposed in the following four steps [78]:

i. Feed-forward computation
ii. Back-propagation to the output layer
iii. Back-propagation to the hidden layer
iv. Weight updates

The algorithm is stopped when the value of the error function has become sufficiently small. The best known drawback of the classical back-propagation algorithm is that it is prone to local minima under some circumstances. The implementation details of the back-propagation algorithm and its mathematical formulation can be found in [53], [77], [78].

In addition to gradient-based training algorithms, meta-heuristics approaches such as evolutionary algorithms (e.g. GAs, differential evaluation (DE), etc.) [80], [81], SA [36], and tabu search (TS) [82] have recently been proposed to solve this network training optimization problem. In this Ph. D. work, we train a MLP NN by a MOGA, i.e. NSGA-II. The training procedure is detailed in Section 4.3.

3.3 Over-fitting and Cross-validation

Assessing the generalization power of a prediction method is essential for reliable prediction. In this regard, NNs show superiority to other methods: after learning from the training dataset, NNs can often correctly infer the unseen data even if the sample data contain noisy information [83]. Nevertheless, a NN can face over-fitting (overtraining) under some circumstances. Over-fitting occurs when the network memorizes the training patterns, but it does not learn. In that case, even though the prediction error on the training set is small, it is high on the testing set. The number of input features (i.e. input neurons), the number of hidden neurons and the number of training samples are all important factors which can cause over-fitting [84].

To prevent over-fitting, a validation set (a fixed set of samples not included in the training set) can be used to do "early stopping". This means using the validation set to detect when over-fitting starts during the training of a neural network; if this occurs, then the training is stopped (early stopping) before convergence to avoid over-fitting. Here, the validation error is used as an estimate of the model generalization error. To this aim, the basic early stopping technique proceeds as follows [85]:

i. Split the input data into three parts: training, validation and testing sets;
ii. Perform training on the training set, and periodically test the trained NN on validation set and compute the validation error rate during training;
iii. Stop training as soon as the validation error starts to go up;
iv. Use the weights the network had in that previous step as the result of training;

v. Perform testing on the testing set using the optimal weights yielded from training.

The number of samples used for training purposes is an important factor needed to guarantee a valid generalization capability. Too few training samples can cause over-fitting, wherein the network performs well on the training data set, but poorly on independent test samples drawn from the same distribution as the training patterns. When the training samples are few, CV is an alternative to be used to avoid over-fitting.

CV provides a simple and effective method for both performance evaluation and model selection, widely applied by the machine learning community. In the context of NN, it is used to evaluate the generalization performance of the NN, i.e., to estimate the prediction error [86]. It is also used for model selection [87] and for determining the optimal network architecture (i.e., the number of hidden neurons) [88].

For NN, the structure of the model influences the learning capability. In practice, the choices of the number of network layers and the number of neurons per layer often come down to a compromise between the generalization error and the learning time [89], [90]. Note that a suitable choice for the global architecture of the network is not a trivial task, if one wants to make a good prediction.

CV is a statistical resampling method which uses multiple training and test subsamples. Different CV techniques such as *k*-fold CV, leave-one-out CV, bootstrap CV, etc., have been proposed in the statistical literature [86]. In the basic *k*-fold CV technique, the input data set is split into a partition of *k* equally (or nearly equally) sized segments or folds. At each round of cross-validation, one among the different folds is excluded from the training set, and only the remaining *k*-1 folds are used for training; the excluded subset is then used for validation. The procedure is repeated until all the *k* folds have been used once for validation and *k*-1 times for training. The prediction error obtained in the validation step is then averaged across all samples. Note that it is sensitive to the specific way in which the dataset has been split [91]. For small *k* values, the bias of *k*-fold CV may become a problem in real-data analysis. For the so called leave-one-out CV, obtained when $k = N$, where *N* is the number of samples, the CV estimator is approximately unbiased for the true prediction error, but it has high variance and it is very computationally intensive for use in NN [46].

In this research line, in Paper I of Part II, a systematic process has been followed in order to identify the optimal NN structure (i.e. the number of hidden neurons) via CV. In this study,

we have used 20-fold CV in order to minimize the bias-variance trade-off while also attaining the required accuracy in feasible computation times [86].

3.4 PIs Estimation by NNs

As we have already mentioned in Section 2.3, the uncertainty in the output of the regression models is caused both by the uncertainty on the model structure, and by the inherent uncertainty in the input datasets, which is quite high for wind and energy demand predictions. In order to quantify and represent the uncertainty of predictions, the commonly applied statistical tools are CIs and PIs.

The usefulness of PIs of various types has been discussed lately, and different methods have been proposed by a number of researchers for the determination of PIs [35]-[37], [72], [92]. Herein, we address the NN-based PIs, i.e. we give a synopsis of the related works using NNs to generate PIs for the target of interest.

The primary techniques for estimating PIs for NN model outputs are the Delta, the Bayesian and the Bootstrap methods [35], [93]. The Bayesian and Delta methods are based on strong mathematical foundations. A comparison of these three methods was given by Khosravi et al. in [35] on different case studies, and by (the author) in [94] where the problem of bus and freeway travel time prediction is considered.

The Delta method is based on a Taylor expansion of the non-linear regression function [95]. This method is capable of generating high quality PIs but at the cost of high computational time in the development stage, because it requires both the calculation of a Jacobian matrix and the unbiased estimation of noise variance.

The Bayesian approach uses Bayesian statistics to express the uncertainty of the neural network parameters in terms of probability distributions, and integrates them to obtain the posterior probability distribution of the target conditional on the observed training set [58], [96]. The axiomatic, strong mathematical foundation makes this method robust and highly repeatable. In the end, NNs trained by a Bayesian-based learning technique have superior generalization power [35]. On the other hand, the computation time required is high, due to the calculation of a Hessian matrix in the development stage (a situation similar to the Delta technique).

The Bootstrap method is frequently used because it is the simplest method among the ones mentioned here [35]. It is a resampling technique that allows assigning measures of accuracy to statistical estimates without requiring the calculation of complex matrices and derivatives [97], [98]. The aim of the resampling is to produce less biased estimates of the true regression of the targets, and to improve the generalization performance of the model [35]. The main disadvantages of this method are: *i*) high computational time when the training sets and neural networks are large; *ii*) with small numbers of input patterns, the individual neural networks tend to be overly trained, leading to poor generalization performance [35], [99].

The common feature of the above mentioned PI estimation methods is that they do not take into account the widths of the intervals in the estimation process [35]. With respect to this point, Khosravi et al. [36] proposed the LUBE, in which they obtain NN-based PIs by considering both CP and PIW in the PI construction phase. These two quantitative measures determine the quality of the estimated PIs. The PICP represents the probability that the set of estimated PIs will contain the true output values, estimated as the proportion of true output values lying within the estimated PIs; PIW simply measures the extension of the interval as the difference of the estimated upper bound and lower bound values. These are in general conflicting measures (wider intervals give larger coverage), and in practice it is important to have narrow PIs with high coverage probability. The mathematical definition of the PICP and PIW measures used are [36]:

$$PICP = \frac{1}{n_p}\sum_{i=1}^{n_p} c_i \tag{11}$$

where $c_i = 1$, if $y_i \in [L(x_i), U(x_i)]$ and otherwise $c_i = 0$,

$$NMPIW = \frac{1}{n_p}\sum_{i=1}^{n_p} \frac{(U(x_i) - L(x_i))}{y_{max} - y_{min}} \tag{12}$$

where $NMPIW$ is the Normalized Mean PIW , and y_{min} and y_{max} represent the true minimum and maximum values of the targets (i.e., the bounds of the range in which the true values fall) in the training set, respectively. Normalization of the PI width by the range of targets makes it possible to objectively compare the PIs, regardless of the techniques used for their estimation or the magnitudes of the true targets.

In the context of the Ph. D. work, our proposed NN-based PIs method does not rely on assumptions of the data being drawn from a given probability distribution. The only

assumption that we make is that the data are independently and identically distributed (i.i.d.), since we perform an empirical and non-parametric approach to the estimation of PIs. In this Ph. D. work, we extend the LUBE method [36] to the multi-objective formulation of the PI estimation problem. More specifically, we use NSGA-II [44] to train MLP NN to concurrently minimize the IW and maximize the CP of the estimated PIs in Pareto optimality sense [100]. Note that the approach proposed in this research work integrates the estimation of the PIs in its learning procedure, while other methods except LUBE construct PIs in two steps (first doing point prediction and then constructing PIs). In Figure 7, the structure of a typical three layer (input, hidden and output) MLP used in this thesis to construct PIs is illustrated: the output neuron above provides the upper bound of the PI and the one below provides the lower bound. By these two output neurons, the NN generates a PI interval for each input pattern.

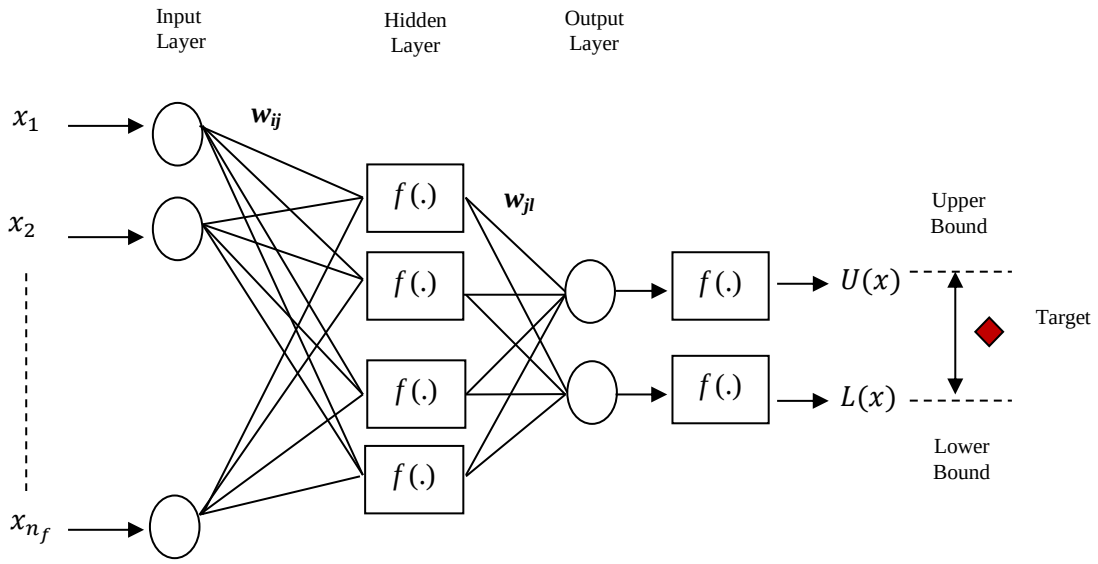


Figure 7. Architecture of a MLP NN model for estimating the lower and upper bounds of PIs.

4. THE MULTI-OBJECTIVE OPTIMIZATION PROBLEM OF TRAINING A NN FOR PREDICTION INTERVALS ESTIMATION

Optimization is the task of finding one or more solutions that minimize (or maximize) one or more specified objectives subjected to all constraints (if any). This Chapter is devoted to single-objective optimization problem, via GAs and SA, and to multi-objective optimization problem (MOP) via MOGA. In particular, we describe NSGA-II, one of the most powerful among evolutionary algorithms (EAs), and the use of NSGA-II to train our NN-based PI MOP with two objectives.

4.1 Single-objective Optimization: Genetic Algorithms and Simulated Annealing

A single-objective optimization problem (SOP) involves a single objective (cost) function and usually results in a single solution or in a set of solutions, called optimal. There exist several optimization methods (algorithms), which can be classified in exact solution techniques [101], [102] and meta-heuristic algorithms [82], [103], [104]. Because of the focus of this thesis work, we here consider on two meta-heuristic algorithms: GAs and SA, which have been used in our experiments for the purpose of comparison.

GA is a directed random search technique, based on the principles of natural selection and genetics, originally proposed by Holland [105]. It can give a result close to the global optimal solution in complex multi-dimensional search spaces in a tractable time. It is one of the most popular evolutionary algorithms (EAs) in diverse research and application fields. The algorithm is first initialized with a population of individual solutions known as chromosomes, each encoded either in a string of binary digits (bits) in the case of binary-coded GAs or in a vector of real-valued variables in the case of real-coded GAs [103]. Each chromosome is associated with one fitness value, evaluated in terms of the objective/fitness function, f, representing the degree of fitness of this chromosome. All fitness values of the population are then used for evaluating the probability of acceptance of individual chromosomes in the next generation, i.e. the chance whether the chromosomes are eliminated or retained in the next generation [106].

There are three major operators at each generation: selection, crossover and mutation. The selection operator is used to select the chromosomes for the next generation: the chromosomes with better fitness values have higher chances to be retained, and those with

poorer fitness values have higher chances to be weeded out. The crossover operator is used to generate the offspring by exchanging certain bits, i.e. information, of paired individuals (chromosomes) chosen from the population, with the expectation that good parents can generate better children [106]. Crossover occurs only with some probability p_c (namely the crossover probability, or crossover rate). When the chromosome pairs are not subject to crossover, they remain unmodified. Mutation is the modification of the value of each bit of a chromosome with some probability p_m (namely the mutation probability or mutation rate). The bits of a chromosome are modified independently, i.e. the mutation of one bit does not affect the mutation of the others. The crossover rate p_c, the mutation rate p_m and the population size Nc are user-specified parameters. Hence, determining the initial p_c and p_m plays an important role for the algorithm to explore the search space. Some studies focused particularly on finding the optimal crossover and/or mutation rates [106], [107]. Note that p_c and p_m might change values along the evolution. A scheme of the standard procedure of a basic GA for a maximization problem is given in Figure 8.

SA, inspired by the physical process of annealing of molten metals, is one of the most popular meta-heuristics for combinatorial optimization problems. It has been established in the 1980s to deal with highly nonlinear problems [108]. One of its advantages is the ability to avoid trapping in local minima, by allowing an occasional uphill move. At each iteration, the SA algorithm reaches some neighboring state $x'$of the current state x, and stochastically decides between moving to state x' or remaining at state x. This is done by sampling a uniform random number r in the range (0, 1) and using a control parameter called the *temperature*. Then, the decision about acceptance or rejection of the new state is made according to the Boltzman Probability Factor (BPF), calculated by

$$exp^{(-\delta/kT)} \tag{13}$$

where δ represents the difference between the objective functions of the newly generated state x' and the current state x, k is the Boltzmann constant which is set to 1 in our experiments, and T is temperature.

Suppose that S is the finite set of all solutions and the objective (cost) function, f, is a real valued function defined on members of S. The aim of SA is to find a state (or solution), $i \in S$, which minimizes f over S [109]. Figure 9 gives a scheme of the standard procedure of a general SA algorithm for a minimization problem [110].

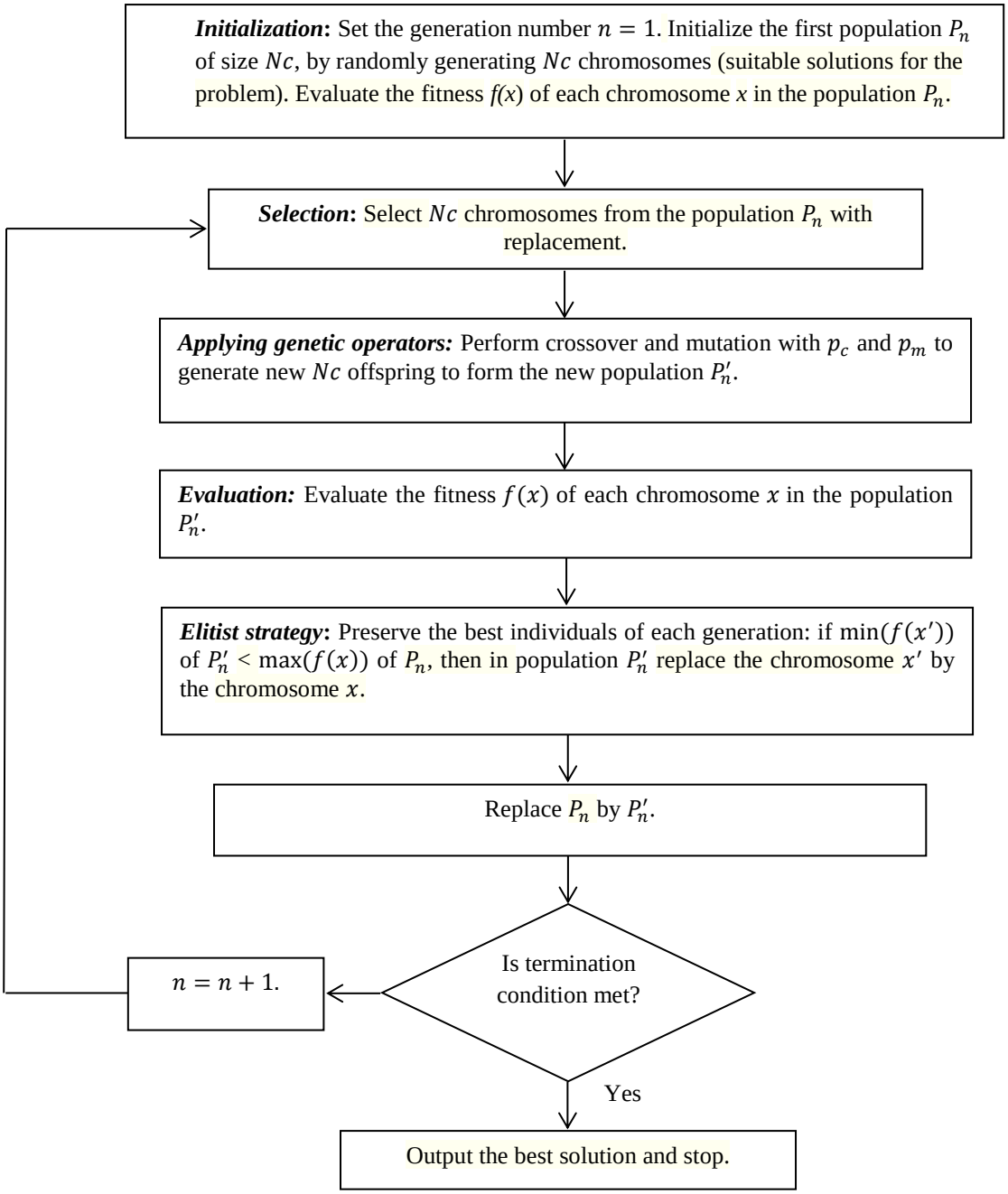


Figure 8. A scheme of the standard procedure of a basic GA for a maximization problem.

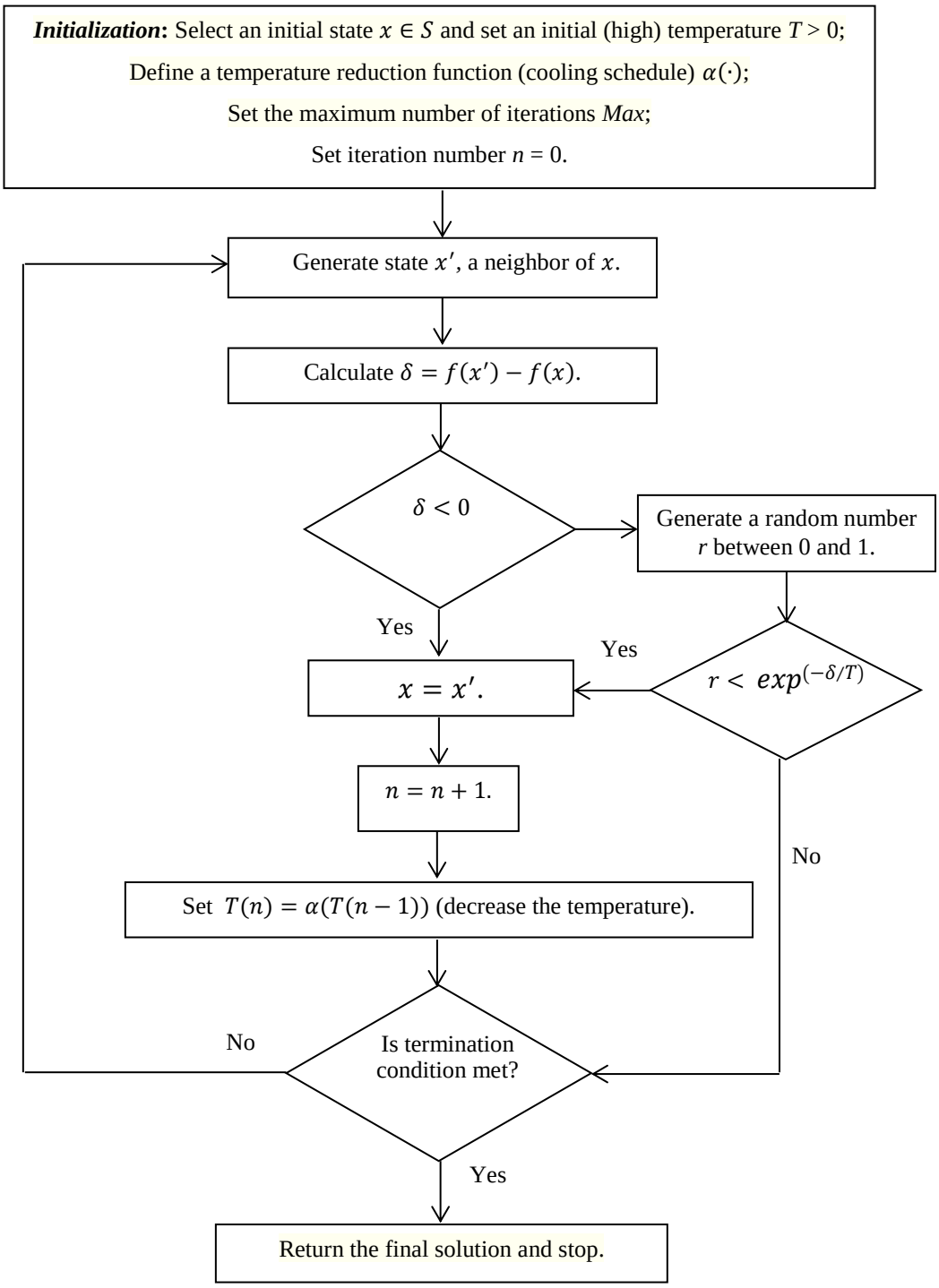


Figure 9. A scheme of the standard procedure of a basic SA for a minimization problem.

Different cooling schedules can be used for temperature reduction: constant thermodynamic speed, exponential, logarithmic, linear, etc. [111]. In our case studies, we have used exponential schedule defined as follows:

$\alpha\big(T(n)\big) = \sigma T(n)$ where σ is a constant factor $(0 < \sigma < 1)$. (14)

The commonly used termination criteria include the following: the maximum number of iterations is reached; no further improvement is achieved for a specific number of consecutive iterations; a very low temperature is reached; or a very small objective function value is found.

Both GAs and SA have been successfully applied to NN. Considerable amount of works show the use of GAs and SA to train a NN, i.e. to obtain optimal NN parameters (weights and biases) [36], [39].

Khosravi et al. [36] have used SOSA to train a NN for generating PIs. They used the weighting approach as shown in (21) to combine two separate objectives (see Section 3.4) PICP and NMPIW in a single-objective function (see (15)). The former objective, to be maximized, represents the probability that the set of estimated PIs will cover the true output values $y(x)$, while the latter, to be minimized, simply measures the extension of the intervals as the difference between the estimated upper bound and lower bound values. The objective function to be minimized is called coverage width-based criterion (CWC) [36], and it is given by

$$CWC = NMPIW\left(1 + \gamma(PICP)\, e^{-\eta(PICP-\mu)}\right) \quad (15)$$

where η and μ are constants. The role of η is to magnify any small difference between μ and PICP. The value of μ gives the nominal confidence level, which is set to 90% in our experiments. Then, η and μ are two parameters determining how much penalty is paid by the PIs with low coverage probability. The function $\gamma(PICP)$ is equal to 1 during training, whereas in the testing of the NN it is given by the following step function:

$$\gamma(PICP) = \begin{cases} 0, & PICP \geq \mu \\ 1, & PICP < \mu \end{cases} \quad (16)$$

In this work, we have performed a comparison among our proposed MOGA framework, SOGA and SOSA. To perform the training via SOSA and SOGA, we have used the CWC, described in (15), as the single-objective cost function. The details and the results of the comparison are given in Paper II of Part II. The details about the implementation of SOSA

and SOGA in our NN-based prediction problem, i.e. of the training (optimization) procedure of a MLP to construct PIs, is given in Chapter 5.

4.2 Multi-objective Optimization: Non-dominated Sorting Genetic Algorithm-II (NSGA-II)

In generality, a MOP consists of more than one objective, and it is associated to a number of equality and inequality constraints, and bounds on the decision variables. In case of multiple objectives, there may not exist one solution, which is the best (global minimum or maximum) with respect to all objectives. In other words, unlike the traditional mathematical setting with a single-objective function, an optimal solution in the sense that one minimizes (or maximizes) all the objective functions simultaneously does not necessarily exist in the MOP. Thus, we encounter a conflict among objectives, which translates into a decision making problem with multi objectives [100]. Ultimately, we try to find good compromises (or "trade-offs") rather than a single solution as in global optimization.

Mathematically, MOP can be written as follows [100]:

$$\textit{Minimise/Maximise } f_m(x), \quad m = 1,2,\dots,M; \tag{17}$$

$$\textit{subject to } \quad g_j(x) \geq 0, \quad j = 1,2,\dots,J; \tag{18}$$

$$h_k(x) = 0, \quad k = 1,2,\dots,K; \tag{19}$$

$$x_i^{(l)} \leq x_i \leq x_i^{(u)} \quad i = 1,2,\dots,I. \tag{20}$$

Given an I-dimensional decision variable vector solution, $x = (x_1, x_2, \dots x_I)^T$ in the solution space R^I, the final goal is to find a vector $x^* \in R^I$ that minimizes a given set of M objective functions $\{f_1(x^*), f_2(x^*), \dots, f_M(x^*)\}$. The solution space is restricted by the constraints in (18) and (19), and the bounds on the decision variables in (20). The solutions satisfying the constraints and variable bounds constitute a feasible decision variable space $\Phi \subset R^I$. In addition to the decision variable space, the objective functions constitute a multi-dimensional objective space, $Z \subset R^M$. The M objective functions $f_m(x)$ must be evaluated in correspondence to x in the search space. Thus, for each solution x in the decision variable space, there exists a point $z \in R^M$ in the objective space, defined by the relation $f(x) = z = (z_1, z_2, \cdots, z_M)^T$ [103].

There are two general approaches to solve the MOPs [112], [113]: one way is to combine the individual objective functions into a single composite function. Determination of a single objective is possible with methods such as utility theory, or with more naïve approaches like the weighted sum method, i.e. by averaging the objectives with a weight vector (see (21)). However, the weighted sum approach relies on the correct selection of the weights or utility functions to characterize the DMs preferences with respect to the different objectives. Small perturbations in the weights can lead to different solutions, and a priori selection of weights does not necessarily guarantee that the final solution will be acceptable. One may have to restate the problem with new weights, and then find a new solution, thus, this increases the computational cost of the whole procedure.

Mathematically the weighting method is described as [103]:

$$Minimize \ \sum_{m=1}^{M} w_m f_m(x) \qquad (21)$$
$$subject\ to\ x \ \in \Phi$$

where $w_m \geq 0 \ for\ all\ m = 1,2, \dots, M$, and $\sum_{m=1}^{M} w_m = 1$.

Although the main advantage of the weighted sum approach is its straightforward implementation, there exist some disadvantages: the selection of the appropriate weights is a challenge; it is impossible to obtain points on non-convex portions of the Pareto optimal set in the criterion space, i.e. this approach does not work correctly for non-convex problems, etc. For a more detailed description of this method, we refer the readers to [103], [112].

The second general approach, which is the focus of this work, is to optimize all the objectives together, thus to determine an entire Pareto optimal solution set or a representative subset. In a typical MOP, there exists a set of solutions $x^* \in R^I$ which are superior to the rest of solutions in the search space, but in which no solution can be regarded superior to any other with respect to all the objective functions. These solutions are known as Pareto-optimal solutions or non-dominated solutions. Thus, a Pareto-optimal set is a set of solutions that are non-dominated with respect to each other when all objectives are considered. In case of a minimization problem, solution x_a is regarded to dominate solution x_b $(x_a \succ x_b)$ if both following conditions are satisfied [100]:

$$\forall i \in \{1,2, \dots, M\}, f_i(x_a) \leq f_i(x_b) \qquad (22)$$

$$\exists j \in \{1,2,\dots,M\}, f_j(x_a) < f_j(x_b) \quad (23)$$

If any of the above two conditions is violated, the solution x_a does not dominate the solution x_b, and x_b is said to be non-dominated by x_a. The solutions that are non-dominated within the entire search space are denoted as Pareto-optimal and constitute the Pareto-optimal set; the corresponding values of the objective functions form the so-called Pareto-optimal front in the objective functions space (see Figure 10).

The majority of existing multi-objective evolutionary algorithms (MOEAs) are based on Pareto dominance [44], [104], [113], [114]. EAs seem particularly suitable to solve MOPs, because they deal simultaneously with a set of possible solutions (the so-called population). This allows us to find several members of the Pareto-optimal set in a single run of the algorithm, instead of having to perform a series of separate runs as in the case of the traditional mathematical programming techniques [104].

To obtain a Pareto-front, another alternative could be to use the ε-constraint method [115], [116]. To perform this method, one has to reformulate the problem as a single-objective one by choosing one objective for optimization, and considering the others as constraints bounded by some allowable levels ε_i. The constraint values, ε_i, are then changed to generate the Pareto-optimal set. This approach requires multiple runs to form the Pareto front and can be time-consuming. In addition, the search is limited to few points in some predefined regions near the fixed constraint values. This may lead to missing some optimal solutions. On the contrary, MOEAs can find, multiple Pareto-optimal solutions in one single run, and the non-dominated solutions in the obtained Pareto-optimal set are well distributed and diverse [112], [115], [116].

The goal of a multi-objective optimization algorithm is to guide the search for solutions in the Pareto-optimal set, while maintaining diversity so as to cover well the Pareto-optimal front and thus allow flexibility in the final decision on the solutions to be actually implemented. The Pareto-optimal set of solutions can provide the DMs the flexibility to select the appropriate solutions, trading-off different preferences on the objectives. The DMs also gain insights into the characteristics of the optimization problem before a final decision is made.
Note that since none of the solutions in the non-dominated set is absolutely better than any other, any one of them can be chosen as a final solution to take decisions. The choice of one solution over the other requires problem knowledge and a number of problem-related

considerations. Thus, one solution chosen by a DM may not be acceptable to another one or in a different context. Therefore, in multi-objective optimization problems, it may be useful to have knowledge about alternative Pareto-optimal solutions, as it provides valuable information about the underlying problem [113].

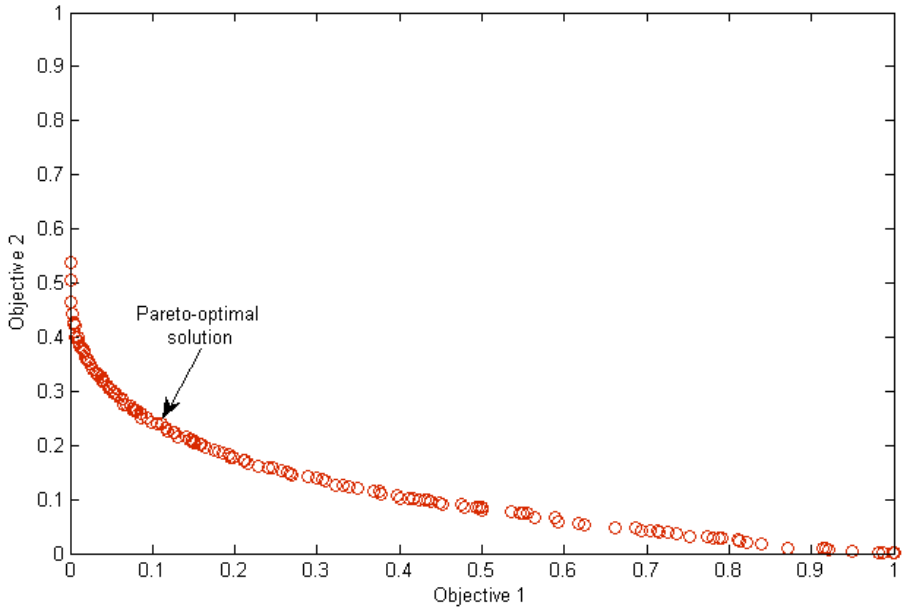


Figure 10. An example of an optimal Pareto front of non-dominated solutions, for illustration purposes.

4.3 Training of NNs by NSGA-II

The NSGA was first proposed by Srinivas and Deb (1994) [113]. Then, they have proposed the improved version, NSGA-II [44], which is so far one of the most popular and powerful MOEAs attempting to find multiple Pareto-optimal solutions in a MOP. This updated version, NSGA-II, has less computational complexity compared to the NSGA. The characteristic feature of NSGA-II is its fast non-dominated sorting, ranking and elitism techniques to find the optimal solutions at each iteration. The pseudo code of the NSGA-II can be found in [44], [104].

The practical implementation of NSGA-II on our PIs construction problem involves two phases: initialization and evolution. These can be summarized as follows:

Initialization phase:

Step 1) Partition the input data into training (D_{train}) and testing (D_{test}) subsets.

Step 2) Define the values of: the maximum number of generations, the number of chromosomes (individuals) Nc in each population, and the initial crossover and mutation probabilities.

Step 3) Set the generation number $n = 1$. Initialize the first population P_n of size Nc, by randomly generating Nc chromosomes. Each chromosome forms a candidate solution by G real-valued genes, where G is the total number of parameters (weights) in the NN. Note that each solution corresponds to a NN.

Step 4) For each input sample x in the training set, evaluate each of the Nc chromosomes in the initial population P_n, i.e. compute the lower and upper bound outputs of each Nc chromosome with G parameters, by performing NN training. Return the values of two objectives 1-PICP and NMPIW for each of the Nc chromosomes.

Step 5) Rank the chromosomes (vectors of G values) in the population P_n by running the fast non-dominated sorting algorithm [44] with respect to the pairs of objective values, and identify the ranked non-dominated fronts $F_1, F_2, \dots, F_k$ where F_1 is the best front, F_2 is the second best front and F_k is the least good front.

Step 6) Apply to P_n a binary tournament selection based on the crowding distance [44], for generating an intermediate population S_n of size Nc.

Step 7) Apply the crossover and mutation operators to S_n, to create the offspring population Q_n of size Nc. Note that mutation probability p_m decreases at each iteration.

Step 8) Apply *Step* 4 onto Q_n and obtain the lower and upper bound outputs. Evaluate each of the Nc chromosomes in the population Q_n. Return the values of the two objectives corresponding to the solutions in Q_n.

Evolution phase:

Step 9) If the maximum number of generations is reached, stop and return P_n. Select the first Pareto front F_1 as the optimal solution set. Otherwise, go to *Step* 10.

Step 10) Combine P_n and Q_n to obtain a union population $R_n = P_n \cup Q_n$.

Step 11) Apply *Steps* 4-5 onto R_n and obtain a sorted union population.

Step 12) Select the Nc best solutions from the sorted union to create the next parent population P_{n+1}.

Step 13) Apply *Steps* 6-8 onto P_{n+1} to obtain Q_{n+1}. Set $n = n + 1$; and go to *Step* 9.

Finally, the best front in terms of ranking of non-dominance and diversity of the individual solutions is chosen. Once the best front is chosen, the testing step is performed on the trained NN with optimal weight values.

The binary tournament selection, mentioned in Step 6, uses the crowded-comparison operator $\prec_n$ as the selection criterion. For solution i in the population, it has two attributes: nondomination rank i_{rank} and crowding distance $i_{distance}$. The crowding distance is a measure of how close an individual is to its neighbors. Large average crowding distance will result in better diversity in the population. For a solution pair, i and j, we have i $\prec_n j$ if $i_{rank} < j_{rank}$ or ($i_{rank} = j_{rank}$ and $i_{distance} > j_{distance}$). That is, if there are two solutions under consideration with different nondomination ranks, we prefer the one with the lower (better) rank. Otherwise, if both solutions have the same rank, i.e. if they both belong to the same non-dominated front, we select the solution which locates in a region with the smallest number of points. For further explanations, we refer the readers to [44].

Note that we have followed the original design of NSGA-II published by Deb et al [44]. In the initialization phase, the *binary tournament selection, recombination*, and *mutation* operators are used to build an offspring population Q_0 of size Nc to combine with its parent population in the later evolution phase. The union population is used in each cycle of the evolution phase. Thus, with respect to the original definition of the NSGA-II, we have performed binary tournament selection, mutation and crossover operators in the initialization phase.

It is worth noting that the *Step* 1 can be modified according to the characteristics of the problem and of the experiment. For exemplification, in Papers IV and VII, a validation process has also been performed. Hence, the dataset has been divided into three parts: training (D_{train}), validation (D_{vald}) and testing (D_{test}).

5. APPLICATIONS

This Chapter is devoted to the description of the applications carried out in Papers I-III of Part II. Paper I considers the problem of scale rate prediction in oil & gas components, whereas the focus of Papers II and III is short-term wind speed prediction. In all three papers, we use the MOGA-NN approach defined in Chapter 4 to estimate PIs for the target of interest.

5.1 Prediction of Scale Deposition Rate in Oil & Gas Equipment

In paper I, we have tackled the prediction of the scale rate in oil & gas components. Degradation to failure of components may cause unplanned costs and production losses through downtime [117]. Prediction of component degradation to failure is important for production availability. In the context of oil & gas industry, scale deposition and corrosion continue to be serious and costly problems, because affecting the operation of the components. Formation of scale on downhole equipment due to produced mineral salts is a common occurrence which is mitigated with chemical treatments or surface modifications [118]. However, when chemical treatments are ineffective or undesirable, the scale buildup should be accounted for and predicted a priori for a given application. Accurate prediction of scale deposition and corrosion can give way to preventive maintenance [119]. In this paper, we focus on the former process.

In oil & gas plant components, scale deposition is influenced by different factors, such as reservoir conditions (temperature, pressure), changes in pH, variation of flow rates, water cut, material structure, etc. [120]. A predictive model is needed to explain the linear or nonlinear mapping between these input (explanatory) variables and the output (the scale deposition rate, hereafter called scale rate). Unlike the classical analytical models based on thermodynamics kinetics and hydrodynamics, or on a combination of these for the prediction of scale deposition in components of production wells, we have here used a MLP NN as prediction method, since we aim at taking into account the variability/uncertainty caused in the output by the uncertain characteristics of the input variables. The originality of the approach is the multi-objective formulation of the problem, which is capable of achieving high coverage with intervals of small width. The multi-objective framework allows considering a set of optimal solutions to select from, according to preferences and to the application purposes.

The case study concerns the scale (deposition) rate on the metal surfaces of equipment used in offshore oil wells. The output variable y is the scale rate; the influencing input variables are temperature (T) and pressure (P), water composition (W) and fluid velocity (V) near the metal surfaces.

We here use a systematic approach, rather than trial-and-error method, for the selection of the optimal number of hidden neurons in the hidden layer. Since the number of input samples (118 samples) in the training set was not enough for selecting the optimal NN structure without facing over-fitting, we have performed *k*-fold, i.e. 20-fold CV. The architecture of the NN consists of one input, one hidden and one output layers. The number of input neurons is 4, the number of hidden neurons is chosen via the CV process, the number of output neurons is 2, one for the lower and one for the upper bound values of the PIs (see Figure 7). As activation functions, the hyperbolic tangent function is used in the hidden layer and the logarithmic sigmoid function is used at the output layer (these choices have been found to give the best results by trial and error, although the results do not show significant sensitivity to them). In order to obtain an optimal NN architecture, 11 different choices for the number of hidden neurons, 5, 7–11, 13, 15, 17, 18, 20, have been explored. As we have performed 20-fold CV, a NN has been trained 20 times with the same number of hidden neurons. Ultimately, we have obtained 220 optimal Pareto fronts including several non-dominated solutions. Note that each solution on a Pareto front obtained by training corresponds to a NN.

In order to evaluate different neural network structures and select the optimal one, the Pareto fronts have been compared in terms of their hypervolume indicators [121]. Note that, if a solution set A has a greater hypervolume than a solution set B, then A is taken to be a better set of solutions than B [122]. The number of hidden neurons providing a significantly larger hypervolume according to classical statistical tests has been evaluated as optimal, and selected for subsequent analyses.

In conclusion, with respect to the statistical test results, we have chosen the optimal number of hidden neurons as 10 for our specific problem. Finally, a solution on the Pareto front obtained by training with 10 hidden neurons has been chosen subjectively, and, then, estimated PIs on the test set, corresponding to this solution, have been plotted for the scale rate values. The solution has been chosen as the one with smallest NMPIW among those with $PICP \geq 90\%$.

5.2 Short-Term Wind Speed Prediction

In Papers II and III, we have tackled the problem of short-term wind forecasting on real case studies including four and three datasets, respectively. The main contributions in Paper II can be summarized as follows: *i*) framing the PI estimation problem, i.e. finding optimal lower and upper bounds of PIs, in a multi-objective framework, and utilizing the powerful NSGA-II algorithm to solve the problem. To the knowledge of the authors, this is the first study proposing such multi-objective formulation for the estimation of NN-based PIs for wind speed prediction; *ii*) analyzing the Pareto front of optimal solutions, and offering several alternatives to the DMs for finally choosing a solution by taking into account the trade-off between risk and robustness; *iii*) showing the application of the method on four different datasets involving different wind speed profiles with seasonality; and *iv*) performing a thorough comparison with both single and multi-objective algorithms.

The hourly wind speeds measured in four different periods in Regina, Saskatchewan have been downloaded from the website [123]. The first dataset comprises wind speeds for the period from 1st of February 2012 to 31st of March 2012 (winter dataset), the second from 1st of July 2012 to 29th of August 2012 (summer dataset), the third from 1st of February 2011 to 30th of June 2011 (w2011 dataset) and the last one from 1st of May 2010 to 30th of September 2010 (w2010 dataset). The four periods have different seasonality and have been selected to represent different patterns and characteristics in the measured time series of wind speeds.

In this study, the method for the estimation of PIs by NN has been applied for short-term wind speed prediction on the four different wind speed datasets (winter, summer, w2011, w2010). Although variability in the testing patterns of the winter and summer datasets are relatively higher compared to the training patterns, MOGA algorithm shows a good accuracy and generalization ability on all datasets.

In order to demonstrate the superiority of the proposed MOGA-NN method if compared to SOGA and SOSA, we have performed a comparison with these methods. In SOSA and SOGA, the CWC (15) has been used as the objective (cost) function. Both the SOGA and SOSA procedures, defined in Section 4.1, have been adapted to our NN-based PI estimation problem. It is worth saying that in SOGA we have used the roulette wheel selection algorithm

[104], [124]. Note that for both SOGA and SOSA, the user specified parameters have been tuned to select the ones giving optimal performances.

For the purpose of comparison, we have also calculated the CWC values of the optimal solutions on the Pareto front yielded by MOGA for both training and testing sets. We have run MOGA, SOSA and SOGA twenty times for each wind dataset. Then, we have used boxplots to analyze the experiment results. From the inspection of boxplots, we can draw the following general conclusion: MOGA algorithm shows more consistent testing results, i.e. better generalization capability, with respect to the CWC value if compared to SOGA and SOSA. For further explanations of the comparisons, we refer the reader to Paper II of Part II.

In addition to the comparisons explained above, we have also applied multi-objective covariance matrix adaptation evolution strategy (MO-CMA-ES) [125] as an alternative multi-objective training algorithm. With respect to the case study results, we can conclude that the optimal Pareto front obtained by NSGA-II is slightly better than the one obtained by MO-CMA-ES.

Finally, a comparison with the ARIMA model has been also performed. Although ARIMA provides high accuracy in terms of CP, PIWs are quite large (around 50% of the target range). Hence, PIs obtained with ARIMA cannot provide useful information in practice, because the uncertainty level is too high to support a reliable and informed decision in typical application contexts. Indeed, for winter and summer datasets, where the test set variability is relatively higher than in w2011 and w2010 datasets, MOGA-NN results in tighter interval widths for the same PICP value.

In conclusion, the tests and comparisons with other methods (SOGA, SOSA, MO-CMA-ES and ARIMA) prove and confirm the superiority of the proposed MOGA-NN method in our specific problem, i.e. short-term wind speed prediction.

For each algorithm, the average CPU times over 20 runs for both training and testing have been recorded using MATLAB on a PC with 4 GB of RAM and a 2.53-GHz processor. Table 5 reports the recorded training CPU times on the winter dataset. The SOSA PI construction time has been recorded for 15000 iterations. The average CPU time for the construction of testing PIs, i.e. for the online prediction of PIs, is very fast for all algorithms, being about 0.05 s. It is needless to say that computational load is dependent on the complexity of the structure

of the model (e.g. number of input neurons, hidden layers, and hidden neurons), the size of the dataset and the performance of the learning algorithm.

Table 4. Descriptive Statistics of CPU times (s) of twenty MOGA, SOSA and SOGA on winter training dataset.

	Mean (s)	**Std (s)**	**Min (s)**	**Max (s)**
MOGA	258.84	9.40	245.49	285.96
SOGA	199.36	12.42	186.42	231.32
SOSA	163.29	6.98	154.52	184.28

For what concerns Paper III, two different machine learning approaches to estimate PIs for time series predictions are considered and compared: MLP NN trained with NSGA-II and ELM combined with the nearest neighbors approach. The proposed approaches are applied for short-term wind speed prediction from a real dataset of hourly wind speed measurements for the region of Regina in Saskatchewan, Canada. It is worth mentioning that in this paper, we have used the same three wind speed datasets (winter, summer, w2011) also used in Paper II. The main contribution of this paper is the proposal of two different machine learning approaches for estimating PIs of time series of wind speed profiles and their comparison based on different criteria. The two approaches differ from each other and also from the approaches previously applied in other studies [35], [36], [92], [126], [127]. The MOGA-NN approach integrates the estimation of the PIs in the learning procedure of the algorithm. The algorithm itself is directly trained to balance the width of the interval and the coverage probability, concurrently optimizing the two quality assessment criteria of the PIs. The ELM algorithm combined with the nearest neighbor approach is trained to fit optimally the time series data (instead of estimating directly the PIs). Conceptually, the ELM algorithm extends the functionality of the standard feed-forward NNs by including different activation functions, and by overcoming computationally expensive learning algorithms, such as back-propagation. In the second step, the PIs are estimated on the basis of the performance of the algorithm on similar training samples in the input space. Both approaches are based on powerful learning algorithms and have proven to be capable of providing good generalization ability and accurate predictions, considering the uncertainties associated to the input data and the model parameters. According to the results of this study, they both look promising for time-series wind speed prediction.

The approaches are shown to yield a similar performance, with different strengths and limitations with respect to the criteria (prediction precision, generalization ability, variability, algorithm complexity, computational load, ease of parameter setting, etc.) used for the comparison. Both algorithms are data-driven and depend highly on the representativeness of the training dataset. Therefore, the quality of PIs can decrease on datasets with large variability and uncertainty in the data.

6. UNCERTAINTY TREATMENT: INTERVAL-BASED ESTIMATED PREDICTION INTERVALS

6.1 Problem Statement

Uncertainty representation and quantification play an important role in every decision making process. The need for an appropriate representation and quantification of uncertainty as part of any analysis that supports an important decision has been widely recognized in the engineering applications. [128].

For practical purposes, uncertainties can be classified in two distinct types [34]: epistemic (state-of-knowledge) and aleatory. The former derives from imprecise model representation of the system behavior, in terms of uncertainty in both the hypotheses assumed (structural uncertainty) and the values of the model parameters (parameter uncertainty). Model uncertainty arises because mathematical models are simplified representations of real systems and, therefore, their results may be affected by error or bias. Aleatory uncertainty describes the inherent variability of the observed physical phenomenon. This type of uncertainty cannot be reduced by conducting exhaustive measurements or by defining a better model, and, thus, it is also named irreducible uncertainty or inherent uncertainty [129].

Uncertainty quantification is the process of representing the uncertainty in the system inputs and parameters, propagating it through the model, and then revealing the resulting uncertainty in the model outcomes [130].

In the literature, methods such as probability theory [131], [132], possibility theory (e.g. fuzzy set theory and in particular type-2 fuzzy sets and interval type-2 fuzzy logic systems) [133], [134], evidence theory [135], interval analysis [136], [137], and MC simulation [138] have been widely used to efficiently represent, aggregate, and propagate different types of uncertainty through computational models.

In this Ph. D. work, with respect to the growing interest in representing and quantifying the uncertainties in distributed power generation systems, we use interval-analysis to represent the uncertainty in input data, typically subject to aleatory uncertainty, particularly in wind-integrated distributed power generation systems. In the framework of this thesis work, aleatory uncertainties concern for instance the time to failure of a component, and wind speed

and load variations, whereas the error associated to the NN regression model is an example of epistemic uncertainty.

6.2 Interval Analysis

Interval analysis is a promising technique for bounding solutions under uncertainty [137]. The uncertain model parameters are described by upper and lower bounds, and the corresponding bounds in the model output are computed using interval functions and interval arithmetic [136]. These bounds contain the true target value with a certain confidence level. The uncertainty in each element x_i of $\boldsymbol{x}$, where $\boldsymbol{x}$ is a vector of inputs in the form $\boldsymbol{x} = [x_1, x_2, \dots, x_n]$, is represented by an interval, and the goal of interval analysis is to construct the smallest interval that exactly contains the resultant possible values for $f(\boldsymbol{x})$ [128].

The interval-valued representation is typically used to reflect the variability in the inputs (e.g. extreme wind speeds in a given area, minimum and maximum of daily temperature, etc.), or their associated uncertainty (e.g. strongly skewed wind speed distributions, etc.), i.e. to express uncertain information associated to the input parameters [139], [140].

In the case study carried out in Paper IV, we use interval analysis to represent the uncertainty in the input, which is wind speed in our specific problem, and propagate it through the model outputs. In other words, uncertainty associated with an unknown input value is represented by a lower and upper bound without the assumption of distribution in-between. Section 6.3 provides the details of this case study.

6.3 Application to Wind speed Prediction Intervals Estimation with Interval Inputs

In Paper IV, we aim at quantifying the uncertainty in the prediction arising from both the input data and the prediction model. We perform prediction with NNs on the basis of uncertain input data expressed in the form of intervals. A MLP NN has been trained to map interval-valued input data into interval outputs, representing the PIs of the real target values. The MLP NN training has been performed by NSGA-II, so that the PIs are optimized both in terms of accuracy (coverage probability) and dimension (width).

Demonstration of the proposed method is given on two case studies: (*i*) a synthetic case study, with 5-minutes simulated data; (*ii*) a real case study, involving hourly wind speed measurements. In both cases, short-term prediction (1-hour and day-ahead, respectively) is

performed taking into account both the uncertainty in the model structure, and the variability (within-hour and within-day, respectively) in the inputs. An interval representation has been given to the hourly and daily inputs by using two different approaches, namely min-max and mean, which quantify in two different ways the within-hour and within-day variability. The NN maps interval-valued input data into an interval output, providing the estimated PIs for the real target.

The originality of the work appears in two aspects: (*i*) while the existing papers on short-term wind speed/power prediction use single-valued data as inputs, obtained as a within-hour [39], [141], or within-day average [142], we give an interval representation to hourly/daily inputs by using two approaches which properly account (in two different ways) for the within-hour/day variability, and (*ii*) we perform a comparison between methods taking into account interval-valued and point-valued (crisp) inputs to demonstrate that the former are more reliable and powerful than the latter in our specific problem.

The wind speed dataset, covering the period from January 1, 2010 till December 30, 2012, has been downloaded from the website [123]. Since hourly data have been collected, 24 wind speed values are available for each day. Fig. 5 in Paper IV shows the behavior of hourly wind speed values only in the first 20 days, for the sake of clarity: one can appreciate the within-day variability in each individual day. The wind speed changes from 0 km/h to 72 km/h with an unstable behavior. From this raw hourly wind speed data, one can obtain daily interval wind speed data with the min-max and mean approach. The so obtained datasets include 1095 intervals among which the first 60% is used for training, 20% for validation and the remaining 20% for testing. The inputs are historical wind speed data $W_{t-1}$and W_{t-2} both for the method considering interval inputs and the one with crisp inputs; the optimal number of inputs has been chosen from an auto-correlation analysis [48].

The basis of our interval computations is interval arithmetic [136] and we have used MATLAB INTLAB Version 6 toolbox for all interval arithmetic calculations. Therefore, we have fed each neuron with an interval $[a, b]$ instead of two single values. Note that we have not applied the "extremal values method" explained in [143] to convert the interval-valued inputs into two single values: we have used only one input neuron for each interval-valued input variable $X(i)$, which is 2 in the wind speed case study; each neuron receives interval-valued inputs and produces interval-valued outputs. Each input variable, i.e. each interval input vector $X(i)$, is described by n_p intervals (as we have n_p samples), i.e.,

$([x_1^-, x_1^+], [x_2^-, x_2^+], \ldots, [x_{n_p}^-, x_{n_p}^+])$ [139]. The output neuron is also described as an interval $[a, b]$ where a and b are real numbers.

In order to show the strength of NSGA-II to train a MLP fed by interval-valued input, we have performed a comparison with SOSA. To perform a comparison between SOSA and the proposed MOGA method, we have run the SOSA by using the same interval-valued wind speed training data. For SOSA, the initial temperature has been determined after a trial and error procedure.

For SOSA, the training process has been repeated five times. The cost function was CWC. According to the training and testing results of SOSA (see Paper IV, Section 4.3), it can be observed that the training and corresponding testing solutions do not show high consistency in terms of coverage probability and interval size among the five runs performed. In other words, there is a high variability among the results of the five runs, e.g. SOSA gives high CP values in one run whereas it generates less accurate PIs in another one: 3 out of 5 runs give CP values smaller than the predetermined nominal confidence level, i.e. 90% in our experiments. This shows a drawback in the SOSA method concerning its robustness on this specific problem.

In conclusion, we have observed that the solutions obtained by MOGA dominate the best ones obtained by SOSA. It is worth pointing out that as both solutions give large interval sizes (around 50%) they cannot provide useful information in practice, because the uncertainty level is too high to support a reliable and informed decision in typical application contexts. However, with the MOGA approach one can select a solution from the Pareto front giving tight PIW with a high CP, which satisfies the predetermined nominal confidence level.

Results of two case studies prove the superiority of the interval-valued input approach with respect to the single-valued one. The former case study turns out to provide higher PICP with lower NMPIW on synthetic data generated from an ARMA model with either Gaussian or Chi-squared innovation distribution. The synthetic data shows a stationary and periodic pattern whereas the wind speed data is highly volatile.

6.4 Application to Adequacy Assessment of a Wind-Integrated Distributed Power Generation System

Paper V addresses the problem of accurate adequacy assessment with quantification of the associated uncertainties, which is crucial for the reliable operations and the sustainability of energy systems. We have presented a framework for conducting the adequacy assessment of a wind-integrated power system accounting for the associated uncertainties in the data (load fluctuations, wind variability, and component failures) and the prediction models. Our adequacy assessment framework leads to the evaluation of the EENS [28], [144], on the basis of interval-valued load and wind power input data. Wind power and load PIs have been estimated by NNs where the training of the network is performed by NSGA-II, so that the PIs are optimized in terms of both accuracy and dimension. Interval-valued EENS results are then obtained to allow informative decision making taking into account the uncertainty in the predictions.

In order to conduct the adequacy assessment of the wind-integrated power system, we use a well-known adequacy index, EENS, which quantifies the capability of the system to meet the demand in the time horizon considered for the analysis. EENS measures the expected value of the energy not supplied due to the lack of available energy through the given time horizon (e.g. one year). It depends on the predicted values for both the system energy production and the power demand, and it is formulated as follow:

$$ENS_k = \sum_{t=1}^{N} Pr(L_t > G_t) \times (L_t - G_t) \quad (24)$$

where ENS_k is the realization of the energy not supplied for the entire horizon in the *k*-th simulation run; t is the equally sized time step (e.g. hour or day); N is the total number of time steps in the considered time horizon, in our case $N = 8736$ for a one year time horizon; G_t is the total power generation available at time step t; L_t is the load demand at time step t; $Pr(L_t > G_t)$, which indicates the probability that the load demand exceeds the available power generation at time step t, is a generalized form of $1(L_t > G_t)$ to handle the interval values of L_t and G_t: when L_t and G_t are crisp values as in the classical adequacy assessment, $Pr(L_t > G_t)$ is reduced to $1(L_t > G_t)$ which equals to 1 if the condition is satisfied, otherwise equals to 0.

Thus, EENS value of the system, i.e. the average amount of the unsupplied energy per year, is estimated as follow:

$$EENS = \frac{\sum_{k=1}^{K} ENS_k}{K} \tag{25}$$

where K is the total number of simulations that has been set to 100 in our experiments.

In the classical estimation of EENS (25), both the predicted value of the generation G_t and of the load L_t at each time step t are assumed to be point estimates, resulting in a point estimate of EENS. Our method is, instead, capable of providing PIs for both the power generation and the load at each time step, to take into account the possible uncertainties in the prediction arising from both the underlying physical processes (wind inherent uncertainty, variability in power demand, etc.) and in the system stochastic behavior (equipment failures, approximations of the system complexities, etc.). A proper adequacy assessment model should take these sources of uncertainty into account, since uncertainty quantification is crucial for a real understanding of the system behavior, and for obtaining reliable results useful for robust decision making. Hence, we aim at a generalization of the EENS in order to include interval estimates of both G_t and L_t.

To this aim, two different strategies are considered for interval-based EENS estimation: a point estimation and an interval estimation. The objective is to know and dominate the impact of the uncertainty in wind and load on the uncertainty in EENS. Both strategies are interval-based, in the sense that the inputs to the evaluation are the short-term PIs for load and for power generation, as obtained by the NN-based PI estimation procedure. The former is based on the probability density function of the continuous random variable $\xi_t = l_t - g_t$, where $l_t \in L_t$ and $g_t \in G_t$ are, respectively, two admissible values of the load demand and power generation at time t, thus $\xi_t \in [max\{0, L_t^- - G_t^+\}, L_t^+ - G_t^-]$ and it results in point estimation of ENNS considering interval-valued input variables, G_t and L_t. On the other hand, the latter takes into account load and power generation PIs in EENS estimation, and it consists in directly using (24) with interval-valued G_t and L_t, thus obtaining as a result an interval evaluation of EENS by directly applying the principles of interval arithmetic [136]. Detailed explanations on the two proposed strategies are given in Paper V of Part II.

The case study consists in the analysis of hourly wind speed data from the region of Regina, Saskatchewan, Canada, from a 9-year period (1 Jan. 2003 to 31 Dec. 2011). Hourly mean

wind speed data are used to determine the time-dependent wind power output of a wind turbine generator (WTG) using its power curve [145], [146]. For load demand, the hourly load fluctuations are modeled using the chronological annual load curve of the IEEE Reliability Test System (RTS) [147] with the scaled annual peak load value.

For comparison purposes, we have considered 6 different scenarios, corresponding to the computation of EENS by taking into account different uncertainty levels in the input parameters, i.e. wind power, load and system state. These scenarios have been called PEENS, interval ENS, ENS LB, ENS mean, ENS UB and ENS actual. ENS LB and ENS UB have been calculated by considering only the LB and UB of the estimated load and wind power PIs, respectively, and by computing a single-valued inputs ENS index. Similarly, to estimate ENS mean, the central values (mean point) of the PIs have been used as input. For computing ENS actual, we have used the actual data sets: ENS actual is, thus, the unknown quantity we would like our estimates to be close to, and it cannot be computed in a real case study; we have calculated it here only for demonstration of the strength of our approach. Simulation results on different scenarios confirm that uncertainties in input data can be properly taken into account to obtain more reliable EENS estimations.

As an alternative method to estimate PIs for wind power, we have used the histogram and empirical cumulative distribution function (CDF) of wind power at time *t* using the historical data. One can see that the PIs obtained by the empirical distribution, i.e. the histogram, do not give accurate and reliable coverage for the target of interest. NN-based PIs obtain the same coverage probability (95%) with lower interval size. One can appreciate that the PIs estimated by the histogram of wind power at time *t* cannot provide useful information in practice, since the uncertainty level in the outcome is too high, i.e. the interval size is too large. On the contrary, the training of the NN with wind speed historical data ensures accounting for the time dependency among successive observations, leading to more accurate predictions.

7. UNCERTAINTY TREATMENT: NN ENSEMBLES

7.1 Construction of an Ensemble of NNs

In general terms, it is well known that an ensemble of different predictors can generate predictions that are more accurate than those obtained by individual predictors [148]. Specifically, a NN ensemble is a learning paradigm where a certain number of NNs are combined to estimate the desired output for the target of interest (see Figure 11) [148]. Typically, a NN ensemble is constructed in two steps: *i*) training a number of individual NNs and *ii*) combining the predictions yielded from these NNs. The aim of assembling a number of NNs into an ensemble is to improve the generalization ability and estimation accuracy of the prediction model.

Considerable research has been carried out both on ensembles and, also, specifically on ensembles of NNs. Traditional NN ensemble techniques have been built via several strategies, such as randomly trying different topologies (different number of hidden layers and neurons) in each individual NN, setting different initial weights or parameters, using different training datasets (e.g. bagging, CV, etc.) or learning algorithms, etc. [148]-[151].

Bagging and boosting are the most prevailing approaches used to produce ensembles [148], [151]. Bagging is based on bootstrap sampling [152], since it produces replicate training sets by sampling with replacement from the training samples [149], [153], [154]. The method works by training the multiple (*m*) models on different data splits (generated by sampling with replacement from the original training dataset), and by averaging their outcomes to obtain the ultimate prediction results on the testing set [153]. The bootstrap method is one of the most widely used statistical methods for standard errors estimation and for construction of CIs and PIs related to the response variable. This is due to its ease of use and to its robustness, and also to the advantages of not requiring assumptions about its probability distribution, and of being efficient even when a small data set is available [97], [154], [155]. The bootstrap is a computational procedure that uses resampling with replacement, in order to reduce uncertainty [97].

In boosting ensembles, the patterns that the earlier classifiers in the series recognized incorrectly are over-represented in the composition of a particular training set, i.e. training samples that are incorrectly predicted by previous classifiers in the series are more often

chosen than samples that were correctly predicted [149], [151], [156]. Thus, boosting aims at producing new classifiers that are more capable to predict samples for which the current ensemble performance is poor [151].

Regarding the combination of the estimated predictions (outputs) of each individual NN, different techniques can be adopted, like a simple arithmetic mean, a weighted mean, a median, a linear combination, local fusion (LF), dynamic integration, etc. [149], [157]. As an exemplification, Baraldi et al. [157] have explored the LF strategies for the aggregation of the outcomes of different ensemble models, whereas Khosravi et al. [150] have combined individual PI forecasts through mean and median calculations.

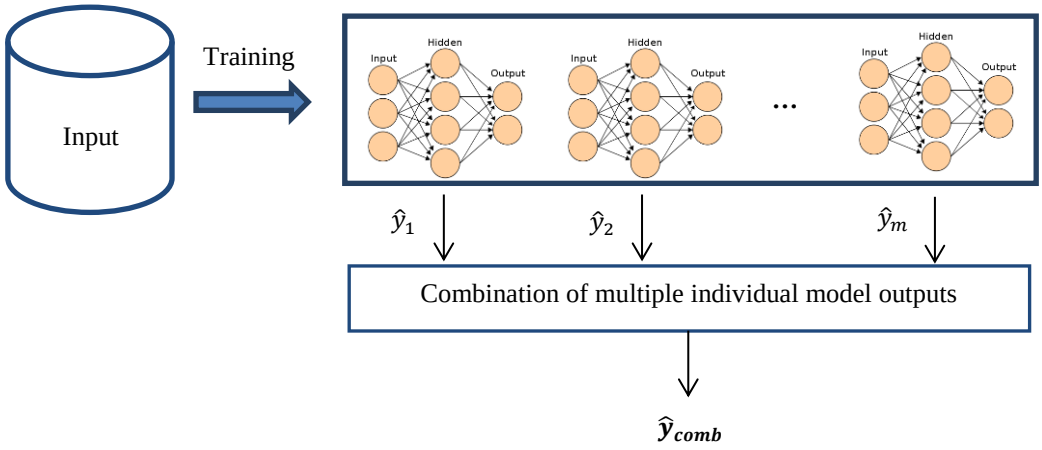


Figure 11. A basic scheme of NN ensemble.

7.2 Application to Wind Power Prediction Intervals Estimation with Interval Wind Speed Inputs

In Paper VI, we propose a novel approach to short-term (1-h ahead) wind power forecasting with uncertainty quantification. The approach can be schematized in two steps: first, short-term estimation of wind speed PIs is performed within a multi-objective optimization framework worked out by NSGA-II; then, the uncertainty in wind speed and the uncertainty in the power curve are combined via a bootstrap sampling technique.

In the present work, we treat the power curve parameters as random variables and account for the epistemic uncertainty by bootstrapping [158], which allows combining also the aleatory uncertainty in the wind speed. The inherent stochasticity in the power curve is motivated by the fact that different wind turbines correspond to specific power curve parameters, which leads to an imprecise and imperfect knowledge of the power curve transformation. A plot of the power curve with parameters V_{ci}, V_r, V_{co}, P_r, i.e. cut-in speed, rated speed, cut-off speed and rated power, is shown in Figure 12.

In the case study, we consider V_{co} and P_r to be fixed (deterministic) values, and respectively equal to the values 30 m/s and 20 MW [159], [160] while V_{ci} and V_r are random variables with distributions F_{ci} and F_r, respectively. More precisely, we sample V_{ci} and V_r from both a uniform and a Gaussian distributions centered around average values of 3.5 and 14.5 m/s, respectively, with a range of uncertainty of [3, 4] and [12, 17] m/s, respectively, defining the domain of the associated distribution (see Figure 12). The two parameters are sampled either from a uniform distribution ($F_{ci} = U[3,4]$ and $F_r = U[12,17]$), or from a Gaussian one ($F_{ci} = N(3.5, (1/6)^2)$ and $F_r = N(14.5, (5/6)^2)$.

Estimation of the wind power PIs based on estimated wind speed PIs are performed as follows:

i. Given the estimated hourly wind speed PIs $[L(x_1), U(x_1)], \dots, [L(x_n), U(x_n)]$ on the testing set, sample two values for the stochastic parameters V_{ci} and V_r from the corresponding distributions, i.e. $V_{ci} \sim F_{ci}$ and $V_r \sim F_r$, and transform all wind speed PIs $[L(x_1), U(x_1)], \dots, [L(x_n), U(x_n)]$ into wind power PIs $[L_p(x_1), U_p(x_1)], \dots [L_p(x_n), U_p(x_n)]$, via the power curve transformation. In the case study, this procedure has been repeated 1000 times.

ii. Aggregate the results of the bootstrap phase by computing, for each element of the testing set, the bootstrapped average wind power PI and the 5th and 95th percentiles of the wind power PI bootstrapped distribution.

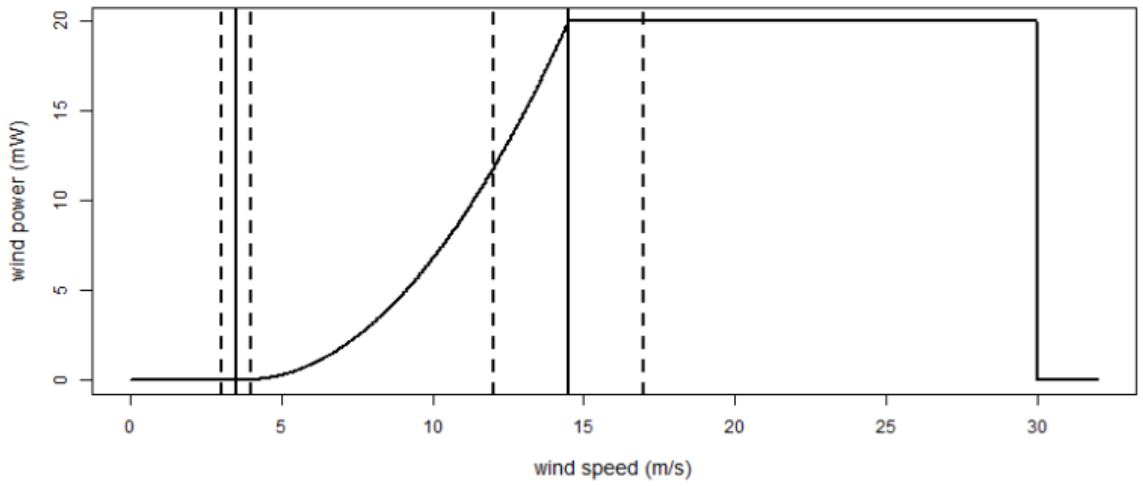


Figure 12. Plot of the power curve g_θ as a function of wind speed. Solid vertical lines correspond to the values of the two stochastic parameters V_{ci} and V_r. Dashed vertical lines identify the domains of the distributions F_{ci} and F_r, respectively.

The user-specified parameters of NN and NSGA-II, and the plots of the resulting average bootstrapped PIs for 1-h ahead wind power prediction are given in the paper.

In short, considering the fact that wind-integrated distributed power generation systems are subject to both epistemic and aleatory uncertainties, this paper presents a novel approach for an adequate treatment (quantification) of both types of uncertainties. The proposed approach quantifies aleatory uncertainty by estimating wind speed PIs, and then transforms them into wind power PIs by using a power curve. In doing so, epistemic uncertainty arising from the imperfect knowledge of the power curve parameters is also taken into account through bootstrap sampling. The procedure manages to effectively decouple aleatory and epistemic uncertainty, and shows a good robustness with respect to the parametric assumptions implicit in the bootstrap. The invariance of the coverage probability by passing from wind speed to wind power PIs has also been shown.

7.3 Application to Short-term Wind Speed Prediction Intervals Estimation

In paper VII, we address the problem of short-term wind speed prediction for wind power production. PIs are considered to account for the uncertainties in the predictions and two non-parametric methods are proposed to construct ensemble models made by NNs to estimate PIs. The proposed method is the enhanced version of the non-parametric MOGA-NN method [38],

[39], here extended to build an ensemble of MLP NNs as base learners. We then apply this method to the problem of short-term wind speed prediction.

We propose two strategies for the construction of the NN ensemble, differing in the partitioning or not of the training dataset, and embedding the *k*-nearest neighbors (*k*-nn) approach in the aggregation phase for the identification of the neighborhoods of a test pattern [157], [161]. The first strategy splits the training dataset into sub-sets with an equal number of samples and, then, each individual NN is trained on a different sub-training set; the second strategy, instead, uses the same training dataset (the entire dataset) for the training of each individual NN. The two methods differ also in the combination method of the individual NNs outputs. Note that in method 2, we obtain an overall Pareto front, hereafter called combined Pareto front, which is obtained by applying non-dominated sorting to the Pareto fronts obtained by the training of each network.

Each individual NN in the ensemble is trained independently to minimize the prediction error with respect to the target. We have used the same architecture (i.e. number of hidden neurons) for each individual NN. The number of hidden neurons has been determined by a trial-and error method. In both methods, the validation set has been used to screen the NNs with respect to their performance in PIs estimation on the validation set

On the real data considered as case study, both methods have obtained superior results compared to those yielded from the individual networks selected in the respective ensembles. Compared to literature methods conceptually and methodologically similar to the present ones, the results obtained show a significant improvement in terms of the quality of the predicted PIs. We can, then, conclude that both ensemble modeling frameworks proposed yield a reliable estimation of the PIs, characterized by a high coverage probability and a small interval size. The reported results demonstrate the practically efficient methods proposed for quantification of uncertainties associated with wind speed prediction.

8. CONCLUSIONS

8.1 Methodological and applicative contributions

The main objective of every electric utility is to meet the energy demand at any time with the lowest possible cost to customers while maintaining acceptable levels of service quality. To this aim, a good knowledge of the future electricity production and consumption stands as a central point. Forecasts of energy demand and production may be included directly in the unit commitment and economic dispatch scheduling process used to ensure that enough generation is available to meet energy demand, or they may simply provide situational awareness for the balancing authority. For this, forecasting methods and models must provide a way to measure of the risk of using the forecasted outcomes in the decision process due to the associated uncertainties.

It is within this context that, in this Ph. D. thesis, we have developed a non-parametric, empirical approach to generate NN-based PIs to account for uncertainty in the prediction due to the variability in the input data and in model approximation errors. As application, we have considered the general problem of adequacy assessment of wind-integrated power systems. In the case studies, we have concentrated particularly on short-term (hour-ahead or day-ahead) wind speed/power and load forecasts, for their relevance to system operations in both the unit commitment and economic dispatch phases.

With reference to the contents addressed in the previous Chapters, the main contributions of the thesis in the domain of adequacy assessment of power networks, particularly in wind-integrated distributed generation systems, presented in Papers I-VII of Part II are formalized as responses to the each research objectives introduced in Section 1.4.

Contributions with respect to the objective 1:

Objective 1: *To develop a prediction method capable of considering the uncertainty in the model parameters affecting the prediction*.

- We propose a multi-objective framework for estimating PIs, optimal in terms of both accuracy (coverage probability) and efficacy (width). More precisely, we propose a multi-objective NN-based PI estimation method to quantify the uncertainties associated to the prediction problem. With the proposed multi-objective framework, we are able to generate a Pareto front of optimal non-dominated solutions. Each solution corresponds to a NN.

DMs can choose any of the solutions on the Pareto front according to his / her preferences as a good compromise in terms of high PICP and low NMPIW.

- Knowledge of PIs allows the DMs and operational planners to quantify the level of uncertainty associated with the forecasts and to consider a multiplicity of solutions/scenarios for the best and worst conditions.
- We use NSGA-II, which is one of the most powerful MOEAs, for NN training. A comparison with another powerful multi-objective optimization algorithm, MO-CMA-ES, has been performed. The comparison results have shown that the PIs produced by NSGA-II are superior to those obtained with MO-CMA-ES, and satisfactory in both objectives of high coverage and small width. It is worth pointing out that it is the first time that NSGA-II is used to solve the problem for finding optimal lower and upper bounds of PIs.
- In order to show the superiority of the proposed multi-objective framework to the single-objective frameworks, particularly to the original LUBE method [36], in Paper II, we have performed comparisons on different datasets. In addition, in Paper II, we have also performed a comparison with a classical time-series regression method, i.e. ARIMA. The results confirm the superiority of our MOGA-NN approach.

Contributions with respect to the objective 2:

Objective 2: *To represent the uncertainty in input data and propagate it through the prediction model onto its results*.

- In order to represent the uncertainty in input data and propagate it through the prediction model onto its results, we present an interval-valued time series prediction modeling framework based on NNs [41]. With the interval-valued representation, one can reflect the variability in the inputs (e.g. extreme wind speeds in a given area, daily peak load, minimum and maximum of daily temperature, etc.), or their associated uncertainty (e.g. strongly skewed wind speed distributions, non-stationary load patterns, etc.).
- We have presented two approaches that can be used to process interval-valued inputs to NNs, which aim at providing more accurate quantification of the input uncertainty in the prediction problem. The experiment results reveal that the interval-valued input approach is capable of capturing the variability in the input data with the required coverage. The results enable different strategies to be planned according to the range of possible outcomes within the interval forecast.

- With respect to the case study comparison results, we can conclude that our method for interval-valued day-ahead wind speed prediction performs better than the one with single-valued inputs, in that we have obtained higher quality PIs.
- Moreover, comparison results carried out between two learning algorithms, SOSA and MOGA (NSGA-II), show the superiority of the latter in training the NN in our specific problem.

Contributions with respect to the objective 3:

Objective 3: To enhance the performance of a NN-based, non-parametric prediction method by an ensemble approach.

- This objective is addressed through the introduction of a novel NN ensemble-modelling framework, by two methods to estimate PIs for short-term wind speed prediction. In the aggregation phase of the selected individual NN results, we have used *k*-nn approach to determine the similar patterns between training and testing sets. This allows us to obtain high accurate results also on the testing set by using the local information coming from the closest patterns of the training sets.
- Both methods demonstrate consistent results and high prediction precision compared to the individual NNs of the ensemble and to conceptually similar methods proposed in the literature [150].
- We can conclude that the NN ensemble approach proposed in Paper VII can provide a significant improvement in the quality of short-term wind speed prediction.

Contributions with respect to the objective 4:

Objective 4: To test the proposed model on real case studies in the context of energy system applications (in particular adequacy assessment).

- In Paper I, data has been obtained from experiments aimed at observing the process of deposition of the scale layer in [117], [119].
- In Papers II and III, the test of the proposed MOGA-NN approach is done on several different datasets concerning short-term wind speed and load forecasting. Wind speed datasets show different wind speed profiles with seasonality measured for the region of Regina in Saskatchewan, Canada. The first dataset comprises wind speeds for the period from 1st of February 2012 to 31st of March 2012; the second from 1st of July 2012 to

29th of August 2012; the third from 1st of February 2011 to 30th of June 2011, and the last one from 1st of May 2010 to 30th of September 2010.

- In Paper IV, the proposed method has been applied on a synthetic case study and on a real case study, in which the data show a high (short-term) variability (within hour and within day) [41]. The real case study includes the wind speed dataset which covers the period from 1st of January 2010 till 30th of December 2012 [123].
- In Paper V, hourly wind speed data from the region of Regina, Saskatchewan, Canada taken, from a 9-year period (1 Jan. 2003 to 31 Dec. 2011) is considered in the case study [123]. Then, hourly mean wind speed data are used to determine the time-dependent wind power output of a wind turbine generator (WTG) using its power curve. For load demand, the hourly load fluctuations are modeled using the chronological annual load curve of the IEEE Reliability Test System (RTS) [10] with the scaled annual peak load value.
- In Paper VII, the hourly wind speeds measured from 1st of February 2003 to 28th of July 2012 in Regina, Saskatchewan, 80000 samples in total [123].
- We can conclude that the case studies on different datasets have let us test the performance of the MOGA-NN method on various datasets having different variability.

Like all the machine learning methods, NNs have some limitations besides their advantages. In general, NNs give high satisfactory performance in forecasting. Their capability to learn the non-linear relationship between input and output and arbitrary function mapping ability make them suitable and promising for forecasting tasks. On the other hand, NNs are data-driven and depend highly on the representativeness of the training dataset, i.e. data driven prediction methods are prone to give less accurate results depending on the high level of variability in the test set, i.e. unseen data, under consideration. Therefore, the prediction accuracy can decrease on test dataset with large variability and uncertainty in the data, with respect to the training. In other words, the difference between training and testing dataset profiles plays an important role in the generalization power of the model. Hence, a data-driven prediction method does not always guarantee to generate high quality predictions on unseen data. Moreover, a NN model can require a computationally intensive procedure for training that requires large computational times. Mostly, the computation time correlates with the network size (i.e. topology), thus the number of parameters to be optimized, and the number of training samples.

It is worth pointing out that tackling a regression problem requires also proper pre-treatment of the input data. How best to select the input variables pertinent to the output variable(s), i.e. feature selection, for inclusion in a model is an important factor that affects both the prediction accuracy and computational cost of the underlying model. For time series forecasting, the number of the previous lags related to the output is also a key factor to be determined properly. In this thesis work, we also address these issues in our case studies, with classical techniques.

8.2 Future Work

To extend the work developed in this thesis, different research directions can be taken into account. Important aspects to investigate and further explore include the following perspectives:

- Due to the critics recently raised in [162] on the limitations of the CWC proposed by Khosravi et al. [35], [36] for evaluating the quality of the estimated PIs, alternative proper scores [163], [164] might be considered. Proper scores, proven in theory and in practice, would allow drawing safe conclusions on the potential superiority of the proposed method.
- Taking into account a combination, i.e. ensemble, of different forecasting methods would help further reducing forecasting errors, thus to improve the forecasting accuracy.
- The implementation of online learning algorithms, which are able to adjust their parameters while novel patterns evolve without retraining the whole algorithm, would also improve forecasting accuracy and quality. The additional use of online wind measurement data has also the potential for improved forecasts, especially where the available dataset is too short to cover all possible patterns, or when the environmental or operational conditions change.
- A novel formulation, particularly for the offshore wind power prediction, can be provided which takes into account both available spatial and temporal information/data, i.e. considering using spatio-temporal correlation.
- Type-2 fuzzy sets can be integrated into the proposed model as an alternative way to represent the input uncertainty.
- The proposed NN-based PIs method can be integrated into a cost model to estimate electricity prices uncertainties.

- Application areas can be extended. For exemplification, for energy demand prediction, energy consumption in buildings can be considered as a case study by considering the potential uncertainties in the context of a Building Energy Management System (BEMS).

REFERENCES

[1] World Energy Council, "World Energy Resources 2013". Online: http://www.worldenergy.org/publications/2013/world-energy-resources-2013-survey.

[2] The World Wind Energy Association, "World Wind Energy Report 2012". Online: http://www.wwindea.org/webimages/WorldWindEnergyReport2012_final.pdf.

[3] BP, "BP Statistical Review of World Energy 2013". Online: http://www.bp.com/en/global/corporate/sustainability/about-our-reporting.html.

[4] A. Maczulak, *Renewable Energy: Sources and Methods*. Infobase Publishing, 2010.

[5] Pierre Pinson, "Wind energy: Forecasting challenges for its operational management", *Statistical Science*, pp. 564‑585, 2013.

[6] J. Twidell and T. Weir, *Renewable Energy Resources*. Taylor & Francis, 2006.

[7] The European Wind Energy Association, "United in tough times - EWEA annual report 2012". Online: http://www.ewea.org/publications/reports/.

[8] World Wind Energy Association, "Key Statistics of World Wind Energy Report 2013", Apr. 2014. Online: http://www.wwindea.org/webimages/WWEA_WorldWindReport KeyFigures_2013.pdf.

[9] Global Wind Energy Council, "Global Wind Report: Annual Market Update 2013", Apr. 2014. Online: http://www.gwec.net/wp-content/uploads/2014/04/GWEC-Global-Wind-Report_9-April-2014.pdf.

[10] The Office of Energy Efficiency and Renewable Energy (EERE). "Advantages and Challenges of Wind Energy".

[11] G. Chicco and P. Mancarella, "Distributed multi-generation: A comprehensive view", *Renewable and Sustainable Energy Reviews*, vol. 13, no. 3, pp. 535‑551, Apr. 2009.

[12] M. H. Bollen and F. Hassan, *Integration of Distributed Generation in the Power System*. John Wiley & Sons, 2011.

[13] T. Ackermann, G. Andersson, and L. Soder, "Electricity market regulations and their impact on distributed generation", in *International Conference on Electric Utility Deregulation and Restructuring and Power Technologies, 2000. Proceedings. DRPT 2000*, pp. 608‑613.

[14] A. F. Zobaa and C. Cecati, "A comprehensive review on distributed power generation", in *International Symposium on Power Electronics, Electrical Drives, Automation and Motion, 2006. SPEEDAM 2006*, 2006, pp. 514‑518.

[15] A. Alarcon-Rodriguez, G. Ault, and S. Galloway, "Multi-objective planning of distributed energy resources: A review of the state-of-the-art", *Renewable and Sustainable Energy Reviews*, vol. 14, no. 5, pp. 1353‑1366, June 2010.

[16] G. Pepermans, J. Driesen, D. Haeseldonckx, R. Belmans, and W. D'haeseleer, "Distributed generation: definition, benefits and issues", *Energy Policy*, vol. 33, no. 6, pp. 787‑798, Apr. 2005.

[17] International Energy Agency, "Distributed Generation in Liberalised Electricty Markets", 2002.

[18] C. L. T. Borges, "An overview of reliability models and methods for distribution systems with renewable energy distributed generation", *Renewable and Sustainable Energy Reviews*, vol. 16, no. 6, pp. 4008‑4015, Aug. 2012.

[19] J. Endrenyi, M. P. Bhavaraju, K. A. Clements, K. J. Dhir, M. F. McCoy, K. Medicherla, N. D. Reppen, L. A. Saluaderi, S. M. Shahidehpour, C. Singh, and J. A. Stratton, "Bulk power system reliability concepts and applications", *IEEE Transactions on Power Systems*, vol. 3, no. 1, pp. 109‑117, Feb. 1988.

[20] Nader Amin Aziz Samaan, "Reliability Assessment of Electric Power Systems Using Genetic Algorithms", Dissertation, Texas A&M University, 2004.
[21] M. C. Mabel, R. E. Raj, and E. Fernandez, "Adequacy evaluation of wind power generation systems", *Energy*, vol. 35, no. 12, pp. 5217–5222, Dec. 2010.
[22] Y. Gao, R. Billinton, and R. Karki, "Composite generation and transmission system adequacy assessment considering wind energy seasonal characteristics", in *IEEE Power Energy Society General Meeting, 2009. PES '09*, 2009, pp. 1–7.
[23] Y. M. Atwa, E. F. El-Saadany, M. M. A. Salama, R. Seethapathy, M. Assam, and S. Conti, "Adequacy Evaluation of Distribution System Including Wind/Solar DG During Different Modes of Operation", *IEEE Transactions on Power Systems*, vol. 26, no. 4, pp. 1945–1952, Nov. 2011.
[24] Roy Billinton and Ronald N. Allan, *Reliability Evalution of Power Systems*, 2nd Edition. New York, NY, USA: Plenum Press, 1996.
[25] J. Wen, Y. Zheng, and F. Donghan, "A review on reliability assessment for wind power", *Renewable and Sustainable Energy Reviews*, vol. 13, no. 9, pp. 2485–2494, Dec. 2009.
[26] R. Billinton, R. Karki, Y. Gao, D. Huang, P. Hu, and W. Wangdee, "Adequacy Assessment Considerations in Wind Integrated Power Systems", *IEEE Transactions on Power Systems*, vol. 27, no. 4, pp. 2297–2305, Nov. 2012.
[27] Y. Gao and R. Billinton, "Adequacy assessment of generating systems containing wind power considering wind speed correlation", *IET Renewable Power Generation*, vol. 3, no. 2, pp. 217, 2009.
[28] Power Systems Engineering Committee, "Reliability Indices for Use in Bulk Power Supply Adequacy Evaluation", *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-97, no. 4, pp. 1097–1103, 1978.
[29] L. Goel and R. Gupta, "A Windows-based Simulation Tool for Reliability Evaluation of Electricity Generating Capacity", *International Journal of Engineering Education*, vol. 13, no. 5, pp. 347–357, 1997.
[30] T. Ackermann, *Wind Power in Power Systems*. John Wiley & Sons, 2005.
[31] A. M. Foley, P. G. Leahy, A. Marvuglia, and E. J. McKeogh, "Current methods and advances in forecasting of wind power generation", *Renewable Energy*, vol. 37, no. 1, pp. 1–8, Jan. 2012.
[32] D. Lew, M. Milligan, G. Jordan, and R. Piwko, "The value of wind power forecasting", presented at the 91st American Meteorological Society Annual Meeting, Washington, DC, 2011.
[33] J. C. Refsgaard, J. P. van der Sluijs, J. Brown, and P. van der Keur, "A framework for dealing with uncertainty due to model structure error", *Advances in Water Resources*, vol. 29, no. 11, pp. 1586–1597, Nov. 2006.
[34] E. Zio and T. Aven, "Uncertainties in smart grids behavior and modeling: What are the risks and vulnerabilities? How to analyze them?", *Energy Policy*, vol. 39, no. 10, pp. 6308–6320, Oct. 2011.
[35] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya, "Comprehensive Review of Neural Network-Based Prediction Intervals and New Advances", *IEEE Transactions on Neural Networks*, vol. 22, no. 9, pp. 1341–1356, sept. 2011.
[36] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya, "Lower Upper Bound Estimation Method for Construction of Neural Network-Based Prediction Intervals", *IEEE Transactions on Neural Networks*, vol. 22, no. 3, pp. 337–346, 2011.
[37] P. Pinson and G. Kariniotakis, "Conditional Prediction Intervals of Wind Power Generation", *IEEE Transactions on Power Systems*, vol. 25, no. 4, pp. 1845–1856,

Nov. 2010.
[38] R. Ak, Y. Li, V. Vitelli, E. Zio, E. López Droguett, and C. Magno Couto Jacinto, "NSGA-II-trained neural network approach to the estimation of prediction intervals of scale deposition rate in oil & gas equipment", *Expert Systems with Applications*, vol. 40, no. 4, pp. 1205‑1212, 2013.
[39] R. Ak, Y-F. Li, V. Vitelli, and E. Zio, "Multi-objective Genetic Algorithm Optimization of a Neural Network for Estimating Wind Speed Prediction Intervals", *Applied Soft Computing*, 2014, (under review).
[40] R. Ak, O. Fink, and E. Zio, "Two Machine Learning Approaches for Short-Term Wind Speed Time Series Prediction", *Special Issue on Neural Networks and Learning Systems Applications in Smart Grid*, 2014, (under review).
[41] R. Ak, V. Vitelli, and E. Zio, "An Interval-Valued Neural Network Approach for Prediction Uncertainty Quantification", *IEEE Transactions on Neural Networks and Learning Systems*, 2014, (under review).
[42] R. Ak, Y-F. Li, V. Vitelli, and E. Zio, "Adequacy Assessment of a Wind-integrated Power System using Neural Network-based Interval Predictions of Wind Power Generation and Load", *International Journal of Electrical Power & Energy Systems*, 2014, (under review).
[43] R. Ak, V. Vitelli, and E. Zio, "Uncertainty modeling in wind power generation prediction by neural networks and bootstrapping", in *Proceedings of ESREL 2013*, 2013, pp. 1‑6.
[44] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II", *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182‑197, 2002.
[45] D. L. Shrestha and D. P. Solomatine, "Machine learning approaches for estimation of prediction interval for the model output", *Neural Networks*, vol. 19, no. 2, pp. 225‑235, 2006.
[46] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Springer, 2009.
[47] C. Chatfield, *Time-Series Forecasting*. CRC Press, 2000.
[48] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, 2013.
[49] S. Makridakis, S. C. Wheelwright, and R. J. Hyndman, *Forecasting Methods and Applications*, 3rd edition. Wiley India Pvt. Limited, 2008.
[50] P. J. Brockwell and R. A. Davis, *Introduction to Time Series and Forecasting*. Taylor & Francis, 2002.
[51] G. P. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model", *Neurocomputing*, vol. 50, pp. 159‑175, Jan. 2003.
[52] R. A. Yaffee and M. McGee, *Introduction to Time Series Analysis and Forecasting: With Applications of SAS and SPSS*. Academic Press, 2000.
[53] A. K. Jain, J. Mao, and K. M. Mohiuddin, "Artificial neural networks: a tutorial", *Computer*, vol. 29, no. 3, pp. 31‑44, 1996.
[54] K. L. Priddy and P. E. Keller, *Artificial Neural Networks: An Introduction*. SPIE Press, 2005.
[55] Vladimir N. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer, 1995.
[56] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications", *Neurocomputing*, vol. 70, no. 1‑3, pp. 489‑501, Dec. 2006.
[57] G.-B. Huang, D. H. Wang, and Y. Lan, "Extreme learning machines: a survey", *Int. J.*

Mach. Learn. & Cyber., vol. 2, no. 2, pp. 107–122, June 2011.
[58] Bishop, C. M., *Neural networks for pattern recognition*. Oxford university press, 1995.
[59] Giebel, G., Brownsword, R., Kariniotakis, G., Denhard, M., and Draxl, C., "The State of the Art in Short-Term Prediction of Wind Power: A Literature Overview", A Literature Overview 2nd Edition. Deliverable D1. 1 of ANEMOS project, 2011.
[60] Y. Zhang, J. Wang, and X. Wang, "Review on probabilistic forecasting of wind power generation", *Renewable and Sustainable Energy Reviews*, vol. 32, pp. 255–270, Apr. 2014.
[61] S. S. Soman, H. Zareipour, O. Malik, and P. Mandal, "A review of wind power and wind speed forecasting methods with different time horizons", in *North American Power Symposium (NAPS), 2010*, 2010, pp. 1–8.
[62] D. C. Park, M. A. El-Sharkawi, I. Marks, R.J., L. E. Atlas, and M. J. Damborg, "Electric load forecasting using an artificial neural network", *IEEE Transactions on Power Systems*, vol. 6, no. 2, pp. 442–449, May 1991.
[63] L. Pérez-Lombard, J. Ortiz, and C. Pout, "A review on buildings energy consumption information", *Energy and Buildings*, vol. 40, no. 3, pp. 394–398, 2008.
[64] H. Zhao and F. Magoulès, "A review on the prediction of building energy consumption", *Renewable and Sustainable Energy Reviews*, vol. 16, no. 6, pp. 3586–3592, Aug. 2012.
[65] S. J. Yao, Y. H. Song, L. Z. Zhang, and X. Y. Cheng, "Wavelet transform and neural networks for short-term electrical load forecasting", *Energy Conversion and Management*, vol. 41, no. 18, pp. 1975–1988, Dec. 2000.
[66] N. Amjady, "Short-term hourly load forecasting using time-series modeling with peak load estimation capability", *IEEE Transactions on Power Systems*, vol. 16, no. 3, pp. 498–505, Aug. 2001.
[67] L. Suganthi and A. A. Samuel, "Energy models for demand forecasting-A review", *Renewable and Sustainable Energy Reviews*, vol. 16, no. 2, pp. 1223–1240, Feb. 2012.
[68] H. Quan, D. Srinivasan, and A. Khosravi, "Short-Term Load and Wind Power Forecasting Using Neural Network-Based Prediction Intervals", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 2, pp. 303–315, Feb. 2014.
[69] P. Pinson, H. Madsen, H. A. Nielsen, G. Papaefthymiou, and B. Klöckl, "From probabilistic forecasts to statistical scenarios of short-term wind power production", *Wind Energ.*, vol. 12, no. 1, pp. 51–62, Jan. 2009.
[70] P. Pinson, "Very-short-term probabilistic forecasting of wind power with generalized logit–normal distributions", *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 61, no. 4, pp. 555–576, Aug. 2012.
[71] H. Sørensen and C. T. Ekstrom, *Introduction to Statistical Data Analysis for the Life Sciences*. CRC Press, 2010.
[72] Tom Heskes, "Practical Confidence and Prediction Intervals", in *Advances in neural information processing systems 9*, vol. 9, Cambridge, MA: MIT Press, 1997, pp. 176–182.
[73] M. W. Gardner and S. R. Dorling, "Artificial neural networks (the multilayer perceptron)-a review of applications in the atmospheric sciences", *Atmospheric Environment*, vol. 32, no. 14–15, pp. 2627–2636, Aug. 1998.
[74] R. J. Howlett and L. C. Jain, *Radial Basis Function Networks 1: Recent Developments in Theory and Applications*. Springer, 2001.
[75] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators", *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

[76] M. H. Hassoun, *Fundamentals of Artificial Neural Networks*. MIT Press, 1995.
[77] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning Internal Representations by Error Propagation", Sept. 1985.
[78] R. Rojas, *Neutral Networks: A Systematic Introduction*. Germany: Springer, 1996.
[79] P. D. R. Rojas, "The Backpropagation Algorithm", in *Neural Networks*, Springer Berlin Heidelberg, 1996, pp. 149‑182.
[80] J. D. Schaffer, D. Whitley, and L. J. Eshelman, "Combinations of genetic algorithms and neural networks: a survey of the state of the art", in *International Workshop on Combinations of Genetic Algorithms and Neural Networks, 1992., COGANN-92*, 1992, pp. 1‑37.
[81] J. Ilonen, J.-K. Kamarainen, and J. Lampinen, "Differential Evolution Training Algorithm for Feed-Forward Neural Networks", *Neural Processing Letters*, vol. 17, no. 1, pp. 93‑105, Feb. 2003.
[82] *Metaheuristic Procedures for Training Neural Networks*. .
[83] G. Zhang, B. Eddy Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks: the state of the art", *International Journal of Forecasting*, vol. 14, no. 1, pp. 35‑62, March 1998.
[84] D. M. Hawkins, "The Problem of Overfitting", *J. Chem. Inf. Comput. Sci.*, vol. 44, no. 1, pp. 1‑12, Jan. 2004.
[85] L. Prechelt, "Early Stopping - But When?", in *Neural Networks: Tricks of the Trade*, G. B. Orr and K.-R. Müller, Éd. Springer Berlin Heidelberg, 1998, pp. 55‑69.
[86] P. Refaeilzadeh, L. Tang, and H. Liu, "Cross-Validation", in *Encyclopedia of Database Systems*, L. LIU and M. T. ÖZSU, Ed. Springer US, 2009, pp. 532‑538.
[87] S. Arlot and A. Celisse, "A survey of cross-validation procedures for model selection", *Statist. Surv.*, vol. 4, pp. 40‑79, 2010.
[88] R. Setiono, "Feedforward Neural Network Construction Using Cross Validation", *Neural Computation*, vol. 13, no. 12, pp. 2865‑2877, Dec. 2001.
[89] A. Khosravi, S. Nahavandi, and D. Creighton, "A Prediction Interval-based Approach to Determine Optimal Structures of Neural Network Metamodels", *Expert Systems with Applications*, vol. 37, no. 3, pp. 2377‑2387, 2010.
[90] U. Anders and O. Korn, "Model selection in neural networks", *Neural Networks*, vol. 12, no. 2, pp. 309‑323, March 1999.
[91] T.-Y. Kwok and D.-Y. Yeung, "Efficient cross-validation for feedforward neural networks", in *IEEE International Conference on Neural Networks, 1995. Proceedings*, 1995, vol. 5, pp. 2789‑2794.
[92] H. Papadopoulos, V. Vovk, and A. Gammerman, "Regression Conformal Prediction with Nearest Neighbours", *Journal of Artificial Intelligence Research*, vol. 40, pp. 815‑840, 2011.
[93] N. Anand Shrivastava and B. Ketan Panigrahi, "Point and prediction interval estimation for electricity markets with machine learning techniques and wavelet transforms", *Neurocomputing*, vol. 118, pp. 301‑310, Oct. 2013.
[94] 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 Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 537‑547, June 2011.
[95] R. D. De Vleaux, 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.

[96] R. Dybowski and V. Gant, *Clinical Applications of Artificial Neural Networks*. Cambridge University Press, 2001.
[97] B. Efron and S. U. D. of Biostatistics, *Bootstrap Methods: Another Look at the Jackknife*. 1977.
[98] A. Khosravi, S. Nahavandi, D. Creighton, and D. Srinivasan, "Optimizing the quality of bootstrap-based prediction intervals", in *The 2011 International Joint Conference on Neural Networks (IJCNN)*, 2011, pp. 3072‑3078.
[99] E. Zio, "A study of the bootstrap method for estimating the accuracy of artificial neural networks in predicting nuclear transient processes", *IEEE Transactions on Nuclear Science*, vol. 53, no. 3, pp. 1460‑1478, June 2006.
[100] Y. Sawaragi, H. Nakayama, and T. Tanino, *Theory of Multiobjective Optimization*. Orlando, FL: Academic Press Inc., 1985.
[101] A. Schrijver, *Theory of Linear and Integer Programming*. John Wiley & Sons, 1998.
[102] P. Toth, "Optimization engineering techniques for the exact solution of NP-hard combinatorial optimization problems", *European Journal of Operational Research*, vol. 125, no. 2, pp. 222‑238, Sep. 2000.
[103] J. Branke, K. Deb, K. Miettinen, and R. Slowinski, *Multiobjective Optimization: Interactive and Evolutionary Approaches*. Springer, 2008.
[104] C. C. Coello, G. B. Lamont, and D. A. van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, 2007.
[105] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. University of Michigan Press, 1975.
[106] Lin, W. Y, Lee, W. Y, and Hong, T. P, "Adapting crossover and mutation rates in genetic algorithms", *J. Inf. Sci. Eng.*, vol. 19, no. 5, pp. 889‑903, 2003.
[107] J. Cervantes and C. R. Stephens, "Optimal Mutation Rates for Genetic Search", in *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, New York, NY, USA, 2006, pp. 1313–1320.
[108] D. T. Pham and D. Karaboga, *Intelligent Optimisation Techniques: Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks*. Springer London, Limited, 2011.
[109] R. W. Eglese, "Simulated annealing: A tool for operational research", *European Journal of Operational Research*, vol. 46, no. 3, pp. 271‑281, June 1990.
[110] D. Henderson, S. H. Jacobson, and A. W. Johnson, "The Theory and Practice of Simulated Annealing", in *Handbook of Metaheuristics*, F. Glover and G. A. Kochenberger, Ed. Springer US, 2003, pp. 287‑319.
[111] Y. Nourani and B. Andresen, "A comparison of simulated annealing cooling strategies", *J. Phys. A: Math. Gen.*, vol. 31, no. 41, pp. 8373, Oct. 1998.
[112] A. Konak, D. W. Coit, and A. E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial", *Reliability Engineering & System Safety*, vol. 91, no. 9, pp. 992‑1007, 2006.
[113] N. Srinivas and K. Deb, "Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms", *Evolutionary Computation*, vol. 2, pp. 221–248, 1994.
[114] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach", *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257‑271, 1999.
[115] Y.-F. Li, N. Pedroni, and E. Zio, "A Memetic Evolutionary Multi-Objective Optimization Method for Environmental Power Unit Commitment", *IEEE Transactions on Power Systems*, vol. 28, no. 3, pp. 2660‑2669, Aug. 2013.

[116] M. A. Abido, "Environmental/economic power dispatch using multiobjective evolutionary algorithms", *IEEE Transactions on Power Systems*, vol. 18, no. 4, pp. 1529‑1537, Nov. 2003.
[117] I. D. Lins, M. C. Moura, E. L. Droguett, E. Zio, and C. M. Jacinto, "Reliability prediction of oil wells by support vector machine with particle swarm optimization for variable selection and hyperparameter tuning", in *Proceedings of the European Safety and Reliability Conference 2011*, Troyes, France, 2011, pp. 1‑10.
[118] M. I. El-Hattab, "Scale Deposition in Surface and Subsurface Production Equipment in the Gulf of Suez", *Journal of Petroleum Technology*, vol. 37, no. 9, Sep. 1985.
[119] I. D. Lins, M. das C. Moura, E. Zio, and E. L. Droguett, "A particle swarm-optimized support vector machine for reliability prediction", *Qual. Reliab. Engng. Int.*, vol. 28, no. 2, pp. 141‑158, March 2012.
[120] R. Nyborg, "Overview of CO2 Corrosion Models for Wells and Pipelines", *CORROSION 2002*, Jan. 2002.
[121] K. Bringmann and T. Friedrich, "Don'T Be Greedy when Calculating Hypervolume Contributions", in *Proceedings of the Tenth ACM SIGEVO Workshop on Foundations of Genetic Algorithms*, New York, NY, USA, 2009, pp. 103–112.
[122] L. While, P. Hingston, L. Barone, and S. Huband, "A faster algorithm for calculating hypervolume", *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 29‑38, Feb. 2006.
[123] Canadian Weather Office, Website: http://www.weatheroffice.gc.ca/canada_e.html, [Accessed: 08 Apr. 2013].
[124] T. Baeck, D. B. Fogel, and Z. Michalewicz, *Evolutionary Computation 1: Basic Algorithms and Operators*. CRC Press, 2000.
[125] C. Igel, N. Hansen, and S. Roth, "Covariance Matrix Adaptation for Multi-objective Optimization", *Evolutionary Computation*, vol. 15, no. 1, pp. 1‑28, March 2007.
[126] G. Li and J. Shi, "On comparing three artificial neural networks for wind speed forecasting", *Applied Energy*, vol. 87, no. 7, pp. 2313‑2320, July 2010.
[127] H. Liu, H. Tian, D. Pan, and Y. Li, "Forecasting models for wind speed using wavelet, wavelet packet, time series and Artificial Neural Networks", *Applied Energy*, vol. 107, pp. 191‑208, July 2013.
[128] J. C. Helton, J. D. Johnson, and W. L. Oberkampf, "An exploration of alternative approaches to the representation of uncertainty in model predictions", *Reliability Engineering & System Safety*, vol. 85, no. 1‑3, pp. 39‑71, July 2004.
[129] H. Agarwal, J. E. Renaud, E. L. Preston, and D. Padmanabhan, "Uncertainty quantification using evidence theory in multidisciplinary design optimization", *Reliability Engineering & System Safety*, vol. 85, no. 1‑3, pp. 281‑294, 2004.
[130] H. Cheng, "Uncertainty Quantification and Uncertainty Reduction Techniques for Large-scale Simulations", Virginia Polytechnic Institute and State University, 2009.
[131] I. Hacking, *The Emergence of Probability: A Philosophical Study of Early Ideas about Probability, Induction and Statistical Inference: A Philosophical Study of Early Induction and Statistical Inference*. Cambridge University Press, 1975.
[132] G. W. Parry and P. W. Winter, "Characterization and Evaluation of Uncertainty in Probabilistic Risk Analysis", *Nucl. Saf.; (United States)*, vol. 22:1, Jan. 1981.
[133] H. J. Zimmermann, *Fuzzy Set Theory-And Its Applications*. Springer, 2001.
[134] Q. Liang and J. M. Mendel, "Interval type-2 fuzzy logic systems: theory and design", *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 5, pp. 535‑550, Oct. 2000.
[135] G. Shafer, *A Mathematical Theory of Evidence*. Princeton University Press, 1976.
[136] R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to Interval Analysis*, 1st ed.

Society for Industrial and Applied Mathematics, 2009.
[137] L. Jaulin, *Applied Interval Analysis: With Examples in Parameter and State Estimation, Robust Control and Robotics*. Springer, 2001.
[138] E. Hofer, M. Kloos, B. Krzykacz-Hausmann, J. Peschke, rg, and M. Woltereck, "An approximate epistemic uncertainty analysis approach in the presence of epistemic and aleatory uncertainties", *Reliability Engineering and System Safety*, vol. 77, no. 3, pp. 229‐238, 2002.
[139] A. M. Roque, C. Maté, J. Arroyo, and Á. Sarabia, "iMLP: Applying Multi-Layer Perceptrons to Interval-Valued Data", *Neural Process. Lett.*, vol. 25, no. 2, pp. 157–169, 2007.
[140] A. L. S. Maia, F. de A. T. de Carvalho, and T. B. Ludermir, "Forecasting models for interval-valued time series", *Neurocomputing*, vol. 71, no. 16‐18, pp. 3344‐3352, Oct. 2008.
[141] A. U. Haque, P. Mandal, M. E. Kaye, J. Meng, L. Chang, and T. Senjyu, "A new strategy for predicting short-term wind speed using soft computing models", *Renewable and Sustainable Energy Reviews*, vol. 16, no. 7, pp. 4563‐4573, Sep. 2012.
[142] H. Bouzgou and N. Benoudjit, "Multiple architecture system for wind speed prediction", *Applied Energy*, vol. 88, no. 7, pp. 2463‐2471, July 2011.
[143] F. Rossi and B. Conan-Guez, " Multi-layer Perceptron on Interval Data", in *Classification, Clustering, and Data Analysis*, P. K. Jajuga, P. A. Sokołowski, and P. H.-H. Bock, Ed. Springer Berlin Heidelberg, 2002, pp. 427‐434.
[144] R. Billinton and R. N. Allan, *Reliability Assessment of Large Electric Power Systems*. Springer, 1988.
[145] Y. Gao, R. Billinton, and R. Karki, "Composite generation and transmission system adequacy assessment considering wind energy seasonal characteristics", in *IEEE Power Energy Society General Meeting, 2009. PES '09*, 2009, pp. 1‐7.
[146] C. G. Justus, W. R. Hargraves, and A. Yalcin, "Nationwide Assessment of Potential Output from Wind-Powered Generators", *Journal of Applied Meteorology*, vol. 15, no. 7, pp. 673‐678, July 1976.
[147] P. M. Subcommittee, "IEEE Reliability Test System", *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-98, no. 6, pp. 2047‐2054, Nov. 1979.
[148] Z.-H. Zhou, J. Wu, and W. Tang, "Ensembling neural networks: Many could be better than all", *Artificial Intelligence*, vol. 137, no. 1‐2, pp. 239‐263, May 2002.
[149] Y. Zhao, J. Gao, and X. Yang, "A survey of neural network ensembles", in *International Conference on Neural Networks and Brain, 2005. ICNN B '05*, 2005, vol. 1, pp. 438‐442.
[150] A. Khosravi and S. Nahavandi, "Combined Nonparametric Prediction Intervals for Wind Power Generation", *IEEE Transactions on Sustainable Energy*, vol. 4, no. 4, pp. 849‐856, 2013.
[151] R. Maclin and D. Opitz, "Popular Ensemble Methods: An Empirical Study", *Journal of Artificial Intelligence Research*, vol. 11, pp. 169‐198, 1999.
[152] D. D. Boos, "Introduction to the Bootstrap World", *Statistical Science*, vol. 18, no. 2, pp. 168‐174, May 2003.
[153] P. Büchlmann and B. Yu, "Analyzing Bagging", *The Annals of Statistics*, vol. 30, no. 4, pp. 927‐961, Aug. 2002.
[154] A. Khosravi, S. Nahavandi, D. Creighton, and D. Srinivasan, "Optimizing the quality of bootstrap-based prediction intervals", in *The 2011 International Joint Conference on Neural Networks (IJCNN)*, 2011, pp. 3072‐3078.

[155] E. Zio, "A study of the bootstrap method for estimating the accuracy of artificial neural networks in predicting nuclear transient processes", *IEEE Transactions on Nuclear Science*, vol. 53, no. 3, pp. 1460‐1478, 2006.
[156] R. E. Schapire, "The Boosting Approach to Machine Learning: An Overview", in *Nonlinear Estimation and Classification*, D. D. Denison, M. H. Hansen, C. C. Holmes, B. Mallick, and B. Yu, Ed. Springer New York, 2003, pp. 149‐171.
[157] P. Baraldi, A. Cammi, F. Mangili, and E. Zio, "Local Fusion of an Ensemble of Models for the Reconstruction of Faulty Signals", *IEEE Transactions on Nuclear Science*, vol. 57, no. 2, pp. 793‐806, Apr. 2010.
[158] B. Efron, "Nonparametric Estimates of Standard Error: The Jackknife, the Bootstrap and Other Methods", *Biometrika*, vol. 68, no. 3, pp. 589‐599, Dec. 1981.
[159] M. Albadi and E. El-Saadany, "Comparative Study on Impacts of Power Curve Model on Capacity Factor Estimation of Pitch-Regulated Turbines", *TJER*, vol. 9, no. 2, pp. 36‐45, 2012.
[160] S. A. Akdağ and Ö. Güler, "A Comparison of Wind Turbine Power Curve Models", *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects*, vol. 33, no. 24, pp. 2257‐2263, 2011.
[161] S. A. Dudani, "The Distance-Weighted k-Nearest-Neighbor Rul", *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-6, no. 4, pp. 325‐327, Apr.1976.
[162] P. Pinson, and J. Tastu, "Discussion of "Prediction intervals for short-term wind farm generation forecasts" and "Combined nonparametric prediction intervals for wind power generation", *IEEE Transactions on Sustainable Energy*, submitted, June 2013 (in press).
[163] J. Brocker, and L. A. Smith, "Scoring probabilistic forecasts: on the importance of being proper," *Weather and Forecasting*, vol. 22, pp. 382–388, 2007.
[164] R. L. Winkler, "A decision-theoretic approach to interval estimation," *Journal of the American Statistical Association*, vol. 67, pp. 187191, 1972.

PAPER I

NSGA-II-trained neural network approach to the estimation of prediction intervals of scale deposition rate in oil & gas equipment

R. Ak, Y. F. Li, V. Vitelli, E. Zio, E. López Droguett, and C. Magno Couto Jacinto. Expert Systems with Applications, vol. 40, no. 4, pp. 1205-1212, March 2013.

NSGA-II-Trained Neural Network Approach to the Estimation of Prediction Intervals of Scale Deposition Rate in Oil & Gas Equipment

Ronay Ak[a], Yanfu Li [a], Valeria Vitelli[a,e], Enrico Zio[a,b] , Enrique López Droguett[c], Carlos Magno Couto Jacinto[d]

[a]Chair on Systems Science and the Energetic Challange, European Foundation for New Energy-Electricité de France École Centrale Paris, Grande Voie des Vignes, Châtenay-Malabry, 92290 France and SUPELEC, Plateau du Moulon - 3 Rue Joliot-Curie, Gif-Sur-Yvette, 91192 France

[b]Department of Energy, Politecnico di Milano, Via Ponzio 34/3 Milan, 20133 Italy

[c]Center for Risk Analysis and Environmental Modeling Federal University of Pernambuco, Recife, Brazil

[d]Petrobras Research Center, CENPES, Petrobras, Rio de Janeiro, Brazil

[e]Department of Biostatistics, University of Oslo, Domus Medica, Sognsvannsveien 9, 0372 Oslo, Norway (current affiliation)

ABSTRACT

Scale deposition can damage equipment in the oil & gas production industry. Hence, the reliable and accurate prediction of the scale deposition rate is critical for production availability. In this study, we consider the problem of predicting the scale deposition rate, providing an indication of the associated prediction uncertainty. We tackle the problem using an empirical modeling approach, based on experimental data. Specifically, we implement a multi-objective genetic algorithm (namely, non-dominated sorting genetic algorithm–II (NSGA-II)) to train a neural network (NN) (i.e. to find its parameters, that is its weights and biases) to provide the prediction intervals (PIs) of the scale deposition rate. The PIs are optimized both in terms of accuracy (coverage probability) and dimension (width). We perform k-fold cross-validation to guide the choice of the NN structure (i.e. the number of hidden neurons). We use hypervolume indicator metric to evaluate the Pareto fronts in the validation step. A case study is considered, with regards to a set of experimental observations: the NSGA-II-trained neural network is shown capable of providing PIs with both high coverage and small width.

Keywords: Prediction intervals, neural networks, multi-objective genetic algorithms, cross-validation, hypervolume, scale deposition rate.

1. INTRODUCTION

Degradation to failure of components may cause unplanned costs and production losses through downtime (Lins et. al., 2011). Then, prediction of component degradation to failure is important for production availability. In the context of oil & gas industry, scale deposition and corrosion continue to be serious and costly problems, because affecting the operation of the components (Moura et. al., 2011). Formation of scale on downhole equipment due to produced mineral salts is a common occurrence which is mitigated with chemical treatments or surface modifications. However, when chemical treatments are ineffective or undesirable, the scale buildup should be accounted for and predicted a priori for a given application. Accurate prediction of scale deposition and corrosion can give way to preventive maintenance. In this paper, we focus on the former process.

In oil & gas plant components, scale deposition is influenced by different factors, such as reservoir conditions (temperature, pressure), changes in pH, variation of flow rates, water cut, material structure, etc. (Nyborg, 2002). A predictive model is needed to explain the linear or nonlinear mapping between these input (explanatory) variables and the output (the scale deposition rate, hereafter called scale rate). In the literature some analytical models based on thermodynamics (Yuan, Todd, & Heriot-Waft 1991), kinetics (Larsen et. al., 2008) and hydrodynamics, or a combination of these (Stamatakis, Stubos, & Muller, 2011) have been proposed for the prediction of scale deposition in components of production wells. The output of these models is typically deterministic, with no consideration given to the variability/uncertainty caused in the output by the uncertain characteristics of the input variables. To account for this, statistical prediction methods based on learning algorithms (neural networks, NNs, support vector machines, SVMs, etc.) have been proposed (Lins et. al., 2011; Moura et. al., 2011; Cottis, Owen, & Turega, 2000).

Due to their capability of learning complex nonlinear relationships among variables from observed data, learning algorithms (e.g. NNs, SVMs, nonlinear regression models, etc.) have been successfully used in many fields of science and engineering. Lins et al. (2011) and Moura et al. (2011) proposed a SVM approach combined with particle swarm optimization (PSO) for reliability prediction in the context of oil production industry. The former work aimed at predicting scale deposition over time; the latter work aimed at predicting time

between failures (TBFs) with simultaneous input variable selection and SVM parameters' tuning by PSO. Cottis, Owen, and Turega (2000) used a conventional multi-layer perceptron NN for the prediction of the corrosion rate of steel in seawater.

In practice, the predictions provided by a learning algorithm like NNs or SVMs are affected by uncertainties (Khosravi et al., 2011a; Khosravi et al., 2011b; Khosravi, Nahavandi, & Creighton, 2010). For this reason, it is important to provide prediction intervals (PIs) of the output. A prediction interval (PI) is an interval estimate for an (unknown) future value of the target. PIs are comprised of lower and upper bounds within which the actual target is expected to lie with a predetermined probability (Khosravi et al., 2011a; Khosravi et al., 2011b; Khosravi, Nahavandi, & Creighton, 2010). There are two conflicting criteria for assessing the quality of the estimated PIs: coverage probability (CP) and prediction interval width (PIW) (Moura et. al., 2011). The prediction interval coverage probability (PICP) represents the probability that the set of estimated PI values will contain a certain percentage of the true output values. The prediction interval width (PIW) simply measures the extension of the interval as the difference of the estimated upper bound and lower bound values. To obtain representative PIs, one should aim at maximizing the CP and minimizing the PIW, simultaneously.

In this paper, we propose the adoption of a multi-objective optimization approach to the construction of PIs for NN predictions of scale rate in oil & gas components. A multi-objective genetic algorithm (namely, non-dominated sorting genetic algorithm–II (NSGA-II)) (Sirinivas & Deb, 1994) is used to train a NN, i.e. optimize its parameters (weights and biases) with respect to accuracy (max) and width (min). A demonstration of the approach and a comparison with the Lower and Upper Bound Estimation (LUBE) method of (Khosravi et al., 2011b) on a synthetic case study of literature is given in (Ak, Li, & Zio, 2012) and testing of the method on a real case study of wind speed prediction is given in (Ak et al., 2012).

Genetic Algorithms (GAs) have been successfully applied in a number of applications of engineering and related fields (Coello, Lamont, & Van Veldhuizen, 2007; Chatterjee & Bandopadhyay, 2012). The major motivation for using the GA search paradigm is due to the following three recognized advantages (Gosselin, Tye-Gingras, & Mathieu-Potvin, 2009): (i) ease of use; (ii) robustness and (iii) capability of exploring large portions of the search space

without falling into a local optimum. Further, GAs are capable of searching solutions from disjoint feasible domains and of operating on irregular functions (i.e. non-continuous and even non-differentiable); for proceeding in the search, GAs do not require the computation of gradients (Ozkol & Komurgoz, 2005).

In order to choose the NN structure (number of hidden neurons) with good generalization performance, a *k*- fold cross-validation is performed. A hypervolume indicator metric is used to compare the Pareto fronts of each cross-validation fold.

The paper is organized as follows: Section 2 briefly introduces the basic concepts of NN and PIs, and the use of NSGA-II for training a NN to estimate PIs. The complete methodology set up for scale rate PIs estimation is illustrated in Section 3. Experimental results on the real case study of scale rate prediction are given in Section 4. Finally, Section 5 concludes the paper with a critical analysis of the results obtained and some ideas of future studies.

2. MODELING FRAMEWORK

In this Section, we describe NN-based PIs estimation in the theoretical framework of multi-objective optimization, and we give the details of our implementation of NSGA-II for tackling the problem at hand.

2.1. PIs

We consider the following mathematical problem of nonlinearregression (Yang, Kavli, Carlin, Clausen, & de Groot, 2002; Zio, 2006):

$$y = f(x; w^*) + \varepsilon(x), \quad \varepsilon(x) \sim N(0, \sigma_\varepsilon^2(x)) \tag{1}$$

where x, y are the input and output vectors of the regression, respectively, and w^* represents the vector of values of the parameters of the model function f, in general nonlinear. The term $\varepsilon(x)$ is a random error with zero mean and variance $\sigma_\varepsilon^2(x) > 0$. For simplicity of illustration, in the following we assume y mono-dimensional. An estimate $\hat{w}$ of w^* is sought by minimizing the quadratic error function on a training set of input/output values $D = \{(x_n, y_n), n = 1,2, \dots, n_p\}$,

$$E(w) = \sum_{i=1}^{n_p} (\hat{y}_i - y_i)^2 \quad (2)$$

where $\hat{y}_i = f(x_i; \hat{w})$ represents the output provided by the model in correspondence of the input x_i, and n_p is the total number of training samples.

We want to quantify the uncertainty associated to the model output estimates, in terms of PIs. A PI is comprised of upper and lower bounds in which a future unknown value of the target is expected to lie with a predetermined confidence level $(1-\propto)$.The formal definition of a PI is thus (Geisser, 1993):

$$Pr(L(x) < y(x) < U(x)) = 1 - \alpha \quad (3)$$

where $L(x)$ and $U(x)$ are respectively the lower and upper bounds of the PI of the output $y(x)$ corresponding to input x; the confidence level $(1-\propto)$ refers to the expected probability that the true value of $y(x)$ lies within the PI, $\big(L(x), U(x)\big)$.

The proposed approach is to train a NN to provide in output the two bounds of the PI corresponding to a given input x. The goodness of the PI estimate attained with the NN-based model is described by two measures of quality: the PI Coverage Probability (PICP) and the Normalized Mean PI Width (NMPIW) (Khosravi et al., 2011b). Their mathematical definitions are:

$$PICP = \frac{1}{n_p} \sum_{i=1}^{n_p} c_i \quad (4)$$

where $c_i = 1$, if $y_i \in [L(x_i), U(x_i)]$ and otherwise $c_i = 0$;

$$NMPIW = \frac{1}{n_p} \sum_{i=1}^{n_p} \frac{(U(x_i) - L(x_i))}{t_{max} - t_{min}} \quad (5)$$

where t_{min} and t_{max} represent the true minimum and maximum values of the outputs (i.e., the bounds of the interval in which the true values fall), respectively.

2.2. Multi-objective Optimization

The development of the NN-based model for PIs estimation implies the optimization of PICP (maximization) and NMPIW (minimization).In other words, the NN structure (number of hidden neurons) and parameters (weights and biases) must be determined so to have the desired PICP with minimum PIW.

In all generality, a multi-objective optimization problem considers a number of objectives, equality and inequality constraints, and bounds on the decision variables. Mathematically the problem can be expressed as follows (Sawaragi, Nakayama, & Tanino, 1985):

$$\textit{Minimise/Maximise } f_m(x), \quad m = 1,2,\dots,M; \tag{6}$$

$$\textit{subject to } g_j(x) \geq 0, \quad j = 1,2,\dots,J; \tag{7}$$

$$h_k(x) = 0, \quad k = 1,2,\dots,K; \tag{8}$$

$$x_i^{(l)} \leq x_i \leq x_i^{(u)} \quad i = 1,2,\dots,I. \tag{9}$$

A solution, $x = \{x_1, x_2, \dots x_I\}$ is an I dimensional decision variable vector in the solution space R^I. The solution space is restricted by the constraints in (7) and (8), and bounds on the decision variables in (9). The final goal is to identify a set of optimal decision variable vectors $x_i^*,\ i = 1,2,\dots,n$ such that each solution included in the set cannot be regarded as better than any other with respect to all the objective functions $f_m(\cdot), m = 1,2,\dots,M$. The concepts of Pareto optimality and dominance drive the comparison among solutions: in case of a minimization problem, solution x_a dominates solution x_b $(x_a \succ x_b)$ if both following conditions are satisfied (Sawaragi et al., 1985):

$$\forall i \in \{1,2,\dots,M\}, f_i(x_a) \leq f_i(x_b) \tag{10}$$

$$\exists j \in \{1,2,\dots,M\}, f_j(x_a) < f_j(x_b) \tag{11}$$

If any of the above two conditions is violated, the solution x_a does not dominate the solution x_b, and x_b is said to be non-dominated by x_a. A solution is said to be Pareto optimal if it is not dominated by any other solution in the solution space. The set of all feasible non-dominated solutions in R^I is referred to as the *Pareto optimal set*, and for a given *Pareto*

optimal set, the corresponding values of the objective functions form the so called *Pareto optimal front* in the objective functions space.

2.3. NSGA-II optimization of a NN for PIs estimation

NSGA-II is one of the most efficient multi-objective evolutionary algorithms (Deb, Agrawal, Pratap, & Meyarivan, 2002). It generates a Pareto optimal solution set, rather than a single solution, via comparison of the qualities of different solutions by using an elitist approach (i.e., a fast non-dominated sorting and crowding-distance estimation procedure Konak, Coit, & Smith, 2006). The practical implementation of NSGA-II on our specific problem involves two phases: initialization and evolution. These can be summarized as follows (Ak et al. 2012b):

2.3.1. Initialization phase

Step 1: Split the input data set into training (D_{train}) and testing (D_{test}) subsets.

Step 2: Fix the maximum number of generations and the number of chromosomes (individuals) Nc in each population. Each chromosome codes a solution by G real-valued genes, where G is the total number of parameters (weights and biases) in the NN: thus, each chromosome represents a NN. Set the generation number $n = 1$. Initialize the first population P_n of size Nc, by randomly generating Nc chromosomes (corresponding to NNs).

Step 3: For each input vector x in the training set, compute the lower and upper bound outputs of the Nc NNs.

Step 4: Evaluate the two objectives PICP and NMPIW for the Nc NNs; then, one pair of values 1-PICP (for minimization) and NMPIW is associated to each of the Nc chromosomes in the population P_n.

Step 5: Rank the chromosomes (vectors of G values) in the population P_n by running the fast non-dominated sorting algorithm (Konak et al., 2006) with respect to the pairs of objective values, and identify the ranked non-dominated fronts $F_1, F_2, \dots, F_k$ where F_1 is the best front, F_2 is the second best front and F_k is the least good front.

Step 6: Apply to P_n a binary tournament selection based on the crowding distance (Konak et al., 2006), for generating an intermediate population S_n of size Nc.

Step 7: Apply the crossover and mutation operators to S_n, to create the offspring population Q_n of size Nc.

Step 8: Apply Step 3 onto Q_n and obtain the lower and upper bound outputs.

Step 9: Evaluate the two objectives in correspondence of the solutions in Q_n, as in Step 4.

2.3.2. Evolution phase

Step 10: If the maximum number of generations is reached, stop and return P_n. Select the first Pareto front F_1 as the optimal solution set. Otherwise, go to Step 11.

Step 11: Combine P_n and Q_n to obtain a union population $R_n = P_n \cup Q_n$.

Step 12: Apply Steps 3-5 onto R_n and obtain a sorted union population.

Step 13: Select the Nc best solutions from the sorted union to create the next parent population P_{n+1}.

Step 14: Apply Steps 6-9 onto P_{n+1} to obtain Q_{n+1}. Set $n = n + 1$; and go to Step 10.

Finally, the best front in terms of ranking of non-dominance and diversity of the individual solutions is chosen. Once the best front is chosen, testing of the trained NN with optimal weight values is performed using the data of the testing set.

3. MODEL IDENTIFICATION

In this study a systematic process is followed in order to identify the optimal NN structure (i.e., the number of hidden neurons) via cross-validation, taking into account both measures of PIs quality (i.e. coverage probability and width) and comparing the set of solutions obtained in each fold in terms of the hypervolume indicator introduced in Bringmann and Friedrich (2009). Fig. 1 shows a general scheme of this process.

3.1. *K*-fold Cross-Validation (CV)

Assessing the prediction accuracy, i.e. the generalization power, of a learning algorithm is essential for reliable prediction. In the case of NN, the structure of the model influences the learning capability. In practice, the choice of the number of network layers and neurons per layer often comes down to a compromise between the generalization error and the learning time (Ileanaˇ, Rotar, & Incze, 2004; Khosravi et al., 2010). Cross-validation (CV) is an approach to evaluate the generalization performance of the NN, and it can be used for

determining the optimal network architecture (i.e., the number of hidden neurons) (Setiono, 2001). CV is a statistical re-sampling method which uses multiple training and test subsamples (Zhang, Hu, Patuwo, & Indro 1999). Different CV techniques such as k-fold CV, leave-one-out CV, bootstrap CV, etc., have been proposed in the statistical literature (Hastie, Tibshirani, & Friedman, 2008). In the basic *k*-fold cross-validation technique, the input data set is split into a partition of *k* equally (or nearly equally) sized segments or folds. At each round of cross-validation, one among the different folds is excluded from the dataset, and only the remaining *k* - 1 folds are used for training; the excluded subset is then used for validation. The procedure is repeated until all the *k* folds have been used once for validation and *k* - 1 times for training. Hence, the advantage of this technique is that, at least in successive rounds, all samples in the input data set are used for validation, while the dimension of the training set is kept high (Setiono, 2001). Fig. 1 demonstrates an example with $k = 3$. The entire data set is divided into 3 folds and in each CV iteration, for training we use a combination of two folds out of three that can be drawn from the whole data set: {2, 3}, {1,3} and {1, 2}. Then, subsets {1}, {2}, and {3} are used for validation, respectively.

The prediction error obtained by using a CV strategy is sensitive to the specific way in which data have been split (Kwok, 1995). For small *k* values, the bias of *k*-fold cross-validation may become a problem in real-data analysis. If k = N, the so-called leave-one-out CV, the cross validation estimator is approximately unbiased for the true prediction error, but it has high variance and it is very computationally intensive for use in NN (Hastie et al., 2008). For this reason, as we shall see, in our case study we use 20-fold cross-validation in order to minimize the bias-variance trade-off while also attaining the required accuracy in feasible computation times (Fushiki, 2011; Refaeilzadeh, Tang, & Liu, 2008; Wada & Kawato, 1992).

3.2. Comparison of Pareto fronts by the hypervolume indicator

The hypervolume indicator has been widely used as a measure to compare Pareto solution sets (fronts) returned by multi-objective optimizers (Bringmann & Friedrich, 2009). Given a Pareto front, it measures the volume of the portion of the objective space dominated by the front. Therefore, different multi-objective algorithms' performances can be compared in terms of the quality of the outcomes, by detecting the dominance between their different Pareto

solution sets (Bringmann & Friedrich, 2009; While, Bradstreet, & Barone, 2012; While, Hingston, Barone, & Huband, 2006).

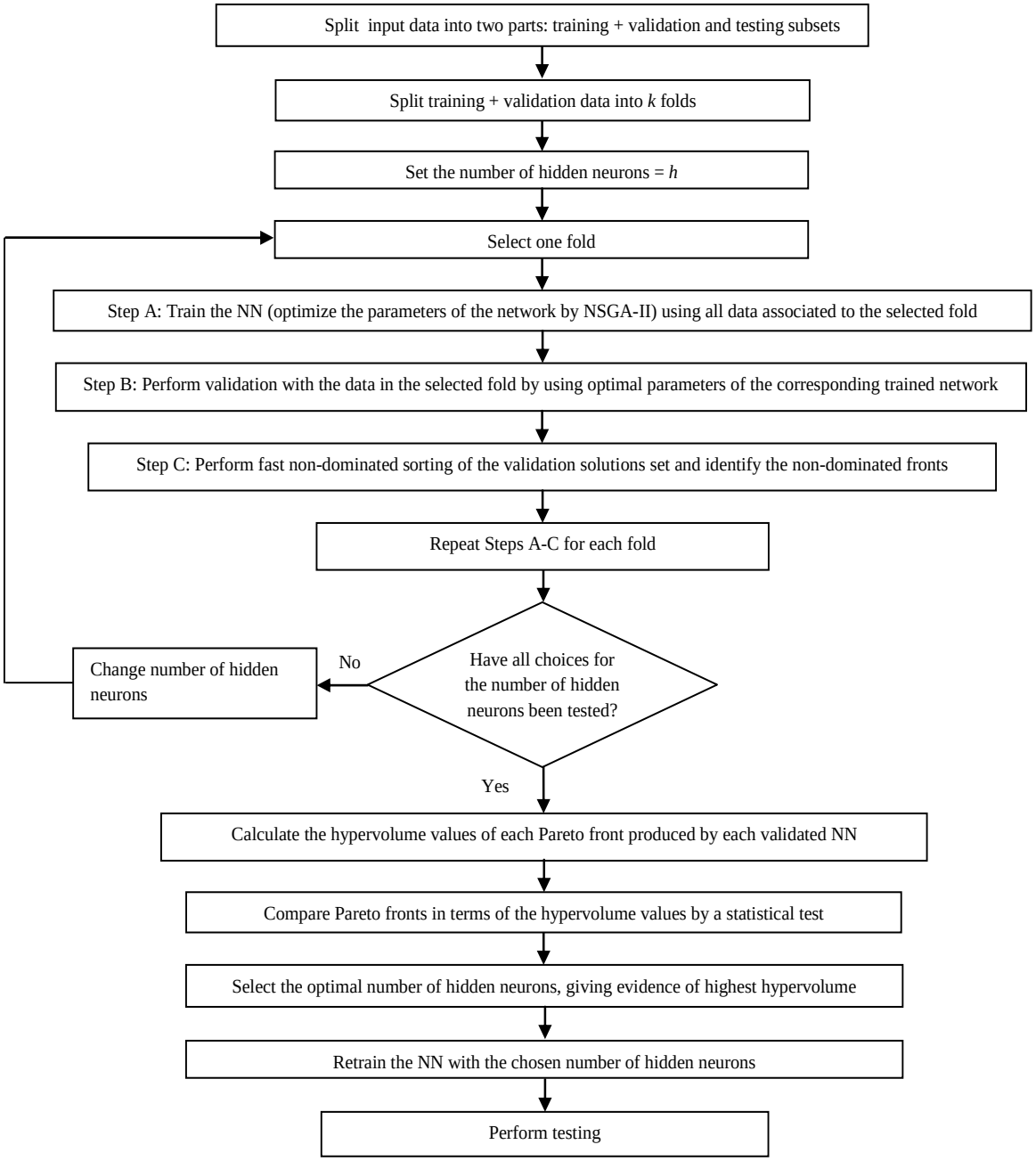


Fig. 1. Flowchart of the methodology.

In our study, we calculate the hypervolume indicator by Monte Carlo simulation (Everson, Fieldsend, & Singh, 2002). A reference point, *R*, is selected as the "worst possible" point in

the objective functions space. An estimate of the hypervolume (in percentage) is then obtained by sampling N uniformly distributed random points within the hyper-cuboids bounded by the reference point R in R^M. Then, the hypervolume indicator estimate is obtained as the percentage of the points dominated by the approximated Pareto front set P (composed by n points in R^M), i.e. in a rejection sampling fashion (Cao, 2008). If a solution set A has a greater hypervolume than a solution set B, then A is taken to be a better set of solutions than B (While et al., 2006).

For the minimization problem of our two (positive) objectives, 1-PICP and NMPIW (M = 2), we split the hypervolume computation by partitioning the objective functions space into three regions with three different reference points of same NMPIW value but different CP. A schematic representation of the objective functions space splitting into three regions and of the position of the three reference points is given in Fig. 3. We fix NMPIW and not CP, because the latter is more important than PIW for our scopes. The overall hypervolume measure is obtained as the weighted sum of the partial hypervolumes in the three regions. By the splitting into three regions, we have given the flexibility to weigh differently the hypervolume measure obtained on different ranges of CP, coherently with the relevance of the corresponding region of the objective functions space.

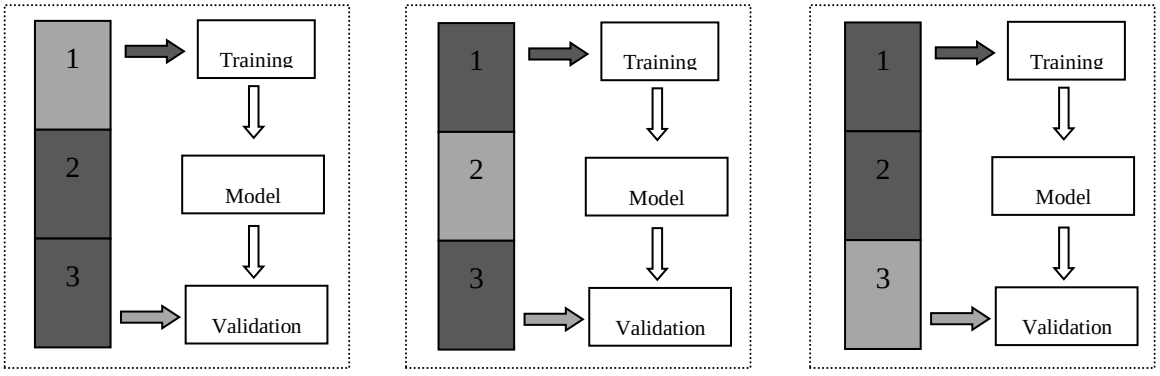


Fig. 2. Scheme of the CV procedure (Refaeilzadeh, Tang, & Liu, 2008).

4. CASE STUDY

The case study concerns the scale (deposition) rate on the metal surfaces of equipment used in offshore oil wells. The output variable *y* is the scale rate; the influencing input variables are: temperature (T) and pressure (P), water composition (W) and fluid velocity (V) near the metal surfaces. Data were obtained from experiments aimed at observing the process of deposition of the scale layer (Lins et al., 2011): if the scale layer achieves a predefined thickness, the equipment fails to properly perform its function. The total data set includes 131 observations; among these, the first 90% of the data (118 observations) are used for training and cross-validation purposes, and the rest is used for testing. All data have been normalized within the range [0.1, 0.9]. In order to perform k-fold cross-validation, the 118 training data are randomly partitioned into 20 subsamples, two of which include 5 samples while the others 6 samples. The architecture of the NN consists of one input, one hidden and one output layers. The number of input neurons is 4, one for each input variable; the number of hidden neurons is chosen via the cross-validation process described in Section 3.1; the number of output neurons is 2, one for the lower and one for the upper bound values of the PIs. As activation functions, the hyperbolic tangent function is used in the hidden layer and the logarithmic sigmoid function is used at the output layer (these choices have been found to give the best results by trial and error, although the results have not shown a strong sensitivity to them). In order to obtain an optimal NN architecture, 11 different choices for the number of hidden neurons, (5, 7–11, 13, 15, 17, 18, 20) have been explored. Hence, 220 NNs have been trained individually to obtain the results shown in Table 1.

After performing 20-fold cross-validation, we obtain 220 Pareto fronts, one for each fold and choice of the number of hidden neurons. The fronts are obtained after NSGA-II training of a NN with the training data associated to the relevant fold. In order to evaluate different neural network structures and select the optimal one, the Pareto fronts are compared in terms of their hypervolume indicators, V_1, V_2, V_3, on the partitioned objective functions space with reference points (1-PICP, NMPIW): R_1 = (0.1, 0.9), R_2 = (0.3, 0.9), R_3 = (1, 0.9) (see Fig. 3). The hypervolume value V_1 indicates the dominated space between (0, 0.9) and (0.1, 0.9), which represents the region of interest in terms of coverage probability; the hypervolume value V_2 indicates the dominated space between (0.1, 0.9) and (0.3, 0.9); the hypervolume value V_3 indicates the dominated space between (0.3, 0.9) and (1, 0.9). The reference value for NMPIW has been fixed to 0.9, because there is no NMPIW value greater than 0.9.

To compare the Pareto fronts, a total hypervolume score, Vscore, for each Pareto front is computed as weighted sum of the V_1, V_2 and V_3 values. To give higher importance to the regions of higher PICP values, in our application we have arbitrarily chosen the weight vector [w_1 w_2 w_3] to be [4/7 2/7 1/7]. Table 1 reports the total hypervolume scores computed on the validation data set, for each choice of the number of hidden neurons (different rows of the Table), and for each fold (different columns of the Table). For each fold and number of hidden neurons, the NN has been trained on the training data corresponding to the fold, and then used for prediction on the validation data included in the fold. Since the so obtained set of solutions does not necessarily form a Pareto front in the objective functions space, one step of non-dominated sorting of the solutions has been performed, before computing the corresponding total hypervolume score. The computation of V_{score} has also been done using the Pareto front resulting from the training of each NN: for the sake of brevity, we do not report the Table of total hypervolume scores obtained from training data, but the results are synthetized in the boxplots in Fig. 4.

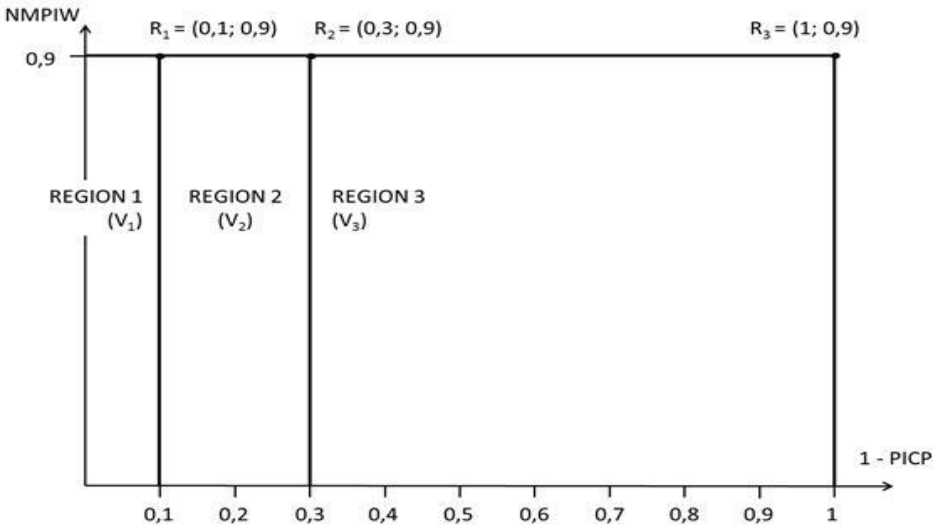


Fig. 3. A schematic representation of the splitting of the objective functions space into three regions for hypervolume computation. The overall hypervolume measure is obtained as the weighted sum of the partial hypervolumes (V_1, V_2 and V_3) computed in the three regions identified by the red lines. The three different reference points (R_1, R_2 and R_3) used in each region for partial hypervolumes computations are also indicated in the figure.

To fix the number of hidden neurons, it is natural to choose that number for which the trained NN leads to the highest total hypervolume score. However, for a given number of

hidden neurons there are 20 different NNs trained on the 20 folds, and hence 20 total hypervolume scores. Due to the variability of the data included in the folds, there is no number of hidden neurons leading to total hypervolume scores consistently superior across all folds. Figures 4 and 5 show the boxplots of the total hypervolume scores for the different numbers of hidden neurons considered, with reference to the training and validation dataset, respectively. From Fig. 4 it is evident that the choice of 10 hidden neurons is optimal, with reference to the training dataset. For confirmation, a pairwise comparison of the median of the 20 total hypevolume scores obtained by the NNs with 10 hidden neurons, with the medians obtained by the NNs with other number of hidden neurons has been performed. The pairwise comparison is conducted by a statistical test, whose aim is rejecting the null hypothesis (H0) of equality of the medians being compared; the test is based on asymptotic normality of the median and roughly equal sample sizes for the two medians being compared, and it is rather insensitive to the underlying distributions of the samples (Chambers et al., 1983; McGill, Tukey, & Larsen, 1978). By fixing the level of each test, i.e. the probability of rejecting a true H0, to 10%, nearly all comparisons allow concluding for the superiority of the total hypervolume score obtained with a choice of 10 hidden neurons.

In Fig. 5, the boxplots of the total hypervolume scores with respect to the validation dataset are shown for all numbers of hidden neurons. The choice of 10 hidden neurons confirms to be one among the best, in terms of higher values of the median and of the lower whisker. Given also the superior performance on the training dataset, the choice of 10 hidden neurons is retained.

After choosing the number of hidden neurons, the NN has been retrained using all data in the training and validation sets, for a total of 118 samples. The first (best) Pareto front found after training includes 50 non-dominated solutions and it is shown in Fig. 6. To verify a posteriori the selection of 10 hidden neurons, retraining has also been performed for other numbers of hidden neurons: the resulting V_1, V_2, V_3 and V_{score} hypervolume values are reported in Table 2. Fig. 7 shows the trend of V_1 (top) and V_{score} (bottom) values with the number of hidden neurons. From inspection of Fig. 7, one can conclude that the choice of 10 hidden neurons corresponds to the highest hypervolume values V_1 (referred to the region of interest) and V_{score}.

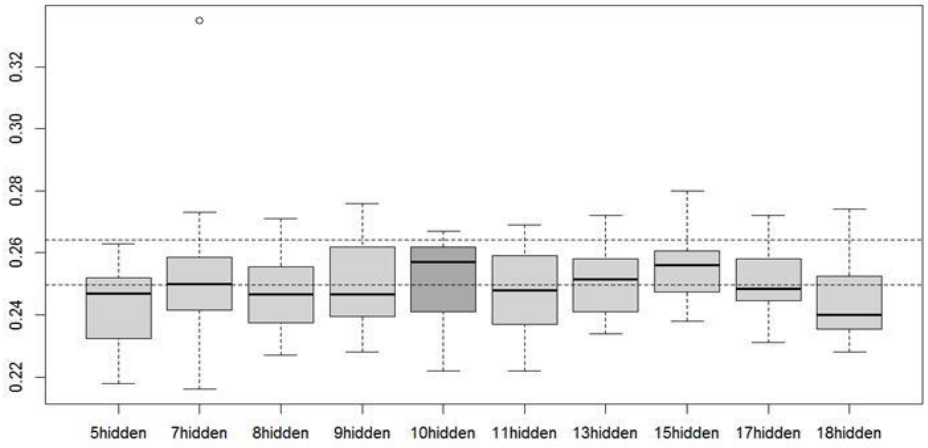


Fig. 4. Boxplots of the total hypervolume scores for different numbers of hidden neurons with respect to the training dataset. Each box extends from Q1 to Q3, where Q1 and Q3 are the first and third quartile of the dataset, respectively; the position of the median is evidenced in each box by a solid horizontal line. The upper (lower) whisker of each boxplot extends to the highest (lowest) value in the dataset smaller (greater) than Q3 + 1.5*IQR (Q1 - 1.5*IQR), where IQR = Q3 – Q1. The boxplot corresponding to 10 hidden neurons is highlighted in dark grey. The horizontal dotted lines are the limits of the 90% confidence interval for the median total hypervolume score obtained with 10 hidden neurons: the medians falling outside these limits are statistically different from the one obtained with 10 hidden neurons.

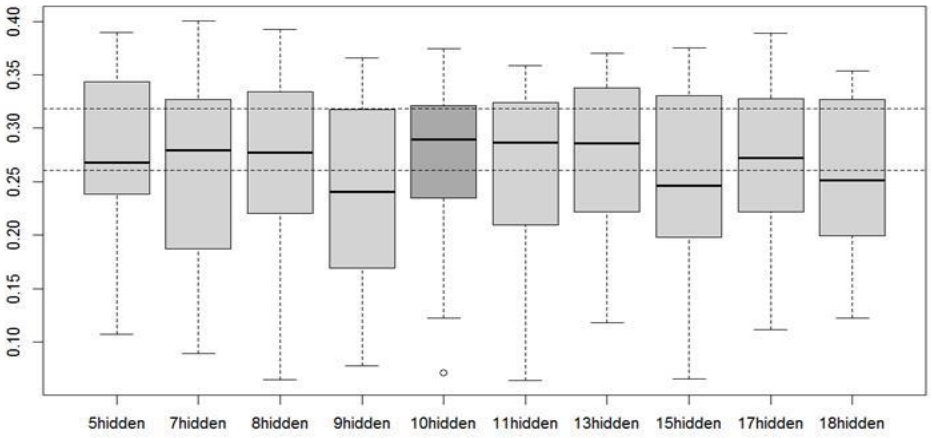


Fig. 5. Boxplots of the total hypervolume scores for different numbers of hidden neurons with respect to the validation dataset. Each box extends from Q_1 to Q_3, where Q_1 and Q_3 are the first and third quartile of the dataset, respectively; the position of the median is evidenced in each box by a solid horizontal line. The upper (lower) whisker of each boxplot extends to the highest (lowest) value in the dataset smaller (greater) than Q_3 + 1.5*IQR (Q_1 - 1.5*IQR), where IQR = Q_3 – Q_1. The boxplot corresponding to 10 hidden neurons is highlighted in dark grey. The horizontal dotted lines are the limits of the 90% confidence interval for the median total hypervolume score obtained with 10 hidden neurons: the medians falling outside these limits are statistically different from the one obtained with 10 hidden neurons.

Finally, Fig. 8 shows the prediction intervals for the scale rate values in the test dataset obtained by the trained NN with 10 hidden neurons and corresponding to a Pareto solution chosen subjectively. The solution has been chosen as the one with smallest NMPIW among those with PICP ≥ 0.9 in Fig. 6. The results on the test dataset give a coverage probability of 100% and an interval width of 0.494.

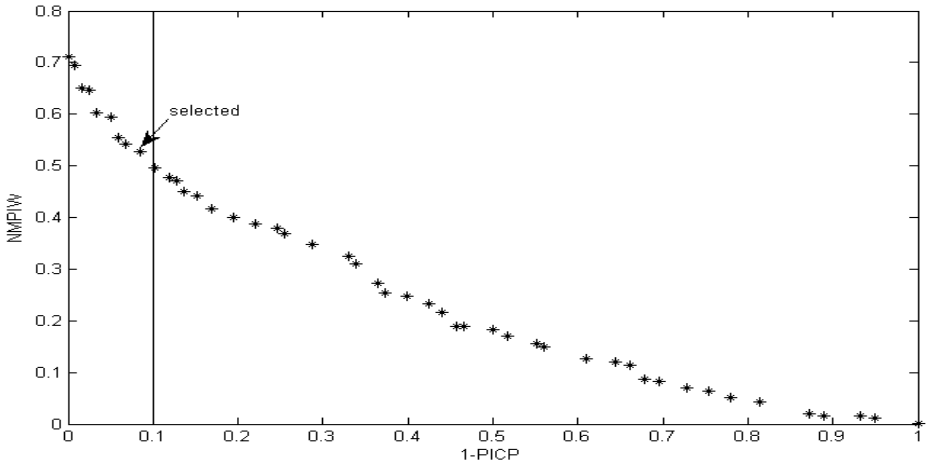


Fig. 6. The best Pareto front obtained by retraining of the NN with the optimal choice of 10 hidden neurons.

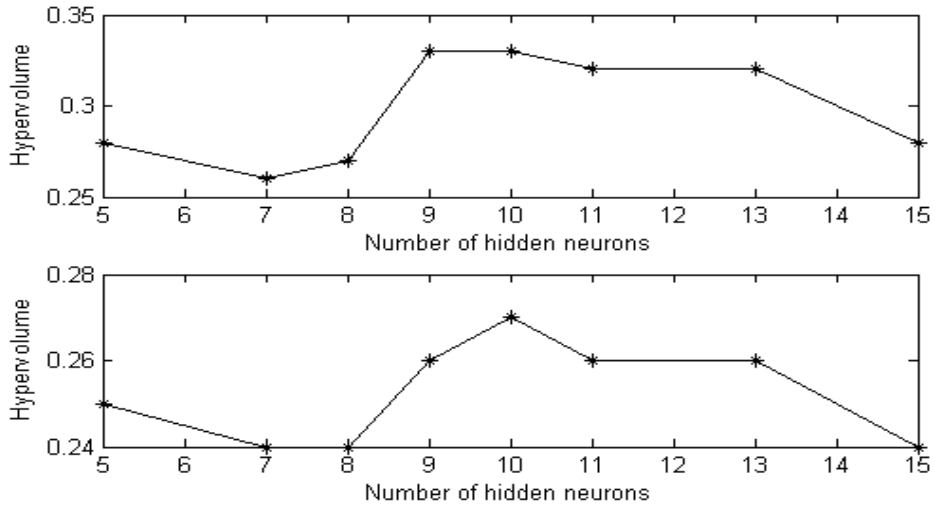


Fig. 7. Hypervolumes values V_1 and V_{score} of the fronts obtained after retraining versus the number of hidden neurons.

Table 1. The hypervolume scores of the Pareto fronts produced after validation of the NN with cross-validation procedure.

n_h	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
5	0.34	0.24	0.24	0.32	0.35	0.20	0.39	0.33	0.25	0.25	0.35	0.33	0.24	0.12	0.35	0.22	0.11	0.28	0.24	0.36
7	0.34	0.30	0.13	0.30	0.28	0.24	0.34	0.32	0.14	0.26	0.39	0.15	0.28	0.20	0.29	0.20	0.09	0.18	0.38	0.40
8	0.32	0.25	0.13	0.29	0.29	0.25	0.34	0.33	0.07	0.22	0.34	0.35	0.24	0.22	0.27	0.21	0.07	0.28	0.39	0.38
9	0.29	0.26	0.12	0.33	0.31	0.18	0.33	0.34	0.16	0.21	0.37	0.14	0.22	0.13	0.27	0.23	0.08	0.23	0.31	0.35
10	0.29	0.29	0.23	0.31	0.31	0.25	0.35	0.31	0.12	0.26	0.37	0.32	0.27	0.13	0.34	0.19	0.07	0.24	0.32	0.37
11	0.33	0.28	0.12	0.31	0.30	0.22	0.32	0.35	0.10	0.21	0.29	0.33	0.25	0.11	0.33	0.21	0.06	0.28	0.32	0.36
13	0.30	0.27	0.12	0.37	0.27	0.17	0.34	0.33	0.13	0.23	0.33	0.35	0.26	0.12	0.30	0.22	0.24	0.30	0.35	0.35
15	0.37	0.30	0.12	0.35	0.29	0.19	0.33	0.30	0.21	0.21	0.33	0.14	0.22	0.13	0.34	0.21	0.07	0.22	0.27	0.36
17	0.33	0.26	0.13	0.33	0.39	0.21	0.32	0.34	0.28	0.24	0.31	0.29	0.24	0.12	0.25	0.18	0.11	0.24	0.31	0.37
18	0.32	0.25	0.12	0.34	0.27	0.24	0.32	0.33	0.27	0.25	0.33	0.35	0.25	0.23	0.13	0.20	0.16	0.20	0.13	0.34
20	0.32	0.31	0.24	0.34	0.27	0.32	0.32	0.31	0.16	0.17	0.39	0.38	0.31	0.12	0.29	0.18	0.11	0.24	0.29	0.34

Table 2. The hypervolume values of the Pareto fronts produced after retraining of the NN with different number of hidden neurons.

	5	**7**	**8**	**9**	**10**	**11**	**13**	**15**
$\mathbf{V_1}$	0.28	0.26	0.27	0.33	0.33	0.32	0.32	0.28
$\mathbf{V_2}$	0.17	0.15	0.14	0.12	0.14	0.13	0.12	0.14
$\mathbf{V_3}$	0.28	0.31	0.30	0.28	0.27	0.28	0.28	0.29
$\mathbf{V_{score}}$	0.25	0.24	0.24	0.26	0.27	0.26	0.26	0.24

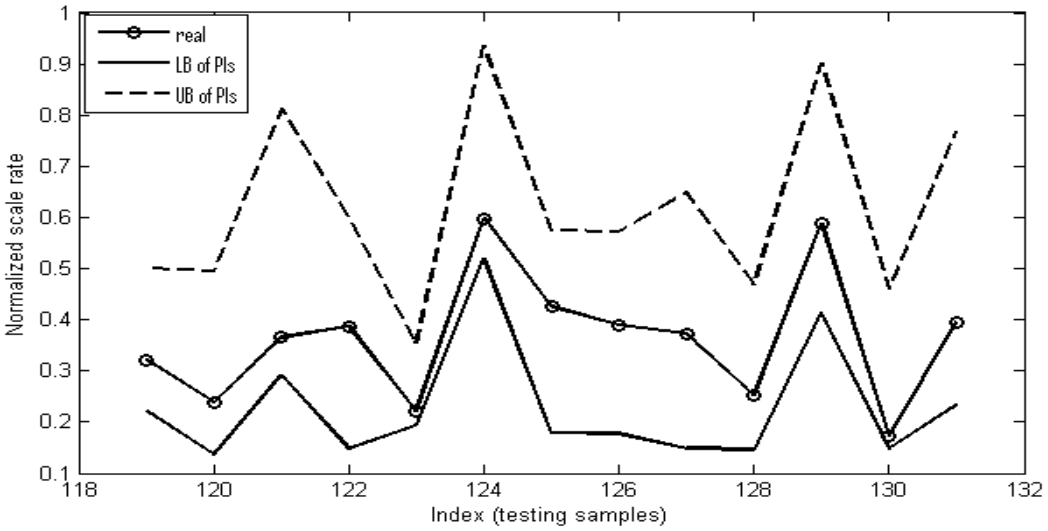


Fig. 8. The prediction intervals for scale rate on the test dataset.

5. CONCLUSION AND FUTURE WORK

A method for the estimation of PIs by NN has been proposed for scale rate prediction. The originality of the approach is the multi-objective formulation of the problem, to achieve high coverage with intervals of small width. The multi-objective framework allows considering a set of optimal solutions to select from, according to preferences and to the application purposes. Moreover, a systematic process for selecting the optimal NN structure (number of hidden neurons) for the problem at hand has been proposed, based on cross-validation analysis and on the comparison of hypervolume indicators. The approach is based on the quantitative evaluation of the superiority of performance attained with the chosen number of hidden neurons with respect to other possible choices, proved via statistical testing.

As future research, we aim at using ensemble methods to further increase the accuracy of the NN-based predictions. Moreover, we aim at exploring different measures for comparing Pareto solutions set.

REFERENCES

Ak, R., Li, Y., & Zio, E. (2012). Estimation of prediction intervals of neural network models by a multi-objective genetic algorithm. In Proceedings of the 10th international Flins conference on uncertainty modeling in knowledge engineering and decision making (Flins 2012), Istanbul, Turkey.

Ak, R., Li, Y., Vitelli, V., & Zio, E. (2012). Estimation of wind speed prediction intervals by multi- objective genetic algorithms and neural networks. In Acts of the XLVI scientific meeting of the italian statistical society, Rome, Italy.

Bringmann, K., & Friedrich, T. (2009). Don't be greedy when calculating hypervolume contributions. In Proceedings of the tenth ACM SIGEVO workshop on Foundations of genetic algorithms (Foga'09) (pp. 103–112), Orlando, USA.

Cao, Y. (2008). Computation of hypervolume indicator. Access date: June 2012. http://www.math works.com/matlabcentral/fileexchange/19651-hypervolumeindicator/content/hypervolume.m.

Chambers, J. M., Cleveland, W. S., Kleiner, B., & Tukey, P. A. (1983). Graphical methods for data analysis. New York: Champman & Hall.

Chatterjee, S., & Bandopadhyay, S. (2012). Reliability estimation using a genetic algorithm-based artificial neural network: An application to a load-haul-dump machine. Expert Systems with Applications, 39, 10943–10951.

Coello, C. A. C., Lamont, G. B., & Van Veldhuizen, D. A. (2007). Evolutionary algorithms for solving multi-objective problems (2nd edition). USA: Springer.

Cottis, R. A., Owen, G., & Turega, M. (2000). Prediction of the corrosion rate of steel in seawater using neural network methods. In NACE international corrosion 2000, Orlando, USA.

Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2002). A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6, 182–197.

Everson, R. M., J. Fieldsend, J. E., & Singh, S. (2002). Full elite sets for multi-objective optimization. In Proceedings of the fifth international conference on adaptive computing in design and manufacture (ACDM 2002) (pp. 87–100) Devon, UK.

Fushiki, T. (2011). Estimation of prediction error by using k-fold cross-validation. Statistics and Computing, 21, 137–146.

Geisser, S. (1993). Predictive inference. An introduction. Chapman & Hall.

Gosselin, L., Tye-Gingras, M., & Mathieu-Potvin, F. (2009). Review of utilization of genetic algorithms in heat transfer problems. International Journal of Heat and Mass Transfer, 52, 2169–2188.

Hastie, T., Tibshirani, R., & Friedman, J. (2008). The elements of statistical learning: Data mining, inference and prediction (2nd ed.). Springer-Verlag.

Ileana, I., Rotar, C., & Incze, A. (2004). The optimization of feed forward neural networks structure using genetic algorithms. In Proceedings of the international conference on theory and applications of mathematics and informatics (ICTAMI 2004), Thessaloniki, Greece.

Khosravi, A., Nahavandi, S., & Creighton, D. (2010). A prediction interval-based approach to determine optimal structures of neural network metamodels. Expert Systems with Applications, 37, 2377–2387.

Khosravi, A., Nahavandi, S., Creighton, D., & Atiya, A. F. (2011a). Comprehensive review of neural network-based prediction intervals and new advances. IEEE Transactions on Neural Networks, 22, 1341–1356.

Khosravi, A., Nahavandi, S., Creighton, D., & Atiya, A. F. (2011b). Lower upper bound estimation method for construction of neural network-based prediction intervals. IEEE Transactions on Neural Networks, 22, 337–346.

Konak, A., Coit, D. W., & Smith, A. E. (2006). Multi-objective optimization using genetic algorithms: A tutorial. Reliability Engineering and System Safety, 91, 992–1007.

Kwok, T-Y. (1995). Efficient cross-validation for feed-forward neural networks. In Proceedings of IEEE international conference on neural networks (pp. 2789–2794), Perth, Australia.

Larsen, T., Randhol, P., Lioliou, M., Josang, L. O., & Ostvold, T. (2008). Kinetics of CaCO3 scale formation during core flooding, In SPE international oilfield scale conference, Aberdeen, UK.

Lins, I. D., Moura, M. C., Droguett, E. L., Zio, E., Jacinto, C. M. (2011). Reliability prediction of oil wells by support vector machine with particle swarm optimization for variable selection and hyperparameter tuning, In Advances in safety, reliability and risk management proceedings of the european safety and reliability conference (Esrel 2011), Troyes, France.

McGill, R., Tukey, J. W., & Larsen, W. A. (1978). Variations of box plots. The American Statistician, 32, 12–16.

Moura, M. C., Lins, I. D., Ferreira, R. J., Droguett, E. L., & Jacinto, C. M. C. (2011). Predictive maintenance policy for oil equipment in case of scaling through support vector machines. In Proceedings of advances in safety, reliability and risk management (Esrel 2011) (pp. 503–507), Troyes, France.

Nyborg, R. (2002). Overview of CO2 corrosion models for wells and pipelines, In NACE international corrosion 2002, Denver, USA.

Ozkol, I., & Komurgoz, G. (2005). Determination of the optimum geometry of the heat exchanger body via a genetic algorithm. Numerical Heat Transfer Part A –Applications, 48, 283–296.

Refaeilzadeh, P., Tang, L., & Liu, H. (2008). Cross-validation. http://www.cse.iitb.ac.in/~tarung/smt/papersppt/ency-cross-validation.pdf., June 2012.

Sawaragi, Y., Nakayama, H., & Tanino, T. (1985). Theory of multi-objective optimization. Mathematics in Science and Engineering, 17, 1–293.

Setiono, R. (2001). Feedforward neural network construction using cross validation. MIT Press Journals Neural Computation, 13, 2865–2877.

Sirinivas, N., & Deb, K. (1994). Multi-objective optimization using non-dominated sorting in genetic algorithms. Journal of Evolutionary Computation, 2, 221–248.

Stamatakis, E., Stubos, A., & Muller, J. (2011). Scale prediction in liquid flow through porous media: A geochemical model for the simulation of CaCO3 deposition at the near-well region. Journal of Geochemical Exploration, 108, 115–125.

Wada, Y., & Kawato, M. (1992). A new information criterion combined with crossvalidation method to estimate generalization capability. Systems and Computers in Japan, 23, 955–965.

While, L., Bradstreet, L., & Barone, L. (2012). A fast way of calculating exact hypervolumes. IEEE Transactions on Evolutionary Computation, 16, 86–95.

While, L., Hingston, P., Barone, L., & Huband, S. (2006). A faster algorithm for calculating hypervolume. IEEE Transactions on Evolutionary Computation, 10, 29–38.

Yang, L., Kavli, T., Carlin, M., Clausen, S., & F. M. de Groot, P. (2002). An evaluation of confidence bound estimation methods for neural networks. International Series in Intelligent Technologies, 18, 71–84.

Yuan, M. D., Todd, A. C., & Heriot-Waft, U. (1991). Prediction of sulfate scaling tendency in oilfield operations. SPE Production Engineering, 6, 63–72.

Zhang, G., Hu, M. Y., Patuwo, B. E., & Indro, D. C. (1999). Artificial neural networks in bankruptcy prediction: General framework and cross-validation analysis. European Journal of Operational Research, 116, 16–32.

Zio, E. (2006). A study of bootstrap method for estimating the accuracy of artificial NNs in predicting nuclear transient processes. IEEE Transactions on Nuclear Science, 53, 1460–1478.

PAPER II

Multi-objective Genetic Algorithm Optimization of a Neural Network for Estimating Wind Speed Prediction Intervals

R. Ak, Y. F. Li, V. Vitelli and E. Zio. (2014), submitted to *Applied Soft Computing* (under review).

Multi-objective Genetic Algorithm Optimization of a Neural Network for Estimating Wind Speed Prediction Intervals

Ronay Ak[a], Yanfu Li [a], Valeria Vitelli[b], Enrico Zio[a,c]

[a]Chair on Systems Science and the Energetic Challenge, European Foundation for New Energy-Electricité de France

École Centrale Paris, Grande Voie des Vignes, Châtenay-Malabry, 92290 France, and SUPELEC, Plateau du Moulon - 3 Rue Joliot-Curie, Gif-Sur-Yvette, 91192 France

[b]Department of Biostatistics, University of Oslo, Domus Medica, Sognsvannsveien 9, 0372 Oslo, Norway.

[c]Department of Energy, Politecnico di Milano, Via Ponzio 34/3 Milan, 20133 Italy

ABSTRACT

In this work, the non-dominated sorting genetic algorithm–II (NSGA-II) is applied to determine the weights of a neural network trained for short-term forecasting of wind speed. More precisely, the neural network is trained to produce the lower and upper bounds of the prediction intervals of wind speed. The objectives driving the search for the optimal values of the neural network weights are the coverage of the prediction intervals (to be maximized) and the width (to be minimized). The method is proved on various wind datasets, involving also other meteorological measurements like air temperature, relative humidity and pressure. Correlation analysis is used to help variable selection for defining the most proper model inputs. The selected neural network model is, then, trained to provide in output the one-hour-ahead prediction of wind speed. The originality of the work lies in proposing a multi-objective framework for estimating wind speed prediction intervals (PIs), optimal both in terms of accuracy (coverage probability) and efficacy (width). A comparison with other single-objective optimization and prediction methods has been carried out, thus showing that the PIs produced by NSGA-II are superior to those obtained with other methods, and satisfactory in both objectives of high coverage and small width.

Keywords: wind energy, short-term wind speed forecasting, prediction intervals, neural networks, multi-objective genetic algorithm.

1. INTRODUCTION

The world energy demand continues to grow and must be satisfied while reducing the environmental impact of energy production. Fossil fuels have been predominantly used for energy production, but they have limited reserves and negative effects on the environment. Then, renewable energy sources are considered and deployed as alternative, reliable and clean forms of energy. The widespread availability of such sources (e.g. wind, sun, etc.) and the sustainability of the production process with reduced negative impacts on the environment, make power production via renewable energy sources a hot topic of research and application.

Among renewable energy sources, wind currently plays a key role in many countries. As a kind of non-polluting renewable energy, wind power has tremendous potential in commercialization and bulk power generation. According to the Half-Year Report 2011 released by The World Wind Energy Association (WWEA) [1], the worldwide wind capacity reached 215000 MW at the end of June 2011 and the global wind capacity grew of 9.3% in the previous six months, and 22.9% on an annual basis (mid-2011 compared to mid-2010). According to the 2011 European Statistics Report of the European Wind Energy Association (EWEA) [2], annual wind power installations in the EU have increased steadily over the past 17 years from 814 MW in 1996 to 9616 MW in 2011, an average annual growth rate of 15.6%. This continuous and rapid growth indicates that wind energy represents a popular solution for meeting the increasing need of electricity, respectful of the environment and sustainable.

In a power network, generated power should cover the power demand at any given time. The power output of a wind turbine is mainly dependent on the local wind speed, and the physical and operating characteristics of the turbine. Wind speed changes according to weather conditions, in time scales ranging from minutes to hours, days and years [3]; then, the wind power output also varies. Wind power variations in short-term time scales have significant effects on power system operations such as regulation, load following, balancing, unit commitment and scheduling [3-7]. Thus, accurate prediction of wind speed and its uncertainty is critical for the safe, reliable and economic operation of the power system.

Wind speed and power forecasting have been tackled in the literature by a variety of methods, including numerical weather prediction (NWP) and statistical models (these latter comprising also artificial intelligence methods like neural networks (NN) and fuzzy logic) [3-8]. Hybrid approaches combining physical and statistical models have also been proposed [9, 10]. While physical models are suited for long-term forecasting (predictions for days, weeks and months ahead), statistical and hybrid approaches are the most promising for short-term forecasting (predictions for seconds, minutes and few hours ahead) [3-10]. Among these, NN are attractive because of their capability of approximating non-linear relationships among multiple variables [4-8].

The vast majority of the existing studies on the use of NN for wind speed prediction aim at providing only point predictions. On the other hand, in practice the accuracy of the point predictions can be significantly affected by the uncertainties in the network structure and input data [11-13], and this is relevant for the design and operation conditions which follow.

Prediction intervals (PIs) can be estimated to provide a measure of the uncertainty in the prediction. PIs are comprised of lower and upper bounds within which the actual target is expected to lie with a predetermined probability [11-13]. There are two competing criteria for assessing the quality of the estimated PIs: coverage probability (CP) and prediction interval width (PIW) [12]. One seeks to simultaneously minimize PIW and maximize CP, which however are conflicting objectives.

In this work, we tackle this problem by adopting a multi-objective genetic algorithm (MOGA) framework, i.e. non-dominated sorting genetic algorithm–II (NSGA-II) [14], to determine the values of the weights of a multi-layer perceptron neural network (MLP NN) trained to estimate the bounds defining the prediction intervals. The work extends the Lower and Upper Bound Estimation (LUBE) method of [12], which combines CP and PIW in one single quality measure for optimization.

Demonstration of the approach is given on a synthetic case study, taken from literature [15] and concerning the short-term (1h ahead) wind speed prediction. Real data on wind and other meteorological parameters related to four different periods of time for the region of Regina, Saskatchewan, Canada, have been downloaded from [16]. The data are first analyzed to

identify correlations among variables and to help defining the structure of the predictive model.

In the case study analyzed, a comparison has been made between the multi-objective method proposed in this paper, the single objective simulated annealing (SOSA) method of [12], a single objective genetic algorithm (SOGA) purposely developed, a baseline method based on autoregressive integrated moving average (ARIMA) and an alternative multi-objective algorithm, the Multi-objective Covariance Matrix Adaptation Evolution Strategy (MO-CMA-ES).

In short, the main contributions of the work can be summarized as: *i*) Framing the PI problem in a multi-objective setting of finding optimal lower and upper bounds of PIs and utilizing the powerful NSGA-II algorithm to solve the problem *ii*) Analyzing the Pareto front of optimal solutions, as offering several alternatives to the decision makers (DMs) for trade-off between risk and robustness *iii*) Showing application of the method on four different datasets involving different wind speed profiles with seasonality, and *iv*) Performing a thorough comparison with both single and multi-objective algorithms.

The paper is organized as follows. Section 2 briefly introduces the basic concepts of NN and PIs, and reviews some existing methods for the construction of PIs from NN outputs. In Section 3, some basic principles of multi-objective optimization and the NSGA-II method are briefly recalled. Section 4 illustrates the use of NSGA-II for training a NN to estimate PIs. Experimental results and comparisons with other methods on the real case study of wind speed prediction are given in Section 5. Finally, Section 6 concludes the paper with a critical analysis of the results obtained and some ideas for future studies.

2. NNS AND PIS

Neural networks (NNs) are a class of nonlinear statistical models inspired by brain architecture, capable of learning complex nonlinear relationships among variables from observed data. This is done by a process of parameter tuning called “training”.
It is common to represent the task of a NN model as one of nonlinear regression of the kind [17, 18]:

$$y = f(x; w^*) + \varepsilon(x), \quad \varepsilon(x) \sim N(0, \sigma_\varepsilon^2(x)) \tag{1}$$

where x, y are the input and output vectors of the regression, respectively, and w^* represents the vector of values of the parameters of the model function f, in general nonlinear. The term $\varepsilon(x)$ is the error associated with f, and is assumed normally distributed with zero mean. For simplicity of illustration, in the following we assume y one-dimensional. An estimate $\hat{w}$ of w^* can be obtained by a training procedure aimed at minimizing the quadratic error function on a training set of input/output values $D = \{(x_n, y_n),\ n = 1,2, \dots, n_p\}$,

$$E(w) = \sum_{i=1}^{n_p} (\hat{y}_i - y_i)^2 \tag{2}$$

where $\hat{y}_i = f(x_i; \hat{w})$ represents the output provided by the NN in correspondence of the input x_i and n_p is the total number of training samples.

A single multiple-input neuron and the information processing through it are illustrated in Fig. 1. Multiple signals $x_1, x_2, \dots, x_{n_f}$ are weighed and fed onto a non-linear, e.g. sigmoid, transfer (activation) function. The multi-layer structure of such neurons (nodes) makes a NN: input signals from a previous layer produce output signals that are distributed to the neurons of the subsequent layer.

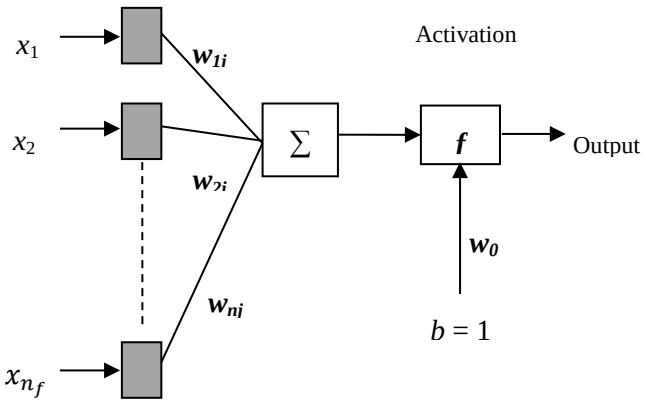


Figure 1. Multiple-input neuron [21].

The output signal H of node j of the hidden layer is given by [19-22]:

$$H_j = f_h\left(w_{0j}b_j + \sum_{i=1}^{n_f} w_{ij}x_i\right) \quad (3)$$

where $j = 1,2,\dots,h$ and h indicates the number of hidden neurons, x_i is the input signal value, $i = 1,2,\cdots,n_f$, n_f is the number of input signals, w_{ij} is the synaptic weight, $f_h()$ is the activation or transfer function and b_j is a bias factor taken as 1. After each hidden neuron computes its activation, it sends its signal to each of the neurons o_l in the output layer. Each output neuron o_l computes its output signal O_l to form the response of the NN for the input pattern received [23]:

$$O_l = f_o\left(w_{0l}b_l + \sum_{j=1}^{h} w_{jl}H_j\right) \quad l = 1,2,\dots,n_o \quad (4)$$

where n_o is the number of output neurons and f_o indicates the activation function used in the output layer.

A PI is defined by upper and lower bounds that include a future unknown value with a predetermined probability, called confidence level $(1-\alpha)$. The formal definition of a PI is the following:

$$Pr(L(x) < y(x) < U(x)) = 1 - \alpha \quad (5)$$

where $L(x)$ and $U(x)$ are respectively the lower and upper bounds of the PI of the output $y(x)$ corresponding to input x; the confidence level $(1-\alpha)$ refers to the expected probability that the true value of $y(x)$ lies within the PI ($L(x)$, $U(x)$).

The main reason for estimating the PI of the NN model output comes from the need of accounting for both the uncertainty in the model structure and the noise in the input data, which affect the point estimates. Two measures are used to evaluate the quality of the PIs: the coverage probability (CP) and the interval width (IW) [11-13]. The prediction interval coverage probability (PICP) represents the probability that the set of estimated PIs will contain the true output values, estimated as the proportion of true output values lying within

the estimated PIs; the prediction interval width (PIW) simply measures the extension of the interval as the difference of the estimated upper bound and lower bound values. These are in general conflicting measures (wider intervals give larger coverage), and in practice it is important to have narrow PIs with high coverage probability [12].

Techniques for estimating PIs for NN model outputs include the Delta, Bayesian, Mean-variance estimation (MVE) and Bootstrap techniques [11]. The Delta method is based on a Taylor expansion of the regression function. This method is capable of generating high quality PIs but at the cost of high computational time in the development stage, because it requires the calculation of a Jacobian matrix and the unbiased estimation of the noise variance [11, 24].

The Bayesian approach uses a Bayesian statistics approach to express the uncertainty of the neural network parameters in terms of probability distributions, and integrates these to obtain the posterior probability distribution of the target conditional on the observed training set [24-26]. The underpinning axiomatic mathematical foundation makes this method robust and highly repeatable. In the end, NN trained by a Bayesian-based learning technique have superior generalization power [11]. On the other hand, the computation time required is high, due to the calculation of a Hessian matrix in the development stage (a situation similar to the Delta technique).

MVE estimates the mean and the variance of the probability distribution of the target as a function of the input, given an assumed target error distribution model [27]. The proposed model is based on the maximum-likelihood formulation of a feed-forward NN [27]. Compared to the aforementioned techniques, the computational burden of this method is negligible both in the development and PI estimation stages. However, the method underestimates the variance of the data, so that the quality and generalization power of the PIs obtained are low [11, 12].

The Bootstrap method is frequently used because it is the simplest method among the ones mentioned here. It is a re-sampling technique that allows assigning measures of accuracy to statistical estimates and does not require the calculation of complex matrices and derivatives [24, 28]. The aim of the re-sampling is to produce less biased estimates of the true regression

of the targets and improve the generalization performance of the model [11]. Main disadvantages are: i) high computational time when the training sets and neural networks are large; ii) with small numbers of input patterns, the individual neural networks tend to be overly trained, leading to poor generalization performance [11, 17].

The common feature of the above PI estimation methods is that they do not take into account the widths of the intervals in the estimation process [11]. With respect to this point, Khosravi et al. [12] proposed a "Lower and Upper Bound Estimation Method (LUBE)" in which the cost function in Eq. (8) to be minimized combines two quantitative measures: PICP and PIW. The mathematical definition of the PICP and PIW measures used are [12]:

$$PICP = \frac{1}{n_p}\sum_{i=1}^{n_p} c_i \tag{6}$$

where $c_i = 1$, if $y_i \in [L(x_i), U(x_i)]$ and otherwise $c_i = 0$,

$$NMPIW = \frac{1}{n_p}\sum_{i=1}^{n_p} \frac{(U(x_i)-L(x_i))}{y_{max}-y_{min}} \tag{7}$$

where $NMPIW$ is the Normalized Mean PIW , and y_{min} and y_{max} represent the true minimum and maximum values of the targets (i.e., the bounds of the range in which the true values fall) in the training set, respectively. Normalization of the PI width by the range of targets makes it possible to objectively compare the PIs, regardless of the techniques used for their estimation or the magnitudes of the true targets.

The cost function proposed in [12] is called coverage width-based criterion (CWC):

$$CWC = NMPIW(1 + \gamma(PICP)\, e^{-\eta(PICP-\mu)}) \tag{8}$$

where η and μ are constants. The role of η is to magnify any small difference between μ and PICP. The value of μ gives the nominal confidence level, which is set to 90% in our experiments. Then, η and μ are two parameters determining how much penalty is paid by the PIs with low coverage probability. The function $\gamma(PICP)$ is equal to 1 during training, whereas in the testing of the NN it is given by the following step function:

$$\gamma(PICP) = \begin{cases} 0, & PICP \geq \mu \\ 1, & PICP < \mu \end{cases} \quad (9)$$

In Fig. 2, a symbolic sketch of the proposed three-layer (input, hidden and output) MLP NN model with two outputs is illustrated: the first output neuron provides the upper bound of the PI and the second the lower bound; by these two output neurons, the NN generates a PI interval for each input pattern.

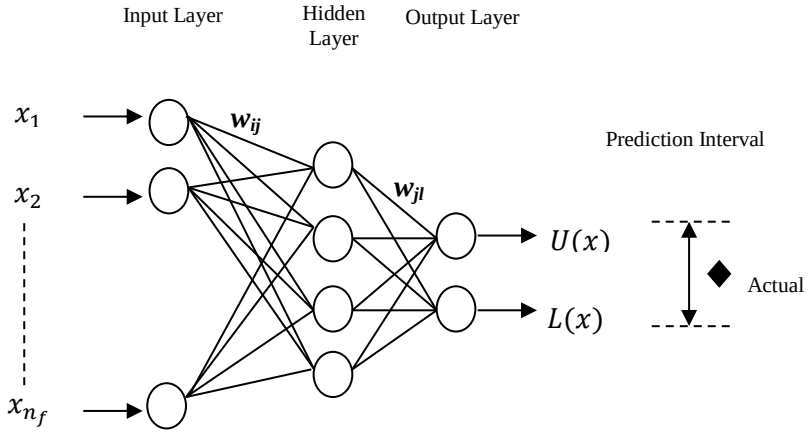


Figure 2. Architecture of a MLP NN model for estimating the lower and upper bounds of PIs [12].

Notice that both the LUBE method and our proposed method directly provide the PIs in output, while the previously described Delta, Bayesian, MVE and Bootstrap methods do so in two steps (first, point estimates calculation and then, further manipulation to get the PIs). Notice also that, while in the LUBE method the PIs are obtained by minimizing the single-objective CWC, our approach consists in estimating the PIs in a multi-objective optimization framework, as described in the following section.

3. MULTI-OBJECTIVE OPTIMIZATION BY NSGA-II

In all generality, a multi-objective optimization problem consists of a number of objectives and is associated with a number of equality and inequality constraints, and bounds on the decision variables. Mathematically the problem can be written as follows [29]:

Minimise/Maximise $f_m(x), \ m = 1,2,\dots,M;$ (10)

$$\textit{subject to} \quad g_j(x) \geq 0, \quad j = 1,2,\dots,J; \tag{11}$$

$$h_k(x) = 0, \quad k = 1,2,\dots,K; \tag{12}$$

$$x_i^{(l)} \leq x_i \leq x_i^{(u)} \quad i = 1,2,\dots,I. \tag{13}$$

A solution, $x = \{x_1, x_2, \dots x_I\}$ is an I dimensional decision variable vector in the solution space R^I. The solution space is restricted by the constraints in (11) and (12), and bounds on the decision variables in (13).

The M objective functions $f_m(x)$ must be evaluated in correspondence of each decision variable vector x in the search space. The final goal is to identify a set of optimal solutions $x^* \in R^I$ in which no solution can be regarded superior to any other with respect to all the objective functions. The comparison of solutions may be performed in terms of the concepts of Pareto optimality and dominance: in case of a minimization problem, solution x_a is regarded to dominate solution x_b ($x_a \succ x_b$) if both following conditions are satisfied [29]:

$$\forall i \in \{1,2,\dots,M\}, f_i(x_a) \leq f_i(x_b) \tag{14}$$

$$\exists j \in \{1,2,\dots,M\}, f_j(x_a) < f_j(x_b) \tag{15}$$

If any of the above two conditions is violated, the solution x_a does not dominate the solution x_b, and x_b is said to be non-dominated by x_a. The solutions that are non-dominated within the entire search space are denoted as Pareto-optimal and constitute the Pareto-optimal set; the corresponding values of the objective functions form the so called Pareto-optimal front in the objective functions space. The goal of a multi-objective optimization algorithm is to guide the search for solutions in the Pareto-optimal set, while maintaining diversity so as to cover well the Pareto-optimal front and thus allow flexibility in the final decision on the solutions to be actually implemented. The Pareto-optimal set of solutions can provide the decision makers (DMs) the flexibility to select the appropriate solutions, trading-off different preferences on the objectives. The decision makers also gain insights into the characteristics of the optimization problem before a final decision is made.

Genetic algorithm (GA) is a popular meta-heuristic approach well-suited for multi-objective problems [30]. It is a population based-search technique inspired by the principles of genetics

and natural selection. Multi-objective GAs (MOGAs) are frequently applied for solving the multi-objective optimization problems, for their ability to find nearly global optima, the ease of use and the robustness [31-33].

We resort to GA for setting the values of the weights of the NN for estimating PIs. This procedure of calibrating the NN model parameters, i.e. the weights, is called training or learning. As a classical learning (training) algorithm, back-propagation has been widely used for performing supervised learning tasks, e.g., the training of NNs [34]. However, finding the optimal weights that minimize the error requires calculating the gradient of the error function, whereas the GA does not require this calculation. The drawbacks of this algorithm have been already discussed in the literature [35, 36]. The most obvious drawback of the back-propagation algorithm is that the performance of the method decreases rapidly as the problem complexity increases [35, 36]. Moreover, the back-propagation algorithm cannot easily be adapted to multi-objective optimization prediction problems.

For the above reasons, we resort to GA for the training of the NN for PI estimation and among the several variations of MOGA in the literature, we adopt the non-dominated Sorting Genetic Algorithm-II (NSGA-II) which is one of the most efficient MOGAs as shown in various comparative studies [14, 30, 31].

4. IMPLEMENTATION OF NSGA-II FOR TRAINING A NN FOR ESTIMATING PIS

In this work, we extend the LUBE method [12] to the multi-objective formulation of the PI estimation problem. More specifically, we use NSGA-II for finding the values of the parameters of the NN which minimize the two objective functions PICP (6) and NMPIW (7) simultaneously, in Pareto optimality sense (for ease of implementation, the maximization of PICP is converted to minimization by subtracting from one, i.e. the objective of the minimization is 1-PICP).

The practical implementation of NSGA-II on our specific problem involves two phases: initialization and evolution. These can be summarized as follows:

Initialization phase:

Step 1) Partition the input data into training (D_{train}) and testing (D_{test}) subsets.

Step 2) Define the values of: the maximum number of generations, the number of chromosomes (individuals) Nc in each population, and the initial crossover and mutation probabilities.

Step 3) Set the generation number $gen = 1$. Initialize the first population P_n of size Nc, by randomly generating Nc chromosomes. Each chromosome forms a candidate solution by G real-valued genes, where G is the total number of parameters (weights) in the NN. Note that each solution corresponds to a NN.

Step 4) For each input sample x in the training set, evaluate each of the Nc chromosomes in the initial population P_n, i.e. compute the lower and upper bound outputs of each Nc chromosome with G parameters, by performing NN training. Return the values of two objectives 1-PICP and NMPIW for each of the Nc chromosomes.

Step 5) Rank the chromosomes (vectors of G values) in the population P_n by running the fast non-dominated sorting algorithm [14] with respect to the pairs of objective values, and identify the ranked non-dominated fronts $F_1, F_2, \dots, F_k$ where F_1 is the best front, F_2 is the second best front and F_k is the least good front.

Step 6) Apply to P_n a binary tournament selection based on the crowding distance [14], for generating an intermediate population S_n of size Nc.

Step 7) Apply the crossover and mutation operators to S_n, to create the offspring population Q_n of size Nc.

Step 8) Apply *Step* 4 onto Q_n and obtain the lower and upper bound outputs. Evaluate each of the Nc chromosomes in the population Q_n. Return the values of the two objectives corresponding to the solutions in Q_n.

Evolution phase:

Step 9) If the maximum number of generations is reached, stop and return P_n. Select the first Pareto front F_1 as the optimal solution set. Otherwise, go to *Step* 10.

Step 10) Combine P_n and Q_n to obtain a union population $R_n = P_n \cup Q_n$.

Step 11) Apply *Steps* 4-5 onto R_n and obtain a sorted union population.

Step 12) Select the Nc best solutions from the sorted union to create the next parent population P_{n+1}.

Step 13) Apply *Steps* 6-8 onto P_{n+1} to obtain Q_{n+1}. Set $gen = gen + 1$; and go to *Step* 9.

Finally, the best front in terms of ranking of non-dominance and diversity of the individual solutions is chosen. Once the best front is chosen, then the testing step is performed on the trained NN with optimal weight values.

The binary tournament selection, mentioned in *Step* 6, uses the crowded-comparison operator $\prec_n$ as the selection criterion [14]. For solution i in the population, it has two attributes: nondomination rank i_{rank} and crowding distance $i_{distance}$. For a solution pair, i and j, we have $i \prec_n j$ if $i_{rank} < j_{rank}$ or ($i_{rank} = j_{rank}$ and $i_{distance} > j_{distance}$). That is, if there are two solutions under consideration with different nondomination ranks, we prefer the one with the lower (better) rank. Otherwise, if both solutions have same ranking, i.e. belong to the same nondominated front, we select the solution which locates in a region with lesser number of points. For further explanations, we refer the readers to [14].

GA uses two operators to generate new solutions from existing ones: crossover (recombination) and mutation (see *Step* 7). Crossover is the key operator for the effectiveness of the GA, and it is used to create two new chromosomes, called offspring, from one selected pair of chromosomes called, parents. We have used the extended intermediate recombination method as a crossover operator [37]. Intermediate recombination can produce any point within a hypercube slightly larger than that defined by the parents [37] and it can only be applied to real-coded GAs [38]. Offspring are produced as follows:

- Randomly select the crossover point (position) $j \in \{1, \dots, S\}$.
- Randomly select the parents $P_1 = (p_1^1, \dots, p_S^1)$ and $P_2 = (p_1^2, \dots, p_S^2)$ depending on the crossover probability.
- Set $C_1 = P_1$ and $C_2 = P_2$. Then, in order to create two offspring $C_1 = (c_1^1, c_2^1, \dots, c_j^1, c_{j+1}^1, \dots, c_S^1)$ and $C_2 = (c_1^2, c_2^2, \dots, c_j^2, c_{j+1}^2, \dots, c_S^2)$, change the genes from j to S according to the following procedure:

$$\left(c_j^1, c_{j+1}^1, \dots, c_S^1\right) = \left(p_j^1, p_{j+1}^1, \dots, p_S^1\right) + r_1 * [\left(p_j^2, p_{j+1}^2 \ \dots, p_S^2\right) - \left(p_j^1, p_{j+1}^1 \ \dots, p_S^1\right) \quad (16)$$

$$\left(c_j^2, c_{j+1}^2, \dots, c_S^2\right) = \left(p_j^1, p_{j+1}^1, \dots, p_S^1\right) + r_2 * [\left(p_j^2, p_{j+1}^2 \ \dots, p_S^2\right) - \left(p_j^1, p_{j+1}^1 \ \dots, p_S^1\right)] \quad (17)$$

where $r_1$and r_2 are two values randomly (uniformly) chosen within the interval [-0.25, 1.25] [38].

Mutation involves the modification of the value of each gene of a solution with a predefined probability P_m (the mutation rate) [39]. For performing mutation, we have used a heuristic method, similar to non-uniform mutation [40], where the mutation probability (rate) P_m decreases at each generation. In our mutation method, the selected gene is replaced with a new real coded value generated by the following algorithm:

$$c_j^i = c_j^i + (rand - 0.5) * 2 \quad i = 1, \dots, Nc \; and \; j = 1, \dots, S \tag{18}$$

where i and j indicate the chromosome and the gene within the chromosome to be mutated, respectively, Nc is the number of chromosomes, and $rand$ indicates a random number value drawn from the standard uniform distribution on the open interval (0,1).

The total computational complexity of the proposed algorithm can be explained in terms of two time-demanding sub-operations: nondominated sorting and fitness evaluation. The time complexity of the nondominated sorting part is $O(MNc^2)$, where M is the number of objectives and Nc is the population size [14]. In the fitness evaluation phase, the NSGA-II is used to train a NN with n_p patterns of training; since for each individual of the population a fitness value is obtained, the process is repeated $Nc \times n_p$ times: hence, the time complexity of this phase is $O(Nc \times n_p)$. In conclusion, the computational complexity of one generation is $O\left(MN^2 + Nc \times n_p\right)$.

5. EXPERIMENTS AND RESULTS

In this Section, results of the application of the proposed method to short-term wind speed forecasting are detailed. The considered wind speed data have been measured in Regina, Saskatchewan, a region of central Canada. Wind farms in Canada are currently responsible for an energy production of 5403 MW, a capacity big enough to power over 1 million homes and equivalent to about 2% of the total electricity demand in Canada [41]. The actual situation in Saskatchewan is characterized by the presence of 4 large wind farms located throughout the

region, with a total capacity of approximately 198 MW. Aside from large wind farms, Saskatchewan residents have installed numerous smaller wind turbines (approximately 200), most of which are characterized by a power production of less than 10 KW [42].

5.1. Pre-Treatment of Input Data

The hourly wind speeds measured in four different periods in Regina, Saskatchewan, (see Fig. 4) have been downloaded from the website [16]. The first dataset comprises wind speeds for the period from 1st of February 2012 to 31st of March 2012 (winter dataset), the second from 1st of July 2012 to 29th of August 2012 (summer dataset), the third from 1st of February 2011 to 30th of June 2011 (w2011 dataset) and the last one from 1st of May 2010 to 30th of September 2010 (w2010 dataset). The four time periods have different seasonality and have been selected to represent different patterns and characteristics in the measured time series of wind speeds.

In addition to the hourly wind speed data, for the winter and summer periods the hourly measurements concerning three meteorological variables (temperature, relative humidity and air pressure) are also available for the same area.

In order to gain insights into the strength of the relationship between the input variables (the meteorological explanatory variables) and the output variable (wind speed), some statistical analyses of the data have been conducted. First, the correlation structure of the data matrix has been explored through various correlation indices and statistical tests [43]. The results obtained by computing Pearson's correlation coefficient are reported in Table 1, and they show that wind speed has in fact weak (lower than 40%) dependences on the meteorological parameters considered, both during summer and during winter. We also performed two different non-parametric tests of no correlation, based on Kendall's and Spearman's statistics [44]: both statistical tests give strong evidence of absence of correlation between wind speed and all other meteorological variables mentioned above, both during summer and winter (*p*-values all below 10%). Finally, also the correlations among meteorological variables have been explored by all these means, and they all resulted to be negligible.

Table 1. Correlation matrix for the explanatory and output variables (winter/summer).

	Temp.	Wind speed	Relative hum.	Air pres.
Temp.	1			
Wind speed	0.362 / 0.140	1		
Relative hum.	-0,506 / -0,758	-0.269 / -0.203	1	
Air pres.	-0,591 / -0,098	-0.282 / -0.333	0.129 / -0,037	1

Secondly, a Principal Component Analysis (PCA) of the meteorological variables was performed using the correlation matrix shown in Table 1 (without the output variable, wind speed). Indeed, when principal components loadings, i.e. the weights in the combinations defining the components, are interpretable and physically meaningful, a possibility is to use as explanatory variables in the model the projections of the original input variables on the principal component space [45]. In this way, the new input variables for the model are less correlated among each other, and possibly more correlated to the target. However, results of PCA (see Table 2) do not show such neat and interpretable loadings. Moreover, the new variables obtained by projection of the explanatory input variables on the first two principal components (which together explain more than 90% of the total variability in the dataset, see the last row of Table 2), do not show an increase in the correlation with the target: $\rho(wind, PC1) = 36.21\%$ in winter and 19.83% in summer; $\rho(wind, PC2) = 24.88\%$ in winter and 13.03% in summer.

All previous considerations support the conclusion that the influence of meteorological variables on the observed wind speed and their mutual dependence, are not a sufficient motivation for including them in the model as explanatory variables. This is not surprising: many models for describing wind condition or wind speed proposed in the literature rely only on past wind speed data [46], or other information concerning wind (e.g. wind direction) [47]. Hence, only historical wind speed values are selected as input variables for the ANN model aimed at providing in output the one-hour-ahead prediction of wind speed.

Table 2. Results of the PCA on meteorological variables (winter/summer).

	1st Principal component loadings	2nd Principal component loadings	3rd Principal component loadings
Temp.	0.677/0.708	--/--	0.735/-0.704
Relative hum.	-0.49/-0.703	-0.762/-0.128	0.423/-0.699
Air pres.	-0.549/--	0.647/0.991	0.53/-0.124
Proportion of explained variance	0.615/0.587	0.291/0.336	0.094/0.076
Cumulative proportion of explained variance	0.615/0.587	0.906 /0.924	1/1

The last choice concerning the model inputs for the NN model is the number of the past wind speed values to consider. First, the analysis of the empirical Autocorrelation Function (ACF; Fig. 5, top, left for winter and right for summer) shows a non-negligible correlation of the wind speed time series, also for high values of the lag. Typically in time series analysis such a consideration leads to the fitting of an autoregressive model, which explains the current value of the target via a linear combination of past values of the target itself [48]. Even if NNs are nonlinear models, this fact can be taken as an indication of the relevance of the past values of the wind ($W_{t-1}, \dots, W_{t-k}$) to explain the current wind speed (W_t). The empirical Partial Autocorrelation Function (PACF; Fig. 5, bottom, left for winter and right for summer) is instead commonly used in time series analysis for model identification, i.e. for the choice of *k* [48]: specifically, PACF at lag *j* is the autocorrelation between W_t and W_{t-j} that is not accounted for by lags 1 through *j*-1, and in autoregressive models of order *k* the PACF is zero at lag *k* + 1 or greater. We thus look for the point on the plot where the PACF essentially becomes zero, and detect the lags at which PACF is not significantly different from zero by a 95% Confidence Interval (CI), whose limits are at $\pm {}^{z_{0.975}}/_{\sqrt{n}}$, where *n* is the dimension of the dataset. The CI limits correspond to the dotted lines in Fig. 5 (bottom): we can see that W_{t-1} and W_{t-2} are highly correlated to W_t, and hence should be used in the prediction, both for the winter and the summer season; indeed, for the winter time series, also W_{t-3} is significantly related to W_t, and should thus be used in the prediction model. In synthesis,

historical wind speed values W_{t-1}, W_{t-2} and W_{t-3} are selected as input variables for predicting W_t in output for the winter season, while during summer only W_{t-1} and W_{t-2} are selected as inputs. Similar to winter and summer datasets, we have also performed ACF and PACF analysis for w2011 and w2010 datasets. In accordance to the results of these analyses, we have selected $W_{t-1,} \dots, W_{t-4}$ and $W_{t-1,} \dots, W_{t-5}$ previous time steps for w2011 and w2010 datasets, respectively, as inputs to NN.

We underline that, given the relatively higher 36.21% value of Table 1 for the correlation between temperature and wind speed observed for the winter dataset, we have also carried out the NN PI estimation by considering also the temperature as an input variable. For the temperature inputs, we have used the same number of previous time steps as for the wind speed variable. The optimal Pareto front obtained is shown in Fig. 3, together with that obtained using the wind speed variable only. It is seen that adding the temperature as input variable did not improve the optimality of the training front, i.e. the Pareto front obtained with only wind speed and the one obtained with both wind speed and temperature are almost the same (see Fig. 3). On the contrary, adding temperature as an input increases the computation time as the number of input neurons increases.

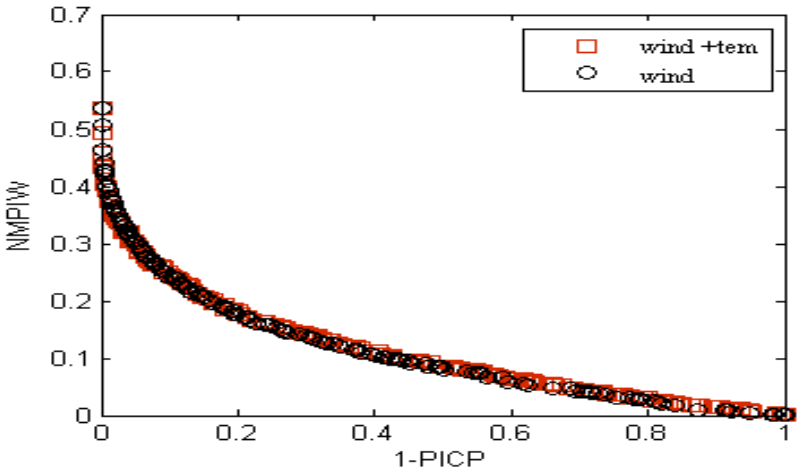


Figure 3. Pareto-fronts obtained by only wind speed input variable (marked as circle), and wind speed and temperature variables (marked as square) on winter dataset.

5.2. NN Training and Testing Results

The first dataset (winter period) includes 1437 samples. The second dataset (summer period) comprises 1438 samples. The third and fourth datasets, which are referred as w2011, w2010

include 3596 and 3667 samples, respectively. For all the datasets, the first 80% of the samples are used for training and the rest for testing. Fig. 4 shows the profile of the normalized four datasets used in this study. All data have been normalized within the range [0.1, 0.9]. The volatile character of the wind speed is clearly observable in all the wind speed profiles.

The architecture of the NN consists of one input, one hidden and one output layers. The number of input neurons is set to 2, 3, 4 and 5 for winter, summer, w2011 and w2010 datasets, respectively, depending on the inputs selected according to the correlation analysis. The number of hidden neurons is set to 10 after a trial-and-error process; the number of output neurons is 2, one for the lower and one for the upper bound values of the PIs. As activation functions, the hyperbolic tangent function is used in the hidden layer and the logarithmic sigmoid function is used at the output layer (these choices have been found to give the best results by trial and error, although the results have not shown a strong sensitivity to them).

For the comparison with other training algorithms, e.g. SOSA, SOGA and MOGA, dedicated parameter tuning has been performed. For SOGA and MOGA, the initial crossover and mutation probabilities have been tuned: crossover probability has been changed from 0.4 to 1 with step size of 0.2; for mutation probability, two alternative values, 0.06 and 1, have been considered. The results show that the performance with initial mutation probability of 1 is worse than with initial mutation probability of 0.06; on the contrary, tuning the crossover probability did not make significant difference in the obtained results. For SOSA, the initial temperature has been tried with values of 5, 200 and 500: it turns out that the SOSA with initial temperature of 200 gives better performance.

Table 3 contains the parameters of the SOSA, SOGA and NSGA-II for the NN training. "MaxGen" indicates the maximum number of generations which is used as a termination condition, and Nc indicates the total number of individuals per population. P_c indicates the crossover probability and is fixed during the run. P_{m_int} is the initial mutation probability and it decreases at each iteration (generation) by the formula:

$$P_{m\_int} \times e^{\left(-\frac{gen}{MaxGen}\right)} \tag{19}$$

Table 3. NSGA-II, SOGA and SOSA parameters used in the experiments

Parameter	**Numerical value**
MaxGen	300
Nc	50
P_{m_int}	0.06
P_c	0.8
μ	0.9
η	50
T_{init}	200
T_{min}	10^{-50}
CWC_{int}	10^{80}
Geometric cooling schedule of SA	$T_{k+1} = T_k * 0.95$

To account for the inherent randomness of NSGA-II, twenty different runs have been performed and an overall best non-dominated Pareto front has been obtained from the twenty individual fronts. To construct such front, the first (best) front of each of twenty runs is collected and the resulting set of solutions is subjected to the fast non-dominated sorting algorithm [14] with respect to the two objective functions values. Then, the ranked non-dominated fronts $F_1, F_2, \dots, F_k$ are identified, where F_1 is the best front, F_2 is the second best front and F_k is the worst front. Solutions in the first (best) front F_1 are then retained as overall best front solutions. Fig. 6 illustrates the overall best front solutions obtained with this procedure from the 20 NSGA-II runs both for winter and w2011 periods.

Given the overall best Pareto set of optimal solutions (i.e. optimal NN weights) one has to pick one (i.e. one trained NN) for use. Two different selection procedures are here employed for choosing a solution, with reference to the Pareto-optimal front of Fig. 6. First, a solution which results in the smallest CWC (see [12] and Eq. 8) is chosen. As a second procedure, the "min-max" method has been used [49]. Min-max method is one of the quantitative criteria performed to select a single best compromise Pareto solution [50]. Let us consider a point f with (f_1, f_2) on the two dimensional Pareto front (see Fig. 3). For each point f, we calculate the relative deviations $z_1 = \left(f_1 - f_1^{min}\right)/f_1^{min}$ and $z_2 = \left(f_2 - f_2^{min}\right)/f_2^{min}$, where f_1^{min} and f_2^{min} are the minimum values of the first and second objective functions on the Pareto front,

respectively. The min-max method amounts to finding the location of f where the maximum relative deviation is minimized. In other words, we seek for the best compromise solution f^* corresponding to the $min[max\{z_1, z_2\}]$ [49, 50]. Note that this method gives a solution that is representative of the center of the Pareto front [49].

Table 4 reports the PICP and NMPIW values of the Pareto front solutions both for the training and testing, selected according to those two different selection methods, i.e. min-max and smallest CWC. The solutions are also marked on the Pareto fronts of winter and w2011 datasets in Fig. 6.

It is observed that the min-max method selects a solution indeed located towards the center of the Pareto-front (see Fig. 6), whereas the smallest CWC selection method gives a solution which has coverage probability (CP) greater than 90%, i.e. the nominal confidence level, with larger interval size (see Table 4). Given the critical importance of the accuracy of the estimated PIs in decision making, interval size should be less influential than CP at least as long as the nominal confidence level is reached [12]. Ideally, solutions giving a CP equal or bigger than the nominal confidence level should have relatively higher credit. Hence, the smallest CWC selection procedure, which is meeting these requirements, is preferable for applied practice.

The optimal values of the NN parameters (weights) obtained in training are used for testing on the last 287 and 719 measurements of the wind speed winter and w2011 datasets, respectively. Figs. 7 and 8 show the prediction intervals for the testing sets of winter and w2011, respectively, estimated by the trained NN corresponding to the Pareto solution resulting in the smallest CWC value. For the sake of visibility, we have plotted first 300 samples of the w2011 testing dataset and estimated PIs (see Fig. 8). The results give a coverage probability of 84% and an average interval width of 0.277 for the winter period, and a coverage probability of 91.4% and an average interval width of 0.265 for the w2011 period (see Table 4).

Table 4. Solutions chosen from the overall Pareto optimal fronts obtained after NN training.

Dataset				**Winter**				**Summer**
	Training		Testing		Training		Testing	
Methods	PICP (%)	NMPIW	PICP (%)	NMPIW	PICP (%)	NMPIW	PICP (%)	NMPIW
Smallest CWC	93.6	0.276	84.0	0.277	94.8	0.323	91.7	0.326
Min-Max	73.0	0.145	65.5	0.144	76.4	0.177	74.0	0.175
Dataset				**w2011**				**w2010**
	Training		Testing		Training		Testing	
Methods	PICP (%)	NMPIW	PICP (%)	NMPIW	PICP (%)	NMPIW	PICP (%)	NMPIW
Smallest CWC	94.2	0.270	91.4	0.265	94.3	0.252	95.2	0.253
Min-Max	74.8	0.152	70.2	0.147	75.6	0.145	76.1	0.146

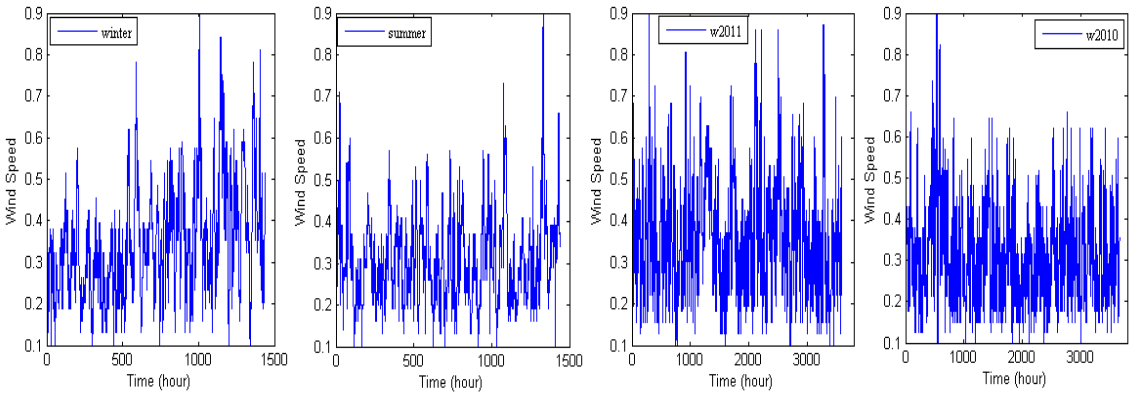
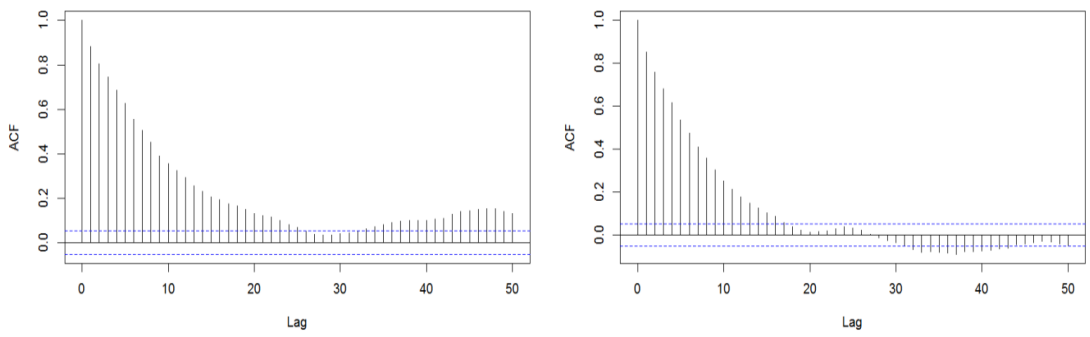


Figure 4. The four (normalized) wind speed datasets used in this study.



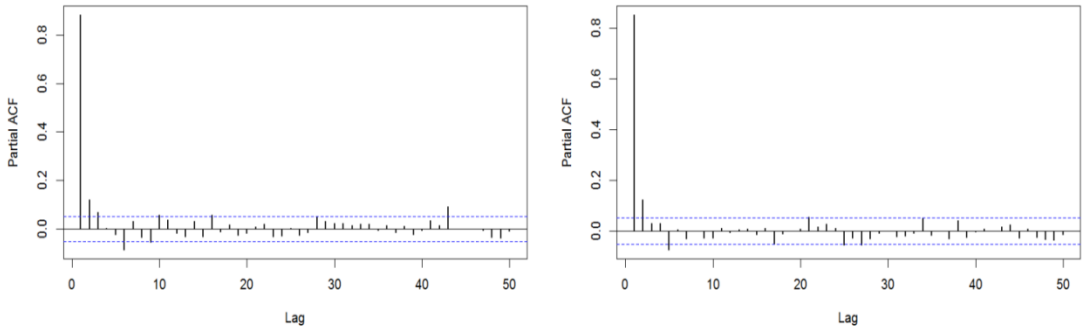


Figure 5. (up) ACF plot for the wind speed time series: winter (left) and summer (right). **(down)** PACF plot for the wind speed time series: winter (left) and summer (right). The X axis shows the hourly lag.

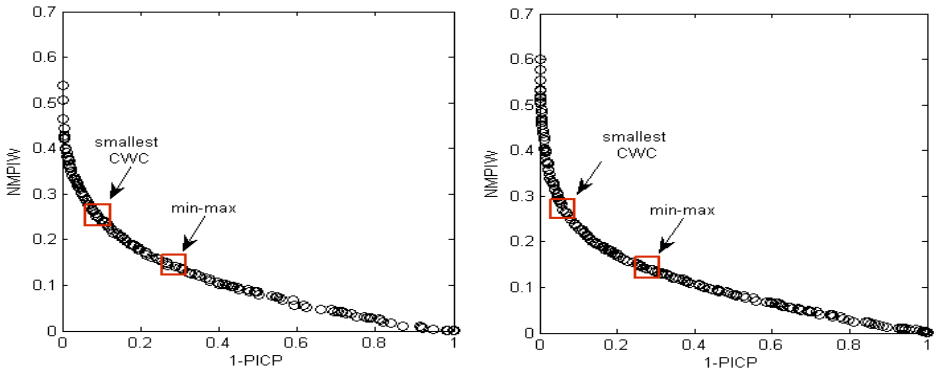


Figure 6. The overall best Pareto front obtained by training of the NN for 1h-ahead wind speed prediction: winter period (left) and w2011 period (right).

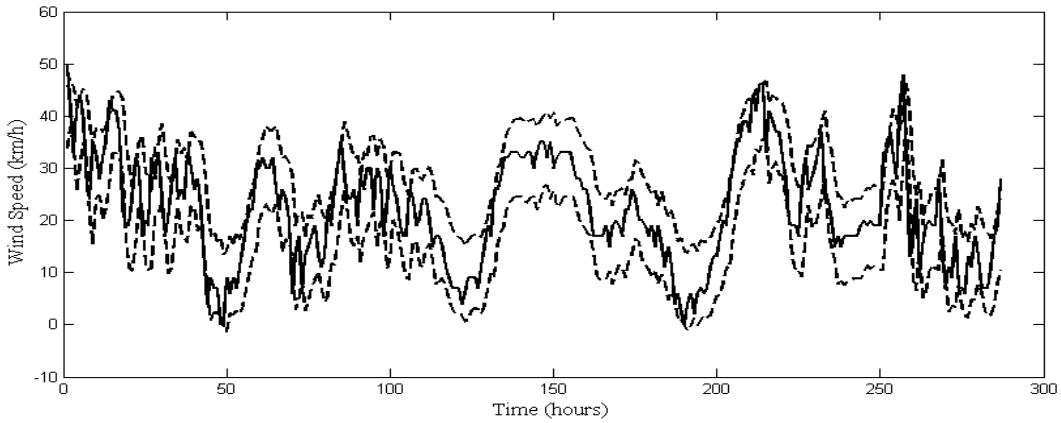


Figure 7. Estimated PIs for 1h ahead wind speed prediction on the testing set (dashed lines), and wind speed data included in the testing set (solid line) for winter dataset.

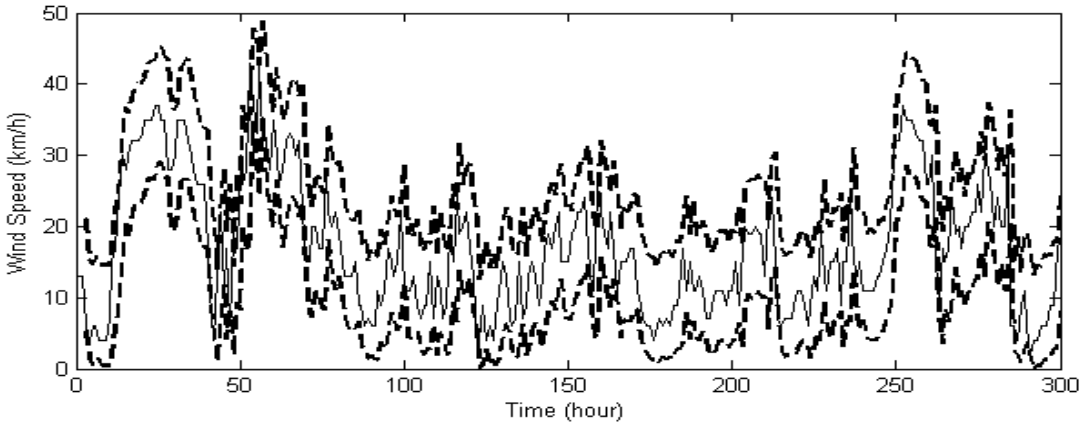


Figure 8. Estimated PIs for 1h ahead wind speed prediction on the testing set (dashed lines), and wind speed data included in the testing set (solid line) for w2011 dataset.

5.3. MOGA comparison with SOSA and SOGA

The single objective genetic algorithm (SOGA) and the single objective simulated annealing (SOSA) procedures, described in [12], have been implemented for comparison. Table 3 contains the parameters of the experiments run for SOSA and SOGA, together with the parameters for the NSGA-II implementation of the MOGA. The "T_{init}", "T_{min}", "Geometric cooling schedule" and "CWC_{int}" are the parameters of the SA optimization technique. "T_{init}" and "T_{min}" represent the starting and finishing temperatures, respectively. The finishing temperature can be used as a termination condition. The geometric cooling schedule sets the decrease of the temperature at each search iteration [12], [51, 52]. Here, we have used a cooling factor of 0.95. CWC_{int} represents the initial value of the CWC: as the temperature drops during the search, the CWC value decreases gradually but not monotonically [12].

In the MOGA and SOGA, the population size is set to 50 and the number of generations to 300, for a total number of evaluations equal to 15000. For fair comparison, SOSA is configured to have equal number of evaluations: therefore, the maximum number of iterations is set to 15000 as termination condition.

For each algorithm, the average CPU times over 20 runs for both training and testing have been recorded using MATLAB on a PC with 4 GB of RAM and a 2.53-GHz processor. Table 5 reports the recorded training CPU times on the winter dataset. The SOSA PI construction time has been recorded for 15000 iterations. The average CPU time for the construction of

testing PIs, i.e. for the online prediction of PIs, is very fast for all algorithms, being about 0.05 s. It is needless to say that computational load is dependent on the complexity of the structure of the model (e.g. number of input neurons, hidden layers, and hidden neurons), the size of the dataset and the performance of the learning algorithm.

Table 5. Descriptive Statistics of CPU times (s) of twenty MOGA, SOSA and SOGA on winter training dataset.

	Mean (s)	**Std (s)**	**Min (s)**	**Max (s)**
MOGA	258.84	9.40	245.49	285.96
SOGA	199.36	12.42	186.42	231.32
SOSA	163.29	6.98	154.52	184.28

As mentioned before, to account for the intrinsic randomness present in the SOSA, SOGA and MOGA optimization procedures, all have been run twenty times. In SOSA and SOGA, the CWC has been used as a cost function. For each of the first (best) fronts found by twenty MOGA runs, a CWC value has been a posteriori calculated by combining the individual PICP and NMPIW values. Then, for each Pareto front, the solution with smallest (best) CWC value is selected among all solutions in the front. This allows obtaining twenty best CWC values, one selected from each Pareto front. After training, we perform the testing of the trained NNs with fixed optimal parameter values (weights and biases). For each solution obtained from training, corresponding CWC values have been also calculated for testing dataset by following the same procedure explained above.

The boxplots of the testing results of the 20 runs of the three different procedures are shown in Fig. 9, where each panel corresponds to the results obtained for the winter, summer, w2011, and w2010 datasets, respectively. The aim of this Figure is to perform a comparison between the three algorithms. A boxplot is an exploratory graphic used to visualize key descriptive statistical measures of the data, such as median and quartiles, and to have an idea of the distribution of the considered variable, i.e. its location, dispersion, and its symmetry or skewness, at a glance [53], [54]. It is also used to make comparisons of these features in two or more datasets. Moreover, in Table 6, the median, mean and standard deviation statistics of

these 20 testing results have also been reported. These statistics have been calculated by considering the outliers among the twenty runs.

From the inspection of boxplots (see Fig. 9) and Table 6, we can draw the following conclusions:

1. MOGA algorithm shows more consistent testing results, i.e. better generalization capability, with respect to the CWC value if compared to SOGA and SOSA.
2. For winter and w2011 datasets, the MOGA boxplots are comparatively shorter (meaning narrower distributions) and have smaller medians than the boxplots of SOGA and SOSA. Also for the summer datasets, MOGA and SOGA seem comparable, but SOSA has higher variability. Indeed, SOSA gives highly variable CWC results, which can be interpreted as unreliable compared to the results of the two other methods. In other words, this fact indicates a higher variability for the estimates of CWC obtained using SOSA algorithm. For w2011 dataset, although SOGA boxplot looks narrower than MOGA and SOSA, its median value is slightly bigger than the one obtained by MOGA.
3. For the w2011 and w2010 datasets, all of the algorithms have quite small CWC values, with quite small standard deviations on testing set. This indicates a higher generalization power of the algorithms with respect to these datasets. However, standard deviations are relatively higher in winter datasets. This can be explained by the fact that w2010 and w2011 datasets show a similar profile of the training and testing sets, and also by the fact that they have a larger training dataset available. Thus, the algorithms are obviously able to generalize the presented patterns well and to transfer them to new unseen data. On the contrary, the testing patterns of the winter and summer datasets show relatively higher variability compared to the training patterns (see Fig. 6). This leads to the fact that for these two datasets, training solutions with CP greater than 90% have CP values lower than 90% on the testing set.
4. Although variability in the testing patterns of the winter and summer datasets are relatively higher compared to the training patterns, MOGA algorithm shows a good accuracy and generalization ability on these datasets.

It is worth saying that, in the MOGA method, the overall best non-dominated Pareto front obtained from the twenty individual fronts is considered as the ultimate Pareto-optimal front (see Fig. 6). Fig. 10 shows the testing solutions corresponding to those on the overall best

non-dominated Pareto front, obtained by MOGA, and twenty individual run results of SOGA and SOSA. These solutions belong to the w2011 dataset. In this plot, the comparison is done in terms of 1-PICP and NMPIW. For the sake of clarity and visibility, the X axis (1-PICP) has been drawn from 0 to 0.15, i.e. for the CP values from 85% to 100%.

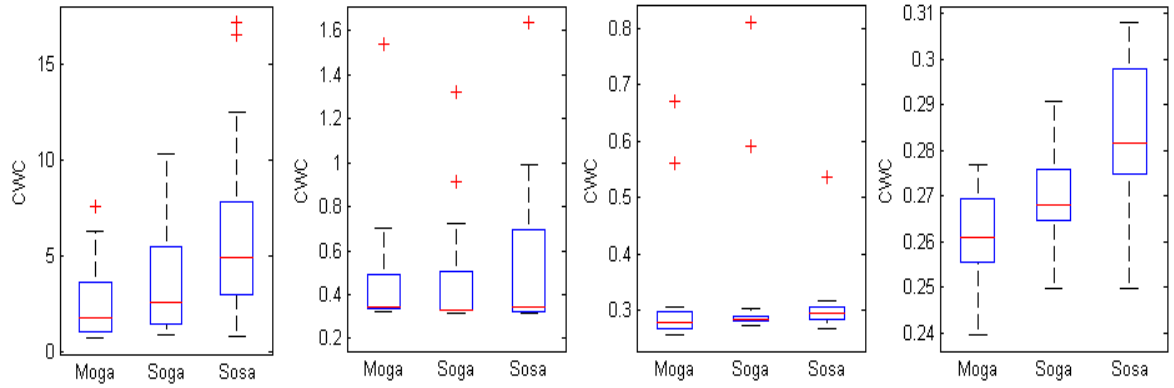


Fig. 9. Boxplot results of twenty SOSA and SOGA runs and twenty best MOGA for NN testing: winter (left), summer (middle-left), w2011 (middle-right) and w2010 (right).

Table 6. Descriptive Statistics of CWC testing results over twenty runs obtained by MOGA, SOSA and SOGA methods, respectively.

	Winter			**Summer**			**w2011**			**w2010**		
	Med	**Mean**	**Std**	**Med**	**Mean**	**Std**	**Med**	**Mean**	**Std**	**Med**	**Mean**	**Std**
MOGA	1.79	2.67	2.16	0.34	0.46	0.29	0.28	0.31	0.11	0.26	0.26	0.01
SOGA	2.57	3.66	2.86	0.33	0.46	0.27	0.29	0.33	0.13	0.27	0.27	0.01
SOSA	4.90	6.16	4.64	0.34	0.51	0.34	0.30	0.31	0.06	0.28	0.28	0.02

From inspection of Fig. 10, it can be observed that both SOGA and SOSA methods never result in coverage probabilities greater than 93% with respect to the w2011 dataset. For PIs with higher CPs (>=95%), SOGA and SOSA do not provide appropriate solutions. Besides, most of the solutions are above the MOGA testing solutions front in the solution space under consideration, this meaning that SOGA and SOSA solutions have larger interval size for the same CP compared to the MOGA ones. For exemplification, a testing solution with 92.9% CP results in 0.283, 0.287, and 0.306 NMPIW values corresponding MOGA, SOGA and SOSA, respectively. One can appreciate that MOGA gives tighter interval size than SOGA and

SOSA. Note that we obtained similar results for all datasets used in the experiments; due to space limitation, only w2011 dataset results have been shown in full.

Finally, we have analyzed the convergence of CWC along the iterations of the NN training procedure. The behavior of CWC as a function of the iterations is shown in Figs. 11 and 12 for SOSA and SOGA methods, respectively. Since the CWC takes extreme values in the first iterations of SOSA, the logarithm of CWC has been plotted in Fig.11. In the case of SOSA, the CWC decreases gradually but non-monotonically due to the structure of the simulated annealing algorithm. In order to show clearly the convergence and non-monotonicity of the SOSA method, a zoom on the behavior of CWC has been also plotted: the right plot in Fig. 11 shows the values of CWC for the last 5000 iterations. It is observed that CWC continues to decreases until it reaches the maximum number of iterations. On the contrary, from inspection of Fig. 12 it is clear that CWC decreases gradually and monotonically in the case of SOGA.

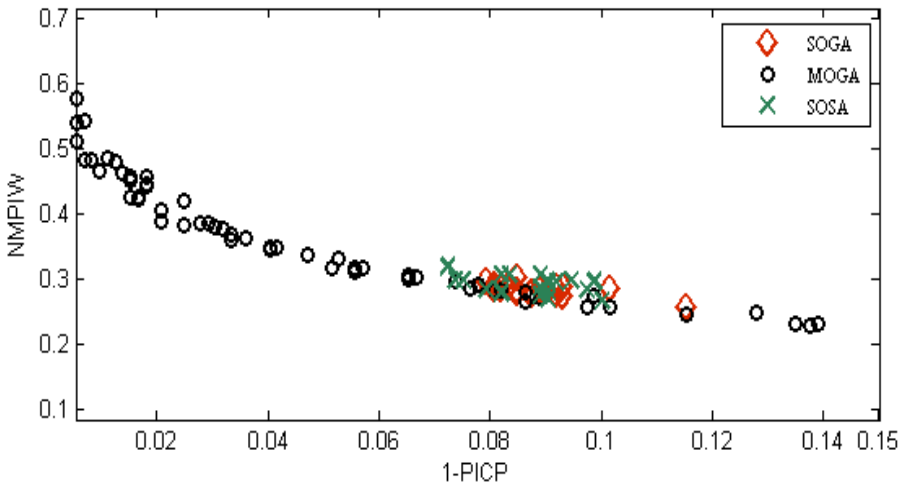


Fig. 10. 20 SOGA (marked as diamond) and SOSA results (marked as cross) versus overall best NSGA-II testing solutions (marked as circle).

Fig. 13 shows the convergence behavior of PICP and NMPIW through the iterations of the MOGA for the winter dataset. Due to space limitation, the similar plots for summer, winter2011 and winter2010 datasets are omitted. To obtain this graph, we have considered the two objectives separately (as if they were two single objectives, even if our research does not really focus on the single-objective solutions), and we have selected the extreme solutions on the front obtained at each iteration. In other words, the solution giving maximum PICP and the one giving minimum NMPIW were selected separately. The motivation behind these last

convergence plots is to show the MOGA algorithm's ability to converge, after a certain number of iterations, to the true optimum, which means respectively 100% PICP and 0 NMPIW. This happens for both the single objectives.

5.4. Comparison with MO-CMA-ES

In addition to the comparisons explained above, we have also applied multi-objective covariance matrix adaptation evolution strategy (MO-CMA-ES) as an alternative multi-objective training algorithm [55, 56]. CMA-ES is a stochastic method for non-linear, non-convex functions in continuous domain. It is regarded as one of the most powerful evolutionary algorithms for real-valued single optimization problems [56]. For details on the methodology and on the implementation procedure, we refer the readers to [55-57]. Here, we have extended the MATLAB source code of single objective CMA-ES published in [57] to the multi-objective framework. In order to have a fair counterpart algorithm to NSGA-II, we have set up the MO-CMA-ES algorithm by modifying the algorithm in Section 4 as follows: in step 3) we replace the random generation of initial population with the sampling from a multivariate normal distribution; in step 7) we replace the genetic operators by the CMA-ES updating schemes. The rest of the steps remain the same.

The parameters used in MO-CMA-ES have been assigned by trial and error as follows: the maximum number of generation and population size has been set to 300 and to 50, same as NSGA-II; the number of parents/points for recombination, i.e. the mu value, has been set equal to the population size; the sigma value has been set to 0.3 and the initial values of the decision variables, i.e. x_{mean}, have been determined with the same formula as we used in NSGA-II. The rest of the parameters are the same as in [57].

Fig. 14 shows a comparison of the optimal Pareto fronts obtained after training of the NN both by NSGA-II and MO-CMA-ES algorithms on the winter and w2011 datasets, respectively. It can be noticed that the optimal Pareto front obtained by NSGA-II is slightly better than the one obtained by MO-CMA-ES.

5.5. Comparison with ARIMA

In order to have a comparison also with a baseline method, we have also generated point predictions and a posteriori calculated PIs by autoregressive integrated moving average (*ARIMA*) model [58]. The ARIMA prediction results have been obtained by the R statistical software package [58]. The best time series model for the winter, summer, w2011, and w2010 datasets have been chosen as ARIMA (3, 0, 0), ARIMA (2, 0, 0), ARIMA (4, 0, 0), and ARIMA (5, 0, 0), respectively. Note that the parameters of the ARIMA model have been chosen not only by considering ACF and PACF results but by also complementing with a trial and error process. Eventually, we have chosen the one giving the smaller Akaike Information Criteria (AIC) value [58].

First, we have calculated point predictions according to the regression functions obtained by R. Then, we have set the confidence level to 90% to obtain PIs. After that, we have calculated both the prediction interval empirical coverage (PICP) and interval width of the estimated PIs on the testing set. The empirical coverage probabilities of the prediction intervals obtained with a confidence level of 90% are reported in Table 7 for all datasets. One can observe that although the confidence level is 90%, the PICP values are quite big. This can be explained with the large interval widths (see Table 7). As the PIWs are quite large (around 50%), they cannot provide useful information in practice, because the uncertainty level is too high to support a reliable and informed decision in typical application contexts. Indeed, for winter and summer datasets, where the test set variability is relatively higher than in w2011 and w2010 datasets, MOGA NN results in tighter interval widths for the same PICP value.

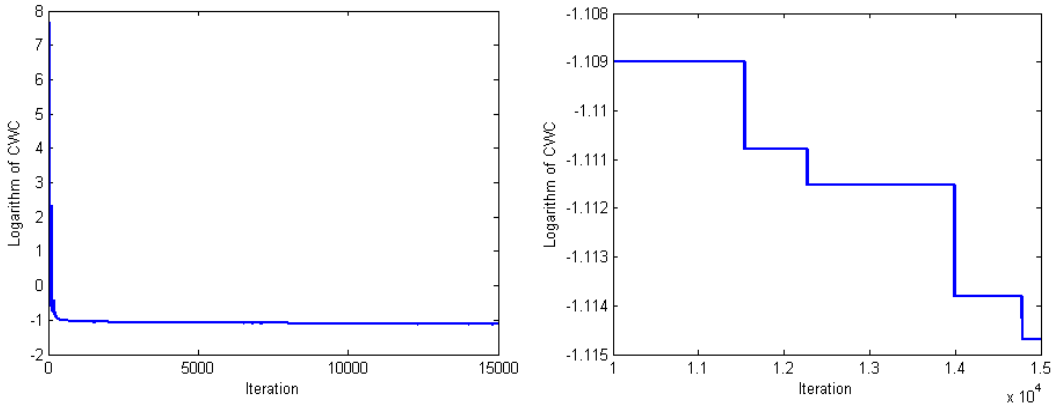


Fig. 11. Evolution of CWC during the training of NN by SOSA algorithm for the winter dataset.

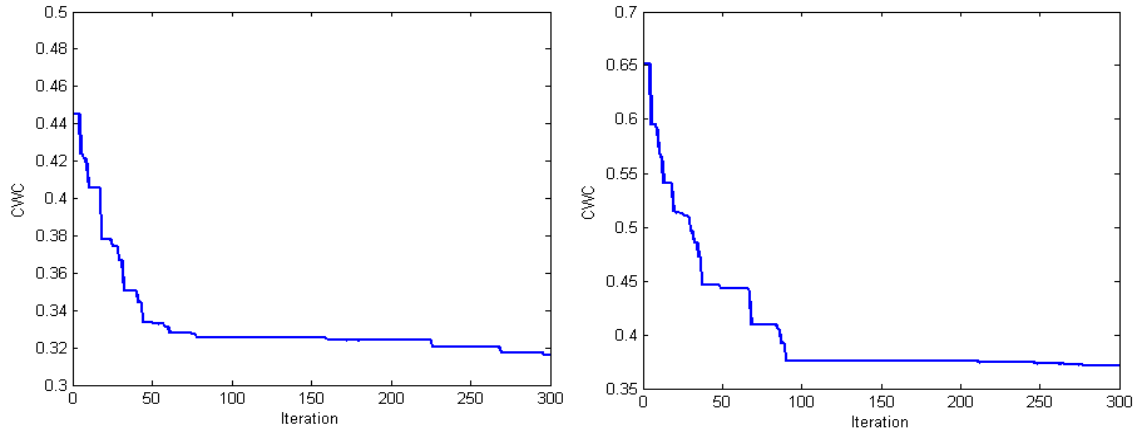


Fig. 12. Evaluation of CWC during the training of NN by SOGA algorithm for winter (left) and summer (right) datasets.

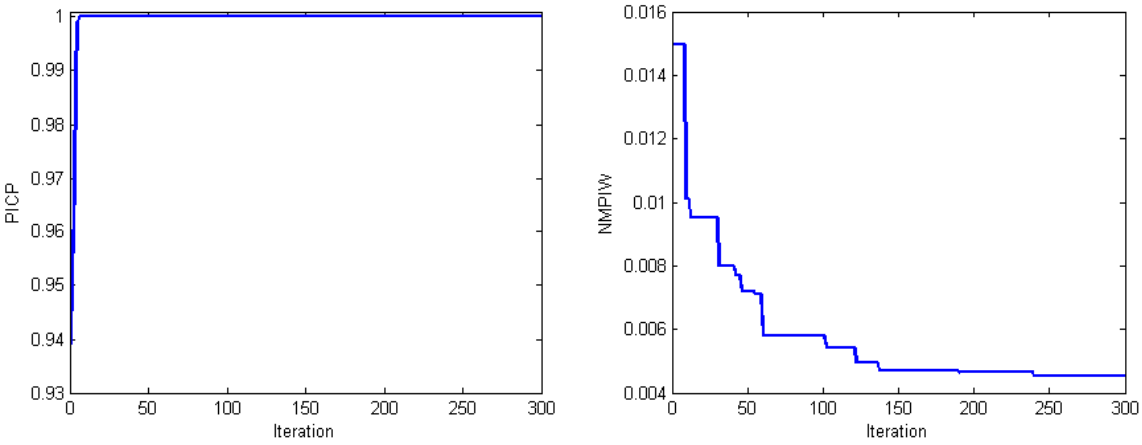


Figure 13. Evaluation of PICP and NMPIW during the training of NN by MOGA algorithm for winter period: PICP (left) and NMPIW (right).

Table 7. PICP and NMPIW values obtained by MOGA and ARIMA with respect to the four datasets.

	Winter		**Summer**		**w2011**		**w2010**	
	PICP (%)	NMPIW	PICP (%)	NMPIW	PICP (%)	NMPIW	PICP (%)	NMPIW
MOGA	98.6	0.463	98.3	0.449	99.6	0.552	99.9	0.457
ARIMA	98.6	0.514	98.3	0.525	99.6	0.500	99.9	0.447

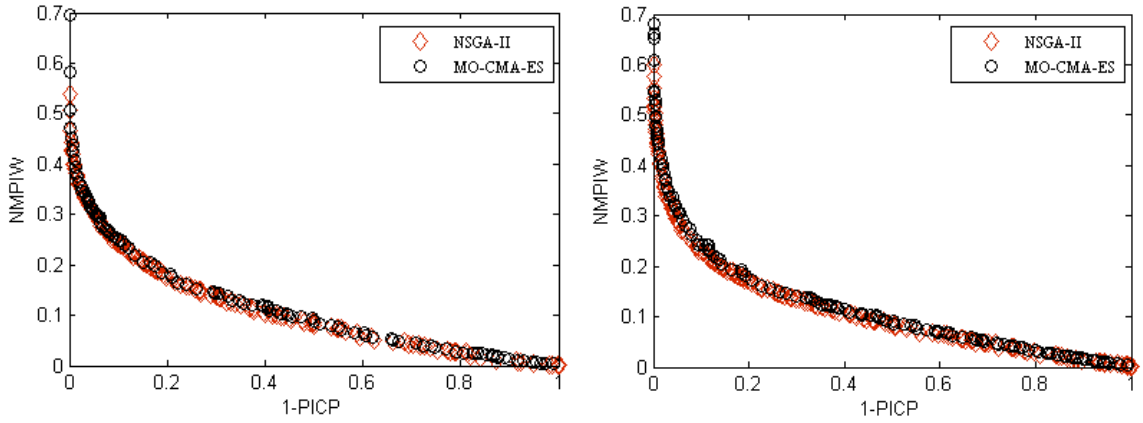


Figure 14. Pareto fronts obtained by training NN by NSGA-II and MO-CMA-ES on winter (left) and w2011 (right) datasets.

6. CONCLUSION

Wind speed prediction is a fundamental issue for wind power generation. The associated uncertainty needs to be properly quantified for reliable decision making in design and operation.

In this study, a method for the estimation of PIs by NN has been applied for short-term wind speed prediction. The wind speed data from four different time periods and related to the region of Regina, Saskatchewan, have been used to demonstrate the capabilities of the proposed method. Within an original multi-objective optimization formulation of the problem of NN training, NSGA-II is capable of estimating NN weights which are optimal in Pareto sense. To the knowledge of the authors, this is the first study proposing such multi-objective formulation for the estimation of NN-based PIs for wind speed prediction. The results obtained confirm the validity of the proposed approach. Tests and comparisons with other methods (SOGA, SOSA, MO-CMA-ES and ARIMA) have also been performed to prove the superiority of our approach.

As for future research, the use of an ensemble of different NNs will be considered to further increase the accuracy of the predictions, and the extension of the approach for prediction of wind power output will be pursued.

REFERENCES

[1] World Wind Energy Association, Half Year Report. Aug. 2011, 1-7. Online, http://www.wwindea.org/home/images/stories/publications/half_year_report_2011_wwea.pdf; Aug. 2011 [Accessed on May 2012].

[2] The European Wind Energy Association, Wind in Power 2011 European Statistics. Online, http://www.ewea.org/fileadmin/ewea_ documents/documents/publications/statistics/Stats_2011.pdf; Feb. 2012 [Accessed on May 2012].

[3] R. G. Kavasseri, K. Seetharaman, Day-ahead wind speed forecasting using f-ARIMA models, Renewable Energy. 34 (2009) 1388-1393.

[4] X. Wang, P. Guo, X. Huang, A review of wind power forecasting models, Energy Procedia. 12 (2011) 770-778.

[5] M. Lei, L. Shiyan, J. Chuanwen, L. Hongling, Z. Yan, A review on the forecasting of wind speed and generated power, Renewable and Sustainable Energy Reviews. 13 (2009) 915-920.

[6] R.S. Tarade, P. K. Katti, A comparative analysis for wind speed prediction, Proceedings of International Conference on Energy, Automation and Signal, Orissa India, Dec. 2011, pp. 556-561.

[7] A. M. Foley, P. G. Leahy, A. Marvuglia, E. J. McKeogh, Current methods and advances in forecasting of wind power generation, Renewable Energy. 37 (2012) 1-8.

[8] W. Zhang, J. Wang, J. Wang, Z. Zhao, M. Tian, Short-term wind speed forecasting based on a hybrid model, Applied Soft Computing. 13 (2013) 3225-3233.

[9] M. C. Alexiadis, P. S. Dokopoulos, H. S. Sahsamanoglou, Wind speed and power forecasting based on spatial correlation models, IEEE Trans. Energy Convers. 14 (1999) 836-842.

[10] I. G. Damousis, M. C. Alexiadis, J. B. Theocharis, P. S. Dokopoulos, A fuzzy model for wind speed prediction and power generation in wind parks using spatial correlation, IEEE Trans. Energy Convers. 19 (2004) 352- 361.

[11] A. Khosravi, S. Nahavandi, D. Creighton, A. F. Atiya, Comprehensive review of neural network-based prediction intervals and new advances, IEEE Transactions on Neural Networks. 22 (2011) 1341-1356.

[12] A. Khosravi, S. Nahavandi, D. Creighton, A. F. Atiya, Lower upper bound estimation method for construction of neural network-based prediction intervals, IEEE Transactions on Neural Networks. 22 (2011) 337-346.

[13] A. Khosravi, S. Nahavandi, D. Creighton, A prediction interval-based approach to determine optimal structures of neural network metamodels, Expert Systems with Applications. 37 (2010) 2377-2387.

[14] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multi-objective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation. 6 (2002) 182-197.

[15] R. Ak, Y. Li, E. Zio, Estimation of prediction intervals of neural network models by a multi-objective genetic algorithm, Proceedings of Flins 2012, Istanbul, Aug. 2012, pp. 1036-1041.

[16] Website: http://www.weatheroffice.gc.ca/canada_e.html, (Dec., 2012).

[17] E. Zio, A study of bootstrap method for estimating the accuracy of artificial NNs in predicting nuclear transient processes, IEEE Transactions on Nuclear Science. 53 (2006) 1460-1478.

[18] L. Yang, T. Kavli, M. Carlin, S. Clausen, P. F. M. de Groot, An evaluation of confidence bound estimation methods for neural networks, Proceedings of ESIT 2000, Sep. 2000, Aachen, Germany, pp. 322-329.

[19] G. Notton, C. Paoli, L. Ivanova, S. Vasileva, M. L. Nivet, Neural network approach to estimate 10-min solar global irradiation values on tilted planes. Renewable Energy, 50 (2013) 576-584.

[20] Q. Zhou, J. Davidson, A. A. Fouad, Application of artificial neural networks in power system security and vulnerability assessment, IEEE Transactions on Power Systems. 9 (1994) 525-532.

[21] J. M. Nazzal, I. M. El-Emary, S. A. Najim, Multilayer perceptron neural network (MLPs) for analyzing the properties of Jordan oil shale, World Applied Sciences Journal. 5 (2008) 546-552.

[22] S. A. Kalogirou, Artificial neural networks in renewable energy systems applications: A review, Renewable and Sustainable Energy Reviews. 5 (2001) 373-401.

[23] N. N. El-Emam, R. H. Al-Rabeh, An intelligent computing technique for fluid flow problems using hybrid adaptive neural network and genetic algorithm, Applied Soft Computing. 11 (2011) 3283-3296.

[24] R. Dybowski, S. J. Roberts, Confidence intervals and prediction intervals for feed-forward neural networks, in: R. Dybowski, V. Gant (Eds.), Clinical Applications of Artificial Neural Networks, Cambridge University Press, 2011, pp. 298- 326.

[25] D. J. C. MacKay, Bayesian interpolation, Neural Computation. 4 (1992) 415-447.

[26] C. P. I. J. Van Hinsbergen, J. W. C. Van Lint, H. J. Van Zuylen, Bayesian committee of neural networks to predict travel times with confidence intervals, Transportation Research Part C, 17 (2009) 498-509.

[27] D. A Nix, A. S. Weigend, Estimating the mean and the variance of the target probability distribution, Proceedings of IEEE Int. Con. Neural Netw. World Congr. Comput. Intell., Orlondo, FL., Jun. 27 - Jul. 2, 1994, pp. 55-60.
[28] R. W. Johnson, An introduction to the bootstrap, Teaching Statistics, 23 (2001) 49-54.
[29] Y. Sawaragi, H. Nakayama, T. Tanino, Theory of Multi-objective Optimization, Orlando, FL, Academic Press Inc., 1985.
[30] A. Konak, D. W. Coit, A. E. Smith, Multi-objective optimization using genetic algorithms: a tutorial, Reliability Engineering and System Safety. 91 (2006) 992-1007.
[31] R. Furtuna, S. Curteanu, F. Leon, Multi-objective optimization of a stacked neural network using an evolutionary hyper-heuristic, Applied Soft Computing. 12 (2012) 133-144.
[32] E. Zitzler, L. Thiele, Multi-objective evolutionary algorithms: a comparative case study and the strength Pareto approach, IEEE Transactions on Evolutionary Computation, 3 (1999) 257-271.
[33] N. Sirinivas, K. Deb, Multi-objective optimization using non-dominated sorting in genetic algorithms, Journal of Evolutionary Computation. 2 (1994) 221-248.
[34] R. Rojas, Neutral Networks: A Systematic Introduction. Springer-Verlag Berlin Heidelberg, Germany, 1996.
[35] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning representations by back-propagating errors, in: T. A. Polk, C. M. Seifert (Eds.), Cognitive modeling, 2002, pp. 213-221.
[36] D. J. Montana, L. Davis, Training Feedforward Neural Networks Using Genetic Algorithms, In IJCAI. 89 (1989) 762-767.
[37] H. Mühlenbein, D. Schlierkamp-Voosen, Predictive models for the breeder genetic algorithm: I. continuous parameter optimization, Evolutionary Computation. 1 (1993) 25-49.
[38] M. Bessaou, P. Siarry, A genetic algorithm with real-value coding to optimize multimodal continuous functions, Structural and Multidisciplinary Optimization. 23 (2001) 63-74.
[39] M. Srinivas, L. M. Patnaik, Adaptive Probabilities of crossover and mutation in genetic algorithms, IEEE Transactions on Systems, Man and Cybernetics. 24 (1994) 656-667.
[40] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, New York, 1992.
[41] Canadian Wind Energy Association (CanWEA), Media Kit 2012. Online, http://www.canwea.ca/pdf/windsight/CanWEA_MediaKit.pdf; 2012 [Accessed on May 2012].
[42] SaskPower Annual Report 2011. Online, http://www.saskpower.com/ news_publications/assets/annual_reports/2011_skpower_annual_report.pdf; 2011 [Accessed on May 2012].
[43] P. Y. Chen, P. M. Popovich (Eds.), Correlation: parametric and nonparametric measures, Sage Publications, no. 137-139, 2002.
[44] M. Hollander, D. A. Wolfe, Nonparametric Statistical Methods, John Wiley & Sons, New York, 1973.
[45] T. Hastie, R. Tibshirani, J. Friedman, The elements of Statistical Learning: Data Mining, Inference and Prediction, second ed., Springer, 2008.
[46] R. Karki, P. Hu, R. Billinton, A simplified wind power generation model for reliability evaluation, IEEE Transactions on Energy Conversion. 21 (2006) 533-540.
[47] N.L. Buccola, T.M. Wood, Empirical models of wind conditions on upper Klamath lake, Oregon, Scientific Investigations Report 2010-5201, Online, http://pubs.usgs.gov/sir/2010/5201/pdf/sir20105201.pdf; 2010, [Accessed on May 2012].
[48] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, Time Series Analysis, Forecasting and Control, fourth ed., Wiley, 2008.
[49] E. Zio, P. Baraldi, and N. Pedroni, Optimal power system generation scheduling by multi-objective genetic algorithms with preferences, Reliability Engineering and System Safety. 94 (2009) 432-444.
[50] A. D. Belegundu, T. R. Chandrupatla, Optimization concepts and applications in engineering, second ed., Cambridge University Press, New York, 2011.
[51] S. Bandyopadhyay, S. Saha, U. Maulik, K. Deb, A Simulated annealing-based multi-objective optimization algorithm: AMOSA. IEEE Transactions on Evolutionary Computation. 12 (2008) 269-283.
[52] E. L. Ulungu, J. Teghem, Ch. Ost, Efficiency of interactive multi-objective simulated annealing through a case study, Journal of the Operational Research Society. 49 (1998) 1044-1050.
[53] F. Mosteller, J. W. Tukey, Data Analysis and Regression: A Second Course in Statistics. Addison-Wesley Series in Behavioral Science: Quantitative Methods, Reading, Mass.: Addison-Wesley, 1977.
[54] J. L. Devore, K. N. Berk. Modern mathematical statistics with applications. Springer, London, 2011.
[55] N. Hansen, A. Ostermeier, Completely derandomized self-adaptation in evolution strategies, Evolutionary Computation. 9 (2001) 159–195.

[56] C. Igel, N. Hansen, S. Roth, Covariance matrix adaptation for multi-objective optimization, Evolutionary computation 15.1 (2007): 1-28.
[57] N. Hansen, The CMA Evolution Strategy: A Tutorial, June 2011. https://www.lri.fr/~hansen/cmatutorial.pdf
[58] R. Shumway, D. Stoffer, Time Series Analysis and Its Applications: With R Examples (Springer Texts in Statistics), Springer-Verlag, New York, 2010.

PAPER III

Two Machine Learning Approaches for Short-Term Wind Speed Time Series Prediction

R. Ak, O. Fink and E. Zio. (2014), submitted to Special Issue on Neural Networks and Learning Systems Applications in Smart Grid (under review).

Two Machine Learning Approaches for Short-Term Wind Speed Time Series Prediction

Ronay Ak[a], Olga Fink[b], and Enrico Zio[a, c], *Senior Member, IEEE*

[a]Chair on Systems Science and the Energetic Challenge, European Foundation for New Energy-Electricité de France, CentraleSupélec, Châtenay-Malabry 92290 and Gif-Sur-Yvette 91192, France

[b]Institute for Transport Planning and Systems, ETH Zurich, Zurich, Switzerland

[c]Department of Energy, Politecnico di Milano, Milan 20133, Italy

ABSTRACT

The increasing liberalization of European electricity markets, the growing proportion of intermittent renewable energy being fed into the energy grids, but also new challenges in the patterns of energy consumption (such as electric mobility) require flexible and intelligent power grids capable of providing efficient, reliable, economical and sustainable energy production and distribution. From the supplier side, particularly, the integration of renewable energy sources (e.g. wind and solar) into the grid imposes an engineering and economic challenge because of the limited ability to control and dispatch these energy sources due to their intermittent characteristics. Time series prediction of wind speed for wind power production is a particularly important and challenging task, whereby Prediction Intervals (PIs) are preferable results of the prediction, rather than point estimates, because they provide information on the confidence in the prediction.

In this paper, two different machine learning approaches to assess PIs of time series predictions are considered and compared: Multi-layer Perceptron Neural Networks (MLP NN) trained with a multi-objective genetic algorithm and Extreme Learning Machines (ELM) combined with the nearest neighbors approach. The proposed approaches are applied for short-term wind speed prediction from a real dataset of hourly wind speed measurements for the region of Regina in Saskatchewan, Canada. Both approaches demonstrate good prediction precision and provide complementary advantages with respect to different evaluation criteria.

Keywords: Extreme learning machines, multilayer perceptron, multi-objective genetic algorithms, prediction intervals, short-term wind speed prediction, wind power production.

1. INTRODUCTION

Smart grid technology induces intelligence in the conventional power grid and comprises many different elements, both at supplier and consumer sides, to provide efficient, reliable, economical and sustainable energy production and distribution. It can be defined as an interconnected network of microgrids with distributed control [1].

On the other hand, the evolution from conventional power grids towards smart grids with integration of distributed renewable energy sources leads to additional uncertainty in the system. Indeed, the challenge of operating power systems reliably and safely increases with the growing proportion of intermittent renewable energy, such as wind and solar, being fed into the energy grids. The inherent variability and uncertainty affecting the renewable energy sources can have a significant impact on power supply, and accurate and reliable predictions of the power output obtainable from these sources are needed on different time scales. Thus, predicting the output of renewable energy sources is critical for integrating them efficiently in the power grid and for dealing with their uncertain and intermittent character.

We consider wind power particularly, whose use has been growing over the last years: the worldwide wind capacity has reached 296 GW by the end of June 2013, out of which 13980 MW have been added in that first half of 2013 [2].

Wind power output mainly depends on the wind speed and on the physical characteristics of the wind turbines. Wind speeds have a very volatile character, which makes their prediction a particularly challenging task. Wind speeds depend on pressure conditions and have different hourly, daily and yearly profiles. Wind power variations in short-term time scales (from seconds to minutes, hours or several days) have significant effects on power system operations such as spot (daily and intraday) market, system management and scheduling of maintenance tasks [3]-[6]. Therefore, short-term prediction of wind speed for wind power production does not only affect system operators, but also electricity companies and wind farm promoters.

Several approaches have been proposed for predicting wind speeds. These include approaches based on physical models for numerical weather prediction, but also statistical and soft

computing approaches, including statistical regression, neural networks and fuzzy logic systems [3]-[7]. Also hybrid approaches have been proposed, combining physical and statistical approaches [7]. The approaches have been developed and applied for predictions performed on different time-scales: short, medium and long term time scales.
Neural networks have been increasingly applied to wind speed prediction tasks, due to their flexibility, self-adaptive learning abilities and the relaxation of the need of physical and phenomenological assumptions for the development of the models [3]-[6], [8]-[10].

Most of the proposed approaches for wind speed predictions provide point estimates of future values. In practice, the accuracy of point estimates can be affected by the uncertainties in the model parameters and input data [11]-[13]. For practical applications, information on the uncertainty in the predictions is necessary to manage properly the energy system.

Prediction intervals (PIs) can be used to provide information on the confidence in the predictions [11]-[13], accounting for both the uncertainty in the model parameters and the noise in the input data.

The main requirement on the quality of the estimated PIs is a high coverage probability that the true values will be within the predicted intervals; on the other hand, to give useful practical information, the intervals need to have small widths. The two requirements are competing, as a small interval will induce a low probability that the true value be within the interval itself, whereas wide intervals may be required to obtain high coverage probability.

In this paper, two different machine learning approaches for estimating PIs of time series predictions are considered and compared. As measures for the quality of predictions, we take the prediction interval coverage probability (PICP) and the prediction interval width (PIW) [12].

In the first approach, a Multi-layer Perceptron Neural Network (MLP NN) is trained by a multi-objective genetic algorithm (MOGA), namely the non-dominated sorting genetic algorithm–II (NSGA-II) [14]. This approach integrates the estimation of the prediction intervals in its learning procedure, and the MLP NN is trained to concurrently minimize the width and maximize the coverage probability of the estimated PIs [15]. The approach is an

extension of the Lower and Upper Bound Estimation (LUBE) approach, proposed in [12].

The second approach combines Extreme Learning Machines (ELM) with the nearest-neighbor approach. It is a newly developed two-step approach: in the first step, the ELM are trained to predict point estimates; in the second step, PIs are quantified based on the performance of the ELM on the nearest neighbors in the training dataset.

These two methods have been selected for consideration because they are two different approaches to PIs estimation: a multi-objective optimization framework for identifying the Pareto front of solutions optimal in terms of PICP and PIW, and exploitation of the local input space performance of the regression algorithm.

The two approaches differ from each other and also from the approaches previously applied in other studies [9]-[12]. The MOGA NN approach integrates the estimation of the PIs in the learning procedure of the algorithm. The algorithm itself is directly trained to balance the width of the interval and the coverage probability, concurrently optimizing the two quality assessment criteria of the PIs. The ELM algorithm combined with the nearest neighbor approach is trained to fit optimally the time series data (contrary to estimating directly the PIs). Conceptually, the ELM algorithm extends the functionality of the standard feed-forward NNs by including different activation functions and overcoming computationally expensive learning algorithms, such as back-propagation. In the second step, the prediction intervals are estimated based on the performance of the algorithm on similar training samples in the input space. Both approaches are based on powerful learning algorithms and have proven capable of providing good generalization ability and accurate predictions, considering the uncertainties associated to the input data and the model parameters. They look promising for time-series wind speed prediction as carried out in this study.

The proposed approaches are applied for short-term wind speed prediction using a real dataset of hourly wind speed measurements for the region of Regina in Saskatchewan, Canada.

The main contribution of this paper is the proposal of two different machine learning approaches for estimating prediction intervals of time series of wind speed profiles and their comparison based on different criteria. The approaches are shown to yield a similar

performance, with different strengths and limitations with respect to the criteria used for the comparison.

The paper extends previous studies, in which either only single machine learning approaches are applied [16] or only point estimates are predicted and compared with different types of neural networks [9], [10]. Apart from those studies, Khosravi et al. [11] reviews and compares different approaches for estimating prediction intervals in different benchmark studies. The main differences of the present work with Khosravi et al. [12] are, on the one hand the application case study, namely the wind speed time series, and on the other hand the approaches applied and compared in the work, which in our case are the novel MOGA NN and ELM combined with a nearest neighbor heuristic.

The remainder of this paper is structured as follows. Section 2 briefly introduces the definition of PIs, the basic concepts of MLP NNs with multi-objective optimization, the basic concepts of Extreme Learning Machines (ELM) in combination with a nearest neighbor approach to estimate PIs. Experimental results on the real case study of wind speed prediction are given in Section 3. In Section 4, the proposed algorithms are compared based on selected criteria and the results are discussed. Finally, Section 5 presents the conclusions of the study.

2. BASICS OF THE TWO MACHINE LEARNING APPROACHES

2.1. Prediction Intervals

Given an input-output process x, $y(x)$, a PI is a statistical estimator composed by lower and upper bounds, $L(x)$ and $U(x)$, that include a future unknown value of the target $\mathrm{y(x)}$ with a predetermined probability, called confidence level and in general indicated with $1-\alpha$ [12], [17]:

$$Pr(L(x) < y(x) < U(x)) = 1 - \alpha \quad (1)$$

The prediction interval coverage probability (PICP) represents the probability that the PIs estimated in correspondence of the different values of x will contain the true output values $y(x)$. This probability is estimated as the proportion of true output values lying within the

estimated PIs:

$$PICP = \frac{1}{n_p}\sum_{i=1}^{n_p} c_i \quad (2)$$

where n_p is the number of samples in the training or testing sets, and $c_i = 1$if $y_i \in [L(x_i), U(x_i)]$ and otherwise $c_i = 0$.

The prediction interval width (PIW) measures the extension of the interval as the difference of the estimated upper bound and lower bound values, $L(x) - U(x)$. We consider the Normalized Mean PIW ($NMPIW$).

$$NMPIW = \frac{1}{n_p}\sum_{i=1}^{n_p} \frac{(U(x_i) - L(x_i))}{y_{max} - y_{min}} \quad (3)$$

where y_{min} and y_{max} represent the true minimum and maximum values of the targets (i.e., the bounds of the range in which the true values fall) in the training set, respectively. Normalization of the PI width by the range of targets makes it possible to objectively compare the PIs, regardless of the techniques used for their estimation or the magnitudes of the true targets.

In general, wider intervals give larger coverage, and in practice it is important to have narrow PIs with high coverage probability [11], [12].

2.2. Quantifying the Prediction Intervals by Multi-Layer Perceptron Neural Networks Trained by MOGA

Multi-layer perceptron (MLP) is one of the most common types of feedforward neural networks proven to be a class of universal approximators [18]. To date, they have been widely used in many practical applications such as optimization [19], pattern recognition [20], clustering [21], prediction [9]-[12], [17], [23], diagnosis [22], etc. In the context of prediction, they have been used as empirical regression models especially for nonlinear regression. NNs are a machine learning algorithm that can theoretically learn the input-output relationship to any degree of precision [17], [18], [23]-[25]. A MLP NN consists of processing units, so

called neurons, ordered into layers: one input layer, one or several hidden layers and one output layer. Each layer comprises a defined number of neurons, which needs to be defined by the users. The neurons are connected by weights. Each layer receives input signals generated by the previous layer, produces output signals through an activation function (e.g. a sigmoid function) and distributes them to the subsequent layer through the neurons [18], [23].

Set h to be the number of hidden neurons, n_f the number of input neurons and n_p the total number of training samples. Then, the output signal H_j of node j of the hidden layer is given by [18], [23]-[25]:

$$H_j = f_h\left(\sum_{k=0}^{n_f} w_{kj} x^k\right) \qquad j = 1, \dots, h \tag{4}$$

where $x^0 = 1$, and for $k = 1,2, \dots, n_f$, x^k is the k-th input vector, $x^k \equiv (x_1^k, x_2^k, \dots, x_{n_p}^k)$, w_{kj} is the synaptic weight and f_h is the activation function used in the hidden layer.
After each hidden neuron output has been computed, the signal is sent to each of the neurons o_l in the output layer. Each output neuron o_l computes its output signal O_l to form the response of the network [18], [23]-[25]:

$$O_l = f_o\left(\sum_{j=0}^{h} w_{jl} H_j\right) \quad l = 1,2, \dots, n_o\ ,\ H_0 = 1 \tag{5}$$

where n_o is the number of output neurons and f_o indicates the activation function used in the output layer.

The values of the weight vector w of the network are optimized during training. Training procedure aims at minimizing the quadratic error function on a training set of input/output values $D = \{(x_i, y_i),\ i = 1,2, \dots, n_p\}$ [17]:

$$E(w) = \sum_{i=1}^{n_p} (O(x_i) - y_i)^2 \tag{6}$$

where $O(x_i)$ is the estimated output value of the network for the i-th input sample x_i.

2.3. Multi-objective Optimization by NSGA-II

In all generality, a multi-objective optimization problem considers a number of objectives, $f_m,\ m = 1, 2, \dots, M$, inequality $g_j,\ j = 1, 2, \dots, J$ and equality $h_k,\ k = 1, 2, \dots, K$ constraints, and bounds on the decision variables $x_i,\ i = 1, 2, \dots, I$. Mathematically the problem can be written as follows [26]-[29]:

$$\textit{Minimise/Maximise} \quad f_m(x), \quad m = 1,2, \dots, M; \tag{7}$$

$$\textit{subject to} \quad g_j(x) \geq 0, \quad j = 1,2, \dots, J; \tag{8}$$

$$h_k(x) = 0, \quad k = 1,2, \dots, K; \tag{9}$$

$$x_i^{(l)} \leq x_i \leq x_i^{(u)} \quad i = 1,2, \dots, I. \tag{10}$$

A solution, $x = \{x_1, x_2, \dots x_I\}$ is an I-dimensional decision variable vector in the solution space R^I, restricted by the constraints (8), (9) and by the bounds on the decision variables (10).

The search for optimality requires that the M objective functions $f_m(x),\ m = 1, 2, \dots, M$ be evaluated in correspondence of the decision variable vector x in the search space. The comparison of solutions during the search is performed in terms of the concept of dominance [26], [27]. Precisely, in case of a minimization problem, solution x_a is regarded to dominate solution x_b $(x_a \succ x_b)$ if the following conditions are satisfied [26]:

$$\forall i \in \{1,2, \dots, M\}: f_i(x_a) \leq f_i(x_b) \ \wedge \tag{11}$$

$$\exists j \in \{1,2, \dots, M\}: f_j(x_a) < f_j(x_b) \tag{12}$$

If any of the above two conditions is violated, the solution x_a does not dominate the solution x_b and x_b is said to be non-dominated by x_a. Eventually, the search aims at identifying a set of optimal solutions $x^* \in R^I$ which are superior to any of the optimal solutions with respect to all objective functions and do not dominate each other. This set of optimal solutions is called Pareto optimal set; the corresponding values of the objective functions form the so called Pareto optimal front in the objective functions space.

In this work, we use GA for the multi-objective optimization. GA is a population based meta-

heuristics inspired by the principles of genetics and natural selection [27]-[29]. It can be used for solving multi-objective optimization problems. The major motivation for using the GA search paradigm is due to the following three recognized advantages [27]-[29]: (*i*) capability of exploring large portions of the search space without falling into a local optimum; (*ii*) ease of use; and (*iii*) robustness. Further, GAs are capable of searching solutions from disjoint feasible domains and of operating on irregular functions (i.e. non-continuous and even non-differentiable); for proceeding in the search, GAs do not require the computation of gradients.

Among the several variations of MOGA in the literature, we select non-dominated Sorting Genetic Algorithm-II (NSGA-II) [14] as the optimization algorithm, because comparative studies [14], [30] have shown that it is one of the most efficient MOGAs.

More specifically, we use NSGA-II for finding the values of the parameters of the NN which minimize the two objective functions PICP (2) and NMPIW (3) simultaneously, in Pareto optimality sense (for ease of implementation, the maximization of PICP is converted to minimization by subtracting from unity, i.e. the objective of the minimization is 1-PICP). The practical implementation of NSGA-II on our specific problem involves two phases: initialization and evolution. These can be summarized as follows:

Initialization phase:

Step 1) Split the input data into training (D_{train}) and testing (D_{test}) subsets.

Step 2) Define the values of: the maximum number of generations, the number of chromosomes (individuals) Nc in each population, and the initial crossover and mutation probabilities.

Step 3) Set the generation number $gen = 1$. Initialize the first population P_n of size Nc by randomly generating Nc chromosomes. Each chromosome forms a candidate solution by G real-valued genes, where G is the total number of parameters (weights) in the NN. Note that each solution corresponds to a NN.

Step 4) For each input sample x in the training set, evaluate each of the Nc chromosomes in the initial population P_n, i.e. compute the lower and upper bound outputs of each Nc chromosome with G parameters by performing NN training. Return the values of two objectives 1-PICP and NMPIW for each of the Nc chromosomes.

Step 5) Rank the chromosomes (vectors of G values) in the population P_n by running the fast

non-dominated sorting algorithm [14] with respect to the pairs of objective values, and identify the ranked non-dominated fronts $F_1, F_2, \dots, F_k$ where F_1 is the best front, F_2 is the second best front and F_k is the least good front.

Step 6) Apply to P_n a binary tournament selection based on the crowding distance [14], for generating an intermediate population S_n of size Nc.

Step 7) Apply the crossover and mutation operators to S_n, to create the offspring population Q_n of size Nc.

Step 8) Apply Step 4 onto Q_n and obtain the lower and upper bound outputs. Evaluate each of the Nc chromosomes in the population Q_n. Return the values of the two objectives corresponding to the solutions in Q_n.

Evolution phase:

Step 9) If the maximum number of generations is reached, stop and return P_n. Select the first Pareto front F_1 as the optimal solution set. Otherwise, go to Step 10.

Step 10) Combine P_n and Q_n to obtain a union population $R_n = P_n \cup Q_n$.

Step 11) Apply Steps 4-5 onto R_n and obtain a sorted union population.

Step 12) Select the Nc best solutions from the sorted union to create the next parent population P_{n+1}.

Step 13) Apply Steps 6-8 onto P_{n+1} to obtain Q_{n+1}. Set $gen = gen + 1$; and go to Step 9.

Finally, the best front in terms of ranking of non-dominance and diversity of the individual solutions is chosen. Once the best front is chosen, then the testing step is performed on the trained NN with optimal weight values.

The binary tournament selection, mentioned in Step 6, uses the crowded-comparison operator $\prec_n$ as the selection criterion [14]. For solution i in the population, it has two attributes: nondomination rank $\mathrm{i}_{\mathrm{rank}}$ and crowding distance $i_{distance}$. For a solution pair, i and j, we have $i \prec_n j$ if $i_{rank} < j_{rank}$ or ($i_{rank} = j_{rank}$ and $i_{distance} > j_{distance}$). That is, if there are two solutions under consideration with different nondomination ranks, we prefer the one with the lower (better) rank. Otherwise, if both solutions have same ranking, i.e. belong to the same nondominated front, we select the solution which locates in a region with least number of points. For further explanations, we refer the readers to [14].

The total computational complexity of the proposed algorithm can be explained by two time demanding sub-operations: non-dominated sorting and fitness evaluation. The time complexity of non-dominated sorting part is $O(MNc^2)$ where M shows the number of objectives and Nc shows the population size [14]. In the fitness evaluation phase, the NSGA-II has been used to train a NN which has n_p input samples. Since for each individual of the population a fitness value is obtained, this process is repeated $Nc \times n_p$ times. Hence, time complexity of this phase is $O(Nc \times n_p)$. In conclusion, the computation complexity of one generation is $O\left(MN^2 + Nc \times n_p\right)$.

2.4. Quantifying the Prediction Intervals with ELM Regression and the Nearest Neighbors Approach

The approach proposed for quantifying prediction intervals combines a regression performed by ELM with a nearest neighbors approach. First, the ELM regression algorithm is trained to provide point estimates and, then, the nearest neighbors approach is applied to quantify the prediction intervals, as proposed in [31]. Actually, both steps of point estimate regression and prediction intervals quantification can be applied independently: therefore, the algorithm applied for the regression task can be selected freely by the user, independently from the approach applied for quantifying the prediction intervals.

In this research, ELM have been applied for the regression task due to their flexibility, computational efficiency and their superior performance demonstrated on several benchmark studies [32], and the nearest neighbors approach has been used for the prediction intervals quantification relying on the k-d tree algorithm [33].

The nearest neighbors approach determines the prediction quantiles empirically, based on the assumption that the performance of the regression algorithm on data patterns in the same region of the input space is similar. This implies the assumption that the local characteristics of the input space determine the prediction performance of the regression algorithm. The similarity of data patterns is defined by the Euclidean distance between them in the input space. The approach is not dependent on any assumption about the distribution of the errors, e.g. that they are normally distributed.

Because the prediction intervals are determined based on the performance of the regression algorithm on the training dataset, prediction intervals can only be predicted for the testing dataset. For the training dataset, only point estimates can be computed. Therefore, the generalization ability of the approach, from training to testing, can only be assessed with respect to the point estimates.

The general procedure of the proposed approach is presented in Fig. 1. In the first step, the regression algorithm is trained to generalize the patterns in the training dataset. After the training phase is finished and the weights are fixed, the algorithm is applied to perform the prediction task on the training dataset. Subsequently, the errors between the actual and the target output are calculated. In the next step, the regression algorithm is applied to the testing dataset. For each of the patterns in the testing dataset, the defined number of nearest neighbors is determined with the specified nearest neighbors algorithm. Subsequently, the errors of the neighboring patterns are sorted in ascending order $\{e_{i,1}, \cdots, e_{i,nn}\}$, where $e_{i,1}$ is the smallest error in the set of neighbors nearest to the testing input x_i and nn is the defined number of nearest neighbors.

The prediction interval can, then, be estimated by adding the prediction error of the quantiles to the point estimate of the specific input pattern:

$$[\hat{y}_i - e_{i,l}, \hat{y}_i + e_{i,u}] \tag{13}$$

where $e_{i,l}$ is the lower interval limit of the training error and $e_{i,u}$ is the upper interval limit of the error.
In the final step, the performance of the estimated prediction intervals can be assessed by calculating the coverage probability (PICP) and the widths of the prediction intervals (PIW).

The described approach is applicable in a very flexible way because the regression algorithm can be selected depending on the requirements of the prediction task and the characteristics of the applied dataset.

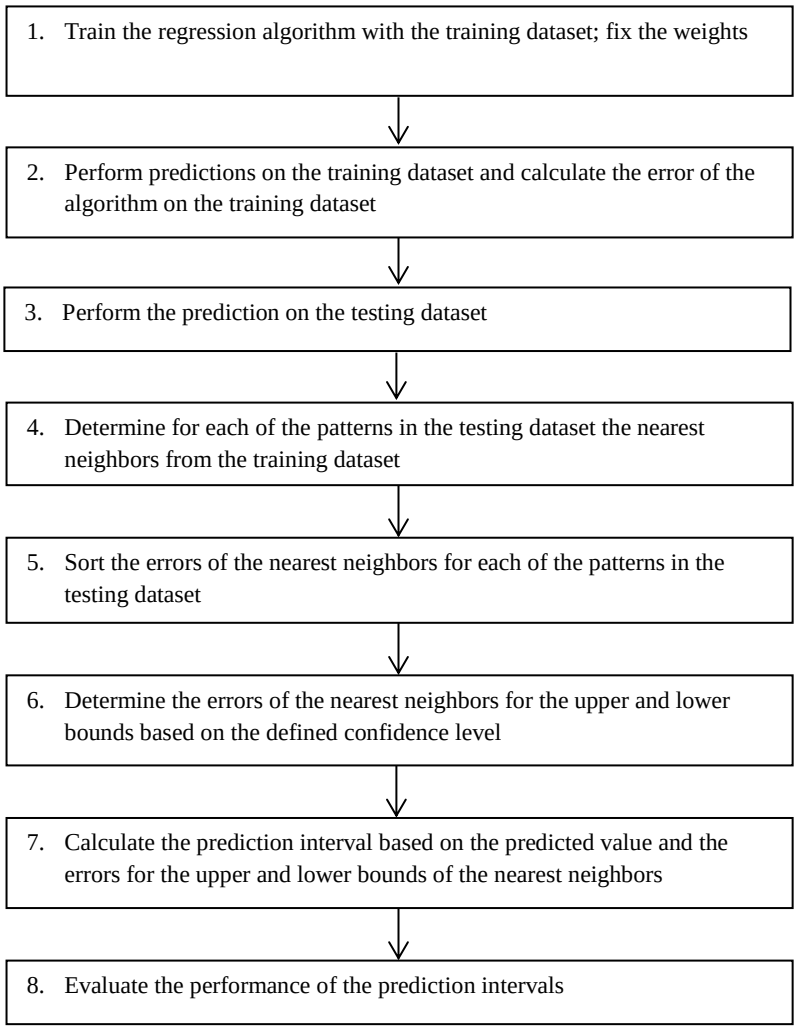


Fig. 1. Procedure for determining prediction intervals.

2.5. General Concepts of Extreme Learning Machines

The extreme learning machines combine the strengths of several machine learning techniques, such as MLP, Radial-Basis-Function networks (RBF) and Support Vector Machines (SVM) thereby providing a uniform learning platform [34]. The ELM are layered feedforward network structures comprising a single hidden layer with flexible activation functions of the hidden nodes (including linear, sigmoidal, polynomial and radial-basis functions). ELM provide a computationally very efficient learning algorithm without iterative parameter adaptation, by integrating a random selection of the hidden nodes and analytic determination of the weights between the output and hidden layers [35].

This learning procedure is not only computationally very efficient, particularly compared to back-propagation learning or other learning procedures of MLP, but also avoids local minima, which is one of the main drawbacks of gradient descent learning approaches [34].

More details on the theoretical concepts and computational algorithms of ELM can be found in [32], [34]-[39].

ELM have been successfully applied to many different applications [40], [41]. Additionally, several extensions and further developments have been introduced to ELM [42], [43].

The major advantages of ELM are that they are computationally efficient and very flexible, achieve good generalization ability, are not prone to local minima, and do not require expert knowledge on algorithm parameter setting and fine-tuning [34]. These are also the reasons for studying the performance of ELM for the prediction of wind speed time series PIs.

3. CASE STUDY AND RESULTS

3.1. Applied Datasets

The proposed prediction algorithms have been applied on three different datasets of hourly wind speeds. The considered wind speed data have been measured in different periods of the year in Regina, Saskatchewan, a region of central Canada [44]. The first dataset comprises wind speeds for the period from 1st of February 2012 to 31st of March 2012, the second from 1st of July 2012 to 29th of August 2012 and the third from 1st of February 2011 to 30th of June 2011.

The first two datasets contain data on a two-month period, whereas the third time period is five months long. The first dataset (winter period) includes 1437 samples; the second dataset (summer period) comprises 1438 samples and the third dataset, referred to as w2011, includes 3596 samples. The three time periods have different seasonality and have been selected to represent different patterns and characteristics in the measured time series of wind speeds.

For the development of the prediction algorithms, in all the three datasets, the first 80% of the

time series data are used for training and the rest 20% are left for testing. For all three datasets, the inputs were normalized to be in the value range between 0.1 and 0.9. Fig. 2 shows the profiles of the three datasets used: the volatile character of the wind speed variable is clearly observable.

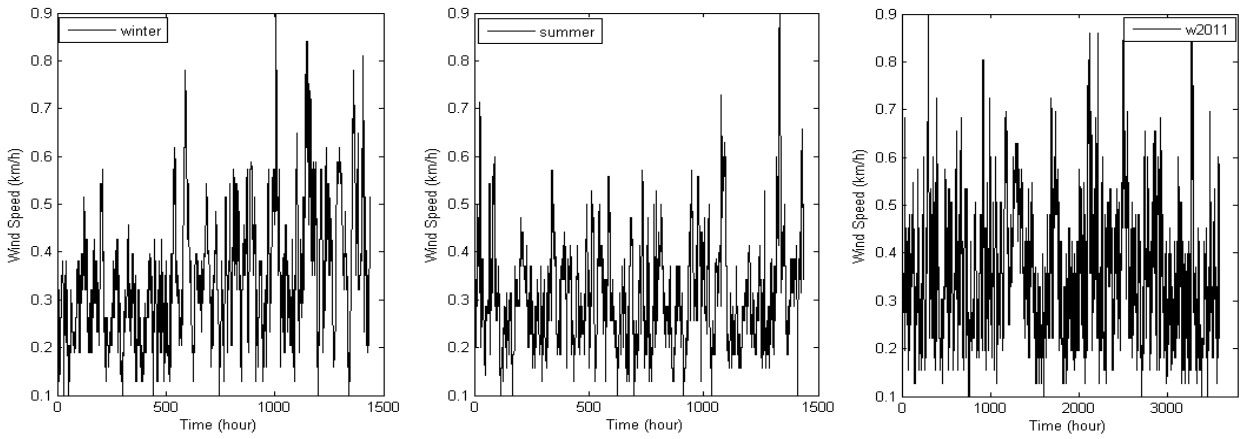


Fig. 2. The wind speed datasets used in this study: winter (left), summer (middle) and w2011 (right).

3.2. Pre-analysis of input data for MOGA MLP NN

In order to select the relevant lagged values of the wind speed ($W_{t-1}, \ldots, W_{t-m}$) to be included as input variables in the prediction model for estimating (W_t), empirical Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) analyses have been done. In time series analysis, this way of selection allows a most effective fitting of an autoregressive model to estimate the value of the output as a linear combination of its past values [45]. Even though NNs are nonlinear models, this indication of the relevance of the past values to estimate the wind speed target of the prediction is useful for the construction of the model. Indeed, various studies [47], [48] have used ACF and PACF to determine the input values of NN that are most related to the prediction values.

In our case, the ACF and PACF results indicate that for predicting W_t in output, wind speed values at previous time steps W_{t-1}, W_{t-2} and W_{t-3}, W_{t-1} and W_{t-2}, $W_{t-1}, W_{t-2}, W_{t-3}$ and W_{t-4} are appropriate input variables for the winter, summer and w2011 datasets, respectively.

3.3. Prediction Results with MOGA MLP NN

The architecture chosen for the NN is the classical one consisting of one input, one hidden and one output layers. The number of input neurons is set to 2 for summer data, to 3 for winter data and to 4 for w2011 data. The number of hidden neurons is set to 10 after a trial-and-error process; the number of output neurons is 2, one for the lower and one for the upper bound values of the PIs. As activation functions, the hyperbolic tangent function is used in the hidden layer and the logarithmic sigmoid function is used at the output layer (these choices have been found to give the best results by trial and error, although the results have not shown a strong sensitivity to them).

Table 1 contains the parameters of the NSGA-II for training the NN. "MaxGen" indicates the maximum number of generations which is used as a termination condition and Nc indicates the total number of individuals per population. P_{c_int} indicates the initial crossover probability and is fixed during the run. P_{m_int} is the initial mutation probability and it decreases at each iteration (generation) by the formula:

$$P_{m\_int} \times e^{(-\frac{gen}{MaxGen})} \tag{14}$$

Before selecting the ultimate initial crossover and mutation probabilities reported in Table 1, a parameter tuning has been performed. Crossover probability has been changed from 0.4 to 1 with step size of 0.2. For mutation probability, 0.06 and 1 values have been set, respectively. The results show that the performance of NSGA-II with the initial mutation probability of 1 is worse than that with the initial mutation probability of 0.06. However, tuning the initial crossover probability did not make any significant difference in the results obtained.

Table 1. NSGA-II Parameters Used in the Experiments.

Parameter	Numerical value
MaxGen	300
Nc	50
P_{m_int}	0.06
P_{c_int}	0.8

To account for the inherent randomness of NSGA-II, twenty different runs have been performed for each dataset and an overall best non-dominated Pareto front has been obtained from the twenty individual fronts. To construct such front, the first (best) front of each of twenty runs is collected and the resulting set of solutions is subjected to the fast non-dominated sorting algorithm [14] with respect to the two objective functions values. Then, the ranked non-dominated fronts $F_1, F_2, \dots, F_k$ are identified, where $F_1$is the best front, F_2 is the second best front and F_k is the worst of the k fronts. Solutions in the first (best) front $F_1$are then retained as overall best front solutions. This procedure gives us the overall best non-dominated Pareto front for the training set. After we have obtained this overall best front, we perform testing using each solution included in it.

While the computational load for the training phase is significant due to the genetic learning algorithm and the iterative back-propagation learning, the computing time for the testing phase is negligible.

Fig. 3 illustrates the overall best Pareto front solutions, obtained with the procedure explained above from the 20 NSGA-II runs for w2011 dataset. Note that in Fig. 3 the X axis indicates "1-PICP". The corresponding testing solutions have been also shown on the plot. It can be observed that the training front (marked in diamonds) and corresponding testing solutions (marked in circles) show high consistency in terms of coverage probability and interval size. Due to space limitation, the similar plots for winter and summer datasets are omitted.

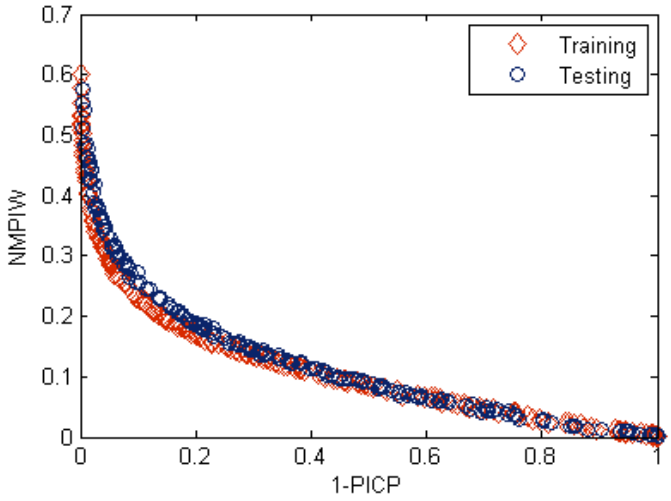


Fig. 3. The overall Pareto front obtained by training of the NN and the corresponding testing solutions for 1h-ahead wind speed prediction using w2011 dataset.

For NSGA-II algorithm, the average CPU times over 20 runs for both training and testing have been registered using MATLAB on a PC with 4 GB of RAM and a 2.53-GHz based processor. For exemplification, the average required CPU times for the MOGA NN method to construct PIs for winter training and test samples are 258.84 s and 0.05 s, respectively. It is seen that the testing phase, i.e. the online prediction of PIs, is very fast.

3.4. Prediction Results with ELM and Nearest Neighbors Approach

The selection of the number of lagged time series inputs for the ELM algorithm can be directly based on the considerations relevant for estimating the PIs. The estimation approach is based on the nearest neighbors approach, whereby the nearest neighbors of a specific pattern are determined by the Euclidean distance calculated on the input values of the specific pattern and its neighboring patterns. For the wind speed datasets considered, the possible value range is comparably small, particularly the maximum value. Hence, the Euclidean distance values for several different patterns will be similar. By addition of more lagged input values, the degree of similarity between the patterns can be decreased.

On this basis, the number of lagged time series values to consider for the wind speed prediction was determined by a wrapping trial and error approach, in which the performance of the algorithm on determining the prediction intervals for the testing dataset was taken as evaluation criterion. For the evaluation, the number of neighbors was varied in the interval [3, 20]. As a result, the number of lagged input values was set to 16. Note that this does not imply a dependence of the output value on 16 lagged time series steps; rather, such a large number of lagged values is required to increase the distance among the patterns, evaluated on a large dimensional space so as to distinguish the neighboring ones. Considering the prediction performance of the algorithm for the point estimates as the selection criterion, a smaller number of lagged input variables could be used, providing better prediction accuracy for point estimates.

Eventually, the number of nearest neighbors was set to 150 by trial and error, searching in the interval [50, 200]. Kd-tree nearest neighbors search [33] has been applied for determining the nearest neighbors of a specific pattern.

Ridge regression was applied within the ELM algorithm [46]. Different regularization factors were applied for the different datasets (0.001 both for the winter and the w2011 datasets and 0.2 for the summer dataset). The reason for different performances of the algorithm with the different regularization factors is due to the different characteristics of the datasets. The regularization factors were determined by trial and error, selecting the best performing parameter values.

To evaluate the prediction results, in the first step, the generalization ability of the ELM algorithm is assessed based on point estimates for training and testing data. For evaluating the prediction accuracy of point estimates, the normalized root mean squared error (NRMSE) is used:

$$NRMSE = \frac{1}{y_{max} - y_{min}} \sqrt{\frac{\sum_{t=1}^{n_p} (y_t - \hat{y}_t)^2}{n_p}} \tag{15}$$

The normalization of the NRMSE is performed over the value range of the entire dataset (total normalized value range of 0.8).

Table 2 shows the NRMSE for the three datasets, for training and testing. The results show a good generalization ability of the ELM algorithm. For the summer dataset, the NRMSE value even decreases for the testing dataset. This can be explained by a higher variability of the wind speeds in the training dataset, compared to the testing dataset. The best training results are obtained on the w2011 dataset and the best testing results on the summer dataset.

As described in the introduction of the methodology, the estimation of the prediction intervals is based on the performance of the algorithm on similar data patterns to those used for training. Therefore, PIs cannot be estimated for the training dataset but only for the testing dataset. Consequently, it is assumed that if the algorithm shows a good generalization ability and does not overfit the training data, this will also induce a good quality of PIs, estimated based on the training dataset prediction accuracy.

Due to the computational efficiency of the ELM, the computing times for the training and also for the testing are negligible.

For evaluating the performance of the proposed approach for estimating the PIs on the testing

dataset, PICP and NMPIW have been generated similarly to the performance evaluation of the MOGA NN approach. The obtained results are presented in Table 3.

Generally, the flexibility of the approach increases with the size and representativeness of the training dataset for the entire dataset.

4. COMPARISON OF THE ALGORITHM PERFORMANCE

4.1. Assessment of the performance of the applied algorithms

There are different criteria that need to be assessed when comparing the performance of machine learning algorithms in the context of a prediction problem. Prediction precision is one of the pivotal criteria when selecting the best performing algorithm.

Several research studies on wind speed prediction used mainly error criteria for assessing and comparing the performance of the applied algorithm. The error criteria included the mean absolute error (MAE), mean absolute percentage error (MAPE), mean square error (MSE) and the root mean square error (RMSE) [7]-[10].

Even though these metrics are mainly suitable for pure point estimates, the NRMSE was also used in this research. Although PIs provide more information on the uncertainty of the prediction to the user, the operator nevertheless requires one operating point and needs to select the most probable value in the interval. Because ELM are trained to provide point estimates, these values are taken for the calculation of the NRMSE.

For the MOGA NN approach, which provides directly the bounds of the PIs, first, the mean values of the estimated intervals corresponding to each solution on the Pareto front have been taken as point predictions. Then, the NRMSE (15) value has been calculated for those point predictions. Note that, for each solution on the Pareto front, we have calculated an NRMSE value. Finally, the median value of these NRMSEs has been considered as the ultimate representative error of the MOGA NN approach with respect to the point predictions a posteriori obtained. The same procedure has been applied for the intervals estimated on the testing set to obtain NRMSE values for the testing set. Another way to convert PIs to point

predictions can be to consider only either the upper bound or the lower bound of the estimated interval, for a more or less robust/conservative prediction. Then, NRMSE values can be similarly calculated for each solution on the front.

The NRMSE results are reported in Table 2 and indicate the good generalization ability: low error values both in training and testing sets indicate the good generalization performance of the MOGA NN approach. Furthermore, the results show that for both algorithms, the NRMSE are in the same value range and do not show significant differences.

NRMSE provides only an incomplete picture on the performance of the algorithms. Because the main focus of the research is on estimating the prediction intervals and not point estimates, the pivotal criterion for assessing the performance of the proposed algorithms is the quality of the estimated PIs. The quality can be assessed by the accuracy and reliability of the prediction [11], [12]. Accuracy can be measured with the coverage probability and reliability is associated with the tightness of the obtained prediction intervals. These two criteria can be assessed by the measures introduced in Section 2, namely the PICP (2) and the NMPIW (3).

The results of these two assessment criteria for both algorithms are displayed in Tables III and IV. Both methods give high coverage probabilities with small interval sizes on the testing sets. This confirms the high prediction performance of the methods.

As these two assessment criteria, i.e. CP and PIW, are competing, in some studies [11], [12], [47], [48] an additional criterion, so-called CWC is introduced. In other words, CWC is a cost function to be minimized combining and weighting the two quantitative measures: PICP and NMPIW. However, CWC is only required if the solutions are clearly not dominated by each other. As demonstrated in the Tables III and IV, for the winter and the summer datasets, the solutions provided by ELM are superior to those provided by MOGA NN and for the w2011 dataset MOGA NN provides dominant solutions. Therefore, a combined criterion, such as CWC is not required in this case.

A further assessment criterion is the generalization ability of the machine learning algorithm. The generalization ability of an algorithm assesses the capability of an algorithm to extract patterns from data and to transfer them to data unseen in the training phase. In this research,

NRMSE of the training and testing data was used to assess this criterion. If an algorithm generalizes well, the performance drop from training to testing data will be small. This behavior can be observed in the NRMSE values obtained with both algorithms (Table 2). Both algorithms were able to extract the typical patterns in the time series and project the extracted pattern to unseen values in the testing datasets.

A further comparison criterion of machine learning algorithms is the computational load, i.e. the efficiency of an algorithm. The efficiency is relevant for both the training phase and in practical applications, particularly for online predictions of a new data sample. From the user point of view, the computational burden of the training phase is relatively less important [12], [47], [48] since the training phase is, usually, only performed once. In cases, where online learning is implemented, the algorithms are usually not retrained, but their parameters are updated. This is, usually, computationally less expensive than a new training phase. Note that computational load is dependent on the type of the selected algorithm, the complexity of the structure of the model (e.g. number of input neurons, hidden layers, and hidden neurons), the size of the dataset and the performance of the learning algorithm.

Compared to ELM, MOGA NN is computationally more expensive with respect to offline computational time. For the application phase, when performing one-step ahead predictions, the online computational time is negligible for both algorithms.

Case study results reported in Tables III and IV indicate that both methods show, in general, a better performance on the w2011 dataset. This can be explained with a similar profile of the training and testing sets in w2011 dataset and a larger training dataset available. The algorithms are obviously able to generalize the presented patterns well and to transfer them to new unseen data. On the contrary, the testing patterns of the winter and summer datasets show relatively higher variability compared to the training patterns (see Fig. 2).

4.2. Discussion of the strengths and limitations of the selected approaches

One of the major advantages of the approach combining a machine learning regression algorithm (done by ELM, in our case) and the nearest neighbors approach (done by kd-tree, in our case) is that for the regression task, the best performing algorithm can be selected. This

makes the approach very flexible because the prediction intervals are determined independently of the applied regression algorithm. Additionally, because prediction quantiles are determined empirically, the approach is not dependent on any assumption about the distribution of the errors, such as the assumption of normally distributed errors.

The applied approach for estimating the prediction intervals is based on the assumption that the performance of the regression algorithm on data patterns in the same region of the input space is similar. This assumption imposes one limitation to the approach: if for a selected system and the pertinent dataset the operating conditions change and the data patterns become dissimilar to those applied for the training data, the distance to the nearest neighbors will increase. However, the novelty of the new pattern will not be reflected in the width of the predicted interval. The challenge of novel or anomalous patterns is common for most of the machine learning techniques. However, for the proposed approach, not only the accuracy of the point estimate will decrease with the degree of novelty of the presented pattern, but also the coverage probability of the prediction interval will decrease. For the wind speed prediction task, the potential of patterns with a very high degree of novelty is limited. Therefore, the described limitation was not witnessed in the case study. However, a similar behavior may be observed in cases where the variability of the testing dataset increases, compared to training dataset. In these cases, the approach may not be very flexible to adapt to the increased variability. This would result in a low flexibility of the approach for high coverage probabilities. This behavior could be observed on the winter dataset.

A further limitation of the approach was observed in the case study because of the comparably small possible value range for wind speeds. This resulted in a high degree of similarity of the Euclidean distance for many patterns. This characteristic of the dataset imposed limitations on the flexibility and accuracy of the applied approach. A possibility to overcome this limitation would be to determine the nearest neighbors based on their similarity in the variability of the input patterns, contrary to the approach applied in this case study based on the Euclidean distance of the patterns in the input space itself. Furthermore, the errors of the training dataset could be weighted, according to the distance of the patterns in the input space.

ELM overcome some of the major limitations and drawbacks of alternative machine learning algorithms. They provide a powerful and efficient learning algorithm and do not require

manual or a computationally very expensive parameter setting, because the parameters of the ELM hidden nodes are not dependent on the target function or the training dataset [34].

Furthermore, the applied approach demonstrates that even though the ELM are not trained to optimize both objectives (PICP and NMPIW) concurrently but only to minimize the error between the target and the prediction, the PIs estimated based on the performance of the algorithm in the local input space provide predictions which are well balanced between both objectives. For some datasets, they even provide better results than the algorithms that were specifically trained based on both objectives.

On the other hand, the MOGA NN approach handles the PI problem in a multi-objective framework. This approach provides a Pareto set of optimal solutions with respect to two objective functions, PICP and NMPIW. This is the main contribution of this method. Knowledge about Pareto optimal set is helpful and provides valuable information about the underlying problem. It gives several optimal solutions and allows the decision makers (DMs) to be aware of the potential risks with respect to the different solutions. The selection of the solution mainly depends on the preferences of the DMs. In other words, the Pareto optimal set of solutions can provide the DMs the flexibility to select the appropriate solutions with different preferences on the objectives. The decision makers also gain insights into the characteristics of the optimization problem before a final decision is made. Obviously, as a final decision, DMs should select one solution from the front to be used in practice.

Moreover, we have used a powerful algorithm, NSGA-II, to train the NN weights. In the literature, the back-propagation has been widely used for performing supervised learning tasks, i.e. the training of NNs [24]. However, this requires calculating the gradient of the error function to find the optimal weights that minimize the estimation error [24], [49], whereas the NSGA-II does not require these derivative calculations. Moreover, existing techniques for estimating PIs for NN algorithm outputs such as Delta and Bayesian methods require the calculation of Jacobian and Hessian matrixes, respectively, and although they are capable of generating high quality PIs, they demand high computational time in the development stage [11]. Compared to Delta and Bayesian methods, NSGA-II is less demanding at the training phase.

The MLP NN algorithm might be prone to give less accurate results, i.e. lower CP values with the same interval size, on the testing set depending on the high level of variability in the test set under consideration. In other words, NNs do not always guarantee to generate high quality PIs on the unseen data. In order to strengthen the generalization ability of the MOGA NN algorithm, a further improvement can be done using ensemble of NNs with different structures by performing also a nearest neighbors approach, similar to the one defined in Section 2.4, to select the neighbors of the test pattern from the set of training patterns. In addition, using a validation set for deciding the actual coverage probability of each solution would result in well-calibrated PIs on the testing set. For the sake of clarity, Table 5 reports a synthesis of the comparison aspects addressed above.

5. CONCLUSION

Effective management of smart grids includes several elements of distributed intelligent control on the supplier and the consumer sides. On the supplier side, a successful integration of renewable power sources and handling of the associated uncertainties are pivotal for the reliability of the power network.

For wind power in particular, a crucial element is to have accurate and stable predictions of wind speeds, concurrently quantifying the associated uncertainties of the predictions. .

In this paper, we have proposed and compared two machine learning approaches, MOGA NN and ELM combined with the nearest neighbors approach for estimating prediction intervals. The algorithms have been applied on a case study of short-term wind speed prediction using a real dataset of hourly wind speed measurements.

Contrary to classical time-series prediction approaches, both proposed approaches generate prediction intervals for the target of interest. Knowledge of PIs allows the decision makers and operational planners to efficiently quantify the level of uncertainty associated with the forecasts and to consider a multiplicity of solutions/scenarios for the best and worst conditions.

Both algorithms show a good accuracy and generalization ability on the conducted case study.

The results do not show significant differences in terms of the quality of the predicted PIs. We can conclude that both methods yield a reliable estimation of the PIs with a high coverage and a relatively small interval size.

The approaches to estimate the PIs are based on very different concepts and can be selected depending on the specific requirements of the user, including quality of the results, generalization ability, computational efficiency, flexibility and ease of use.

Generally, if an algorithm is specifically trained to optimize two objectives, it is expected to be superior to an algorithm that was trained to optimize a simple error criterion. However, this could not be observed in this research. The presented results indicate that the generalization ability of the ELM on the training dataset is representative of the performance on new data patterns and multi-objective optimization is, therefore, not required in this case.
Both applied algorithms are data-driven and depend highly on the representativeness of the training dataset. Therefore, the quality of PIs can decrease on datasets with large variability and uncertainty in the data.

A possible direction of future research is to implement online learning algorithms that are able to adjust their parameters while novel patterns evolve, without retraining the whole algorithm. This would be particularly useful for applications, in which the available dataset is too short to cover all possible patterns or in which the environmental or operational conditions change.

REFERENCES

[1] *Magazine*, vol. 8, no.1, pp. 18-28, 2010.
[2] World Wind Energy Association, Half Year Report. Oct. 2013, 1-22. http://www.wwindea.org/webimages/WorldWindEnergyReport2012_final.pdf; [Accessed on October 2013].
[3] X. Wang, P. Guo, and X. Huang, "A review of wind power forecasting models," *Energy Procedia*, vol. 12, pp. 770-778, 2011.
[4] M. Lei, L. Shiyan, J. Chuanwen, L. Hongling, and Z. Yan, "A review on the forecasting of wind speed and generated power," *Renewable and Sustainable Energy Reviews*, vol. 13, pp. 915-920, 2009.
[5] R.S. Tarade and P. K. Katti, "A comparative analysis for wind speed prediction," in *Proc. International Conference on Energy, Automation and Signal*, Orissa India, Dec. 2011, pp. 556-561.
[6] A. M. Foley, P. G. Leahy, A. Marvuglia, and E. J. McKeogh, "Current methods and advances in forecasting of wind power generation," *Renewable Energy*, vol. 37, pp. 1-8, 2012.
[7] I. G. Damousis, M. C. Alexiadis, J. B. Theocharis, P. S. Dokopoulos, "A fuzzy model for wind speed prediction and power generation in wind parks using spatial correlation," *IEEE Trans. Energy Convers.*, vol. 19 pp. 352- 361, 2004.
[8] W. Zhang, J. Wang, J. Wang, Z. Zhao, and M. Tian, "Short-term wind speed forecasting based on a hybrid model," *Applied Soft Computing*, vol. 13, pp. 3225-3233, 2013.

[9] L. Gong and J. Shi, "On comparing three artificial neural networks for wind speed forecasting," *Applied Energy*, vol. 87, pp. 2313-2320, 2010.
[10] H. Liu, H. Q. Tian, D. F. Pan, and Y. F. Li, "Forecasting models for wind speed using wavelet, wavelet packet, time series and Artificial Neural Networks," Applied Energy, vol. 107, pp. 191-208, 2013.
[11] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya, "Comprehensive review of neural network-based prediction intervals and new advances," IEEE Transactions on Neural Networks, vol. 22, no. 9, pp. 1341-1356, Sep. 2011.
[12] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya, "Lower Upper Bound Estimation Method for Construction of Neural Network-Based Prediction Intervals," IEEE Transactions on Neural Networks, vol. 22, no. 3, pp. 337-346, March 2011.
[13] P. Pinson and G. Kariniotakis, "Conditional Prediction Intervals on Wind Power Generation," IEEE Transactions on Power Systems, vol. 25, no. 4, pp. 1845-1856, Nov. 2010.
[14] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-II," IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182-197, Apr. 2002.
[15] R. Ak, Y. F. Li, V. Vitelli, E. Zio, E. López Droguett, and C. Magno Couto Jacinto, "NSGA-II-trained neural network approach to the estimation of prediction intervals of scale deposition rate in oil & gas equipment," Expert Systems with Applications, vol. 40, no. 4, pp. 1205-1212, March 2013.
[16] S. Salcedo-Sanz, E. G. Ortiz-Garcia, A. M. Pérez-Bellido, A. Portilla-Figueras, and L. Prieto, "Short term wind speed prediction based on evolutionary support vector regression algorithms," Expert Systems with Applications, vol. 38, no. 4, pp. 4052-4057, 2011.
[17] D. L. Shrestha and D. P. Solomatine, "Machine learning approaches for estimation of prediction interval for the model output," Neural Networks, vol. 19, no. 2, pp. 225-235, 2006.
[18] K. Hornik, M. Stinchcombe, and H. White. "Multilayer feedforward networks are universal approximators," Neural Networks, vol. 2, no.5, pp. 359-366, 1989.
[19] X-S. Zhang, Neural Networks in Optimization. The Netherlands, Kluwer Academic Publisher, 2000.
[20] C. M. Bishop, Neural networks for Pattern Recognition. New York, Oxford university press, 1995, pp. 1-477.
[21] P. Arabie, J. H. Lawrence, and G. De Soete, eds. Clustering and Classification. Singapore, World Scientific Publishing, 1996.
[22] T. Sorsa, H. N. Koivo, and H. Koivisto. "Neural networks in process fault diagnosis," IEEE Transactions on Systems, Man and Cybernetics, vol. 21, no. 4, pp. 815-825, Jul/Aug. 1991.
[23] D. Svozil, V. Kvasnicka, and J. Pospichal, "Introduction to multi-layer feed-forward neural networks," Chemometrics and Intelligent Laboratory Systems, vol. 39, pp. 43-62, 1997.
[24] R. Rojas, Neutral Networks: A Systematic Introduction. Springer, 1996. Chapter 7.
[25] G. B. Huang, "Learning capability and storage capacity of two-hidden-layer feedforward networks," IEEE Transactions on Neural Networks, vol. 14, no. 2, pp. 274-281, Mar. 2003.
[26] Y. Sawaragi, H. Nakayama, and T. Tanino. Theory of Multiobjective Optimization, Orlando, FL: Academic Press Inc., 1985, pp. 1-296.
[27] N. Sirinivas, and K. Deb, "Multi-objective optimization using non-dominated sorting in genetic algorithms," Journal of Evolutionary Computation, vol. 2, pp. 221-248, 1994.
[28] C. A. C. Coello, G. B. Lamont, and D. A. Van Veldhuizen. Evolutionary algorithms for solving multi-objective problems, 2nd ed. D. E. Goldberg and J.R. Koza, Eds., USA: Springer, 2007.
[29] A. Konak D.W. Coit and A.E. Smith "Multi-objective optimization using genetic algorithms: A tutorial," Reliability Engineering & System Safety, vol. 91, no. 9, pp. 992–1007, Sep. 2006.
[30] M. T. Jensen, "Reducing the run-time complexity of multiobjective EAs: The NSGA-II and other algorithms," IEEE Transactions on Evolutionary Computation, vol. 7, no. 5, pp. 503-515, Oct. 2003.
[31] S. Briesemeister, J. Rahnenführer, and O. Kohlbacher, "No longer confidential: Estimating the confidence of individual regression predictions," PloS one, vol. 7, no. 11, e48723, Nov. 2012.
[32] G.B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 42, no. 2, pp. 513-529, Apr. 2012.
[33] J.H. Friedman, J.L. Bentley, and R.A. Finkel, "An algorithm for finding best matches in logarithmic expected time," ACM Transactions on Mathematical Software (TOMS) vol. 3, no. 3, pp. 209-226, 1997.
[34] G.B. Huang, Q.Y. Zhu, and C.K. Siew, "Extreme learning machine: Theory and applications," Neurocomputing, vol. 70, pp. 489-501, 2006.
[35] G.B. Huang, D. Wang, and Y. Lan, "Extreme learning machines: a survey," International Journal of Machine Learning and Cybernetics, vol. 2, no. 2, pp. 107-122, 2011.
[36] G.B. Huang and L. Chen, "Convex incremental extreme learning machine," Neurocomputing, vol. 70, no.16, pp. 3056-3062, 2007.

[37] G.B. Huang and L. Chen, "Enhanced random search based incremental extreme learning machine," Neurocomputing, vol.71, no.16, pp.3460-3468, 2008.
[38] G.B. Huang, L. Chen, and C.K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," IEEE Transactions on Neural Networks, vol. 17, pp. 879-892, July 2006.
[39] G.B. Huang, Q.Y. Zhu, and C.K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in Proc. IEEE International Joint Conference on Neural Networks, vol. 2, pp. 985-990, 2004.
[40] B. P. Chacko, V. R. V. Krishnan, G. Raju, and P. B. Anto, "Handwritten character recognition using wavelet energy and extreme learning machine," International Journal of Machine Learning and Cybernetics, vol. 3, no.2, pp. 149–161, June 2012.
[41] O. Fink, E. Zio, and U. Weidmann, "Extreme learning machines for predicting operation disruption events in railway systems," in Proc. European Safety and Reliability Conference (ESREL), Amsterdam, Netherlands, 2013.
[42] Y. Lan, Y.C. Soh, and G.B. Huang, "Ensemble of online sequential extreme learning machine," Neurocomputing, vol. 72, no. 13-15, pp. 3391 – 3395, Aug. 2009.
[43] Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten, and A. Lendasse, "Op-elm: Optimally pruned extreme learning machine," IEEE Transactions on Neural Networks, vol. 21, no. 1, pp. 158–162, Jan. 2010.
[44] Website: http://www.weatheroffice.gc.ca/canada_e.html, (Dec., 2012).
[45] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel. Time Series Analysis, Forecasting and Control, fourth ed., Wiley, 2008.
[46] A.E. Hoerl and R.W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," Technometrics, vol. 12, pp. 55-67, 1970.
[47] Q. Hao, D. Srinivasan, and A. Khosravi. "Short-Term Load and Wind Power Forecasting Using Neural Network-Based Prediction Intervals,"
[48] A. Khosravi and S. Nahavandi, "Combined Nonparametric Prediction Intervals for Wind Power Generation," IEEE Transactions on Sustainable Energy, vol. 4, no. 4, pp. 849-856, Oct. 2013.
[49] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," in Cognitive Modeling, Chapter 8. T. A. Polk and C. M. Seifert, Eds., 2002, pp. 213-220.

PAPER IV

An Interval-Valued Neural Network Approach for Prediction Uncertainty Quantification

R. Ak, V. Vitelli and E. Zio. (2014), resubmitted to *IEEE Transactions on Neural Networks and Learning Systems* (under review).

An Interval-Valued Neural Network Approach for Prediction Uncertainty Quantification

Ronay Ak[a], Valeria Vitelli[b], and Enrico Zio[a, c], *Senior Member, IEEE*

[a]Chair on Systems Science and the Energetic Challenge, European Foundation for New Energy-Electricité de France, CentraleSupélec, Châtenay-Malabry 92290 and Gif-Sur-Yvette 91192, France

[b]Department of Biostatistics, University of Oslo, Oslo, Norway

[c]Department of Energy, Politecnico di Milano, Milan 20133, Italy

ABSTRACT

We consider the task of performing prediction with neural networks on the basis of uncertain input data expressed in the form of intervals. We aim at quantifying the uncertainty in the prediction arising from both the input data and the prediction model. A multi-layer perceptron neural network (NN) is trained to map interval-valued input data into interval outputs, representing the prediction intervals (PIs) of the real target values. The NN training is performed by non-dominated sorting genetic algorithm–II (NSGA-II), so that the PIs are optimized both in terms of accuracy (coverage probability) and dimension (width). Demonstration of the proposed method is given on two case studies: (i) a synthetic case study, in which the data have been generated with a 5-min time frequency from an Auto-Regressive Moving Average (ARMA) model with either Gaussian or Chi-squared innovation distribution; (ii) a real case study, in which experimental data consist in wind speed measurements with a time-step of 1-hour. Comparisons are given with a crisp (single-valued) approach. The results show that the crisp approach is less reliable than the interval-valued input approach in terms of capturing the variability in input.

Keywords: Interval-valued neural networks, multi-objective genetic-algorithm, prediction intervals, short-term wind speed forecasting, uncertainty.

1. INTRODUCTION

Prediction plays a crucial role in every decision-making process, and for this reason it should take into account any source of uncertainty that may affect its outcome. Prediction uncertainty can arise due to measurement errors, lack of knowledge in input data, and model approximation errors (e.g. due to imperfections in the model formulation) [1]-[3]. For practical purposes, uncertainties can be classified in two distinct types [3]: epistemic and aleatory. The former derives from imprecise model representation of the system behavior, in terms of uncertainty in both the hypotheses assumed (structural uncertainty) and the values of the model parameters (parameter uncertainty) [4].The latter describes the inherent variability of the observed physical phenomenon, and it is therefore also named stochastic uncertainty, irreducible uncertainty, or inherent uncertainty [5].

Uncertainty quantification is the process of representing the uncertainty in the system inputs and parameters, propagating it through the model, and then revealing the resulting uncertainty in the model outcomes [2].

In the literature, methods such as probability modeling [6], Neural Networks-based prediction intervals estimation [7]-[11], conformal prediction [12], [13], interval analysis [14]-[16], fuzzy set theory [17], and in particular type-2 fuzzy sets [18] and interval type-2 fuzzy logic systems [19], as well as L-fuzzy mathematical morphology [20] and extensions of fuzzy mathematical morphology [21], Monte Carlo simulation [22], and Latin hypercube sampling [23] have been used to efficiently represent, aggregate, and propagate different types of uncertainty through computational models. Interval analysis is a powerful technique for bounding solutions under uncertainty. The uncertain model parameters are described by upper and lower bounds, and the corresponding bounds in the model output are computed using interval functions and interval arithmetic [24]. These bounds contain the true target value with a certain confidence level. The interval-valued representation can also be used to reflect the variability in the inputs (e.g. extreme wind speeds in a given area, minimum and maximum of daily temperature, etc.), or their associated uncertainty (e.g. strongly skewed wind speed distributions, etc.), i.e. to express the uncertain information associated to the input parameters [14]-[16], [25].

In this paper, we present an interval-valued time series prediction modeling framework based on a data-driven learning approach, more specifically a multi-layer perceptron neural network (NN). Demonstration of the proposed method is given on two case studies: (i) a synthetic case study, with 5-minutes simulated data; (ii) a real case study, involving hourly wind speed measurements. In both cases, short-term prediction (1-hour and day-ahead, respectively) is performed taking into account both the uncertainty in the model structure, and the variability (within-hour and within-day, respectively) in the inputs.

The wind speed prediction case study has been chosen because of its relevance for wind power production. Wind power variations in short-term time scales have significant effects on power system operations such as regulation, load following, balancing, unit commitment and scheduling [8], [26], [27]. Thus, accurate prediction of wind speed and its uncertainty is critical for the safe, reliable and economic operation of power systems [26], [27]. Prediction Intervals (PIs) are preferable results of the prediction, rather than point estimates, because they provide information on the confidence in the prediction [7]-[11].

An interval representation has been given to the hourly and daily inputs by using two different approaches (see Section 4), which quantify in two different ways the within-hour and within-day variability. The network maps interval-valued input data into an interval output, providing the estimated prediction intervals (PIs) for the real target. PIs are comprised of lower and upper bounds within which the actual target is expected to lie with a predetermined probability [7]-[11]. The NN prediction model is trained by a multi-objective genetic algorithm (MOGA) (the powerful non-dominated sorting genetic algorithm-II, NSGA-II), so that the PIs are optimal both in terms of accuracy (coverage probability) and dimension (width).

The prediction interval coverage probability (PICP) represents the probability that the set of estimated PI values will contain a certain percentage of the true output values. Prediction interval width (PIW) simply measures the extension of the interval as the difference of the estimated upper and lower bound values. The network uses interval-valued data but its weights and biases are crisp (i.e. single-valued). The NSGA-II training procedure generates Pareto-optimal solution sets, which include non-dominated solutions for the two objectives (PICP and PIW).

The originality of the work appears in two aspects: (i) while the existing papers on short-term wind speed/power prediction use single-valued data as inputs, obtained as a within-hour [11], [26] or within-day average [28], [29], we give an interval representation to hourly/daily inputs by using two approaches (see Section 4), which properly account (in two different ways) for the within-hour/day variability; (ii) we handle the PIs problem in a multi-objective framework [11], [30], whereas the existing relevant methods for wind speed/power prediction consider only one objective for optimization. It is worth recalling that in [11], we have performed a comparison with single-objective genetic algorithm (SOGA) and single-objective simulated annealing (SOSA) methods. SOSA has been proposed in support of the LUBE method in [7]. The comparison results show that the PIs produced by NSGA-II compare well with those obtained by LUBE and are satisfactory in both objectives of high coverage and small width. In [30], we have implemented the NSGA-II to train a NN to provide the PIs of the scale deposition rate. We have performed k-fold cross-validation to guide the choice of the NN structure (i.e. the number of hidden neurons) with good generalization performance. We have used a hypervolume indicator metric to compare the Pareto fronts obtained in each cross-validation fold. All these analyses have been done with single-valued inputs in both works.

The paper is organized as follows. Section 2 introduces the basic concepts of interval-valued NNs for PIs estimation. In Section 3, basic principles of multi-objective optimization are briefly recalled and the use of NSGA-II for training a NN to estimate PIs is illustrated. Experimental results on the synthetic case study and on the real case study concerning wind speed prediction are given in Section 4. Finally, Section 5 concludes the paper with a critical analysis of the results and some ideas for future studies.

2. NEURAL NETWORKS AND PREDICTION INTERVALS

Neural networks (NNs) are a class of nonlinear statistical models inspired by brain architecture, capable of learning complex nonlinear relationships among variables from observed data. This is done by a process of parameter tuning called "training". It is common to think of a NN model as a way of solving a nonlinear regression problem of the kind [31], [32]:

$$y = f(x; w^*) + \varepsilon(x), \quad \varepsilon(x) \sim N(0, \sigma_\varepsilon^2(x)) \tag{1}$$

where x, y are the input and output vectors of the regression, respectively, and w^* represents the vector of values of the parameters of the model function f, in general nonlinear. The term $\varepsilon(x)$ is the error associated with the regression model f, and it is assumed normally distributed with zero mean. For simplicity of illustration, in the following we assume y one-dimensional. An estimate $\hat{w}$ of w^* can be obtained by a training procedure aimed at minimizing the quadratic error function on a training set of input/output values $D = \{(x_i, y_i), i = 1,2, \dots, n_p\}$,

$$E(w) = \sum_{i=1}^{n_p} (\hat{y}_i - y_i)^2 \quad (2)$$

where $\hat{y}_i = f(x_i; \hat{w})$ represents the output provided by the NN in correspondence of the input x_i and n_p is the total number of training samples.

A PI is a statistical estimator composed by upper and lower bounds that include a future unknown value of the target $y(x)$ with a predetermined probability, called confidence level in literature [7]-[11].

To evaluate the quality of the PIs, we take the prediction interval coverage probability (PICP) and the prediction interval width (PIW) [7], [10] as measures: the former represents the probability that the set of estimated PIs will contain the true output values $y(x)$ (to be maximized), and the latter simply measures the extension of the interval as the difference of the estimated upper bound and lower bound values (to be minimized). In general, these two measures are conflicting (i.e., wider intervals give larger coverage), but in practice it is important to have narrow PIs with high coverage probability [7].

When interval-valued data [24] are used as input, each input pattern x_i is represented as an interval $x_i = [x_i^-, x_i^+]$ where $x_i^- \leq x_i^+$ are the lower and upper bounds (real values) of the input interval, respectively. Each estimated output value $\hat{y}_i$ corresponding to the $i-th$ sample x_i is, then, described by an interval as well, $\hat{y}_i = [\hat{y}_i^-, \hat{y}_i^+]$, where $\hat{y}_i^- \leq \hat{y}_i^+$ are the estimated lower and upper bounds of the PI in output, respectively.

The mathematical formulation of the PICP and PIW measures given by [7] is modified for

interval-valued input and output data:

$$PICP = \frac{1}{n_p}\sum_{i=1}^{n_p} c_i \tag{3}$$

where n_p is the number of training samples in the considered input dataset, and

$$c_i = \begin{cases} 1 & y_i \subseteq [\hat{y}_i^-, \hat{y}_i^+] \\ \frac{diam(y_i \cap \hat{y}_i)}{diam(y_i)} & y_i \not\subset [\hat{y}_i^-, \hat{y}_i^+] \wedge y_i \cap \hat{y}_i \neq \emptyset \\ 0 & otherwise \end{cases} \tag{4}$$

where $y_i = [y_i^-, y_i^+]$, $y_i^- \leq y_i^+$ are the lower and upper bounds (true values) of the output interval, respectively, and $diam()$ indicates the width of the interval. More precisely, (4) means that if the interval-valued real target is covered by the estimated PI, i.e. if the target is a subinterval of the estimated PI, then c_i is equal to 1. If the estimated PI does not cover the entire real target, but the intersection of the two is not empty, then c_i is equal to the ratio between $diam(y_i \cap \hat{y}_i)$ and the width of the interval y_i, and in that case c_i takes a values smaller than 1. Finally, if the estimated PI does not cover the entire real target and the intersection of the two is empty, then the coverage c_i of the $i-th$ sample is 0. This calculation corresponds to the probabilistic assumption that the target y_i can take any value in $[y_i^-, y_i^+]$ with uniform probability, i.e. that each point in $[y_i^-, y_i^+]$ is equally likely to be a possible value of *y*.

For PIW, we consider the normalized quantity:

$$NMPIW = \frac{1}{n_p}\frac{\sum_{i=1}^{n_p}(\hat{y}_i^+ - \hat{y}_i^-)}{y_{max} - y_{min}} \tag{5}$$

where NMPIW stands for Normalized Mean PIW, and y_{min} and y_{max} represent the minimum and maximum values of the true targets (i.e., the bounds of the range in which the true values fall). Normalization of the PI width by the range of targets makes it possible to objectively compare the PIs, regardless of the techniques used for their estimation or the magnitudes of the true targets.

3. NON-DOMINATED SORTING GENETIC ALGORITHM-II (NSGA-II) MULTI-OBJECTIVE OPTIMIZATION FOR NEURAL NETWORK TRAINING

The problem of finding PIs optimal both in terms of coverage probability and width can be formulated in a multi-objective optimization framework considering the two conflicting objectives PICP and NMPIW.

3.1. Multi-objective Optimization by NSGA-II

In all generality, a multi-objective optimization problem considers a number of objectives, $f_m,\ m = 1, 2, \dots, M$, inequality $g_j,\ j = 1, 2, \dots, J$ and equality $h_k,\ k = 1, 2, \dots, K$ constraints, and bounds on the decision variables $x_i,\ i = 1, 2, \dots, I$. Mathematically the problem can be written as follows [33]:

$$\textit{Minimise/Maximise} \quad f_m(x), \qquad m = 1,2, \dots, M; \tag{6}$$

$$\textit{subject to} \quad g_j(x) \geq 0, \quad j = 1,2, \dots, J; \tag{7}$$

$$h_k(x) = 0, \quad k = 1,2, \dots, K; \tag{8}$$

$$x_i^{(l)} \leq x_i \leq x_i^{(u)} \quad i = 1,2, \dots, I. \tag{9}$$

A solution, $x = \{x_1, x_2, \dots x_I\}$ is an I-dimensional decision variable vector in the solution space R^I, restricted by the constraints (7), (8) and by the bounds on the decision variables (9). The search for optimality requires that the M objective functions $f_m(x),\ m = 1, 2, \dots, M$ be evaluated in correspondence of the decision variable vector x in the search space. The comparison of solutions during the search is performed in terms of the concept of dominance [33]. Precisely, in case of a minimization problem, solution x_a is regarded to dominate solution x_b *($x_a \succ x_b$)* if the following conditions are satisfied:

$$\forall i \in \{1,2, \dots, M\}: f_i(x_a) \leq f_i(x_b) \ \wedge \tag{10}$$

$$\exists j \in \{1,2, \dots, M\}: f_j(x_a) < f_j(x_b) \tag{11}$$

If any of the above two conditions is violated, the solution x_a does not dominate the solution x_b, and x_b is said to be non-dominated by x_a. Eventually, the search aims at identifying a set of optimal solutions $x^* \in R^I$ which are superior to any other solution in the search space with

respect to all objective functions, and which do not dominate each other. This set of optimal solutions is called Pareto optimal set; the corresponding values of the objective functions form the so called Pareto-optimal front in the objective functions space.

In this work, we use GA for the multi-objective optimization. GA is a population based meta-heuristics inspired by the principles of genetics and natural selection [34]. It can be used for solving multi-objective optimization problems [35], [36]. Among the several options for MOGA, we adopt NSGA-II, as comparative studies show that it is very efficient [34], [37].

3.2. Implementation of NSGA-II for training a NN for Estimating PIs

In this work, we extend the method described in [7] to a multi-objective framework for estimating output PIs from interval-valued inputs. More specifically, we use NSGA-II for finding the values of the parameters of the NN which optimize two objective functions PICP (3) and NMPIW (5) in a Pareto optimality sense (for ease of implementation, the maximization of PICP is converted to minimization by subtracting from one, i.e. the objective of the minimization is 1-PICP).

The practical implementation of NSGA-II on our specific problem involves two phases: initialization and evolution. These can be summarized as follows (for more details on the NSGA-II implementation see [30]):

Initialization phase:

Step 1: Split the input data into training (D_{train}) and testing (D_{test}) subsets.
Step 2: Fix the maximum number of generations and the number of chromosomes (individuals) Nc in each population; each chromosome codes a solution by G real-valued genes, where G is the total number of parameters (weights) in the NN. Set the generation number $n = 1$. Initialize the first population P_n of size Nc, by randomly generating Nc chromosomes.
Step 3: For each input vector x in the training set, compute the lower and upper bound outputs of the Nc NNs, each one with G parameters.
Step 4: Evaluate the two objectives PICP and NMPIW for the Nc NNs (one pair of values 1-PICP and NMPIW for each of the Nc chromosomes in the population P_n).

Step 5: Rank the chromosomes (vectors of G values) in the population P_n by running the fast non-dominated sorting algorithm [37] with respect to the pairs of objective values, and identify the ranked non-dominated fronts $F_1, F_2, \dots, F_k$ where F_1 is the best front, F_2 is the second best front and F_k is the least good front.

Step 6: Apply to P_n a binary tournament selection based on the crowding distance [37], for generating an intermediate population S_n of size Nc.

Step 7: Apply the crossover and mutation operators to S_n, to create the offspring population Q_n of size Nc.

Step 8: Apply Step 3 onto Q_n and obtain the lower and upper bound outputs.

Step 9: Evaluate the two objectives in correspondence of the solutions in Q_n, as in Step 4.

Evolution phase:

Step 10: If the maximum number of generations is reached, stop and return P_n. Select the first Pareto front F_1 as the optimal solution set. Otherwise, go to Step 11.

Step 11: Combine P_n and Q_n to obtain a union population $R_n = P_n \cup Q_n$.

Step 12: Apply Steps 3-5 onto R_n and obtain a sorted union population.

Step 13: Select the Nc best solutions from the sorted union to create the next parent population P_{n+1}.

Step 14: Apply Steps 6-9 onto P_{n+1} to obtain Q_{n+1}. Set $n = n + 1$; and go to Step 10.

Finally, the best front in terms of non-dominance and diversity of the individual solutions is chosen. Once the best front is chosen, the testing step is performed on the trained NN with optimal weight values.

The total computational complexity of the proposed algorithm depends on two sub-operations: non-dominated sorting and fitness evaluation. The time complexity of non-dominated sorting is $O(MNc^2)$, where M is the number of objectives and Nc is the population size [37]. In the fitness evaluation phase, NSGA-II is used to train a NN which has n_p input samples. Since for each individual of the population a fitness value is obtained, this process is repeated $Nc \times n_p$ times. Hence, time complexity of this phase is $O(Nc \times n_p)$. In conclusion, the computational complexity of one generation is $O\left(MNc^2 + Nc \times n_p\right)$.

4. EXPERIMENTS AND RESULTS

Two case studies have been considered: a synthetic case study, consisting of four time series datasets generated according to different input variability scenarios, and a real case study concerning time series of wind speed data. The synthetic time series datasets have been generated with a 5-min time frequency from an Auto-Regressive Moving Average (ARMA) model with either Gaussian or Chi-squared innovation distribution. For what concerns the real case study, hourly measurements of wind speed for a period of 3 years (from 2010 to 2012) related to Regina, a region of Canada, have been used [38].

The synthetic case study is aimed at considering hourly data and the effects of within-hour variability. Hourly interval input data is obtained from the 5-min time series data by two different approaches, which we refer to as "min-max" and "mean": the former obtains hourly intervals by taking the minimum and the maximum values of the 5-min time series data within each hour; the latter, instead, obtains one-standard deviation intervals $[\overline{x}_i - s_i, \overline{x}_i + s_i]$ by computing the sample mean $(\overline{x}_i)$ and standard deviation (s_i) of each 12 within-hour 5-min data sample. Single-valued (crisp) hourly input have also been obtained as a within-hour average, i.e. by taking the mean of each 12 within-hour 5-min data sample, for comparison. The wind speed case study considers the effect of within-day variability, and min-max and mean approaches are applied to the 24 within-day hourly data samples.

The architecture of the NN model consists of one input, one hidden and one output layers. The number of input neurons is set to 2 for both case studies, since an auto-correlation analysis [39] has shown that the historical past values x_{t-1} and x_{t-2} should be used as input variables for predicting x_t in output. The number of hidden neurons is set to 10 for the synthetic case study and to 15 for the real case study, after a trial-and-error process. The number of output neurons is 1 in the input-interval case, since in this case a single neuron provides an interval in output; conversely, in order to estimate PIs starting from crisp input data, the number of output neurons must be set equal to 2, to provide the lower and upper bounds. As activation functions, the hyperbolic tangent function is used in the hidden layer and the logarithmic sigmoid function is used at the output layer. We remark that all arithmetic calculations throughout the estimation process of the interval-valued NN have been performed according to interval arithmetic (interval product, sum, etc.).

To account for the inherent randomness of NSGA-II, 5 different runs of this algorithm have been performed and an overall best non-dominated Pareto front has been obtained from the 5 individual fronts. To construct such best non-dominated front, the first (best) front of each of the 5 runs is collected, and the resulting set of solutions is subjected to the fast non-dominated sorting algorithm [37] with respect to the two objective functions. Then, the ranked non-dominated fronts $F_1, F_2, \dots, F_k$ are identified, where F_1 is the best front, F_2 is the second best front and F_k is the worst front. Solutions in the first (best) front F_1 are then retained as the overall best front solutions. This procedure gives us the overall best non-dominated Pareto front for the training set. After we have obtained this overall best front, we perform testing using each solution included in it.

For the first case study, the first 80% of the input data have been used for training and the rest for testing. For the second, a validation process has been performed. So the dataset has been divided into three parts: the first 60% is used for training, 20% for validation and the remaining 20% for testing. All data have been normalized within the range [0.1, 0.9].

Table 1 contains the parameters of the NSGA-II for training the NN. "MaxGen" indicates the maximum number of generations which is used as a termination condition and Nc indicates the total number of individuals per population. P_c indicates the crossover probability and is fixed during the run. P_{m_int} is the initial mutation probability and it decreases at each iteration (generation) by the formula:

$$P_{m\_int} \times e^{\left(-\frac{gen}{MaxGen}\right)} \tag{12}$$

4.1. Synthetic Case Study

Four synthetic datasets have been generated according to the following model:

$$y(t) = f(t) + \delta(t), \tag{13}$$

where $f(t)$ is the deterministic component and $\delta(t)$ is the stochastic one, and the time horizon is 50 days which makes 1200 hours. The deterministic component has the following

expression:

$$f(t) = 10 + 1.5 * sin\left(\frac{2\pi t}{T_1}\right) + sin\left(\frac{2\pi t}{T_2}\right), \quad (14)$$

where the period T_1 of the first periodic component has been set equal to 1 week, while T_2 is 1 day. The stochastic component $\delta(t)$ of the generating model in (13) is given by an $ARMA(p, q)$ model [39], with $p = 2$ autoregressive terms, with same coefficients $\phi_1 = \phi_2 = 0.1$, and $q = 1$ innovation term with coefficient given by $\varphi_1 = 0.05$. Four different scenarios are then considered, which differ in the distribution chosen for the innovation term, and in the higher or lower innovation variability: in two of the four scenarios the innovation is Gaussian, and has variance equal to 1 and 9 respectively, while in the other two scenarios the innovation has a Chi-squared distribution, with 2 or 5 degrees of freedom (corresponding to a variance equal to 4 and 10, respectively). We thus generate four different 5-min time series datasets, from which we will obtain either crisp or interval hourly data.

Table 1. NSGA-II and SOSA Parameters Used in the Experiments

Parameter	**Numerical value**
MaxGen	300
Nc	50
P_{m_int}	0.06
μ	0.9
η	50
T_{init}	200
T_{min}	10^{-50}
CWC_{int}	10^{80}
Geometric cooling schedule of SA	$T_{k+1} = T_k * 0.95$

Fig. 1 illustrates the testing solutions corresponding to the first (best) Pareto front found after training the NN on interval data constructed by the min-max approach (left) and mean approach (right). The plots show the solutions for the data generated from a Gaussian distribution. On each plot, two testing fronts are illustrated: the ones where solutions are marked as circles have been obtained after training the NN on the interval data showing

higher variability, while the ones with solutions marked as diamonds have been obtained after training the NN on the interval data having lower variability. Testing solutions obtained with data showing a lower variability are better than the ones with higher variability; hence, we can conclude that a higher variability in the input data may cause less reliable prediction results, and should thus be properly taken into account. Pareto fronts of solutions obtained for the data generated from a Chi-squared distribution are similar, and the results robust with respect to the choice of the innovation distribution.

Given the overall best Pareto set of optimal model solutions (i.e. optimal NN weights), it is necessary to select one NN model for use. For exemplification purposes, a solution is here subjectively chosen as a good compromise in terms of high PICP and low NMPIW. The selected solution is characterized by 95% CP and a NMPIW equal to 0.420 for the min-max approach applied to lower variability Gaussian data. The results on the testing set give a coverage probability of 95.5 % and an interval width of 0.412. Fig. 2 shows 1-hour-ahead PIs for the selected Pareto solution, estimated on the testing set by the trained NN; the interval-valued targets included in the testing set are also shown in the figure.

Moreover, we also plot in Fig. 3 the 5-min original time series data (testing set), corresponding to the generating scenario with Gaussian distribution and low variability, together with the estimated PIs corresponding to the selected solution: the solid line shows the 5-min original time series data, while the dashed lines are the PIs, estimated starting from interval input data constructed with the min-max approach within each hour. Since the time step for the estimated PIs is 1 hour, in order to compare them to the 5-min original time series data, we have shown in Fig. 3 the same lower and upper bounds within each hour; thus, the PIs appear as a step function if compared to the original 5-min data.

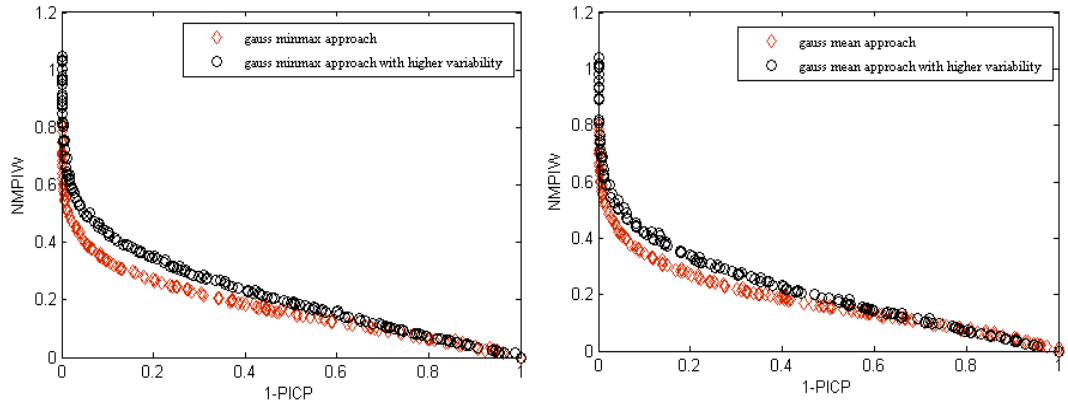


Fig. 1. Testing solutions for the Gaussian time series: min-max approach (left) and mean approach (right).

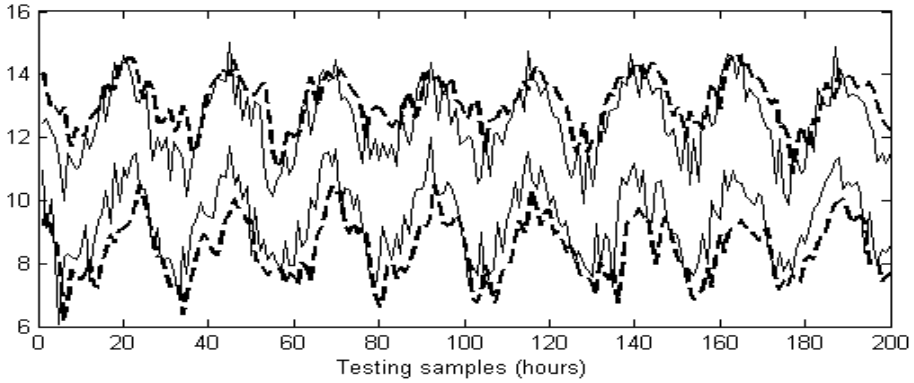


Fig. 2. Estimated PIs for 1-h ahead prediction on the testing set (dashed lines), and interval-valued input data (target) constructed by the min-max approach from the Gaussian distribution scenario with lower variability on the testing set (solid lines).

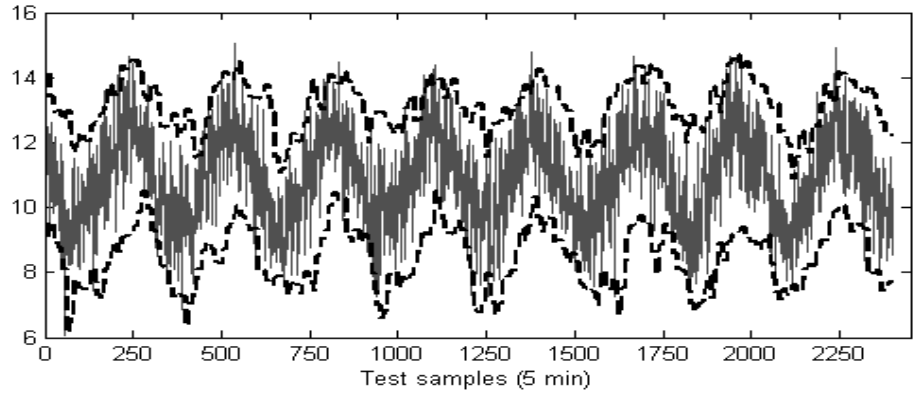


Fig. 3. Estimated PIs for 1-h ahead prediction on the testing set (dashed lines), and the original 5-min time series data on the testing set (solid line) obtained in the Gaussian distribution scenario with lower variability.

In order to compare the Pareto front optimal solutions obtained with crisp and interval-valued inputs, a new normalized measure of the mean prediction interval width, named NMPIW*, has been a posteriori calculated as follows:

$$NMPIW^* = \frac{RT}{RRT} \times \frac{NRT}{0.8} \times NMPIW \quad (15)$$

where RT, RRT and NRT represent, respectively, the range of target (i.e., the range of the non-normalized hourly training data in input), the range of real target (i.e., the range of the non-normalized 5-min original time series data over the training set), and the range of normalized target (i.e., the range of the normalized hourly training data in input, $y_{max} - y_{min}$). Note that, unless the synthetic scenario changes, RRT takes the same value for min-max, mean and crisp approaches. The idea behind renormalization is to be able to compare PIs estimated from both interval and crisp approaches with respect to 5-min original time series data. As NMPIW for each solution on the Pareto front has been calculated by dividing the mean prediction interval width (MPIW) by the range of the training set in question, which is different for the two approaches, the Pareto fronts corresponding to the two approaches are not comparable. In order to analyze the performance of each approach with respect to 5-min original time series data, one should carry out a renormalization process which takes into account the range of the dataset involved in the comparison, and which leads the estimated PIs to a common unit of measure. As a numerical example for the calculation of NMPIW*, we have considered a testing solution, obtained on the synthetic data generated from the Gaussian distribution with lower variability and with the crisp approach, reported in Fig. 4. The selected solution results in a coverage probability of 91% and an interval width of 0.328 on the testing. The values of RT, RRT and NRT are 6.87, 11.383, and 0.647, respectively. Thus, by using (16), we have obtained NMPIW* as follows:

$$NMPIW^* = \frac{6.87}{11.383} \times \frac{0.647}{0.8} \times 0.328 = 0.16 \quad (16)$$

Moreover, for each solution on each Pareto front, a PICP* value has been a posteriori calculated. Equations (3) and (4) have been used with y_i representing non-normalized 5-min original time series data, and with $c_i = 1$, if $y_i \in [L(x_i), U(x_i)]$ and otherwise $c_i = 0$, where $L(x_i)$ and $U(x_i)$ indicate de-normalized lower and upper bounds of the estimated PIs. Since

estimated PIs have been obtained with hourly input data, while original data have a 5-min time frequency, in order to a posteriori calculate PICP* with respect to the original data we have assumed the same lower and upper bounds, $[L(x_i),U(x_i)]$, for each 5-min time step within each hour. Renormalization allows us to convert current Pareto fronts to new ones whose coverage probability and interval size are calculated according to the 5-min dataset, and are comparable across different (crisp and interval) approaches.

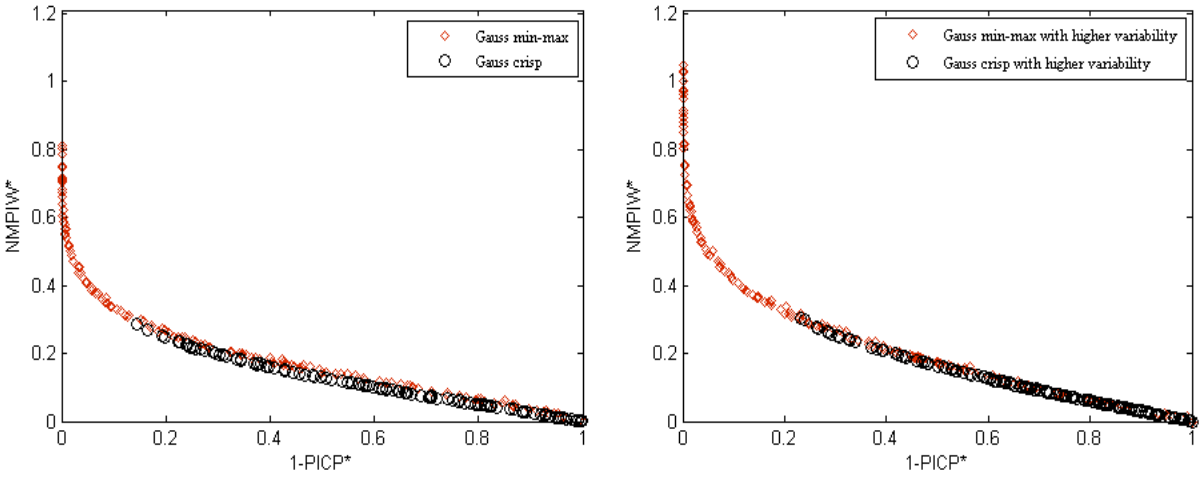


Fig. 4. Testing solutions obtained in the synthetic case study with interval-valued (min-max approach) and crisp approaches: data have been generated from the Gaussian distribution with lower (left) and higher variability (right).

In Fig. 4, a comparison between the testing fronts obtained with interval-valued and crisp inputs are illustrated. Solutions have been plotted according to the renormalized measures, i.e. the axes of the plots correspond to the new quantities NMPIW* and 1-PICP*, so that they can be compared. It can be appreciated that the solutions obtained with a crisp approach never result in coverage probabilities greater than 90% with respect to the original data. Furthermore, when the variability in the original data increases (right plots), the crisp approach gives less reliable results in terms of coverage probability, which is smaller than 80%. However, a model should take the within hour variability (high or low) into account and be capable of properly capturing it. Predictions resulting in a coverage probability lower than expected show the poor prediction power of the crisp approach, which cannot be considered a reliable support to decision making in the presence of high variability.

4.2. Real Case Study: Short-term Wind Speed Prediction

In this Section, results of the application of the proposed method to short-term wind speed forecasting with interval-input data are detailed. The dataset considered for the analysis consists in hourly wind speed data measured in Regina, Saskatchewan, a region of central Canada. Wind farms in Canada are currently responsible of an energy production of 5403 MW, a capacity big enough to power over 1 million homes and equivalent to about 2% of the total electricity demand in Canada [40]. The actual situation in Saskatchewan is characterized by the presence of 4 large wind farms located throughout the region, with a total capacity of approximately 198 MW [41].

The wind speed dataset, covering the period from January 1, 2010 till December 30, 2012, has been downloaded from the website [38]. Since hourly data have been collected, 24 wind speed values are available for each day. Fig. 5 shows the behavior of hourly wind speed values only in the first 20 days, for the sake of clarity: one can appreciate the within-day variability in each individual day. The wind speed changes from 0 km/h to 72 km/h with an unstable behavior. From this raw hourly wind speed data, one can obtain daily interval wind speed data with the min-max and mean approach described at the beginning of Section 4. The so obtained datasets include 1095 intervals among which the first 60% is used for training, 20% for validation and the remaining 20% for testing.

The procedure described in Sections II and III has been applied for day-ahead wind speed prediction, both with interval and crisp inputs. Crisp results are reported for comparison, in terms of daily averages of the raw hourly data, with the same data splitting for training, validation and testing sets. The inputs are historical wind speed data $W_{t-1}$and W_{t-2} both for interval and crisp inputs; the optimal number of inputs has been chosen from an auto-correlation analysis [39].

When an optimal solution is selected from the front obtained by optimizing the NN on the basis of the training data, it is possible that the CP resulting from the application of this optimal NN to unseen data is lower than the one obtained on the training data. Thus, a validation set has been also selected, to test the generalization power of the proposed method. In other words, the aim is to test whether the selection of the solution with the required CP on

the training data will result in well-calibrated PIs on the validation data or not. Fig. 6 shows the values of PICP and NMPIW obtained on the validation set along the iterations of the MOGA (for the min-max approach). To obtain these graphs, at each iteration an optimal solution has been selected from the training front, it has been used on the validation set, and the corresponding PICP and NMPIW values have been recorded. The motivation behind these plots is to show the capability of the MOGA algorithm to generate reliable predictions on unseen data.

Table 2. NSGA-II and SOSA Parameters Used in the Experiments

Training		**Validation**	
PICP (%)	NMPIW	PICP (%)	NMPIW
90.1	0.440	91.0	0.470
93.2	0.487	94.3	0.514
92.1	0.466	93.1	0.494
90.6	0.452	91.6	0.482
91.6	0.456	92.5	0.486
90.3	0.446	92.0	0.474
94.3	0.529	96.0	0.562
93.6	0.493	94.4	0.526

Table 2 reports the PICP and NMPIW values of the selected training and validation solutions corresponding to those having coverage probability between 90% and 95% on the overall best non-dominated Pareto front. These solutions are obtained by the min-max approach. From inspection both of Table 2 and the profiles of both objectives on the training and validation sets shown in Fig. 6, we can observe that the training and testing results do not show significant difference. The PICP evaluation is coherent with NMPIW; hence, we can conclude that the proposed method results in well-calibrated PIs not only on the training set but also on the validation set.

In Fig. 7, the testing solutions obtained with the interval-valued min-max and mean approaches, and with crisp inputs, are illustrated. The figure has been plotted according to the renormalized solutions, as explained in Section 4.1, i.e. the axes of the plot correspond to the new quantities NMPIW* and 1-PICP*. As already appreciated in the synthetic case study,

one can notice that the solutions obtained with a crisp approach do not result in a coverage probability larger than 95% with respect to the original data. Furthermore, looking at the solutions in Fig. 7 which show a CP greater than 90%, the ones corresponding to the crisp approach give larger interval size. Since in practice it is important to have narrow PIs with high coverage probability, an interval-inputs approach is more suited to reliable decision making.

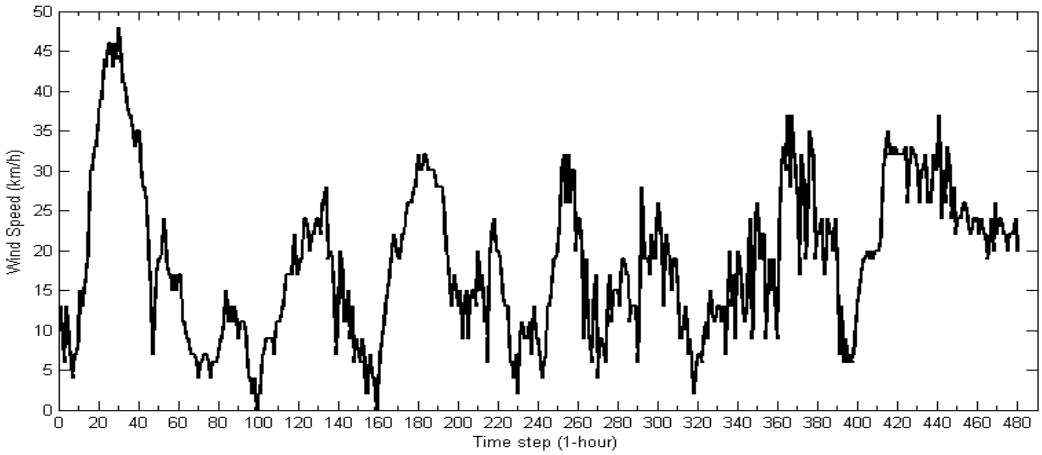


Fig. 5. The raw hourly wind speed dataset used in this study: first 20 days.

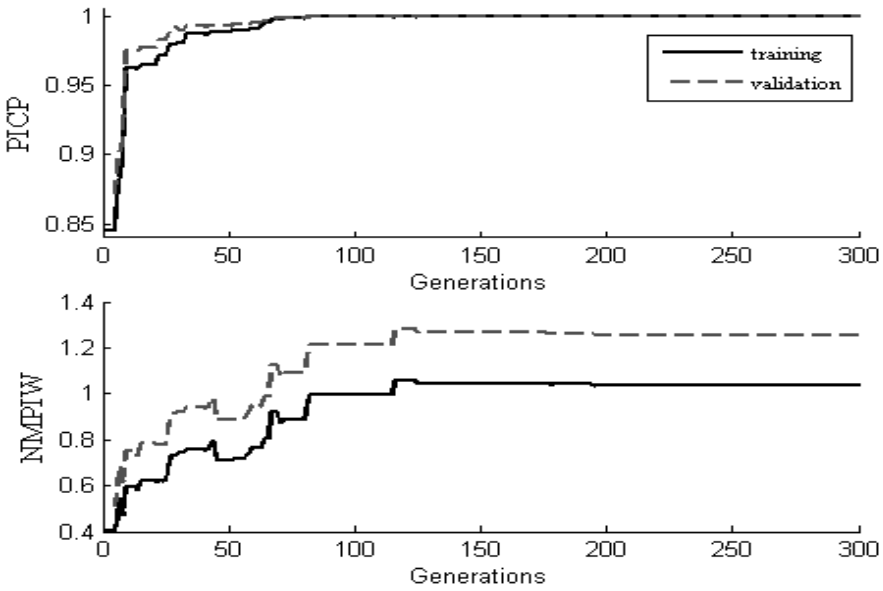


Fig. 6. Evaluation of PICP (top) and NMPIW (bottom) with respect to training and validation sets along MOGA iterations, considering interval inputs obtained with a min-max approach.

From the overall best Pareto set of optimal solutions (i.e. optimal NN weights) obtained after training the network on the interval input data constructed with the min-max and mean approaches, a solution must be chosen. The selection of the solution might be accomplished by setting a constraint on one of the objective and choosing the optimal value for the other one, or by considering some other methods to weigh the two objectives [42]. In general, the selection should represent the preferences of the decision makers (DMs). Here, for simplicity's sake, we do not introduce any specific formal method of preference assignment but subjectively choose a good compromise solution: for the min-max approach, the results give a coverage probability of 92.1% and interval width of 0.466 on the training, and a coverage probability of 93.9% and interval width of 0.48 on the testing. For the mean approach, the selected solution results in a coverage probability of 91.7% and interval width of 0.424 on the training, and a coverage probability of 93% and interval width of 0.437on the testing.

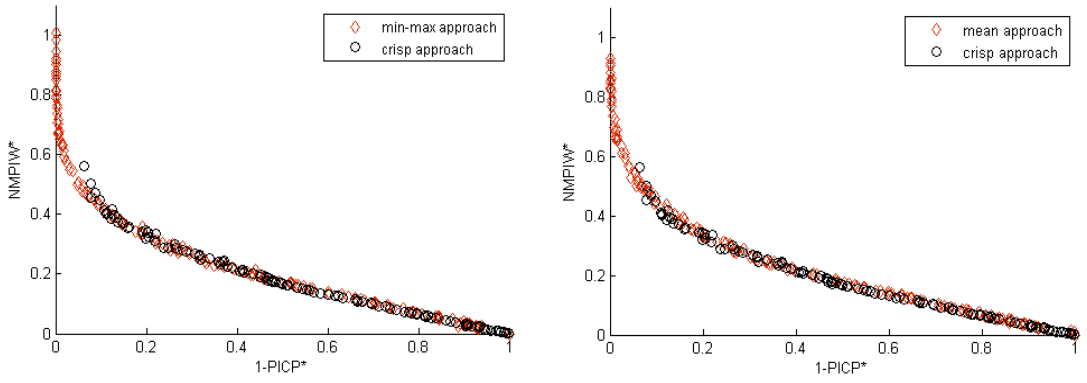


Fig. 7. Comparison between crisp and interval-valued approaches testing solutions, after renormalization, for day-ahead wind speed prediction: min-max with respect to crisp approach comparison (left), and mean with respect to crisp approach comparison (right).

Figs. 8 and 9 report day-ahead PIs (dashed lines) for the selected Pareto solutions, with respect to the mean and min-max approaches respectively, estimated on the testing set by the trained NN. The interval-valued targets (solid lines) included in the testing set are also shown in the figures. As wind speed cannot be negative, to reflect the real physical phenomena the negative lower bounds of the PIs have been replaced with zeros. From inspection of the figures, we observe that the target profile of the mean approach is more accurate if compared to that of the min-max approach. However, the peak points have been covered relatively better by the min-max approach if compared to the mean. Hence, which one would be

preferably chosen depends on the application. The mean approach might be considered more similar to classical methods for short-term wind speed/power prediction using single-valued data as inputs, obtained as a within-hour or within-day average. By this approach we can add information to the single-valued averages, and thus we can include in the model the potential uncertainty caused by the data itself showing a within hour/day variability. Hence, the mean approach is a well-suited interval inputs alternative to the classical crisp inputs one, and it might be considered more feasible in practice.

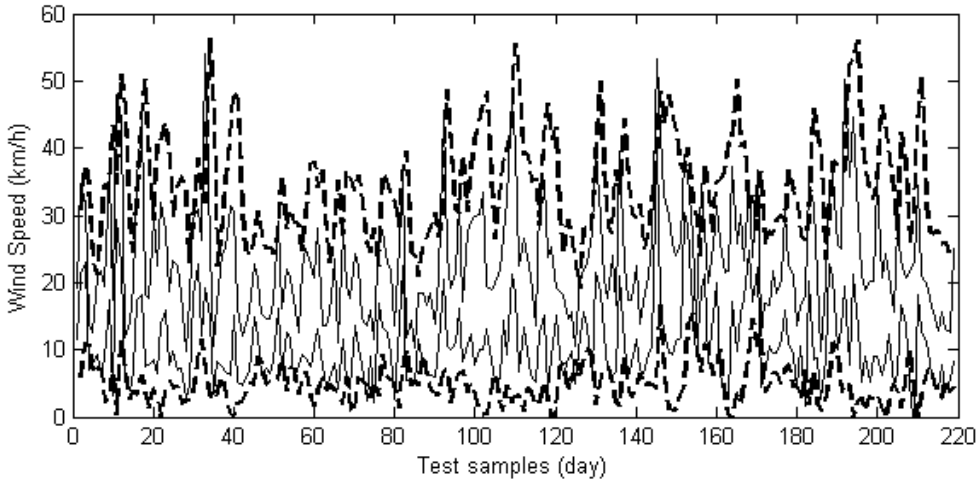


Fig. 8. Estimated PIs with interval inputs for day-ahead wind speed prediction on the testing set (dashed lines), and interval-valued wind speed data (constructed by the mean approach) included in the testing set (solid line).

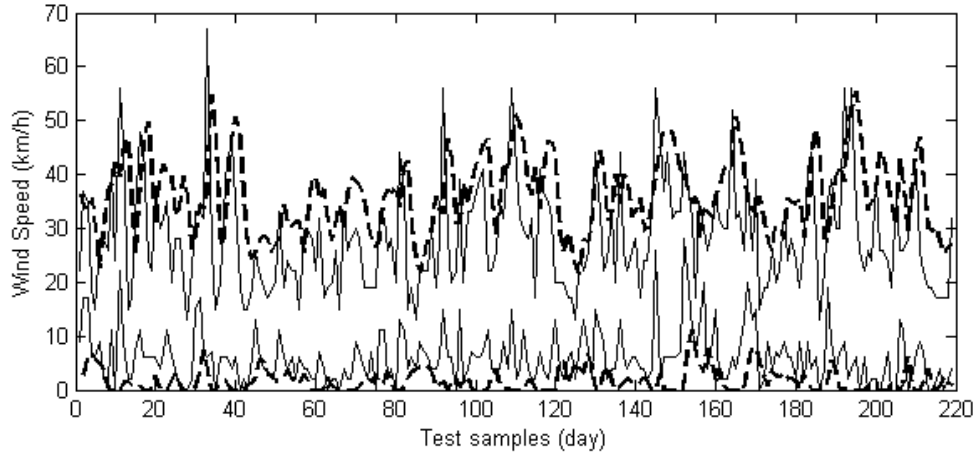


Fig. 9. Estimated PIs (dashed lines) with interval inputs for day-ahead wind speed prediction on the testing set and interval-valued wind speed data (constructed by the min-max approach) included in the testing set (solid line).

In order to compare the interval-valued and crisp approaches in a clear way, we have shown the PIs obtained by both approaches in one Figure (see Figs. 10 and 11). In Fig. 10, we have shown the estimated day-ahead PIs corresponding to the selected solutions obtained by mean and crisp approaches, respectively, on the daily crisp wind speed testing set by the trained NN. The solutions have been selected from the overall best Pareto set of optimal solutions obtained by mean and crisp approaches. These solutions result in 91.8% CP* and 0.483 NMPIW* for the mean approach, and has 91.3% CP and 0.495 NMPIW for the crisp approach, on the testing dataset. It is clear that the solution obtained by the mean approach dominates the one obtained by the crisp approach. Note that PICP* and NMPIW* values have been a posteriori calculated only for the mean approach; as the crisp approach has been trained with the crisp daily wind speed training set, it is not necessary to convert PICP and NMPIW to PICP* and NMPIW* values.

Similarly, Fig. 11 has been plotted by considering a posteriori calculated PICP* and NMPIW* values (see Fig. 7) corresponding to the two solutions selected from the overall best Pareto fronts of min-max and crisp approaches, respectively. These solutions result in 91.4% CP* and 0.452 NMPIW* for min-max approach, and 91.2 % CP* with 0.472 NMPIW* for crisp approach, on the testing dataset (raw hourly wind speed data). It is obvious that the solution obtained by min-max approach is superior to the one obtained by crisp approach. In other words, we have obtained higher quality PIs with interval-valued input approach. Note that this comparison is done on the raw hourly wind speed dataset. Since the time step for the estimated PIs is 1 day, in order to compare them to the hourly original time series data, we have shown in Fig. 11 the same lower and upper bounds within each day; thus, the PIs appear as a step function if compared to the original 1-hour data. Due to space limitations we have only plotted the estimated PIs obtained by min-max approach.

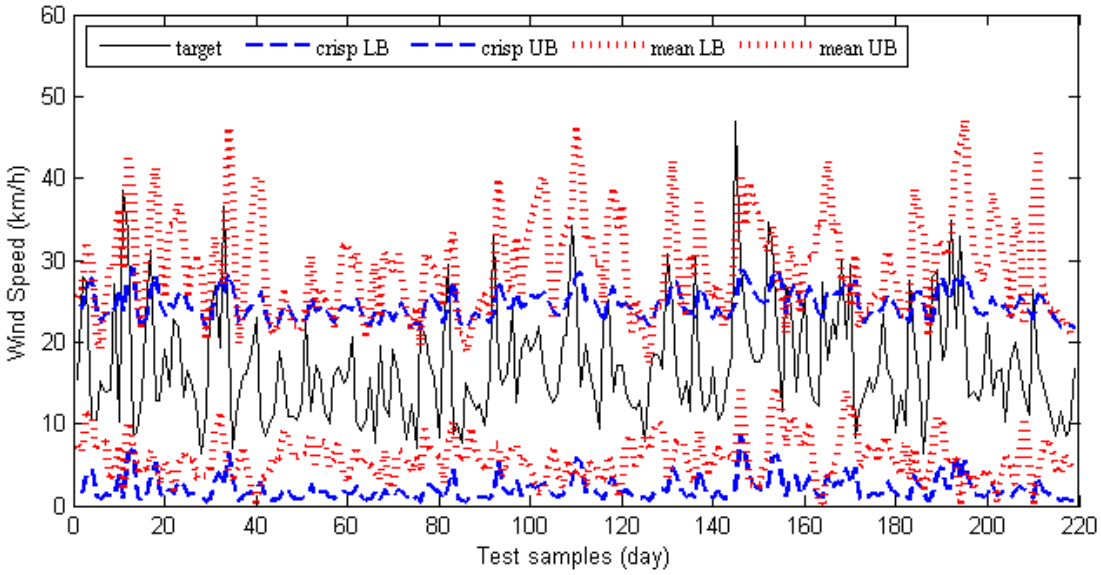


Fig. 10. Estimated PIs with interval (dotted red lines) and crisp (dashed blue lines) inputs for day-ahead wind speed prediction on the testing set and single-valued (crisp) daily wind speed data included in the testing set (solid line).

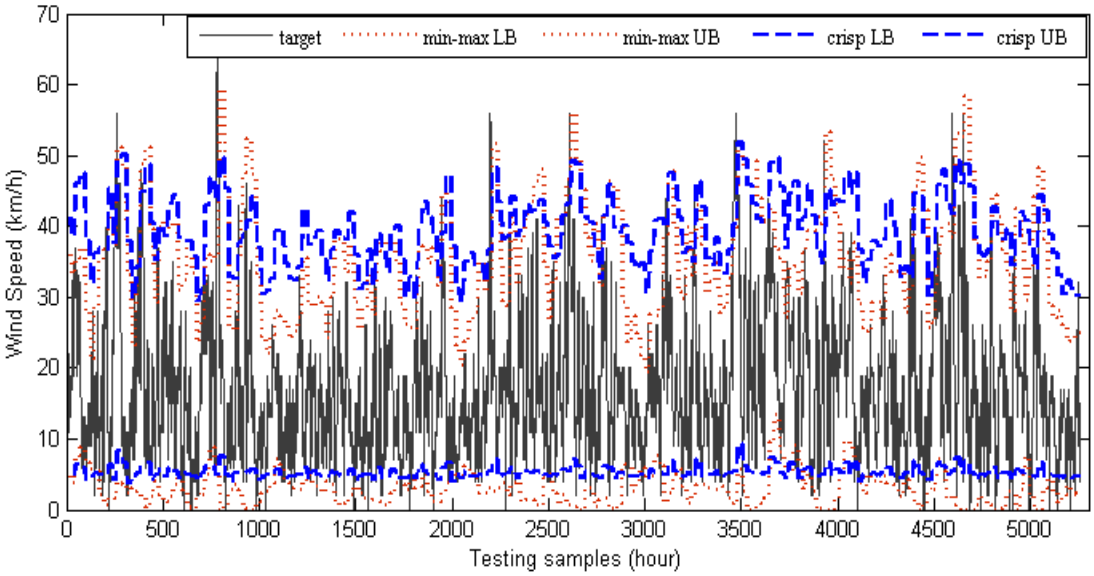


Fig. 11. Estimated PIs with interval (dotted red lines) and crisp (dashed blue lines) inputs for day-ahead wind speed prediction on the testing set and single-valued (crisp) raw hourly wind speed data (solid line).

From the results illustrated in Figs. 10 and 11, one might comment that the PIs obtained with the interval inputs approach are capable of capturing the peak points (highest and lowest) of the target of interest (hourly data). Although there are some highly extreme values dropping out of the estimated PIs, the interval approach leads to better coverage of the intermittent

characteristic of wind speed than the crisp approach. In other words, the interval approach manages to describe more efficiently the short-term variability of wind speed.

4.3. Comparison with single-objective simulated annealing (SOSA) method

In this section, we present the results from a comparison with a method called "Lower and Upper Bound Estimation (LUBE)" proposed by Khosravi et al. in [7] to estimate PIs with single-valued (crisp) inputs. In their paper, the authors have used single-objective simulated annealing algorithm (SOSA) to train the NN and adopted the cost function defined in (17), which combines PICP and NMPIW, to be minimized. The cost function proposed in [7] is called coverage width-based criterion (CWC):

$$CWC = NMPIW(1 + \gamma(PICP)\, e^{-\eta(PICP-\mu)}) \tag{17}$$

where η and μ are constants. The role of η is to magnify any small difference between μ and PICP. The value of μ gives the nominal confidence level, which is set to 90% in our experiments (see Table 1). Then, η and μ are two parameters determining how much penalty is paid by the PIs with low coverage probability. The function $\gamma(PICP)$ is equal to 1 during training, whereas in the testing of the NN is given by the following step function:

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

To perform a comparison between SOSA and the proposed MOGA method, we have run the SOSA by using the same interval-valued wind speed training data. For SOSA, the initial temperature has been determined after a trial and error procedure. It has been tried with values of 5, 200 and 500: it turns out that the SOSA with initial temperature of 200 gives best performance. Table 1 contains the parameters of the SOSA; the maximum number of generation has been set to 500.

The training process has been repeated five times. Training and testing results in each run have been reported in Table 3. Due to space limitation, we have put only min-max approach results.

According to the results reported in Table 3, it can be observed that the training and corresponding testing solutions do not show high consistency in terms of coverage probability and interval size among the five runs performed. In other words, there is a high difference among the results: SOSA gives high CP value in one run whereas it generates less accurate PIs in another one: 3 out of 5 runs give CP values smaller than the predetermined nominal confidence level, i.e. 90% in our experiments. Thus CWC values are quite high for those runs. This shows a drawback about SOSA method robustness on this specific problem.

For comparison purpose, we have selected the run giving the smallest CWC value on the training set, which is 0.649. Note that in previous works of literature the mean or median value of several runs has been used as prediction result [7], [42]. The selected run results in 93.8% CP and 0.567 NMPIW on the training set, and a coverage probability of 95.6% and interval width of 0.578 on the testing. By comparison, we have selected a solution from the overall best Pareto front obtained by MOGA min-max approach. This selected solution gives a coverage probability of 94.3 % and interval width of 0.529 on the training, and a coverage probability of 96.6 % and interval width of 0.546 on the testing. For what concerns the mean approach, we have observed similar results: 2 out of the 5 runs have given CP less than 90% both on training and testing sets. The runs resulting in coverage probability bigger than 90% have quite large interval widths (above 50%). We have selected a run which has the smallest CWC value: it has a coverage probability of 93.1% with 0.520 NMPIW on the training and 94.7% CP and interval width of 0.531 on the testing datasets. On the contrary, the MOGA method has given a solution with 93.3% CP with 0.440 interval size on the training, and 94.4% CP with 0.453 interval size on the testing set.

It is clear that the solutions obtained by MOGA dominate the best ones obtained by SOSA. It is worth pointing out that as both solutions obtained by min-max method give large interval sizes (around 50%) they cannot provide useful information in practice, because the uncertainty level is too high to support a reliable and informed decision in typical application contexts. However, with the MOGA approach one can select a solution from the Pareto front giving tight PIW with a high CP, which satisfies the predetermined nominal confidence level. In short, from the results reported in Table 3, one can conclude that the SOSA method does not give high quality PIs with respect to the interval-valued time series forecasting case study considered in this work.

Table 3. PICP and NMPIW Values Obtained by SOSA with Respect to Wind Speed Dataset (Training / Testing)

SOSA METHOD	PICP (%)	NMPIW	CWC
1	93.8 / 95.6	0.567 / 0.578	0.649 / 0.578
2	71.7 / 73.8	0.300 / 0.312	2897 / 1032
3	72.0 / 75.2	0.297 / 0.310	2425 / 519.6
4	75.5 / 76.3	0.317 / 0.328	439.0 / 311.3
5	92.1 / 95.1	0.725 / 0.752	0.978 / 0.752

5. CONCLUSIONS

The goal of the research presented in this paper is to quantitatively represent the uncertainty in neural networks predictions of time series data, originating both from variability in the input and in the prediction model itself. The application focus has been on wind speed, whose forecasting is crucial for the energy market, system adequacy and service quality in power grid with integrated wind energy systems. Accuracy of predictions of power supply and quantitative information on the related uncertainty is relevant both for the power providers and the system operators.

Specifically, we have presented two approaches that can be used to process interval-valued inputs with multi-layer perceptron neural networks. The method has been applied on a synthetic case study and on a real case study, in which the data show a high (short-term) variability (within hour and within day). The results obtained reveal that the interval-valued input approach is capable of capturing the variability in the input data with the required coverage. The results enable different strategies to be planned according to the range of possible outcomes within the interval forecast.

As for future research, the use of an ensemble of different NNs will be considered to further increase the accuracy of the predictions, and type-2 fuzzy sets can be integrated into the proposed model as an alternative way to represent the input uncertainty.

REFERENCES

[1] J. C. Refsgaard, J. P. van der Sluijs, J. Brown, and P. van der Keur, "A framework for dealing with uncertainty due to model structure error," *Advances in Water Resources*, vol. 29, no. 11, pp. 1586-1597, Nov. 2006.
[2] H. Cheng, "Uncertainty quantification and uncertainty reduction techniques for large-scale simulations," Ph.D. dissertation, Virginia Polytechnic Institute and State University, Virginia, 2009.
[3] E. Zio and T. Aven, "Uncertainties in smart grids behavior and modeling: What are the risks and vulnerabilities? How to analyze them?," *Energy Policy*, vol. 39, no. 10, pp. 6308-6320, Oct. 2011.
[4] N. Pedroni, E. Zio, and G. E. Apostolakis, "Comparison of bootstrapped artificial neural networks and quadratic response surfaces for the estimation of the functional failure probability of a thermal–hydraulic passive system", Reliability Engineering and System Safety, vol. 95, no. 4, pp. 386-395, Apr. 2010.
[5] H. Agarwal, J. E. Renaud, E. L. Preston, and D. Padmanabhan, "Uncertainty quantification using evidence theory in multidisciplinary design optimization," Reliability Engineering and System Safety, vol. 85, no. 1-3, pp. 281-294, July-Sep. 2004.
[6] J. C. Helton, "Uncertainty and sensitivity analysis in the presence of stochastic and subjective uncertainty," Journal of Statistical Computation and Simulation, vol. 57, no. 1-4, pp. 3-76, 1997.
[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 Transactions on Neural Networks, vol. 22, no. 3, pp. 337-346, March 2011.
[8] H. Quan, D. Srinivasan, and A. Khosravi. "Short-Term Load and Wind Power Forecasting Using Neural Network-Based Prediction Intervals," IEEE Transactions on Neural Networks and Learning Systems, vol. 25, no. 2, pp. 303-315, Feb. 2014.
[9] A. Khosravi, S. Nahavandi, and D. Creighton, "A prediction interval-based approach to determine optimal structures of neural network metamodels," Expert Systems with Applications, vol. 37, no. 3, pp. 2377-2387, March 2010.
[10] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya, "Comprehensive Review of Neural Network-Based Prediction Intervals and New Advances," IEEE Transactions on Neural Networks, vol. 22, no. 9, pp. 1341-1356, Sept. 2011.
[11] R. Ak, Y-F. Li, V. Vitelli, and E. Zio, "Multi-objective Genetic Algorithm Optimization of a Neural Network for Estimating Wind Speed Prediction Intervals," Applied Soft Computing (resubmitted), 2014.
[12] H. Papadopoulos, V. Vovk, and A. Gammerman, "Regression Conformal Prediction with Nearest Neighbours," Journal of Artificial Intelligence Research, vol. 40, pp. 815-840, 2011.
[13] H. Papadopoulos and H. Haralambous, "Reliable Prediction Intervals with Regression Neural Networks," Neural Networks, vol. 24, no. 8, pp. 842-851, 2011.
[14] L. V. Barboza, G. P. Dimuro, and R. H. S. Reiser, "Towards interval analysis of the load uncertainty in power electric systems," in Proc. International Conference on Probabilistic Methods Applied to Power Systems, 2004, pp. 538-544.
[15] A. M. Roque, C. Maté, J. Arroyo, and Á. Sarabia, "iMLP: Applying Multi-Layer Perceptrons to Interval-Valued Data," Neural Processing Letters, vol. 25, no. 2, pp. 157–169, Apr. 2007.
[16] A. L. S. Maia, F. A .T. de Carvalho, and T. B. Ludermir, "Forecasting models for interval-valued time series," Neurocomputing, vol. 71, no. 16, pp. 3344-3352, 2008.
[17] H. J. Zimmermann, Fuzzy Set Theory-And Its Applications. 4th ed., USA: Kluwer Academic Publishers, 2001, pp. 1-514.
[18] D. Zhai and J. M. Mendel, "Uncertainty measures for general Type-2 fuzzy sets," Information Sciences, vol. 181, no. 3, pp. 503-518, Feb. 2011.
[19] L. Qilian and J. M. Mendel, "Interval type-2 fuzzy logic systems: theory and design," IEEE Transactions on Fuzzy Systems, vol. 8, no. 5, pp. 535-550, 2000.
[20] S. Peter, et al., "Interval-Valued and Intuitionistic Fuzzy Mathematical Morphologies as Special Cases of\ mathbb {L}-Fuzzy Mathematical Morphology," Journal of Mathematical Imaging and Vision, vol. 43, no. 1, pp. 50-71, 2012.
[21] M. Nachtegael, P. Sussner, T. Mélange, and E. E. Kerre, "On the role of complete lattices in mathematical morphology: From tool to uncertainty model," Information Sciences, vol. 181, no. 10, pp. 1971-1988, 2011.
[22] E. Hofer, M. Kloos, B. Krzykacz-Hausmann, J. Peschke, and M. Woltereck, "An approximate epistemic uncertainty analysis approach in the presence of epistemic and aleatory uncertainties", Reliability Engineering and System Safety, vol. 77, no. 3, pp. 229-238, Sep. 2002.
[23] J. C. Helton and F. J. Davis, "Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems," Reliability Engineering and System Safety, vol. 81, no 1, pp. 23-69, July 2003.

[24] R. E. Moore, R. B. Kearfott, and M. J. Cloud. Introduction to Interval Analysis, Society for Industrial and Applied Mathematics. 1st ed., USA, 2009, pp. 1-235.
[25] J. Arroyo and C. Maté, "Introducing interval time series: Accuracy measures," in Proc. COMPSTAT, 2006, pp. 1139-1146.
[26] A. U. Haque, P. Mandal, M. E. Kaye, J. Meng, L. Chang, and T. Senjyu, "A new strategy for predicting short-term wind speed using soft computing models," Renewable and Sustainable Energy Reviews, vol. 16, no. 7, pp. 4563-4573, 2012.
[27] J. Jaesung and R. P. Broadwater, "Current status and future advances for wind speed and power forecasting," Renewable and Sustainable Energy Reviews, vol. 31, pp. 762-777, March 2014.
[28] H. Bouzgou and N. Benoudjit, "Multiple architecture system for wind speed prediction," Applied Energy, vol. 88, no. 7, pp. 2463-2471, 2011.
[29] A. More and M. C. Deo, "Forecasting wind with neural networks," Marine Structures, vol. 16, no. 1, pp. 35-49, Jan. 2003.
[30] R. Ak, Y. F. Li, V. Vitelli, E. Zio, E. López Droguett, and C. Magno Couto Jacinto, "NSGA-II-trained neural network approach to the estimation of prediction intervals of scale deposition rate in oil & gas equipment," Expert Systems with Applications, vol. 40, no. 4, pp. 1205-1212, March 2013.
[31] E. Zio, "A study of the bootstrap method for estimating the accuracy of artificial neural networks in predicting nuclear transient processes," IEEE Transactions on Nuclear Science, vol. 53, no. 3, pp. 1460-1478, June 2006.
[32] D. L. Shrestha and D. P. Solomatine, "Machine learning approaches for estimation of prediction interval for the model output," Neural Networks, vol. 19, no. 2, pp. 225-235, 2006.
[33] Y. Sawaragi, H. Nakayama, and T. Tanino. Theory of Multiobjective Optimization. Orlando, FL: Academic Press Inc., 1985, pp. 1-296.
[34] A. Konak, D. W. Coit, and A. E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial," Reliability Engineering and System Safety, vol. 91, no. 9, pp. 992-1007, Sep. 2006.
[35] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," IEEE Transactions on Evolutionary Computation, vol. 3, no. 4, pp. 257-271, Nov. 1999.
[36] N. Srinivas and K. Deb, "Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms," Evolutionary Computation, vol. 2, no. 3, pp. 221–248, 1994.
[37] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II ," IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182-197, Apr. 2002.
[38] Canadian Weather Office, 2012. Available: http://www.weatheroffice.gc.ca/canada_e.html. [Accessed: 08-avr-2013].
[39] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, Time Series Analysis: Forecasting and Control. 4th ed., Wiley, 2008.
[40] Media Kit 2012, Canadian Wind Energy Association (CanWEA). Available: http://www.canwea.ca/pdf/windsight/CanWEA_MediaKit.pdf
[41] SaskPower Annual Report 2011. Available: http://www.saskpower.com/news_publications/assets/annual_reports/2011_skpower_annual_report.pdf
[42] E. Zio, P. Baraldi, and N. Pedroni, "Optimal power system generation scheduling by multi-objective genetic algorithms with preferences," Reliability Engineering and System Safety, vol. 94, pp. 432-444, 2009.

PAPER V

Adequacy Assessment of a Wind-integrated Power System using Neural Network - based Interval Predictions of Wind Power Generation and Load

R. Ak, Y. F. Li, V. Vitelli and E. Zio. (2014), submitted to *International Journal of Electrical Power & Energy Systems* (under review).

Adequacy Assessment of a Wind-integrated Power System using Neural Network - based Interval Predictions of Wind Power Generation and Load

Ronay Ak[a], Yanfu Li [a], Valeria Vitelli[b], Enrico Zio[a,c]

[a]Chair on Systems Science and the Energetic Challenge, European Foundation for New Energy-Electricité de France

École Centrale Paris, Grande Voie des Vignes, Châtenay-Malabry, 92290 France, and SUPELEC, Plateau du Moulon - 3 Rue Joliot-Curie, Gif-Sur-Yvette, 91192 France

[b]Department of Biostatistics, University of Oslo, Domus Medica, Sognsvannsveien 9, 0372 Oslo, Norway.

[c]Department of Energy, Politecnico di Milano, Via Ponzio 34/3 Milan, 20133 Italy

ABSTRACT

In this paper, we present a modeling and simulation framework for conducting the adequacy assessment of a wind-integrated power system accounting for the associated uncertainties. A multi-perceptron artificial neural network (MLP NN) is trained by a non-dominated sorting genetic algorithm–II (NSGA-II) to forecast point-values and prediction intervals (PIs) of the wind power and load. The output of the assessment is given in terms of point-valued and interval-valued Expected Energy Not Supplied (EENS). We consider different scenarios of wind power and load levels, to explore the influence of the uncertainty in wind and load predictions on the estimation of system adequacy.

Keywords: Adequacy assessment, multi-objective genetic algorithm, neural networks, prediction intervals.

1. INTRODUCTION

The adequacy assessment of a power system is challenging due to the many uncertainties associated, for example, to fluctuations in energy demand, to the prediction of future weather conditions (e.g. wind speed, solar irradiation, etc.), to possible equipment (e.g. generators, lines, etc.) unavailability, to failures in electric power transactions, to errors (operator errors, dispatcher and relay malfunctions), and to other relevant issues [1]-[3].

In this paper, we present a modeling and simulation framework for conducting the adequacy assessment of a wind-integrated power system accounting for uncertainties in the data and prediction models. A widely used adequacy index, the Expected Energy Not Supplied (EENS), is evaluated as output of the assessment. EENS measures the failure of the system to meet the demand by the cumulative amount of energy that is not provided to the customers, over the time horizon of interest for the analysis [4], [5].

Several works in the literature calculate EENS for the adequacy assessment of a power network [6]-[8]. The originality of the present work lies in proposing not only point-valued results, like the works previously mentioned, but also interval-valued results to inform the decision makers (DMs) on the uncertainty in the predictions. Uncertainties are here considered due to load fluctuations, wind variability, and component failures.

A case study is considered in which hourly wind speed data from the region of Regina, Saskatchewan, Canada are taken, from a 9-year period (1 Jan. 2003 to 31 Dec. 2011) [9]. Hourly mean wind speed data are used to determine the time-dependent wind power output of a wind turbine generator (WTG) using its power curve [7]. For load demand, the hourly load fluctuations are modeled using the chronological annual load curve of the IEEE Reliability Test System (RTS) [10] with the scaled annual peak load value.

The generating units in the power system are represented by two-state models, describing operation and failure, and they are sampled by sequential Monte Carlo simulation. The inputs to estimate the EENS are the Prediction Interval (PIs) for 1-hour ahead wind power and load. These values are provided by the use of multi-perceptron artificial neural networks (MLP NNs) trained by the non-dominated sorting genetic algorithm–II (NSGA-II) [11]: the lower

and upper bounds of the NN-based PIs are optimal both in terms of coverage probability (PICP) and width (PIW). The NSGA-II training procedure generates Pareto-optimal solution sets, which include non-dominated solutions for the two objectives (PICP and PIW). One solution has, then, to be selected among the ones in the Pareto optimal set according to the preferences on the objectives.

The method proposed for estimation of PIs for 1-hour ahead wind power and load is the extension of a single-objective optimization method, called Lower and Upper Bound Estimation (LUBE) approach, proposed in [12]. The strength of the proposed method has been already shown in [13] via comparison with the original LUBE method based on a single-objective genetic algorithm, and with a baseline method, i.e. ARIMA. In [13], we have carried out a case study on four different single-valued wind speed datasets involving different wind speed profiles with seasonality. The results have confirmed the superiority of the NN-based PIs estimation approach trained by NSGA-II on the other methods considered in the comparison.

The paper is organized as follows. Section 2 briefly introduces the definition of PIs and the use of NSGA-II for training a NN to estimate PIs. In Section 3, the methodology for interval-based estimation of EENS is given. Experimental results on the case study are given in Section 4. Finally, Section 5 presents the conclusions of the study.

2. METHODOLOGY TO ESTIMATE LOAD AND WIND POWER PIS

In the following sub-sections, the main phases of the methodology are described. The application of the framework is shown on a case study taken from literature [7]. In Fig. 1, a flowchart of the methodology for the adequacy assessment of wind-integrated power systems is depicted.

2.1. Wind Power Generation

Hourly wind speed data have been collected for the region of Regina, Saskatchewan, Canada for a 9-year period (1 Jan. 2003 to 31 Dec. 2011) [9]. Since wind power is a function of wind speed, forecasts of power are generally derived from wind speed. In order to conduct the

adequacy assessment over one-year time horizon, for each hour in the year (8736 h) the hourly means are calculated over 9 years of wind speed values. The so obtained one-year time series of wind speed V(t), t = 1, ..., 8736, are then transformed in wind power $P(t)$ values using a quadratic characteristic curve (power curve) of literature [14], [15].

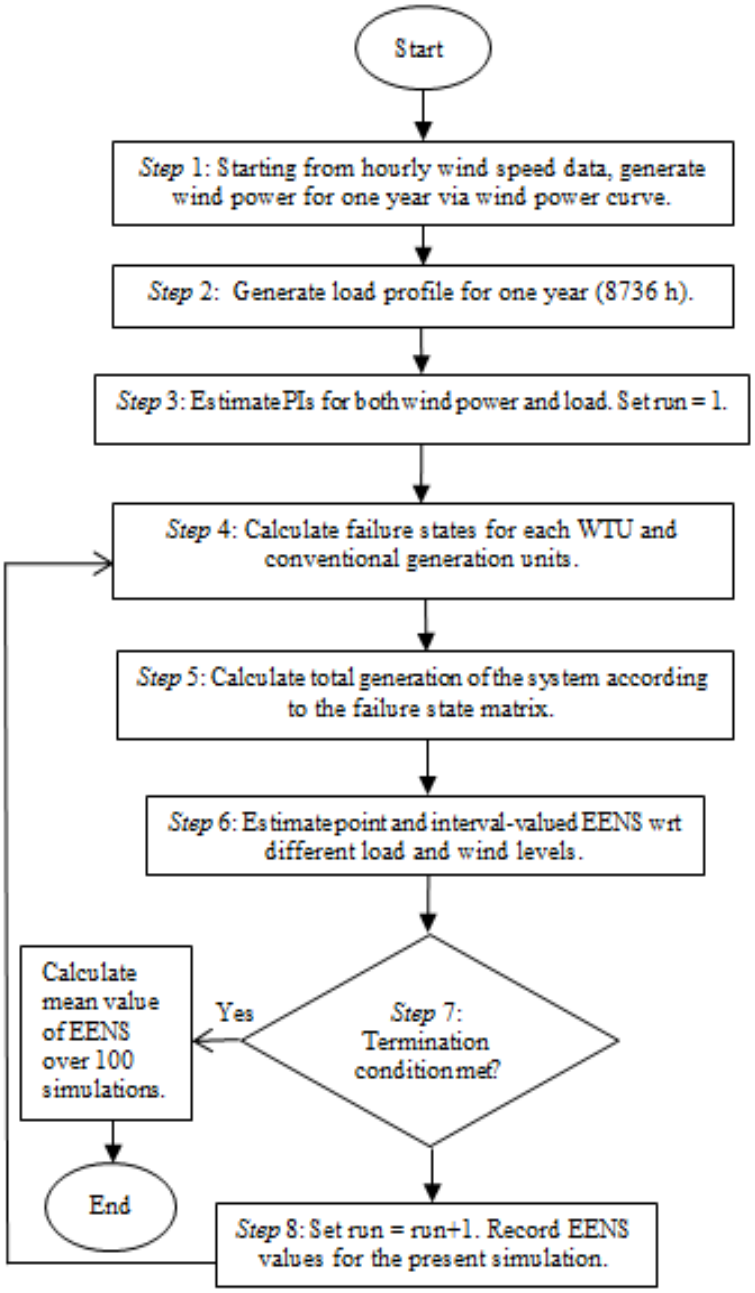


Fig. 1. Flowchart of the proposed methodology.

2.2. Load Modeling

The load duration curve (LDC) on an annual basis (8736 h) is created by manipulating the hourly load values from the IEEE-RTS [10]. One year (8736 h) load data, i.e. a load value $L(t)$ for each hour t = 1, ..., 8736, have been generated with the following formula [16]:

$$L(t) = \bar{L}(t) + \bar{L}(t)\left(\frac{\sigma}{100}\right)X^{norm} \quad (1)$$

where $\bar{L}(t)$ is the expected value of load for hour t, calculated using the following equation:

$$\bar{L}(t) = P_w(t) \times P_d(t) \times P_h(t) \times L_{max} \quad (2)$$

where L_{max} is the peak load in a year, $P_w(t)$ is the weekly peak load as a percentage of the annual peak, $P_d(t)$ is the daily peak load as a percentage of the weekly peak and $P_h(t)$ is the hourly peak load as a percentage of the daily peak. The system peak load L_{max} is set to 185 MW [7]. σ is the load forecasting uncertainty error (standard deviation) expressed as a percentage of the hourly peak load, and X^{norm} is defined as [16]:

$$X^{norm} = \sqrt{-2\,ln(R_1)}\,cos(2\pi R_2) \quad (3)$$

where R_1 and R_2 are two random numbers drawn from the standard uniform distribution on the open interval (0,1), and X^{norm} is a normally distributed random number [16], [17]. The load forecasting error σ is set to 5%.

2.3. Estimation of NN-based PIs

Based on the hourly wind power and load values over a 1-year horizon, we define a data-driven strategy to perform short term (1-hour ahead) prediction of both load and wind power with uncertainty quantification. Not only a point estimate of the target, but also PIs are computed.

PIs estimation is performed under the assumption of statistical independence of the input data, both wind speed and load. On the other hand, we do not rely on assumptions that the data are drawn from a given probability distribution and, hence, here we perform an empirical and

non-parametric approach to the estimation of PIs.

In order to estimate PIs for 1-hour ahead wind power and load prediction, we use MLP NNs [18] which are a class of nonlinear statistical models inspired by brain architecture. Multi-layer perceptron (MLP) is one of the most common types of feedforward neural networks proven to be a class of universal approximators [18]. MLP NNs are capable of learning complex nonlinear relationships among variables from observed data, by a process of parameter tuning called "training" [18]–[20]. MLP NNs have been successfully used in many practical applications for prediction, especially for nonlinear problems [21]. In the area of load and wind speed/power forecasting, literature on NN forecasting models [13], [22], [23] shows that the high complexity and nonlinearity of power systems are such that the application of classical forecasting techniques may not be reliable. It is known that short-term load and wind speed/power prediction plays a critical role for day-ahead electricity markets where wind power penetration is relatively high; medium-term forecasting (days to weeks) is relevant for the unit commitment and maintenance operations; and, long-term forecasting (months to years) is useful for planning and policy making [24]: thus, the necessity of reliable forecasting tools for wind speed prediction. On the other hand, the NN-based approaches to PIs estimation have become popular, and this area of research has been established and well accepted due to the superiority of these approaches on classical regression models for complex prediction problems [12], [13], [23], [25]. For this reason, we have chosen Multi-layer Perceptron NN due to its capability of learning complex nonlinear relationships between input and output variables from observed data: many successful experiments show its superiority in terms of forecasting performance compared to classical methods. It is worth mentioning that there exist also probabilistic methods based on quantile regression that can perform forecasting taking into account the associated uncertainty [26], [27] among which some parametric probabilistic forecasting methods [28].

In Fig. 2 the structure of a typical three layer (input, hidden and output) NN is illustrated. The neurons are connected by weights. Each layer receives input signals generated by the previous layer, produces output signals through an activation function (e.g., a sigmoid transfer or activation function), and distributes them to the subsequent layer [18], [20]. The first output neuron provides the upper bound of the PIs, and the second the lower bound.

Set h to be the number of hidden neurons, n_f the number of input neurons and n_p the total number of training samples. Then, the output signal H_j of node j of the hidden layer is given by [18], [20]:

$$H_j = f_h\left(\sum_{k=0}^{n_f} w_{kj} x^k\right) \qquad j = 1, \dots, h \tag{4}$$

where $x^0 = 1$, and for $k = 1,2, \dots, n_f$, x^k is the *k-th* input vector, $x^k \equiv (x_1^k, x_2^k, \dots, x_{n_p}^k)$, w_{kj} is the synaptic weight, and f_h is the activation function used in the hidden layer.

After each hidden neuron output has been computed, the signal is sent to each of the neurons o_l in the output layer. Each output neuron o_l computes its output signal O_l to form the response of the network [18], [20]:

$$O_l = f_o\left(\sum_{j=0}^{h} w_{jl} H_j\right) \quad l = 1,2, \dots, n_o\ ,\ H_0 = 1 \tag{5}$$

where n_o is the number of output neurons and f_o indicates the activation function used in the output layer.

The values of the weight vector w of the network are optimized during training. Training procedure aims at minimizing the quadratic error function on a training set of input/output values $D = \{(x_i, y_i),\ i = 1,2, \dots, n_p\}$.

$$E(w) = \sum_{i=1}^{n_p} (O(x_i) - y_i)^2 \tag{6}$$

where x and y are the input and target vectors respectively, $O(x_i)$ is the estimated output value of the network for the i-*th* input sample x_i. It is worth mentioning that in the case study of the present work, the inputs $x^k,\ k = 1,2, \dots, n_f$ to the NN are the historical values of wind power and load data, respectively. More precisely, we have used x_{t-1}, x_{t-2}, x_{t-3}, and x_{t-4} wind power values of previous time steps as input variables to predict PIs for wind power x_t, i.e. for $y(x_t)$, in output. The same procedure has been followed for the estimation of the load PIs at time *t*.

PI is comprised of upper and lower bounds in which a future unknown value of the target is expected to lie with a predetermined confidence level $(1-\propto)$ [12]. We evaluate the "goodness" of the PIs by estimating the empirical PIs coverage probability (PICP), which one wants to maximize, and the interval width (PIW), which one wants to minimize.

The mathematical definitions of the PICP and PIW measures are [12]:

$$PICP = \frac{1}{n_p}\sum_{i=1}^{n_p} c_i \tag{7}$$

where $c_i = 1$, if $y_i \in [L(x_i), U(x_i)]$ and otherwise $c_i = 0$,

$$NMPIW = \frac{1}{n_p}\sum_{i=1}^{n_p} \frac{(U(x_i)-L(x_i))}{y_{max}-y_{min}} \tag{8}$$

where NMPIW is the Normalized Mean PIW, and y_{min} and y_{max} represent the true minimum and maximum values of the targets y (i.e., the bounds of the range in which the true values fall) in the training set, respectively. Normalization of the PI width by the range of targets makes it possible to objectively compare the PIs, regardless of the techniques used for their estimation or the magnitudes of the true targets.

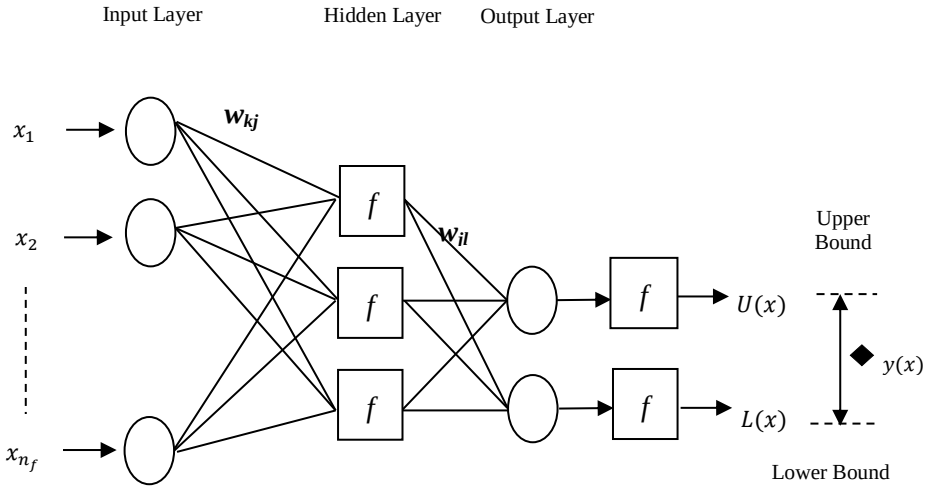


Fig. 2. Architecture of a NN model for estimating the lower and upper bounds of PIs.

The PIs estimation problem is addressed by taking into account these two conflicting objectives within a multi-objective framework. For this, we solve the problem by a MOGA (multi-objective genetic algorithm for its ability to find nearly global optima, the ease of use and the robustness [11], [29]. More specifically, we use NSGA-II [11] to optimize the parameters (i.e. the weights $\widehat{w}$) of the NN with respect to both PICP and PIW objectives. More precisely, the NN is trained by NSGA-II to produce the lower and upper bounds of the PIs for short-term forecasting (1-hour ahead) of wind power and load. For the details of the practical implementation of NSGA-II for NN-based PIs estimation see [30]. Among the several variations of MOGA in the literature, we select NSGA-II as the optimization tool, because comparative studies [11] have shown that it is one of the most efficient MOGAs.

The training by NSGA-II is justified by the fact that the back-propagation, widely used for performing supervised learning tasks like the training of NNs, would require calculating the gradient of the error function to find the optimal weights that minimize the estimation error, whereas the NSGA-II does not require these derivative calculations. Moreover, existing techniques for estimating PIs for NN algorithm outputs such as Delta and Bayesian methods require the calculation of Jacobian and Hessian matrices, respectively, and although they are capable of generating high quality PIs, they demand high computational time in the development stage [25]. Compared to Delta and Bayesian methods, NSGA-II is less demanding at the training phase. Also the proposed approach integrates the estimation of the prediction intervals in its learning procedure while several methods construct PIs in two steps (first doing point prediction and then constructing PIs).

To obtain a Pareto-front, another alternative way could be to use the ε-constraint method in the literature [31], [32]. To perform this method, one has to reformulate the problem as a single-objective one by choosing one objective for optimization and considering the other as a constraint. The constraint value is changed to generate the Pareto-optimal set. This approach requires multiple runs to form the Pareto and this can be time-consuming [31], [32]. In addition, the search is limited to few points in some predefined regions near the fixed constraint values. This may lead to missing some optimal solutions. On the contrary, approach using NSGA-II can find, multiple Pareto-optimal solutions in one single run and the nondominated solutions in the obtained Pareto-optimal set are well distributed and diverse [11], [29], [31], [32].

3. METHODOLOGY TO ESTIMATE EENS

In order to conduct the adequacy assessment of the wind-integrated power system, we use the well-known adequacy index, EENS, which quantifies the capability of the system to meet the demand in the time horizon considered for the analysis. EENS measures the expected value of the energy not supplied due to the lack of available energy through the given time horizon (e.g. one year). It depends on the predicted values for both the system energy production and the power demand, and it is formulated as follow [33], [34]:

$$ENS_k = \sum_{t=1}^{N} Pr(L_t > G_t) \times (L_t - G_t) \quad (9)$$

where ENS_k is the realization of the energy not supplied for the entire horizon the *k*-th simulation run; t is the equally sized time step (e.g. hour or day); N is the total number of time steps in the considered time horizon, in our case N = 8736 for a one year time horizon, G_t is the total power generation available at time step t; L_t is the load demand at time step t; $Pr(L_t > G_t)$, which indicates the probability that the load demand exceeds the available power generation at time step t, is a generalized form of $1(L_t > G_t)$ to handle the interval values of L_t and G_t: when L_t and G_t are crisp values as in the classical adequacy assessment, $Pr(L_t > G_t)$ is reduced to $1(L_t > G_t)$ which equals to 1 if the condition is satisfied, otherwise equals to 0.

Thus, EENS value of the system, i.e. the average amount of the unsupplied energy per year, is estimated as follow:

$$EENS = \frac{\sum_{k=1}^{K} ENS_k}{K} \quad (10)$$

where *K* is the total number of simulations that has been set to 100 in our experiments.

In the classical definition of ENS given in (9), both the predicted value of the generation G_t and of the load L_t at each time step t are assumed to be point estimates, resulting in a point estimate of EENS (see (10)). Our method is, instead, capable of providing PIs for both the power generation and the load at each time step, to take into account the possible uncertainties in the prediction arising from both the underlying physical processes (wind inherent

uncertainty, variability in power demand, etc.) and in the system stochastic behavior (equipment failures, approximations of the system complexities, etc.). A proper adequacy assessment model should take these sources of uncertainty into account, since uncertainty quantification is crucial for a real understanding of the system behavior, and for obtaining reliable results useful for robust decision making. Hence, we aim at a generalization of the ENS formulation given in (9), in order to include interval estimates of both G_t and L_t.

Two different strategies are considered for interval-based EENS estimation: a point estimation and an interval estimation. They are both interval-based, in the sense that the inputs to the evaluation are the short-term PIs for load and for power generation, as obtained by the NN-based estimation procedure described in the previous section.

3.1. Interval-based Interval Estimation of EENS

One possible strategy for taking into account load and power generation PIs in EENS estimation consists in directly using (9) with interval-valued G_t and L_t, thus obtaining as a result an interval evaluation of EENS by directly applying the principles of interval arithmetic [35]. In other words, all arithmetic calculations throughout the evaluation process of the interval-valued G_t and L_t are performed according to interval arithmetic (interval product, sum, intersection, etc.). Moreover, an assumption is made in the computation of $Pr(L_t > G_t)$ in the case of interval- valued L_t and G_t: due to lack of further information, a uniform probability is assumed for the actual (unknown) values of both load and power generation being anywhere inside the intervals of $\mathrm{L_t}$ and $\mathrm{G_t}$, respectively.

Since the expected value of a random variable lies in the range where the random variable lies, and since EENS is (in general) the expected value of a random variable over a given time horizon, we are here giving a new and probabilistically coherent definition of EENS when it is evaluated based on interval load and power values. The relationship between our estimates and the range for the expected value of the EENS random variable is to be fully studied, and will be the objective of our future speculations.

Precisely, total load and total generation at time t are defined as $L_t = [L_t^-,\ L_t^+]$ and $G_t = [G_t^-, G_t^+]$, respectively, where L_t^- and G_t^- indicate the lower bounds, and L_t^+ and G_t^+ indicate

the upper bounds of the intervals of the two quantities.

$(L_t - G_t)$ in (9) is calculated as follows, in accordance with the interval arithmetic rules [35]:

$$(L_t - G_t) = [L_t^- - G_t^+, L_t^+ - G_t^-] \quad (11)$$

More precisely, (11) shows an interval arithmetic operation for the difference of two intervals, L_t and G_t [35]. The difference $(L_t - G_t)$ is one of the terms on the right side of (9). Since we used interval-valued, i.e. estimated, PIs, for load and power, (11) has been given to explain how we calculated the $(L_t - G_t)$ in (9) by using interval-valued arithmetic.

Numbers in Fig. 3 illustrate the possible relationships between load and total generation with respect to two different cases describing possible load and generation at time t. Note that, in order to calculate EENS, we are interested in the subintervals where load may be larger than total power generation. For example, in Fig. 3(a), number 1 indicates that load takes a value inside the interval $[L_t^-, L_a]$. This means that even load takes the maximum value, L_a, the total generation will be greater than load, so EENS will be 0. On the other side, if load and generation take a value inside intervals $[L_a, L_t^+]$ and $[G_t^-, G_a]$, respectively, which have been indicated by number 2 in Fig. 3(a), this may lead to an EENS value bigger than 0. The following statements have been given to explain these relationships and corresponding EENS calculations.

For the case in Fig. 3(a), $Pr(L_t > G_t)$ is calculated as follows by considering subintervals 2 and 3:

$$Pr(L_t > G_t) = p_{2l} \times p_{2g} \times 1/2 \quad (12)$$

where p_{1l}, p_{2l}, and p_{2g}, p_{3g} are fractions of the intervals L_t and G_t, respectively. Specifically, if $diam(\,)$ indicates the length of an interval, p_{1l} is the fraction of $diam(L_t^-,\ L_a)$ over the length of the entire interval, $diam(L_t^-,\ L_t^+)$:

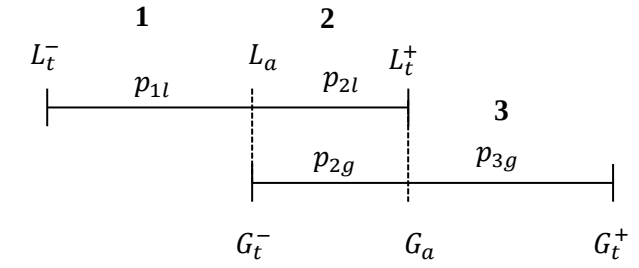
$$p_{1l} = \frac{diam(L_t^-, L_a)}{diam(L_t^-, L_t^+)} \quad (13)$$

This fraction corresponds to the probability of the actual (unknown) load (or generation) being within that part (subinterval) of the interval, because of the assumption of uniform distribution of the actual load (or generation) within the estimated intervals. In fact, we can formally derive (12) and (13) by directly using the probability density function of a uniform random variable [36]: if U is a uniform random variable on the interval (m, n), then its probability density function $\phi(u)$ is given by

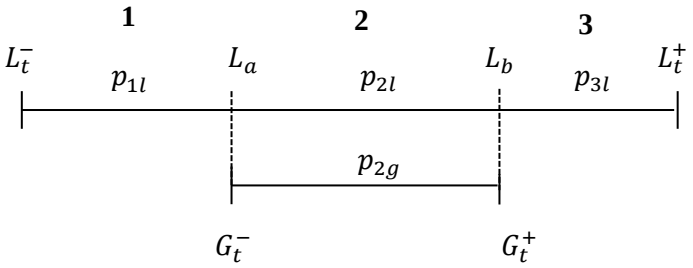
$$\phi(u) = \begin{cases} \frac{1}{m-n}, & if\ m < u < n \\ 0, & otherwise \end{cases} \tag{14}$$

If $[a, b]$ is a subinterval of (m, n), then the probability of U falling within the interval $[a, b]$ depends only on the length of $[a, b]$ with respect to (m, n). Specifically [36]:

$$Pr(a \le U \le b) = \int_a^b \frac{du}{m-n} = \frac{a-b}{m-n} \tag{15}$$



(a)



(b)

Fig. 3. Two different cases describing possible load and generation at time *t*.

Equation (12) has been written in order to explicit the calculation of the term $Pr(L_t > G_t)$ in (9). This calculation, for exemplification purposes, has been carried out by considering a certain relationship between load and generation, the one shown in Fig. 3(a), where the lower bound of load is smaller than the lower bound of total power generation, and the upper bound of load is smaller than the upper bound of total power generation. So, for the case in Fig. 3(a), if load takes a value inside the interval $[L_a,\ L_t^+]$, and total generation takes a value inside the interval $[G_t^-, G_a]$, the computed value of ENS can be bigger than 0 with positive probability. Equation (12) calculates this probability according to a classical formulation of ENS, given in (9).

Ultimately, for the case in Fig. 3(a), with respect to the interval-valued load and total generation at time step t for the k-th simulation, (9) has been modified as follows:

$$ENS_t = p_{2l} \times p_{2g} \times 1/2 \times [L_t^- - G_t^+, L_t^+ - G_t^-] \quad (16)$$

$$ENS_k = \sum_{t=1}^{N} ENS_t \quad (17)$$

Note that in this example, as the $L_t^- - G_t^+$ value is smaller than 0, we set it to 0 to consider only the subintervals, e.g. $[L_a,\ L_t^+]$, which may result in unserved energy. In other words, when $L_t^- - G_t^+$ is negative, this means that we have enough energy to meet the demand, so EENS becomes 0. For exemplification, let the intervals for load and total generation be [5, 15] and [10, 20], respectively. According to (11), from the $L_t - G_t$ subtraction we obtain [-15, 5] as a result. However, in our calculation we only consider the interval [0, 5] as possibly resulting in unserved energy. This means that we can have maximum 5 MWh of unserved energy at time t, and this corresponds to the interval $[L_a,\ L_t^+]$.
If the load and the generation intervals are as in Fig. 3(b), $Pr(L_t > G_t)$ is calculated similarly as follows:

$$Pr(L_t > G_t) = p_{3l} + p_{2l} \times p_{2g} \times 1/2, \quad (18)$$

where p_{3l}, p_{2l}, and p_{2g} are the intervals fractions as defined above. This calculation corresponds to the probabilistic assumption that load and generation can take any value in $[L_t^-,\ L_t^+]$ and $[G_t^-, G_t^+]$, respectively, with uniform probability, i.e. each point in $[L_t^-, L_t^+]$ and

$[G_t^-, G_t^+]$ is equally likely to be a possible value of L and G, respectively. In another words, to perform decision making and to use in practice, one should select a final crisp value of load (power) within the interval $L_t = [L_t^-,\ L_t^+]$. By selecting a uniform distribution, we gave equal chances to all the values inside the interval.

Finally, one should calculate the mean value of the estimated ENS values for each simulation run and, thus, obtain the expected (average) amount of the unsupplied energy (load) (EENS) of the system (see (10)) over the study period which is one year in our case.

It is worth mentioning that we performed some simulations, with Gamma and Gaussian distributions inside the intervals, in order to assess whether the results were much influenced by these choices with respect to the use of a Uniform distribution instead: the answer is no, there is no significance influence.

3.2. Interval-based Point Estimation of EENS

As explained in Section 3.1, load and power generation, provided by NNs as PIs, can be directly used for EENS estimation. One possible strategy, leading to an interval estimation of EENS, has already been described in the previous section. An alternative way to generalize EENS to the interval case leads to obtaining a point estimate of the adequacy index. This strategy is based on the probability density function of the continuous random variable $\xi_t = l_t - g_t$, where $l_t \in L_t$ and $g_t \in G_t$ are, respectively, two admissible values of the load demand and power generation at time t, thus $\xi_t \in [max\{0, L_t^- - G_t^+\}, L_t^+ - G_t^-]$. Any value assumed by ξ_t represents a possible amount of energy that cannot be supplied by the power system at time t to meet the demand: hence, a point estimate of EENS (PEENS) at time t can be obtained by computing the expected value of ξ_t over the intervals of admissible values for load and power, L_t and G_t, respectively. This is indeed a probabilistic approach, since the assumption of uniform distribution of the energy values within L_t and G_t has again to be made. Moreover, uncertainty quantification is taken into account, because the load and power PIs are used in the EENS estimation process. The obtained final estimate of EENS is a single value, which may give a more interpretable result.

According to this strategy, the PEENS of the system for the k-th simulation can be calculated

as follows [34]:

$$PEENS_k = \int_{max\{0,\ L_t^- - G_t^+\}}^{L_t^+ - G_t^-} \xi_t \ Pr(\xi_t > 0) d\xi_t \quad (19)$$

Note that $Pr(\xi_t > 0)$ is a value obtained by computing an integral over the relevant domains of the two variables L_t and G_t under the assumption of a uniform distribution. We wrote it inside the integral for giving the general definition of EENS. Indeed, this is not easy to compute in general, so instead of computing $Pr(\xi_t > 0)$ directly, we found it easier for carrying out the computations to modify the domain and integrate it together with ξ_t.

From this general formulation we can derive the following expressions, for the examples shown in Fig. 3 (Fig. 3(a) and 3(b), respectively):

$$PEENS_k = \int_{G_t^-}^{L_t^+} \int_{G_t^-}^{L_t} (L_t - G_t) \frac{1}{W_L} \frac{1}{W_G} dG_t dL_t \quad (20)$$

$$PEENS_k = \int_{G_t^-}^{G_t^+} \int_{G_t}^{L_t^+} (L_t - G_t) \frac{1}{W_L} \frac{1}{W_G} dL_t \, dG_t \quad (21)$$

where $W_L = L_t^+ - L_t^-$ and $W_G = G_t^+ - G_t^-$, and we directly computed the integrals assuming a uniform probability density function for both random variables L_t and G_t. In general, for any of the possible cases of interval-valued load and generation at each time step, we can derive an analytic expression for the interval-based point estimate of EENS. We do not report the explicit EENS calculations in each case, for the sake of brevity.

As mentioned in the previous section, to estimate the expected point estimation of EENS (EPEENS) from all the simulation runs over the study period, the following formulation holds:

$$EPEENS = \frac{\sum_{k=1}^{K} PEENS_k}{K} \quad (22)$$

4. EXPERIMENTAL RESULTS

The proposed approach has been tested on the RBTS (Roy Billinton test system) system [37]. The RBTS system consists of 11 conventional generation units with a total capacity of 240

MW. A wind farm with 20 identical WTG units has been added to the RBTS system. Each WTG is assumed to have a rated capacity of 2 MW and cut-in, rated and cut-out speeds of 14.4 km/h, 36 km/h and 80 km/h, respectively. In Fig. 4, the system topology of the RBTS system is shown.

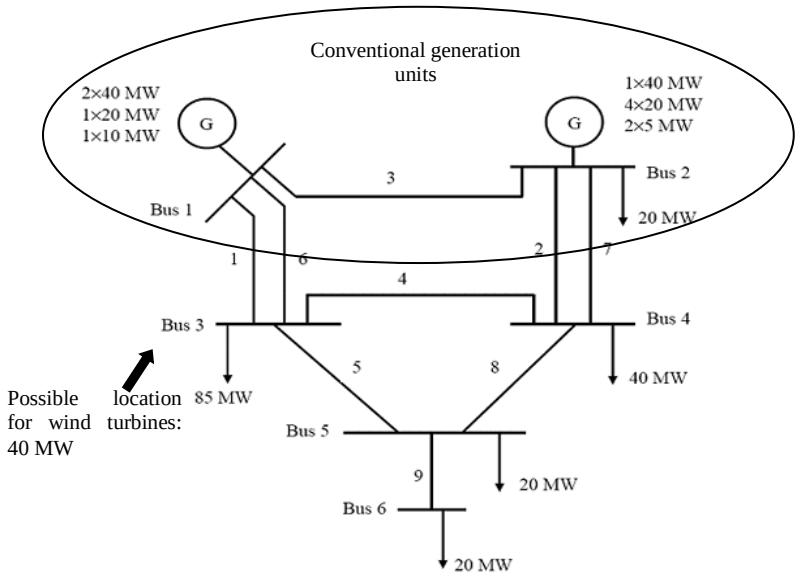


Fig. 4. Single line diagram of the RBTS [7].

4.1. Failure Modeling

With respect to the failure and repair behavior, the system components are considered to be independent and with only two states: up and down.

It is assumed that all components are initially in the up state. For a generic component *i* (such as generator, transformer, line, etc.), both time-to-failure (TTF_i) and time-to-repair (TTR_i) follow an exponential distribution. By randomly sampling two numbers R_1 and R_2 from a uniform distribution on (0, 1), the sampled values of the state residence time are calculated as follows:

$$TTF = -MTTF \times ln(R_1) \quad (23)$$

$$TTR = -MTTR \times ln(R_2) \quad (24)$$

where MTTF and MTTR are the means of the respective exponential distributions.

In order to generate a failure state matrix for the system in question, first, we have sampled the duration of the current state for each component i for a 1-year time horizon by using (23) or (24), i.e. if the current state is up we have used (23), otherwise (24). Then, we have combined them in a matrix to obtain a sequence of system states. We have assumed that the availability of each component is independent of each other and also independent of load value.

Fig. 5 shows a simulated component operating/restoration history, i.e. the transition history from the upstate to the down state [16], [38].

Table 1 reports conventional generating unit ratings and reliability data [37]. For an individual wind turbine, the failure and repair rates are set to 0.0005/hr and 0.013 /hr, respectively [2].

4.2. Data Description and NN Parameters

Hourly wind speed time data for the period 2003-2011 (9 year series) have been measured in Regina, Saskatchewan, a region of central Canada [9]. These 9 years data have been used to calculate hourly mean wind speed values. The one year time series of wind speed have then been transformed in a time series of wind power through the characteristic curve (power curve) of a wind turbine, defined in Section 2.1. One year (8736 h) load data, i.e. load profile over 1 year with 1-h time step, have been generated according to the load model described in Section 2.2. Fig. 6 shows raw time series data sets, for both total wind power of WTG units, with a maximum value of 37.36 MW and load, with a maximum value of 196.88 MW. Both time series data sets show remarkable fluctuations along time.

The architecture of the NN model used consists of one input, one hidden and one output layers. The number of input neurons is set to 4 for both load and wind power PIs estimations, since an auto-correlation analysis [39] has shown that the historical past values x_{t-1}, x_{t-2}, x_{t-3}, and x_{t-4} should be used as input variables for predicting x_t in output; the number of hidden neurons is set to 10 after a trial-and-error process; the number of output neurons is set to 2, to provide the lower and upper bounds. As activation functions, the hyperbolic tangent function in the hidden layer and the logarithmic sigmoid function in the output layer have been found to give the most satisfactory results.

To account for the inherent randomness of NSGA-II, five different runs have been performed and an overall best non-dominated Pareto front has been obtained from the five individual fronts. All data have been normalized within the range [0.1, 0.9].

Table 1. Conventional Generation Units' Reliability Data [37]

Unit size (MW)	Type	No. of units	MTTF (hr)	MTTR (hr)
5	hydro	2	4380	45
10	thermal	1	2190	45
20	hydro	4	3650	55
20	thermal	1	1752	45
40	hydro	1	2920	60
40	thermal	2	1460	45

Table 2 contains the parameters of the NSGA-II for training the NN. "MaxGen" indicates the maximum number of generations which is used as a termination condition and Nc indicates the total number of individuals per population. P_c indicates the crossover probability and is fixed during the run. P_{m_int} is the initial mutation probability and it decreases at each iteration (generation) by the formula:

$$P_{m\_int} \times e^{(-\frac{gen}{MaxGen})} \tag{25}$$

Table 2. NSGA-II Parameters Used in the Experiments

Parameter	Numerical value
MaxGen	300
Nc	50
P_c	0.8
P_{m_int}	0.06

4.3. Estimated PIs

The multi-objective NSGA-II with PI coverage probability and width provides Pareto sets of solutions (one for the wind power and one for the load), i.e. optimal NN models (weights); it is, then, necessary to select the optimal sets of weights to use in the NN models for prediction (see Fig. 7).

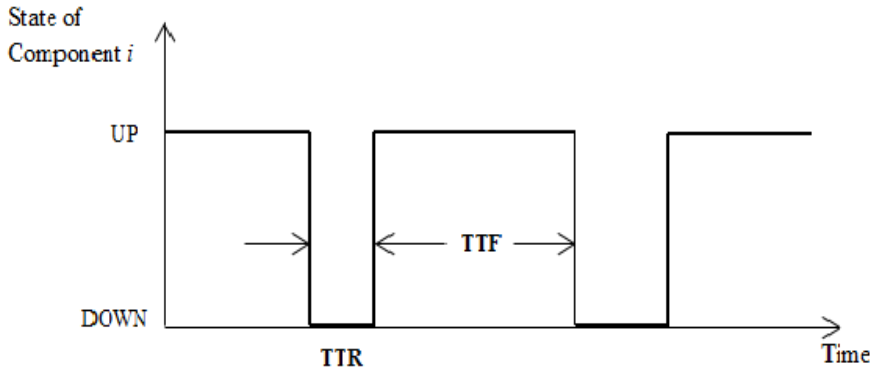


Fig. 5. Example of an operating/repair scenario for a component.

In practice, the selection of the solution mainly depends on the preferences of the DMs. In addition to DMs' subjective choices, some quantitative procedures can be also performed. Zio, et al. [40] have described three methods to choose a compromise solution from a Pareto-optimal front. Each method results in a different solution which locates differently in the Pareto-frontier. More precisely, one solution might be towards the center of the Pareto-front, so then gives lower PICP but on the other hand narrower PIWs; whereas another might has higher coverage probability with larger interval size.

In the light of the methods defined in [40], Ak, et al. [13] have employed two different selection procedures for choosing a solution, with reference to the Pareto-optimal front obtained after training.

For exemplification purposes, solutions are here subjectively chosen as a good compromise in terms of high PICP and low NMPIW. The selected solutions are characterized by 95 % PICP and a NMPIW equal to 0.265 for the load prediction, and 95 % PICP with a NMPIW equal to

0.19 for the wind power prediction, respectively. Note that in Fig. 7, the X axis indicates "1-PICP".

Fig. 8 shows 1-hour ahead PIs for the selected Pareto solutions, marked in rectangles in Fig. 8, estimated by the trained NNs for wind power from one turbine and load predictions. For the sake of clarity of visualization, a zoom on the first 250 hours has been plotted.

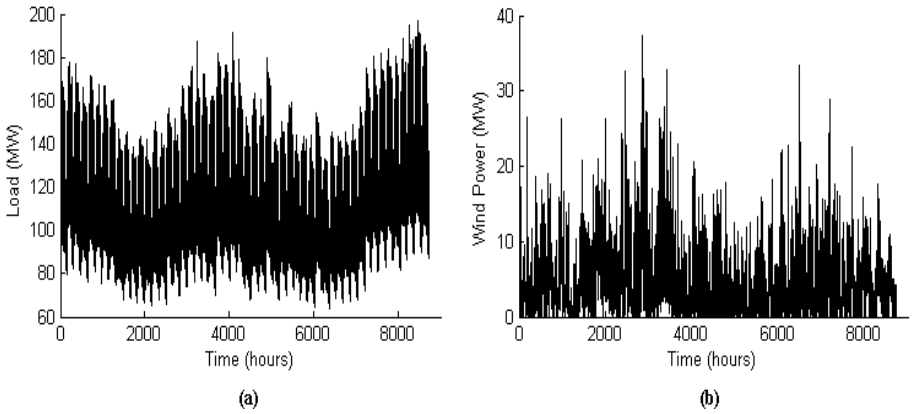


Fig. 6. The wind power time series set and load curve over 1 year used in this study: (a) load (b) wind power.

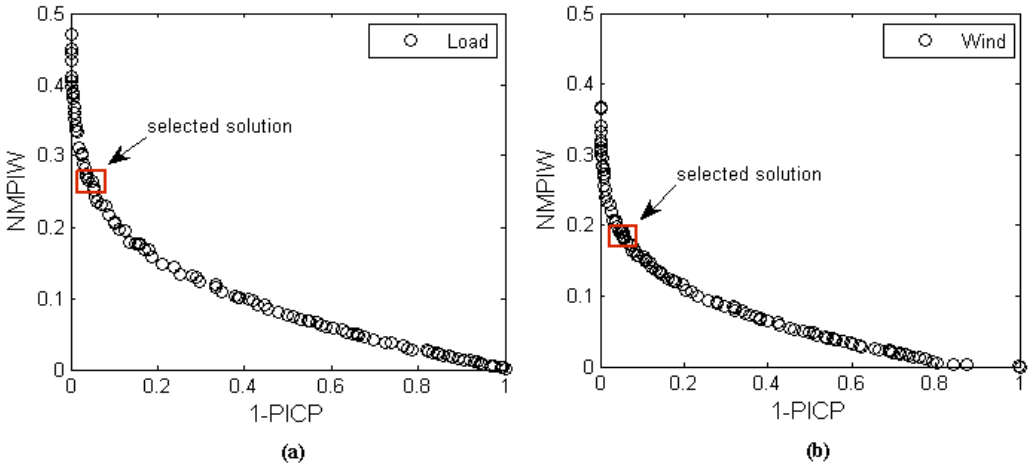


Fig. 7. The overall best Pareto front obtained by training of the NNs for 1h-ahead load and wind power predictions: (a) load (b) wind power.

As an alternative method to estimate PIs for wind power, we have used the histogram and empirical cumulative distribution function (cdf) of wind power at time t using the historical data. For exemplification of this analysis, we have used only winter data with respect to the

seasonality in the entire dataset which is one year in our EENS estimation. More precisely, for each time instant (hour) t in a day (1, 2,…, 24), we have collected 89 historical data samples over three months period of winter. By using these historical wind power data samples, we have constructed the histogram and empirical cdf for each hour t, so we have obtained 24 histograms. Fig. 9 shows the histogram and the cumulative distribution function of the wind power at hour 3 in any winter day.

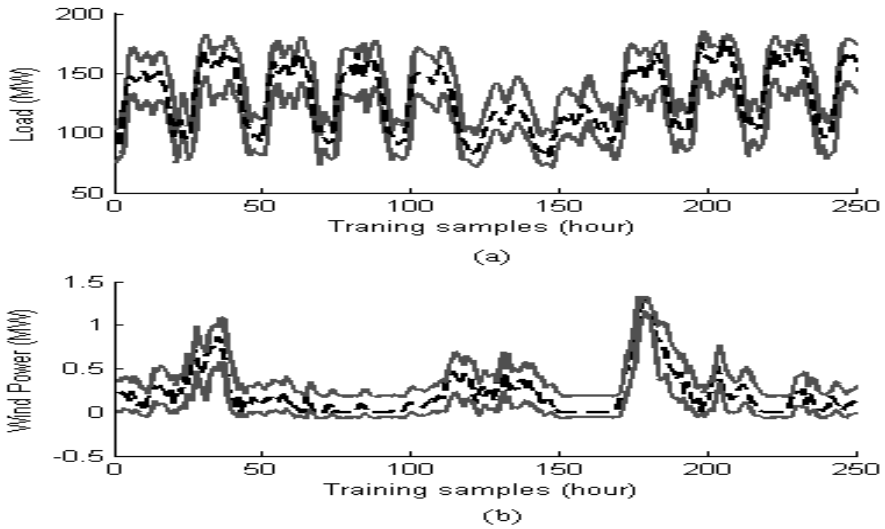


Fig. 8. Estimated PIs (solid lines) over a 1-year time horizon and target data (dashed lines): (a) 1h-ahead load (b) 1h-ahead wind power from one turbine.

To obtain PIs for each hour *t*, we have set the confidence level to 95% and recorded the lower and upper bounds corresponding to this confidence level from the histogram of the hour in question. For each day, we have used the same PIs obtained for each hour of the day. On this basis, we have calculated both the prediction interval coverage probability (PICP) and interval width of the estimated PIs on the entire testing set (target): the PICP is 95% with a NMPIW equal to 0.448.

For comparison purpose, we have selected a Pareto optimal solution obtained by the trained NN with 95 % PICP that corresponds to NMPIW equal to 0.19. Fig. 10 shows 1-hour ahead PIs for the winter period obtained by the selected Pareto solution and the histogram. For the sake of clarity of visualization, a zoom on the 24 hours has been plotted. Note that due to the high frequency of 0 as a value of the wind power for each hour t, the lower bound of the PIs estimated by the empirical distribution is also 0 for each hour t (see Figs. 9 and 10).

From the inspection of Fig. 10, one can see that the PIs obtained by the empirical distribution, i.e. the histogram, do not give accurate and reliable coverage for the target of interest. NN-based PIs obtain the same coverage probability (95 %) with lower interval size. One can appreciate that the PIs estimated by the histogram of wind power at time t cannot provide useful information in practice, since the uncertainty level in the outcome is too high, i.e. the interval size is too large. On the contrary, the training of the NN with wind speed historical data ensures accounting for the time dependency among successive observations, leading to more accurate predictions.

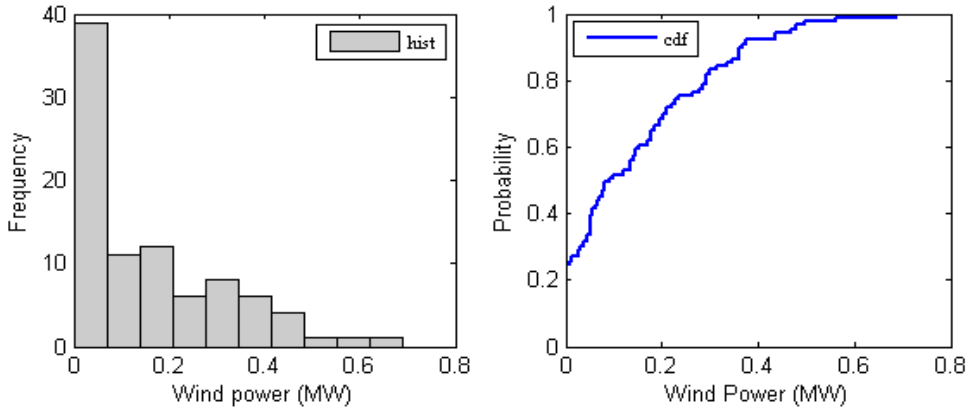


Fig. 9. The histogram (left) and the cumulative distribution function (right) of the wind power at hour 3 in any winter day.

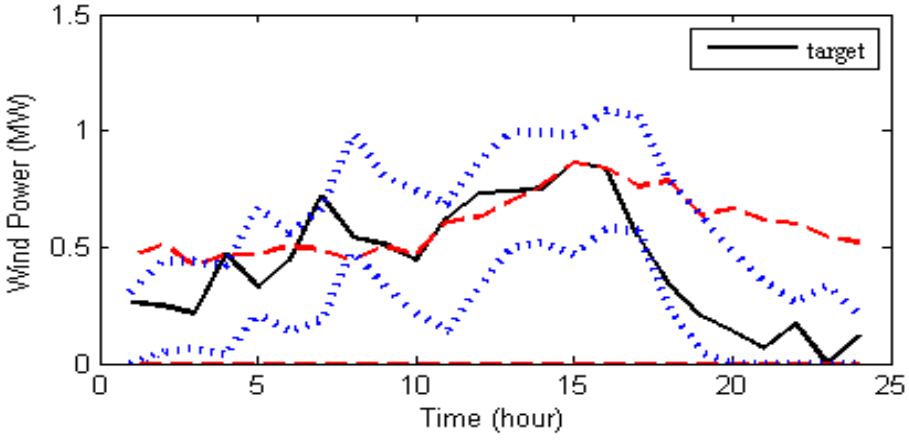


Fig. 10. Estimated PIs by the trained NN (dotted lines) and by the empirical distribution (dashed lines) over winter dataset, and true target data (solid line).

4.4. Estimated EENS

To estimate the overall EENS accounting for failures and repairs of the components, we performed 100 repetitions. In each repetition, a new matrix of the up and down states of the components is generated. Then, for each repetition the assessment process is followed with the same estimated load and wind power PIs and conventional units' generation capacity. Fig. 11 shows the ENS results obtained according to the methods explained in Sections III-A and III-B. It can be noticed that the estimated PIs of ENS include the point estimations (PEENS) of the expected energy not supplied for each simulation (see Section 3.2). With respect to the results shown in Fig. 11(a), we have estimated the expected energy not supplied, i.e. interval EENS, for the interval ENS via (10): [18000, 26788.13]. Note that this interval includes the EPEENS value reported in Table 3. We can interpret the PEENS value as the expected value of the interval ENS for each simulation run (see Fig. 11 and Section 3). Although both approaches are based on interval-valued inputs, interval-valued ENS carries more information, i.e. reflects the worst and best cases of unavailable energy during the given time horizon, and it provides an indication of how the uncertainties in input affect the output quantities.

Table 3. Descriptive Statistics of EENS over 100 Simulations

	Mean	**Std dev.**
EENS_LB	19278.93	11090.22
EENS_UB	25521.84	14663.75
EENS_mean	22381.89	12859.20
EPEENS	22419.12	12888.83
EENS_actual	22769.24	13147.55
EENS_MC	21320.00	13590.00

The ENS values in Fig. 11(b) have been obtained by considering 6 different scenarios, corresponding to the different uncertainty levels in the input parameters, i.e. wind power, load and system state. These scenarios have been called PEENS, interval ENS, ENS LB, ENS mean, ENS UB and ENS actual. The former two are also shown in Fig. 11(a) separately. ENS

LB and ENS UB have been calculated by considering only the LB and UB of the estimated load and wind power PIs, respectively, and by computing a single-valued inputs ENS index. Similarly, to estimate ENS mean, the central values (mean point) of the PIs have been used as input. For computing ENS actual, we have used the actual data sets shown in Fig. 6: ENS actual is, thus, the unknown quantity we would like our estimates to be close to, and it cannot be computed in a real case study; we have calculated it here only for demonstration of the strength of our approach.

Note that, differently from the PEENS and interval ENS, the values of ENS LB, ENS UB, and ENS mean are calculated with single-valued load and wind power inputs. Table 3 reports the mean and standard deviations of the ENS LB, ENS UB, ENS mean, PEENS and ENS actual results over 100 simulations, so it gives the expected energy not supplied calculated by (10) and (22), over one year period for each scenario. Note that, being capable of properly accounting also for uncertainties, EPEENS is the closest to EENS actual on average (and with comparable variability).

EENS is equal to zero when there is no failure of conventional generators (see Fig. 11), because at any time t over the given time horizon, the total available generation capacity, $G(t)$ is bigger than the total load $L(t)$, i.e. the maximum load value is less than 200 MW, whereas the total capacity of conventional generation units (GUs) is 240 MW. In case of failures, EENS takes different non-zero values according to the load and wind levels. This means that if there is no failure in the system, the system is able to meet the total energy demand. Note that load and wind levels do not change through the runs. Hence, there is only one factor that can affect the system EENS value through the runs and this factor is the failure of the GUs. Thus, the magnitude of the EENS values (see Table 3) shows the effect of the failures on the system adequacy.

For our specific problem, a component can be evaluated as critical if its failure rate is relatively larger and, at the same time, its generation capacity is bigger than the others. In our case study, two conventional generation units have biggest failure rates and generation capacities (see Table 1). In case of the failure of these two components, the system might not be able to meet the demand, thus determining a positive EENS value. In order to identify the most critical GUs, i.e. those which contribute more to EENS of the system in case of their

failure, one can apply a possible quantitative strategy, based on the link between the resulting EENS value and the GUs failure rate.

We, here, briefly outline the methodology for assessing the GUs impact on EENS, even if it is beyond the scope of the present paper:
Step 1) For each simulation, assign a value to each generation unit i, calculated by the multiplication of a weight value that is proportional to the amount of time in which the unit was OFF in that simulation; this weight could be, for instance, w_i= (time the unit i was OFF)/(total time of the simulation), and the power generation capacity of this unit i, C_i.
Step 2) For each unit i, sum up all the values $w_i \times C_i$ computed in Step 1 along the runs, and divide by the sum of the simulation runs, i.e. 100: the higher this value, the more the unit contributes to the EENS of the system (i.e. the more the unit is critical).
One can, then, use the results of such analysis to identify the components (by calculation of the component importance indices) which have high contribution to the expected annual energy not supplied. Also, if the estimated EENS is unacceptable, technical interventions may be needed, e.g. new generation units might be added to the system.

As each scenario carries different information, ultimately the decision makers are supposed to select the one which gives a more interpretable result for their final decisions/actions. Fig. 12 shows the boxplots of the differences obtained by the subtraction of ENS_actual from the ENS_LB, ENS_UB, ENS_mean and PEENS, respectively. A boxplot is an exploratory graphic used to visualize key statistical measures, such as median and quartiles, and to have an idea about the distribution of a data set, i.e. the location, dispersion, and symmetry or skewness of the data set, at a glance [41], [42]. It is also used to make comparisons of these features in two or more data sets. The boxplots dif_mean and dif_point are comparatively shorter (meaning narrower distributions) than the boxplots dif_LB and dif_UB. This fact indicates a higher variability for the estimates of ENS obtained using PIs LB and UB, compared to the ones based on PEENS and mean. In other words, ENS mean and PEENS show comparable results, which are also more consistent with respect to the actual values of ENS throughout the simulations (not just on average, as we could already conclude from Table 3). Since the PEENS is estimated on the basis of the load and power PIs, i.e. it takes into account the uncertainties in the inputs, it is more precise and reliable compared to the others. Hence, among all the possible estimates of ENS that could be obtained, PEENS shows

more promising and trustable results in capturing the actual ENS by considering the uncertain inputs.

On the basis of the comparisons shown in Fig. 11 and Table 3, where load demand and wind power generation take different values according to the considered scenario, the conclusion that different load and wind levels result in different EENS can be drawn. From the results reported in Table 3 we can observe that, for values of the load corresponding to the upper bound of PIs, a bigger EENS is obtained compared to the one obtained in other scenarios. It is worth to remark that, in the same scenario, an increase in the wind level would reduce the EENS. If we consider only the LB and UB of the estimated wind power, and we look at load PIs, total wind power covers 1.3 % and 5.8 % of the total load during the given time horizon (1-year) for the LB scenario and UB scenario, respectively. This is due to the low wind power penetration for both scenarios. Therefore, the change in the wind power penetration from the LB to the UB of the wind power PIs, does not play a significant role in the variation of the EENS value, whereas the change from LB to UB of load causes a larger amount of energy not supplied. Then, when we consider the LB scenario of load, it is expected to obtain a lower EENS value (see Table 3). In other words, when the wind penetration level is low and total load is relatively high, the combined impact of the uncertainties of the load and the system component failures on the EENS value is more dominant than the wind variability, as expected.

Ultimately, having an estimate of EENS with an associated variability helps the decision makers in managing the system on the basis of a more realistic/reliable adequacy assessment.

4.5. Comparison with a Method based on MC Simulation

In this section we discuss the results obtained by estimating EENS using a probabilistic Monte Carlo (MC) simulation method [43]. Load and wind speed are assumed to be random variables. In order to determine the probability density function (pdf) for wind, we have used the same wind speed time series data set, i.e. hourly mean wind speed values, described in Section 4.2. In order to sample the load in each repetition, we have used (1-3) described in Section 2.2.

For wind power, first we have generated the wind speed values from the corresponding probability density functions (pdfs). To do so, we have split the entire data set into four parts with respect to the seasonality. In other words, each subset represents a season (winter, spring, summer and autumn). Table 4 reports the characteristics of the pdfs used for wind speed. It is worth saying that as the wind speed is random, the wind power output is also random. However, we have generated wind power values, $P(t)$, using the power curve whose parameters have been given in Section 4.

In each repetition, a new matrix of the up and down states of the GUs is generated using the failure model defined in Section 4.1. Note that, load and wind speed values also change through the runs. Then, for each repetition the assessment process is followed with the sampled load and the wind power values generated from the sampled wind speeds.

It is worth saying that for each subset of date, normal, gamma and weibull distributions have been fit with respect to the shape of the histogram. Then, the ultimate pdfs have been chosen according to the best fits. All choices are reported in Table 4.

From the inspection of the results reported in Table 3, we can conclude that EENS_MC estimated by the probabilistic MC approach gives a point value within the EENS_LB and EENS_UB. However, the EENS_MC is less accurate with respect to the actual EENS if compared to the EPEENS value.

Using this MC method, we can represent random behavior of both load and wind speed, and hence wind power, in each run. However, when using MC simulation, it is very important to choose an appropriate probability distribution function to sample from.

An alternative methodology, similar to our proposal, but based on the sampling of wind speed and load values from a proper probability distribution, could be stated as follows: the pdfs both for load and wind speed could be estimated by using the time series data we use for PIs estimation [43], [44]. Then, in each run of the simulation, load and wind speed values can be sampled for 1-year time horizon by using the pdfs. The sampled wind speed values can be transformed to the wind power values by using the characteristic curve (power curve) of a wind turbine. These samples are used as inputs to NN to estimate PIs of load and wind power. When computing interval-based point EENS through (18-20), the previously estimated pdfs are used to approximate the integrals, rather than making a uniform distribution assumption.

Thus, for each time step *t*, load and wind power output values are sampled within the intervals based on the pdf functions. Finally, this process is repeated for each run. Note that this methodology does not generate an interval valued EENS, but it provides a point EENS with an associated variability measure given by Monte Carlo sampling.

This second version of the MC approach could provide more information but at a higher computational cost. This is because in MC estimates randomization is needed over all unknowns in the problem, and this means for our case study randomizing over wind, load, and failures of units (as we are currently doing). Hence in this perspective, a comparison between this alternative and our approach should be carried out with caution. In addition, compared to the methodology proposed in the present work, this alternative seems redundant: it might not be needed to estimate new PIs in each repetition, since these PIs would be based on data sampled from a common pdf, and thus they are expected to be consistent. This is the main reason why we have assumed the estimated PIs to be the same for each repetition.

Finally, an MC-based EENS distribution will heavily depend on assumptions made on the load and power distributions, while our approach can be used based on the assumption, uniform distribution, in absence of enough information to build proper pdfs, without losing accuracy, robustness and generalization in realistic applications.

Table 4. PDFs Used in the MC Experiment to Sample Wind

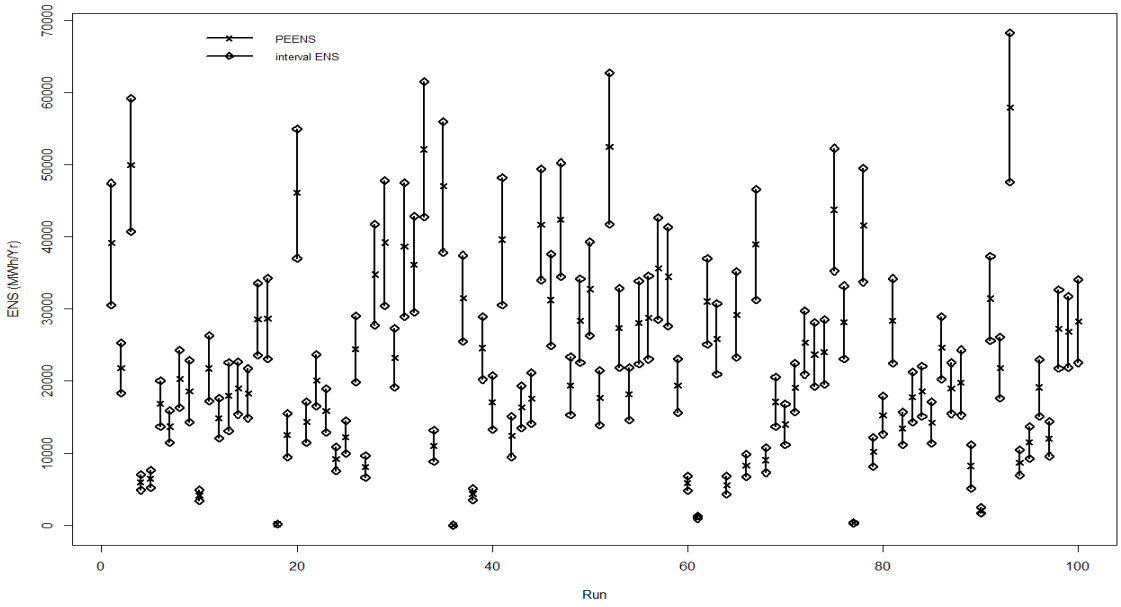
Season	**Pdfs**
Winter	Normal (18.2, 3.68)
Spring	8 + Gamrna (1.67, 7.31)
Summer	6 + Gamrna (1.62, 6.46)
Autumn	7 + Weibull (11.3, 2.84)

5. CONCLUSION

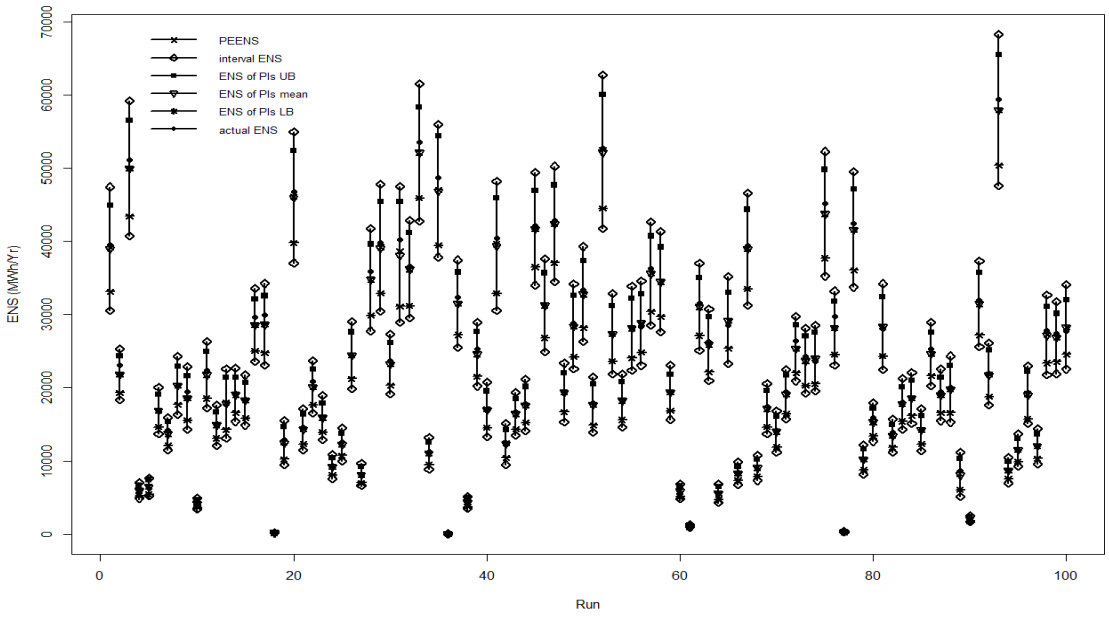
A method which calculates the EENS value for a wind-integrated power network based on interval-valued load and wind power input data has been proposed. The objective is to know and dominate the impact of the uncertainty in wind and load on the uncertainty in EENS.

Simulation results on different scenarios confirm that uncertainties in input data can be properly taken into account to obtain more reliable EENS estimations.

The presented expected annual energy not supplied can be integrated with a cost model whose results help the decision makers to take operational level decisions and do medium-term and long-term strategic planning.



(a)



(b)

Fig. 11. The ENS results over 100 runs: (a) interval-valued ENS and PEENS (b) comparisons of different scenarios.

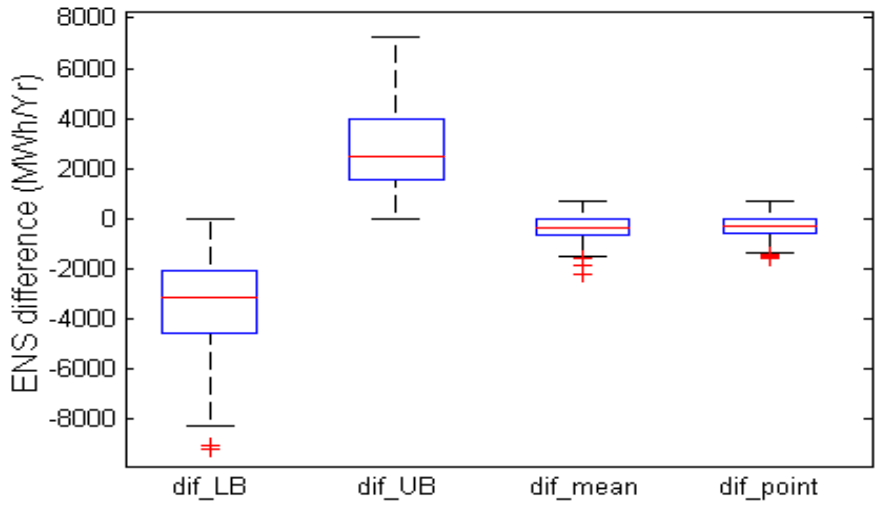


Fig. 12. Boxplots of the differences obtained by the subtraction of ENS_actual from the ENS_LB, ENS_UB, ENS_mean and PEENS, respectively.

REFERENCES

[1] E. Zio and T. Aven, "Uncertainties in smart grids behavior and modeling: What are the risks and vulnerabilities? How to analyze them?," *Energy Policy*, vol. 39, no. 10, pp. 6308-6320, Oct. 2011.

[2] Y-F. Li and E. Zio, "A multi-state model for the reliability assessment of a distributed generation system via universal generating function," *Reliability Engineering & System Safety*, vol. 106, pp. 28-36, Oct. 2012.
[3] Z. Wei, T. Tao, D. ZhuoShu, and E. Zio, "A dynamic particle filter- support vector regression method for reliability prediction*," Reliability Engineering & System Safety*, vol. 119, pp. 109-116, Nov. 2013.
[4] R., Billinton and R. N. Allan, *Reliability Evaluation of Power Systems*. 2nd ed., New York: Plenum Press, 1996.
[5] G. A. Koeppel, "Reliability considerations of future energy systems: multi-carrier systems and the effect of energy storage," Ph.D. dissertation, Power System Laboratory, Swiss Federal Institute of Technology, Zurich, 2007.
[6] J. Wen, Y. Zheng, and F. Donghan, "A review on reliability assessment for wind power," *Renewable and Sustainable Energy Reviews*, vol. 13, no. 9, pp. 2485–2494, Dec. 2009.
[7] Y. Gao, R. Billinton, and R. Karki, "Composite generation and transmission system adequacy assessment considering wind energy seasonal characteristics," IEEE Power Energy Society General Meeting, PES '09 IEEE, Calgary, Canada, July 26-30, 2009.
[8] B. Falahati, Y. Fu, Z. Darabi, and L. Wu, "Reliability assessment of power systems considering the large-scale PHEV integration," Vehicle Power and Propulsion Conference (VPPC), 2011 IEEE, Chicago, USA, Sep. 6-9, 2011.
[9] Canadian Weather Office. [Online]. Available: http: //www.weatheroffice.gc.ca/canada_e.html. [Accessed: 01-Jan-2013].
[10] Reliability Test System Task Force of the Application of Probability Methods Subcommittee, "IEEE Reliability Test System*," IEEE Trans. on Power Apparatus and Systems*, vol. PAS-98, no. 6, pp. 2047–2054, Nov. 1979.
[11] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II ," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, Apr. 2002.
[12] 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. on Neural Networks*, vol. 22, no. 3, pp. 337-346, March 2011.
[13] R. Ak, Y. F. Li, V. Vitelli, and E. Zio, "Multi-objective Genetic Algorithm Optimization of a Neural Network for Estimating Wind Speed Prediction Intervals," *Applied Soft Computing* (under review), 2014.
[14] R. Karki, P. Hu, and R. Billinton, "A simplified wind power generation model for reliability evaluation," *IEEE Trans. on Energy Conversion*, vol. 21, no. 2, pp. 533–540, June 2006.
[15] C.G. Justus, W.R. Hargraves, and A. Yalcin, "Nationwide assessment of potential output from wind-powered generators," *Journal of Applied Meteorology*, vol. 15, no.7, pp. 673–678, July 1976.
[16] A.B. Rodrigues and M.G. Da Silva, "Probabilistic assessment of available transfer capability based on Monte Carlo method with sequential simulation," *IEEE Transactions on Power Systems*, vol. 22, no. 1, pp. 484–492, Feb. 2007.
[17] R. Billinton and W. Li, Reliability Assessment of Electrical Power Systems Using Monte Carlo Methods. New York: Plenum, 1994.
[18] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.
[19] D.L. Shrestha and D.P. Solomatine, "Machine learning approaches for estimation of prediction interval for the model output," Neural Networks vol. 19, no. 2, pp. 225–235, 2006.
[20] D. Svozil, V. Kvasnicka, and J. Pospichal, "Introduction to Multi-layer Feed-forward Neural Networks," *Chemometrics and Intelligent Laboratory Systems*, vol. 39, no. 1, pp. 43-62, 1997.
[21] Z. Guoqiang, B. E. Patuwo, and M. Y. Hu. "Forecasting with artificial neural networks: The state of the art," *International journal of forecasting*, vol. 14, no. 1, pp. 35-62, 1998.
[22] A. A. da Silva and L. S. Moulin, "Confidence intervals for neural network based short-term load forecasting," *IEEE Transactions on Power Systems*, vol. 15, no. 4, pp. 1191-1196, 2000.
[23] H. Quan, D. Srinivasan, and A. Khosravi, "Short-Term Load and Wind Power Forecasting Using Neural Network-Based Prediction Intervals," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 2, pp. 303-315, Feb. 2014.
[24] Zhang, Yao, Jianxue Wang, and Xifan Wang. "Review on probabilistic forecasting of wind power generation," Renewable and Sustainable Energy Reviews, 32 (2014): 255-270.
[25] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya, "Comprehensive review of neural network-based prediction intervals and new advances," *IEEE Transactions on Neural Networks*, vol. 22, no. 9, pp. 1341-1356, Sept. 2011.
[26] J. B. Bremnes, "Probabilistic wind power forecasts using local quantile regression," Wind Energy, vol. 7, no. 1, pp. 47–54, 2004.

[27] P. Pinson, H. Madsen, H. A. Nielsen, G. Papaefthymiou, and B. Klöckl, "From probabilistic forecasts to statistical scenarios of short-term wind power production," Wind energy, vol. 12, no. 1, pp. 51-62, 2009.
[28] P. Pinson, "Very-short-term probabilistic forecasting of wind power with generalized logit–normal distributions," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 61, no. 4, pp. 555-576, 2012.
[29] A. Konak, D.W. Coit, and A.E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial," *Reliability Engineering & System Safety*, vol. 91, no. 9, pp. 992–1007, Sep. 2006.
[30] R. Ak, Y. F. Li, V. Vitelli, E. Zio, E. López Droguett, and C. Magno Couto Jacinto, "NSGA-II-trained neural network approach to the estimation of prediction intervals of scale deposition rate in oil & gas equipment," *Expert Systems with Applications*, vol. 40, no. 4, pp. 1205-1212, March 2013.
[31] Y-F. Li, N. Pedroni, and E. Zio, "A Memetic Evolutionary Multi-Objective Optimization Method for Environmental Power Unit Commitment*," IEEE Transactions on Power Systems*, vol. 28, no.3, pp. 2660-2669, Aug. 2013.
[32] M. A. Abido, "Environmental/economic power dispatch using multiobjective evolutionary algorithms," *IEEE Transactions on Power Systems*, vol. 18, no. 4, pp. 1529-1537, Nov. 2003.
[33] R., Billinton and R. N. Allan, *Reliability Assessment of Large Electric Power Systems.* Springer, 1988.
[34] Power Systems Engineering Committee, "Reliability indices for use in bulk power supply adequacy evaluation," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-97, no. 4, pp. 1097–1103, July/Aug. 1978.
[35] R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to Interval Analysis, Society for Industrial and Applied Mathematics*. 1st ed., USA, 2009, pp. 1-235.
[36] S. M. Ross, *Introduction to Probability Models*. 10th Ed., Elsevier Inc., USA, 2010.
[37] R. Billinton, S. Kumar, N. Chowdhury, K. Chu, K. Debnath, L. Goel, E. Khan, P. Kos, G. Nourbakhsh,, J. Oteng-Adjei, "A reliability test system for educational purposes-basic data," *IEEE Trans. on Power Systems*, vol. 4, no. 3, pp. 1238–1244, Aug. 1989.
[38] R. Billinton and P. Wang, "Teaching distribution system reliability evaluation using monte carlo simulation," *IEEE Transactions on Power Systems*, vol. 14, no. 2, pp. 397-403, May 1999.
[39] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, Time Series Analysis: Forecasting and Control. 4th ed., Wiley, 2008.
[40] E. Zio, P. Baraldi, and N. Pedroni, "Optimal power system generation scheduling by multi-objective genetic algorithms with preferences," Reliability Engineering and System Safety, vol. 94, pp. 432-444, 2009.
[41] J. L. Devore and K. N. Berk. Modern mathematical statistics with applications. Springer, London, 2011.
[42] F. Mosteller and J. W. Tukey, Data Analysis and Regression: A Second Course in Statistics. Addison-Wesley Series in Behavioral Science: Quantitative Methods, Reading, Mass.: Addison-Wesley, 1977, 1.
[43] Y. M. Atwa, E. F. El-Saadany, and A-C. Guise. "Supply adequacy assessment of distribution system including wind-based DG during different modes of operation," IEEE Transactions on Power Systems, vol. 25, no. 1, pp. 78-86, Feb. 2010.
[44] N. Maisonneuve and G. Gross, "A Production Simulation Tool for Systems With Integrated Wind Energy Resources," IEEE Transactions on Power Systems, vol.26, no.4, pp.2285, 2292, Nov. 2011.

PAPER VI

Uncertainty Modeling in Wind Power Generation Prediction by Neural Networks and Bootstrapping

R. Ak, V. Vitelli and E. Zio. *In Proc Esrel 2013 Conference*, 29 Sept. – 2 Oct. 2013, Amsterdam.

Uncertainty Modeling in Wind Power Generation Prediction by Neural Networks and Bootstrapping

Ronay Ak[a], Valeria Vitelli[b], and Enrico Zio[a, c]

[a]Chair on Systems Science and the Energetic Challenge, European Foundation for New Energy-Electricité de France, CentraleSupélec, Châtenay-Malabry 92290 and Gif-Sur-Yvette 91192, France

[b]Department of Biostatistics, University of Oslo, Oslo, Norway

[c]Department of Energy, Politecnico di Milano, Milan 20133, Italy

ABSTRACT

Accurate short-term wind power forecasting with quantification of the associated uncertainty is crucial for the management of energy systems including wind power generation. On top of the inherent uncertainty in wind speed, it is necessary to account also for the uncertainty in the relationship between wind speed and the corresponding power production, typically described by a power curve whose characteristic parameters are not precisely known in practice. In this paper, we propose a novel approach to wind power forecasting with uncertainty quantification. The approach can be schematized in two steps: first, short-term estimation of wind speed prediction intervals (PIs) is performed within a multi-objective optimization framework worked out by non-dominated sorting genetic algorithm–II (NSGA-II); then, the uncertainty in wind speed and the uncertainty in the power curve are combined via a bootstrap sampling technique, thus obtaining wind power PIs with same coverage as the wind speed PIs.

1. INTRODUCTION

Power production via renewable energy sources is a hot topic of research and application. This is due to both the widespread availability of such sources (e.g. wind, sun, etc.) and to the sustainability of the associated production process. Among renewable energy sources, wind power is widely recognized as one of the most promising, because of its tremendous potential in commercialization and bulk power generation.

The management of wind power generation systems relies on short-term wind power generation forecasting, which must also provide a measure of the associated uncertainty. Two uncertainty sources can be considered: the inherent uncertainty in wind speed, due to the intermittent and unstable nature of wind (aleatory uncertainty); the uncertainty in the relationship between wind power and wind speed (epistemic uncertainty) (Helton 1994). The latter uncertainty is mainly due to the parameters defining the power curve (cut-in, rated and cut-off speeds, and rated power), which can be different for each single turbine within a wind farm (Novoa & Jin 2011).

In the present work, we treat the power curve parameters as random variables and account for the epistemic uncertainty by bootstrapping (Efron 1981), which allows combining also the aleatory uncertainty in the wind speed.

To do so, we first perform short-term forecasting of wind speed in a multi-objective optimization framework, where the non-dominated sorting genetic algorithm–II (NSGA-II) (Deb et al., 2002) is applied to optimize the weights of a neural network (NN) for estimating the prediction intervals (PIs) of wind speed. We, then, combine the uncertainty in wind speed forecasting with the uncertainty in the power curve via a bootstrap sampling technique. This results in obtaining wind power PIs with the associated uncertainty. By a precise probabilistic formulation, we show that the coverage probability of the wind power PIs obtained is the same as the one of wind speed PIs. Moreover, we test the robustness of the procedure with respect to the form of the distributions for the power curve random parameters.

The rest of the paper is organized as follows. In Section 2, the methodology for NN-based wind speed PIs estimation and for bootstrap-based wind power PIs estimation is introduced

and described. In Section 3, a case study is carried out to test the effectiveness of the proposed approach. Finally, in the Conclusion Section some final remarks are given.

2. METHODOLOGY

2.1. Estimation of Wind Speed PIs by NSGA-II

A PI is comprised of upper and lower bounds in which a future unknown value of the target is expected to lie with a predetermined confidence level $(1-\propto)$.The formal definition of a PI is thus (Geisser, 1993):

$$P(L(x) < y(x) < U(x)) = 1-\propto \quad (1)$$

where $L(x)$ and $U(x)$ indicate respectively the lower and upper bounds of the PI of the output $y(x)$ corresponding to input x; the confidence level $(1-\propto)$ refers to the expected probability that the true value of $y(x)$ lies within the PI, $[L(x), U(x)]$.

In order to provide wind speed PIs, we use multi perceptron artificial neural networks (NNs) (Korbicz et al. 2004) which are a class of nonlinear statistical models inspired by brain architecture, capable of learning complex nonlinear relationships among variables from observed data (Hornik et al. 1989), by a process of parameter tuning called "training". It is common to represent the task of such a NN model as one of nonlinear regression of the kind (Zio 2006, Shrestha & Solomatine 2006):

$$y(x) = f(x; w) + \varepsilon(x), \quad \varepsilon(x) \sim N(0, \sigma_\varepsilon^2(x)) \quad (2)$$

where x, $y(x)$ are the input and output vectors of the regression, which in our case represent measured historical wind speeds at time $t-1, t-2, \ldots t-r$ and the true target at time t, respectively. w represents the vector of values of the parameters of the model function f, in general nonlinear. The term $\varepsilon(x)$ is the error associated to the regression model f, and it is assumed normally distributed with zero mean.

We evaluate the PIs by the coverage probability of the prediction intervals (CP), which one wants to maximize, and the interval width (PIW), which one wants to minimize. The mathematical definitions of the PICP and PIW used in this work are (Khosravi et al. 2011):

$$PICP = \frac{1}{n_p}\sum_{i=1}^{n_p} c_i \quad (3)$$

where n_p is the number of samples in the training or testing sets, and $c_i = 1$, if $y_i \in [L(x_i), U(x_i)]$ and $c_i = 0$ otherwise. $L(x_i)$ and $U(x_i)$ are the estimated lower and upper bounds of the prediction interval in output, in correspondence of the input x_i.

$$NMPIW = \frac{1}{n_p}\sum_{i=1}^{n_p} \frac{(U(x_i)-L(x_i))}{t_{max}-t_{min}} \quad (4)$$

where NMPIW is the Normalized Mean PIW, and t_{min} and t_{max} represent the true minimum and maximum values of the targets (i.e., the bounds of the range in which the true values fall) in the training set, respectively. Normalization of the PI width by the range of targets makes it possible to objectively compare the PIs, regardless of the techniques used for their estimation or the magnitudes of the true targets.

The PIs estimation problem is addressed by taking into account the two conflicting objectives in a multi-objective framework. For this, we use NSGA-II, which is one of the most efficient multi-objective genetic algorithms (MOGAs) (Konak et al. 2006, Deb et al. 2002), to optimize the parameters (i.e. the weights) of the network taking into account both objectives. More precisely, the neural network is trained by NSGA-II to produce the lower and upper bounds of the prediction intervals for short-term forecasting (1-hour ahead) of wind speed. The practical implementation of NSGA-II on our specific problem involves two phases: initialization and evolution. These can be summarized as follows:

Initialization phase:

Step 1: Split the input data into training (D_{train}) and testing (D_{test}) subsets.

Step 2: Fix the maximum number of generations and the number of chromosomes (individuals) Nc in each population; each chromosome codes a solution by G real-valued genes, where G is the total number of parameters (weights) in the NN. Set the generation number $n = 1$. Initialize the first population P_n of size Nc, by randomly generating Nc chromosomes.

Step 3: For each input vector x in the training set, compute the lower and upper bound outputs of the Nc NNs, each one with G parameters.

Step 4: Evaluate the two objectives PICP and NMPIW for the Nc NNs (one pair of values 1-PICP and NMPIW for each of the Nc chromosomes in the population P_n).

Step 5: Rank the chromosomes (vectors of G values) in the population P_n by running the fast non-dominated sorting algorithm (Deb et al. 2002) with respect to the pairs of objective values, and identify the ranked non-dominated fronts $F_1, F_2, \dots, F_k$ where F_1 is the best front, F_2 is the second best front and F_k is the least good front.

Step 6: Apply to P_n a binary tournament selection based on the crowding distance (Deb et al. 2002), for generating an intermediate population S_n of size Nc.

Step 7: Apply the crossover and mutation operators to S_n, to create the offspring population Q_n of size Nc.

Step 8: Apply Step 3 onto Q_n and obtain the lower and upper bound outputs.

Step 9: Evaluate the two objectives in correspondence of the solutions in Q_n, as in Step 4.

Evolution phase:

Step 10: If the maximum number of generations is reached, stop and return P_n. Select the first Pareto front F_1 as the optimal solution set. Otherwise, go to Step 11.

Step 11: Combine P_n and Q_n to obtain a union population $R_n = P_n \cup Q_n$.

Step 12: Apply Steps 3-5 onto R_n and obtain a sorted union population.

Step 13: Select the Nc best solutions from the sorted union to create the next parent population P_{n+1}.

Step 14: Apply Steps 6-9 onto P_{n+1} to obtain Q_{n+1}. Set $n = n + 1$; and go to Step 10.

Finally, the best front in terms of ranking of non-dominance and diversity of the individual solutions is chosen. Once the best front of solutions is obtained, then the testing step is performed on the trained NN with optimal weight values.

2.2. Wind Power PIs Estimation

The wind power value $p(x)$ depends on the wind speed $y(x)$. Suppose that [$L_p(x)$, $U_p(x)$] is the PI associated to the wind power value $p(x)$ in correspondence of the input x, i.e. to the wind speed value $y(x)$. Then, the following property must hold:

$$P\left(L_p(x) \leq p(x) \leq U_p(x)\right) = 1 - \propto_p, \quad (5)$$

where $1 - \propto_p \epsilon [0,1]$ is the coverage probability.

Our working hypothesis stands on the fact that both wind power values and PIs depend on the wind speed values and PIs, respectively, via a non-monotonic transformation, namely the power curve. In this hypothesis, the rest of the subsection is devoted to the following two issues:

1. assess the value of $1 - \propto_p$ given the coverage probability of the PI associated to the wind speed $y(x)$;
2. develop a bootstrap-based approach to the estimation of $[L_p(x), U_p(x)]$.

In order to assess the coverage probability of wind power PIs, we have to take into account the fact that they have been obtained via a power curve transformation, which means:

$$L_p(x) = g_\theta\big(L(x)\big) \quad (6)$$

$$U_p(x) = g_\theta\big(U(x)\big) \quad (7)$$

where $[L(x), U(x)]$ is the PI for the wind speed value $y(x)$ associated to the input x, with associated coverage probability $1 - \propto_s$, while g_θ is a quadratic power curve transformation given by the following expression (Justus et al. 1976):

$$g_\theta(V) = \begin{cases} 0 & if\ V \leq V_{ci}\ or\ V > V_{co} \\ P_r(a + bV + cV^2) & if\ V_{ci} < V < V_r \\ P_r & if\ V_r < V \leq V_{co} \end{cases} \quad (8)$$

with

$$a = -\frac{V_{ci}(V_{ci}+V_r).(V_{ci}^2+2V_{ci}V_r-V_r^2)}{2(V_{ci}-V_r)^2.V_r^2} \quad (9)$$

$$b = \frac{V_{ci}^4+4V_{ci}^3.V_r+6V_{ci}^2V_r^2-2V_{ci}V_r^3-V_r^4}{2(V_{ci}-V_r)^2.V_r^3} \quad (10)$$

$$c = -\frac{V_{ci}^3+3V_{ci}^2.V_r+3V_{ci}V_r^2-3V_r^3}{2(V_{ci}-V_r)^2.V_r^3} \quad (11)$$

and with $\theta = (V_{ci}, V_r, V_{co}, P_r)$ being the vector of parameters defining the power curve, i.e. cut-in speed, rated speed, cut-off speed and rated power. A plot of the power curve g_θ is shown in Figure 1. In the following, we will consider V_{co} and P_r to be fixed (deterministic) values, and respectively equal to the values 30 m/s and 20 MW (Albadi & El-Saadany 2012, Akdag & Guler 2010), while V_{ci} and V_r are random variables with distributions F_{ci} and F_r, respectively. The inherent stochasticity in the power curve is motivated by the fact that different wind turbines correspond to specific power curve parameters, which leads to an imprecise and imperfect knowledge of the power curve transformation.

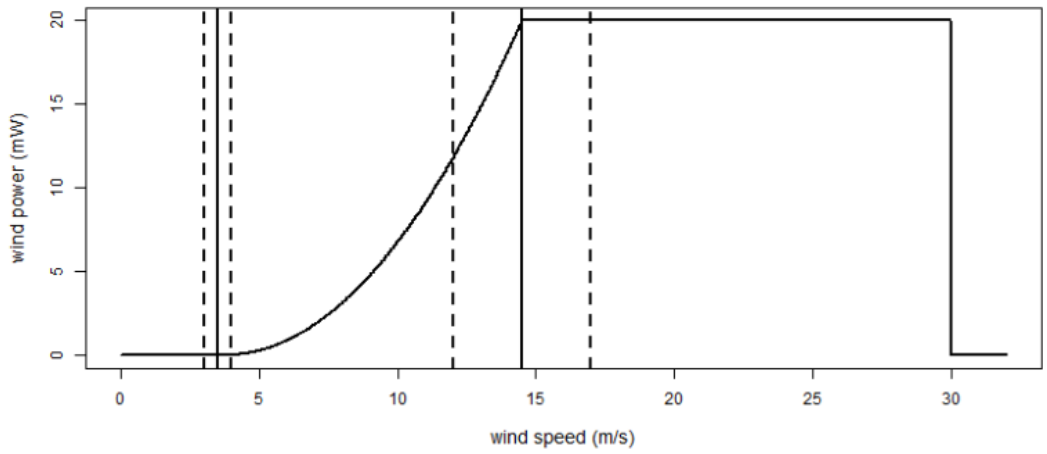


Figure 1. Plot of the power curve g_θ as a function of wind speed. Solid vertical lines correspond to the values of the two stochastic parameters V_{ci} and V_r. Dashed vertical lines identify the domains of the distributions F_{ci} and F_r, respectively.

The following chain of identities holds:

$$P\left(L_p(x) \le p(x) \le U_p(x)\right) = P\left(g_\theta(L(x)) \le p(x) \le g_\theta(U(x))\right) =$$
$$= \int_{\theta\epsilon\Theta} P\left(g_\theta(L(x)) \le p(x) \le g_\theta(U(x))\middle|\theta = \theta_0\right).P(\theta = \theta_0)\,.d\theta = \quad (12)$$
$$= \int_{\theta\epsilon\Theta} P\left(g_\theta^{-1}\left(g_\theta(L(x))\right) \le y(x) \le g_\theta^{-1}\left(g_\theta(U(x))\right)\middle|\theta = \theta_0\right).P(\theta = \theta_0).d\theta =$$
$$= \int_{\theta\in\Theta}(1-\propto_s)\,P(\theta = \theta_0)d\theta = (1-\propto_s).$$

The first and second equalities in Equation (12) derive from our working hypothesis, the third one from the theorem of total probability, the fourth one from the definition of coverage probability for wind speed PIs, and the last one stands because we integrate out the parameter vector θ over the whole probability space Θ.

Hence, we can conclude that $\propto_p = \propto_s$, i.e. the coverage probability is maintained while passing from wind speed PIs to wind power PIs via a wind power curve transformation.

We remark that, in general, $P\big(f(L) \leq f(x) \leq f(U)\big) = P(L \leq x \leq U)$ if and only if f is a strictly monotonic function, because in this case the existence of the inverse is ensured. The power curve transformation g_θ, whose definition is given in Equation (8), is non-monotonic, but it is monotonic when restricted to the open subset of the co-domain $(0, P_r)$. Note that the co-domain of the power curve is given by the closure of the latter subset, i.e. $[0, P_r]$. Hence, we can restrict our analysis to the open subset $(0, P_r)$, and treat the non-monotonicity issue as a border issue, a posteriori restricting the obtained wind power PIs to their domain of admissibility (note that this is usually done in the context of PIs estimation when the target of interest is a bounded variable, e.g. a proportion).

We now move to the problem of estimating the wind power PIs $[L_p(x_1), U_p(x_1)]$, … ,$[L_p(x_n), U_p(x_n)]$ corresponding to the testing set $\{x_i, y_i\}$, for $i = 1, \dots, n$.

Since the parameter vector θ is a multivariate random variable, the wind power PIs estimation process provides a distribution of intervals accounting for the parameters stochasticity. To get such a distribution, parametric bootstrap (Efron 1981, Shao & Tu 1995) is used. Parametric bootstrap is a technique which allows generating a sample for each parameter, and then estimating some relevant quantities concerning the target of interest.

More precisely, given the estimated wind speed PIs $[L(x_1), U(x_1)], \dots, [L(x_n), U(x_n)]$ in the testing set, the parametric bootstrap sampling technique is articulated in the following two steps:

1. Bootstrap Phase:

Sample two values for the stochastic parameters V_{ci} and V_r from the corresponding distributions, i.e. $V_{ci} \sim F_{ci}$ and $V_r \sim F_r$, and transform all wind speed PIs $[L(x_1), U(x_1)], \ldots, [L(x_n), U(x_n)]$ into wind power PIs $[L_p(x_1), U_p(x_1)], \ldots, [L_p(x_n), U_p(x_n)]$ via the power curve transformation g_θ defined in (8), and using the previously sampled parameter values. Repeat the sampling until a sufficient number of sets of wind power PIs has been obtained.

2. Aggregation Phase:

Aggregate the results of the bootstrap phase by computing, for each element of the testing set, the bootstrapped average wind power PI and the 5th and 95th percentiles of the wind power PI bootstrapped distribution.

This bootstrapping technique allows obtaining a set of wind power PIs, which accounts for both the aleatory uncertainty intrinsic in wind power generation and the variability associated to wind power PIs themselves, thus expressing the epistemic uncertainty related to the power curve estimation procedure. The aleatory uncertainty can be expressed by showing the bootstrapped average wind power PIs, while the epistemic one can be summarized in the 5th and 95th percentiles of the wind power PIs bootstrapped distribution.

3. CASE STUDY

The wind speed data used in this study have been measured for Regina, Saskatchewan, a region of central Canada (Canadian Weather Office, 2012) over a period of two months from 1st of February 2012 to 31st of March 2012. The total data set includes 1437 samples (see Fig. 2), among which the first 80% (the first 1150 samples) are used for training and the rest for testing. The architecture of the multi-perceptron NN consists of one input, one hidden and one output layers. The number of input neurons is 3 corresponding to the wind speed values of the previous three time steps (W_{t-1}, W_{t-2} and W_{t-3}); the number of hidden neurons is set to 10 after a trial-and-error process; the number of output neurons is 2, one for the lower and one for the upper bound values of the PIs. As activation functions, the hyperbolic tangent function is used in the hidden layer and the logarithmic sigmoid function is used at the output layer (these choices have been found to give the best results by trial and error, although the results

have not shown a strong sensitivity to them). The training of the NN weights is done by NSGA-II to maximize PICP (Equation 3) while minimizing (Equation 4). All data have been normalized within the range [0.1, 0.9]. To account for the inherent randomness of NSGA-II, twenty different runs have been performed and an overall best non-dominated Pareto front has been obtained from the twenty individual fronts.

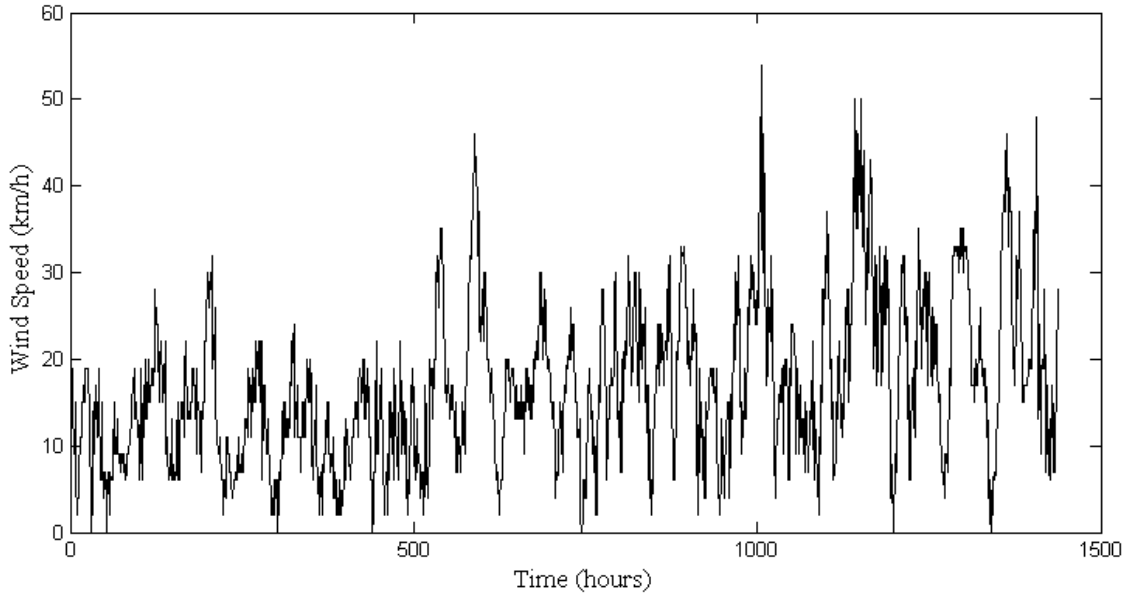


Figure 2. The wind speed data set used in this study.

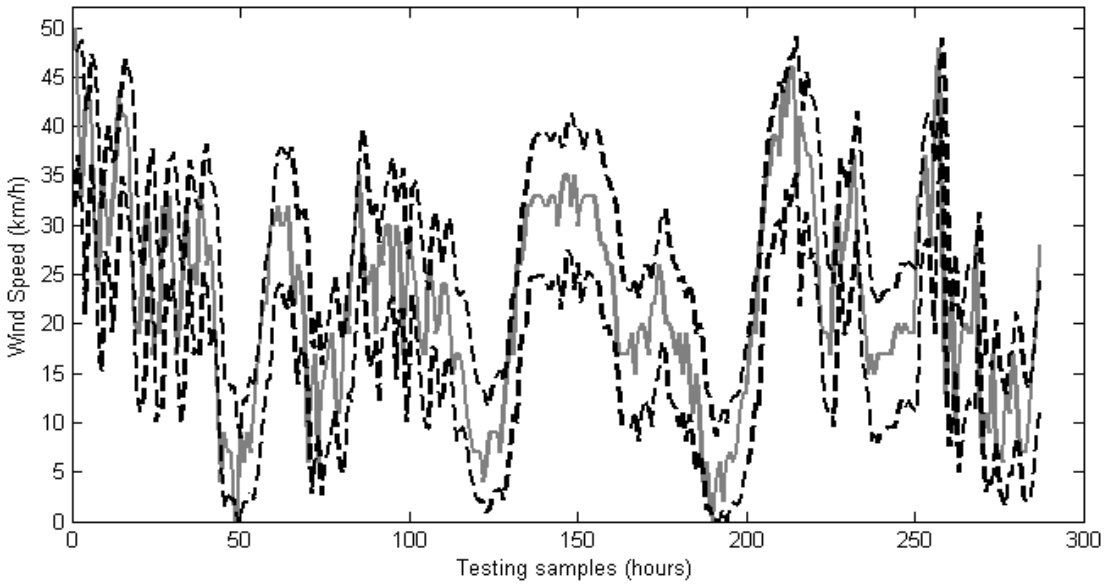


Figure 3. Estimated PIs for 1-hour ahead wind speed prediction on the test data set (dashed lines), and wind speed target data included in the test data set (solid line).

Given the overall best Pareto set of optimal solutions (i.e. optimal NN weights), one has to pick one (i.e. one trained NN) for use. We take a solution subjectively chosen, because judged to provide a good compromise in terms of high PICP and low NMPIW. The selected solution has 90% CP and 0.242 NMPIW on the training, and 82% CP and 0.255 NMPIW on the testing. Figure 3 shows wind speed target data (testing set) together with the estimated PIs corresponding to the selected solution.

The bootstrapping estimation technique described in the previous section is then applied to the estimated wind speed PIs (testing set) shown in Figure 3 to obtain wind power PIs. The number of bootstrap replicates has been set equal to 1000. In order to test the robustness of this bootstrapping technique with respect to the parametric assumption concerning the distribution of the power curve parameters, we sample V_{ci} and V_r from both a uniform and a Gaussian distributions centered around average values of 3.5 and 14.5 m/s, respectively, with a range of uncertainty of [3, 4] and [12, 17] m/s, respectively, defining the domain of the associated distribution (see Figure 1). Then, the two parameters are sampled either from a uniform distribution ($F_{ci} = U[3,4]$ and $F_r = U[12,17]$), or from a Gaussian one ($F_{ci} = N(3.5, (1/6)^2)$ and $F_r = N(14.5, (5/6)^2)$.

The resulting average bootstrapped PIs for 1-hour ahead wind power prediction, obtained by applying to the wind speed PIs of the testing data set the bootstrapping scheme described in the previous section, are shown in Figure 4. From inspection, the robustness of the bootstrapping procedure with respect to the distribution hypothesis can be appreciated. The results are also compared with the ones obtained by fixing the stochastic parameters defining the power curve to their average values; in this case, the uncertainty is evidently underestimated.

In Figures 5 and 6, we finally show the bootstrapped distributions of the wind power PIs obtained by uniform and Gaussian sampling, respectively. The bootstrapped distributions are shown by the 5^{th} and 95^{th} percentiles (dotted and dashed lines, respectively). By looking at these plots, some considerations can be made: first, the bootstrapping technique allows us to efficiently decouple epistemic (PIs distribution) and aleatory (PIs width) uncertainty. Secondly, the epistemic uncertainty that generates a variability into the PIs bounds, described by the percentiles in Figures 5 and 6, is smaller than the aleatory uncertainty, quantified via

the PIs width: this can be appreciated in the fact that the 95th percentile of the lower bound bootstrapped distribution is never greater than the 5th percentile of the upper bound bootstrapped distribution; or, in other words, by the fact that the PIs width is in general bigger than the uncertainty associated to the PIs themselves.

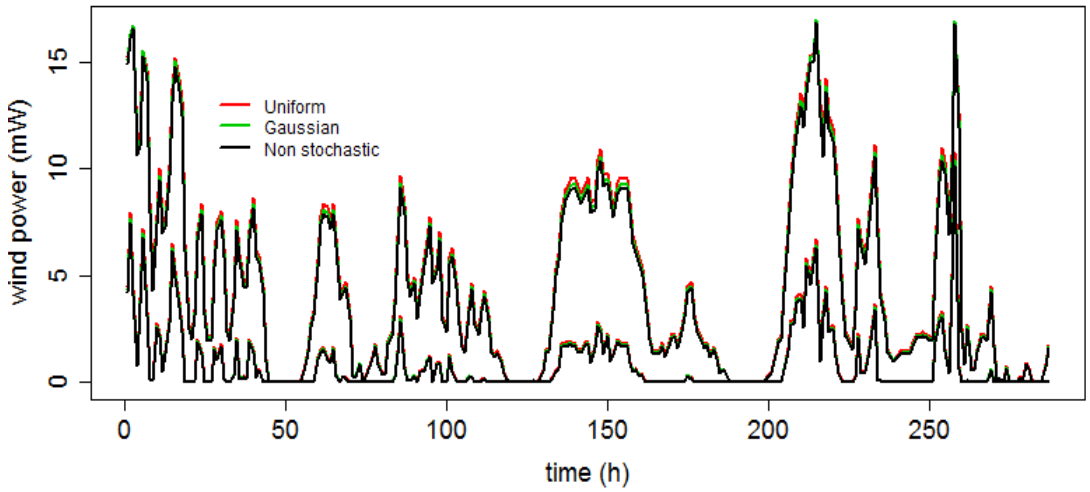


Figure 4. Average bootstrapped PIs for 1-hour ahead wind power prediction on the testing data set, obtained by sampling the power curve stochastic parameters from a uniform distribution (red lines), from a Gaussian distribution (green lines), and by fixing them to their average values (black lines).

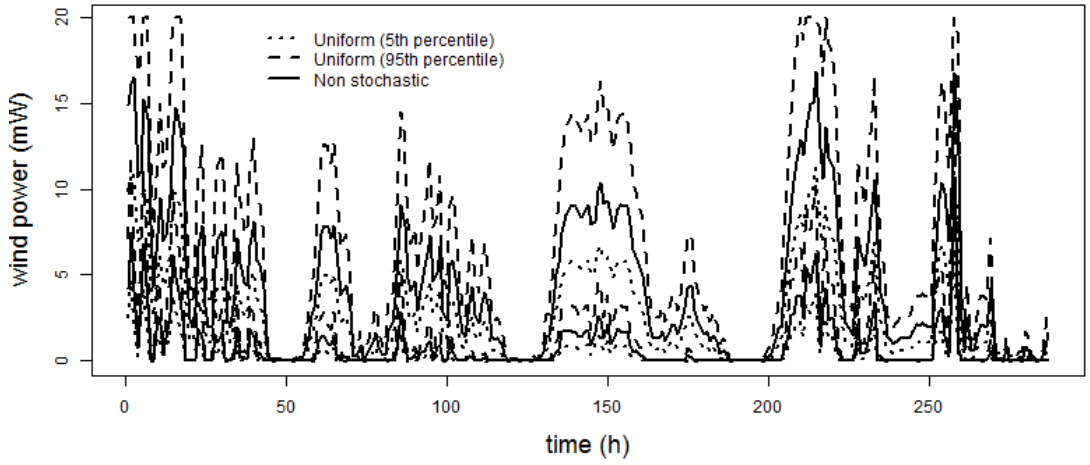


Figure 5. 5th (dotted lines) and 95th (dashed lines) percentiles of the bootstrapped distribution of 1-hour ahead wind power PIs obtained by sampling the power curve stochastic parameters from a uniform distribution, together with wind power PIs obtained by fixing the parameters to their average values (solid line).

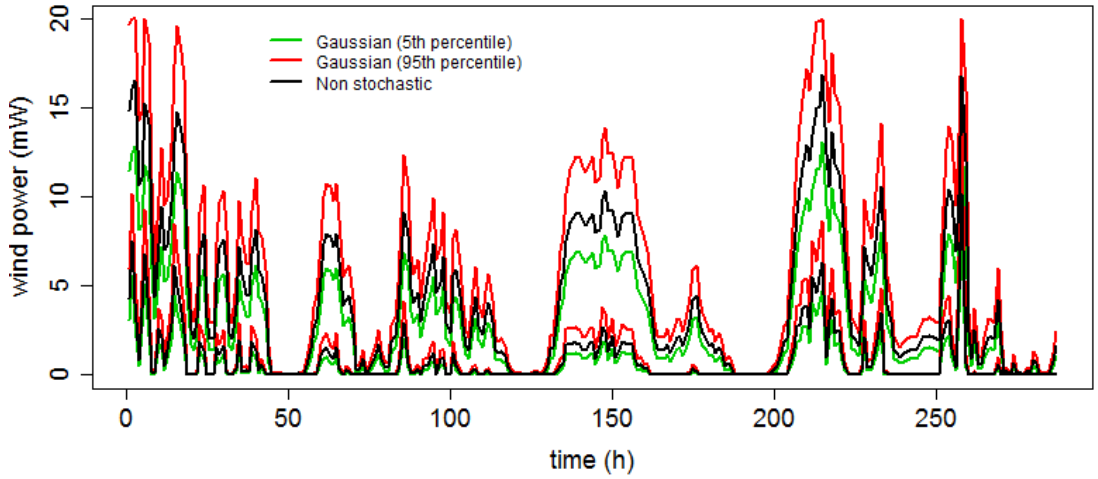


Figure 6. 5th (green lines) and 95th (red lines) percentiles of the bootstrapped distribution of 1-hour ahead wind power PIs obtained by sampling the power curve stochastic parameters from a Gaussian distribution, together with wind power PIs obtained by fixing the parameters to their average values (black lines).

4. CONCLUSIONS

In this work, we presented a novel approach to wind power PIs estimation, taking into account both aleatory and epistemic uncertainty. The proposed approach quantifies aleatory uncertainty by estimating wind speed PIs, and then transforms them into wind power PIs by using a power curve. In doing so, epistemic uncertainty arising from the imperfect knowledge of the power curve parameters is also taken into account through bootstrap sampling. The procedure manages to effectively decouple aleatory and epistemic uncertainty, and moreover shows a good robustness with respect to the parametric assumptions implicit in the bootstrap. The invariance of the coverage probability by passing from wind speed to wind power PIs has also been shown.

REFERENCES

Akdag, A. A. & Güler, Ö. 2011. Comparison of wind turbine power curve models. *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects* 33 (24): 2257-2263.

Albadi, M. & El-Saadany, E. 2012. Comparative study on impacts of power curve model on capacity factor estimation of pitch-regulated turbines. *TJER* 9: 36–45.

Canadian Weather Office. 2012. Website: http://www.weatheroffice.gc.ca/canada_e.html, (accessed Dec., 2012).

Deb, K., Pratap, A., Agarwal, S. & Meyarivan, T. 2002. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6 (2): 182–197.

Efron, B. 1981. Nonparametric estimates of standard error: the jackknife, the bootstrap and other methods. *Biometrika* 68 (3): 589–599.

Geisser, S. 1993. *Predictive Inference: An Introduction*. Chapman and Hall.

Helton, J.C., 1994. Treatment of uncertainty in performance assessments for complex systems. *Risk Analysis* 14 (4): 483–511.

Hornik, K., Stinchcombe, M. & White, H. 1989. Multilayer feedforward networks are universal approximators. *Neural Networks* 2: 359–366.

Justus, C.G., Hargraves, W.R. & Yalcin, A. 1976. Nationwide assessment of potential output from wind-powered generators. *Journal of Applied Meteorology* 15: 673–678.

Khosravi, A., Nahavandi, S., Creighton, D. & Atiya, A.F. 2011. Lower upper bound estimation method for construction of neural network-based prediction intervals. *IEEE Transactions on Neural Networks* 22 (3): 337–346.

Konak, A., Coit, D.W. & Smith, A.E. 2006. Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety* 91(9): 992–1007.

Korbicz, J., Koscielny, J.M., Kowalczuk, Z. & Cholewa, W. 2004. *Fault diagnosis: models, artificial intelligence, applications*. Berlin: Springer-Verlag.

Novoa, C. & Jin, T. 2011. Reliability centered planning for distributed generation considering wind power volatility. *Electric Power Systems Research* 81(8): 1654–1661.

Shao, J. & Tu, D. 1995. *The Jacknife and Bootstrap*. Springer Series in Statistics, New York.

Shrestha, D.L. & Solomatine, D.P. 2006. Machine learning approaches for estimation of prediction interval for the model output. *Neural Networks* 19: 225–235.

Zio, E. 2006. A study of the bootstrap method for estimating the accuracy of artificial neural networks in predicting nuclear transient processes. *IEEE Transactions on Nuclear Science* 53 (3): 1460–1478.

PAPER VII

Ensembles of Neural Networks for Estimating Short-term Prediction Intervals of Wind Speed

R. Ak, V. Vitelli, E. Zio, (2014) (*under preparation*).

Ensembles of Neural Networks for Estimating Short-term Prediction Intervals of Wind Speed

Ronay Ak[a], Valeria Vitelli[b], and Enrico Zio[a, c]

[a]Chair on Systems Science and the Energetic Challenge, European Foundation for New Energy-Electricité de France, CentraleSupélec, Châtenay-Malabry 92290 and Gif-Sur-Yvette 91192, France

[b]Department of Biostatistics, University of Oslo, Oslo, Norway

[c]Department of Energy, Politecnico di Milano, Milan 20133, Italy

ABSTRACT

In this paper, we address the problem of wind speed prediction for wind power production. Prediction intervals (PIs) are considered to account for the uncertainties in the predictions and two non-parametric methods are proposed to construct ensemble models made by Neural Networks (NNs) to estimate PIs. Short-term (1-h ahead) wind speed prediction on a real dataset of hourly wind speed measurements for the region of Regina in Saskatchewan, Canada, is considered as case study. Both methods proposed for NNs ensemble construction demonstrate consistent results and high prediction precision, compared both to the individual NNs of the ensembles and to conceptually similar estimation methods proposed in the literature.

Keywords: multi-perceptron neural networks, ensemble, multi-objective, wind speed prediction, prediction intervals.

1. INTRODUCTION

The efficient and reliable use of renewable energy sources continues to be a most important issue for world sustainable energy management. Significant amount of investments are being made to replace existing energy sources with new and renewable ones. This must be done while reliably and safely operating power systems, under the challenging conditions brought by intermittent renewable energy (e.g. wind, solar, etc.) fed into the energy grids.

For planning and operational purposes, then, accurate and robust prediction of the power generated by renewable energy sources becomes critical to guarantee the adequacy of the generation system, particularly for intraday energy trading: errors in prediction of the renewable energy sources can impact significantly on subsequent operations.

Among renewable energy sources, wind energy represents a popular, clean (contributes to less carbon-intensive energy production) and sustainable solution. Over the past decade, wind energy has received fast-growing attention throughout the world, and the utilization of wind power has increased dramatically [1]. Furthermore, the use of wind energy is expected to continue to increase: the World Wind Energy Association (WWEA) has predicted a possible wind capacity of more than 700000 MW by 2020 [2]. These projections enhance the importance of the reliable integration of large amount of wind energy to the power grid, without harming its reliability.

The planning and operation for the generation of energy from wind requires accurate mapping and prediction of wind speed, with the volatile and stochastic character of wind speed posing additional challenges for predictability. The prediction model must be capable of providing in output also a quantification of the uncertainty associated to the prediction, for informed decision-making. To this aim, in the present work we propose a novel framework to estimate prediction intervals (PIs). The framework proposed allows developing an ensemble of Neural Network (NN) models.

In general terms, it is well known that an ensemble of different predictors can generate predictions that are more accurate than those obtained by individual predictors [3]. Specifically, a NN ensemble is a learning paradigm where a certain number of NNs are combined to estimate the desired output for the target of interest (see Fig. 1) [3]. Typically, a

NN ensemble is constructed in two steps: *i*) training a number of individual NNs and *ii*) combining the predictions yielded from these NNs. The aim of assembling a number of NNs into an ensemble is to improve the generalization ability and estimation accuracy of the prediction model.

Considerable research has been done on ensembles and, also, specifically on ensembles of NNs. Traditional NN ensemble techniques have been built via several strategies, such as randomly trying different topologies (different number of hidden layers and neurons) in each individual NN, setting different initial weights or parameters, using different training datasets (e.g. bagging, cross-validation, etc.) or learning algorithms, etc. [4]-[8].

Bagging and boosting are the most prevailing approaches used to produce ensembles [3], [7]. Bagging is based on bootstrap sampling [5], since it produces replicate training sets by sampling with replacement from the training samples [6], [9], [10]. The method works by training the multiple (*m*) models on different data splits (generated by sampling with replacement from the original training dataset) and by averaging their outcomes to obtain the ultimate prediction results on the testing set [9]. In boosting ensembles, the patterns that the earlier classifiers in the series recognized incorrectly are over-represented in the composition of a particular training set, i.e. training samples that are incorrectly predicted by previous classifiers in the series are more often chosen than samples that were correctly predicted [6], [7], [11]. Thus, boosting aims at producing new classifiers that are more capable to predict samples for which the current ensemble performance is poor [7].

Regarding the combination of the estimated predictions (outputs) of each individual NN, different techniques can be adopted, like a simple arithmetic mean, a weighted mean, a median, a linear combination, local fusion (LF), dynamic integration, etc. [6], [12]. As an exemplification, Baraldi et al. [12] have explored the LF strategies for the aggregation of the outcomes of different ensemble models, whereas Khosravi et al. [4] have combined individual PI forecasts through mean and median calculations.

In our previous works [13], [14] a simple multi-layer perceptron neural network (MLP NN) model has been used to estimate PIs. We have introduced a multi-objective framework to estimate PIs, which are dominant-optimal in terms of both interval width (PIW) and coverage probability (CP). More precisely, a MLP NN is trained by a MOGA, namely the non-

dominated sorting genetic algorithm–II (NSGA-II) [15]: this approach integrates the estimation of the PIs in a learning procedure where the MLP NN is trained to concurrently minimize the PIs width and maximize their coverage probability. In this paper, we propose an enhanced version of the non-parametric multi-objective genetic algorithm (MOGA)-NN method, which has been originally proposed by Ak et al. in [13], [14], here extended to build an ensemble of MLP NNs as base learners and we apply it to the problem of short-term wind speed prediction. The case study considered is that of [14], where we have considered short-term wind speed prediction on four different wind speed datasets involving different wind speed profiles with seasonality.

We propose two NN ensemble methods, differing in the partitioning or not of the training dataset, and embedding the *k*-nearest neighbors (*k*-nn) approach in the aggregation phase for the identification of the neighborhoods of a test pattern [12], [16], [17]. The first strategy splits the training dataset into sub-sets with an equal number of samples and, then, each individual NN is trained on a different sub-training set; the second strategy, instead, uses the same training dataset (the entire dataset) for the training of each individual NN. The two methods differ also in the combination method of the individual NNs outputs.

The rest of the paper is organized as follows. The basics of MLP NNs modelling and the definition of PIs are given in Section 2. Section 3 provides the details of the methods proposed in the present work for the ensemble of NNs. Section 4 presents the data and parameters used in the experiments, and the results of the case study. Finally, Section 5 concludes the paper.

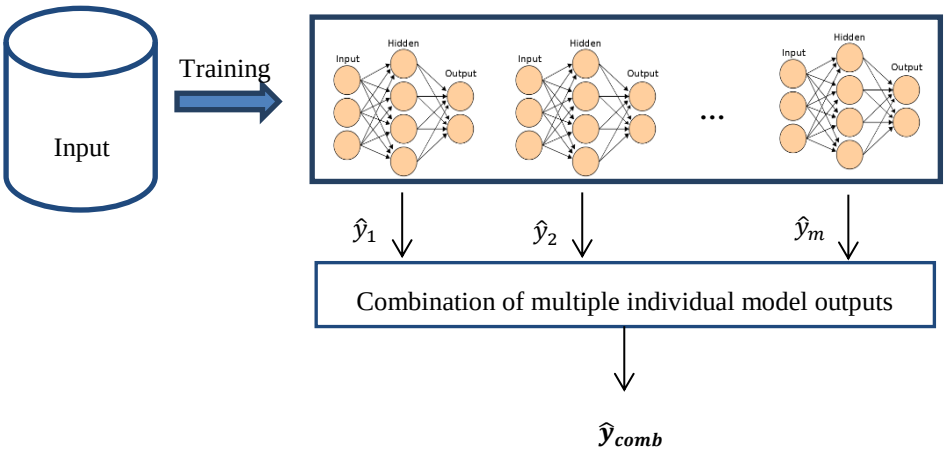


Figure 1. A basic scheme of NN ensemble.

2. MLP NNs and PIs

NNs are universal functional approximators often used as regression models capable of learning non-linear patterns from historical data [18]. MLP NNs are a common and popular type of feed-forward NNs [18], [19]. A MLP NN consists of processing units, the so called neurons, that are ordered into layers: one input layer, one or several hidden layers and one output layer. The nodes are connected by weights. Each layer receives input signals generated by the previous layer, it produces output signals through an activation function (e.g. a sigmoid transfer, or activation, function), and it distributes them to the subsequent layer through the neurons.

The prediction accuracy of a NN can depend on several factors, such as the network topology, the level of variability and uncertainty in the input data, the number of data samples for training, the learning algorithm, the set of initial parameters, etc. For the theoretical basics of the NN modelling, we refer the reader to [19]-[21].

Figure 2 shows the scheme of a three-layered MLP NN used in the present work to construct PIs. The first output neuron provides the upper bound and the second the lower bound. With these two output neurons, the NN generates a PI interval for each input pattern.

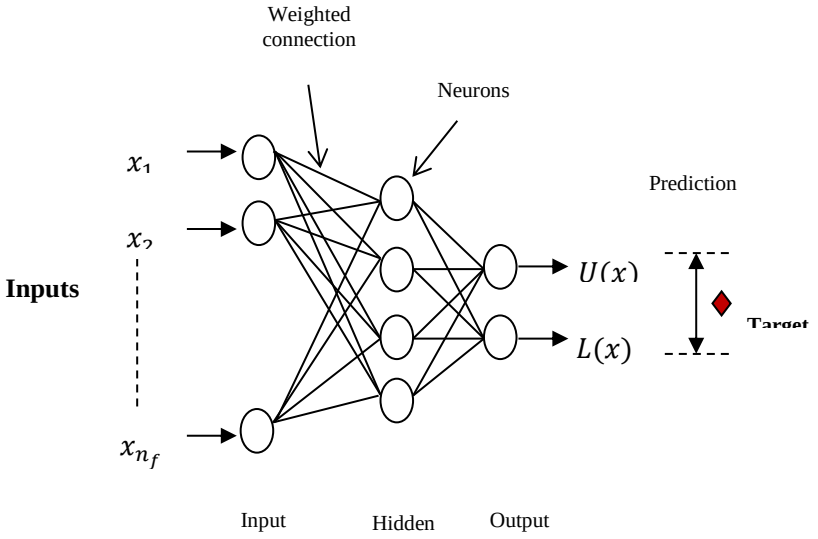


Figure 2. Sketch of a three-layered MLP NN architecture.

A PI is a statistical estimator composed by upper and lower bounds that include a future unknown value of the target $y(x)$ with a predetermined probability, called confidence level and in general indicated with $1-\alpha$ [22], [23]. The formal definition of a PI can be, thus, given in the following form:

$$Pr(L(x) < y(x) < U(x)) = 1 - \alpha \quad (1)$$

where $L(x)$ and U(x) are the estimators of the lower and upper bounds of the PI corresponding to input x, and the confidence level $(1-\alpha)$ is the probability that the true unknown value of $y(x)$ lies within the interval $[L(x), U(x)]$.

The prediction interval coverage probability (PICP) represents the probability that the set of estimated PIs will contain the true output values and it is estimated as the proportion of true output values lying within the estimated PIs. Prediction interval width (PIW) simply measures the extension of the interval as the difference between the estimated lower and upper bound values. These are in general conflicting measures (wider intervals give larger coverage), and in practice it is important to have narrow PIs with high coverage probability. The mathematical definition of the PICP and PIW measures here employed are, respectively [23]:

$$PICP = \frac{1}{n_p}\sum_{i=1}^{n_p} c_i \quad (2)$$

where n_p is the number of samples in the training or testing sets, and $c_i = 1$, if $y_i \in [L(x_i), U(x_i)]$ and otherwise $c_i = 0$,

$$NMPIW = \frac{1}{n_p}\sum_{i=1}^{n_p} \frac{(U(x_i)-L(x_i))}{y_{max}-y_{min}} \quad (3)$$

where $NMPIW$ is the Normalized Mean PIW, and y_{min} and y_{max} represent the true minimum and maximum values of the target (i.e., the bounds of the range in which the true values fall) in the training set, respectively. Normalization of the PI width by the range of targets makes it possible to objectively compare the PIs, regardless of the techniques used for their estimation or the magnitudes of the true targets.

Khosravi et al. [23] proposed a "Lower and Upper Bound Estimation Method (LUBE)", in which they obtain NN-based PIs by considering both CP and PIW in the PI construction phase. They perform the training by using a single-objective PI-based cost function called coverage width criterion (CWC), in which they combine two separate objectives: PICP and NMPIW. The CWC objective function is given by [23]:

$$CWC = NMPIW(1 + \gamma(PICP)\, e^{-\eta(PICP-\mu)}) \quad (4)$$

where η and μ are constants. The role of η is to magnify any small difference between μ and PICP. The value of μ gives the nominal confidence level, which is set to 90% in our experiments. Then, η and μ are two parameters determining how much penalty is paid by the PIs with low coverage probability. The function $\gamma(PICP)$ is equal to 1 during training.

In this work, in order to obtain PIs optimal in terms of both interval size and coverage, we proceed as in [13] and [14] using NSGA-II, which is one of the most powerful multi-objective evolutionary algorithms (MOEAs): NSGA-II finds the values of the parameters of the NN which minimize the two objective functions PICP (2) and NMPIW (3) simultaneously, in Pareto optimality sense (for ease of implementation, the maximization of PICP is converted to minimization by subtracting from unity, i.e. the objective of the minimization is 1-PICP). The practical implementation of NSGA-II on our specific problem can be found in [13], [14].

Then, in our framework for NN ensemble construction, we use the CWC of [23] as a measure to rank the individual NNs on the validation set. We stress that we do not use CWC (4) during the training of the individual networks, but, rather, the multi-objective formulation of the PI estimation problem in terms of (2) and (3).

3. PI ESTIMATION VIA AN ENSEMBLE OF MOGA-BASED NNS

In order to construct PIs using an ensemble of NNs we propose two methods. The first one is classical and consists of partitioning the training dataset into several sub-sets, and then performing the training of each individual NN with different training sets. The second method uses the same training data set for each individual NN, which differs for the initial weights randomization. With method 2, we obtain an overall Pareto front, hereafter called combined

Pareto front, which is obtained by applying non-dominated sorting to the Pareto fronts obtained by the training of each network.

The implementing procedures for the methods are synthesized below.

Method 1:

Step 1: Divide the input dataset into training (D_{tr}), validation (D_{vald}) and testing (D_{test}) sets.

Step 2: Split the training set D_{tr} into n_{total} sub-sets with equal number of samples.

Step 3: Set the number of hidden neurons and other initial parameters.

Step 4: Train n_{total} MOGA-based NNs by assigning a training set to each network.

Step 5: After training the n_{total} NNs, select one solution from each Pareto front. This solution is selected by the rule described below:

- First, select the solutions on the Pareto front giving PICP greater than 90%.
- Apply the "weighted average" approach [24] to the selected solutions and select one final solution among them.

Step 6: After the selection of a single solution from each optimal Pareto front, perform validation using the parameters of each selected NN.

Step 7: Calculate the value of CWC on the validation set, CWC_{vald}, for each of the n_{total} NNs, and rank them on the basis of their CWC_{vald} values.

Step 8: Select the n_{best} NNs giving the smallest CWC_{vald} and discard the others.

Step 9: For each testing sample *i* in the testing dataset, find the *k*-nn in the training dataset of each selected n_{best} NN.

Method 2:

Step 1: Divide the input data set into training (D_{tr}), validation (D_{vald}) and testing (D_{test}) sets.

Step 2: Set the number of hidden neurons and other initial parameters.

Step 3: Train *m* MOGA-based NNs using D_{tr}, where each network uses the full training set and differs only in its random initial weight settings. Thus, obtain *m* optimal Pareto fronts.

Step 4: Obtain an overall best Pareto front from the *m* optimal Pareto fronts.

Step 5: Perform validation using D_{vald} with the solutions on the overall best Pareto front.

Step 6: Obtain the non-dominated solutions on the validation front obtained in Step 5.

Step 7: Calculate the CWC_{vald} value of the non-dominated solutions and sort (rank) them with respect to their CWC_{vald} values.

Step 8: Select the n_{best} NNs giving the smallest CWC_{vald} and discard the others.
Step 9: For each testing sample *i* in the testing dataset, determine the *k*-nn in the training dataset of each selected n_{best} NNs.

Combination of the outputs

Step 10: Combine the lower and upper bounds of the selected *k*-nn by mean, median and weighted mean calculations, respectively.
Step 11: Perform testing with the selected n_{best} NNs on the testing set. Then, compare the estimated PI results of the selected n_{best} NNs with the combined PI results.

Each individual NN in the ensemble is trained independently to minimize the prediction error with respect to the target. We have used the same architecture (i.e. number of hidden neurons) for each individual NN. The number of hidden neurons has been determined by a trial-and error method.

In both methods, the validation set has been used to screen the NNs with respect to their performance in PIs estimation on the validation set (see Step 7). In other words, in method 1, we have ranked the n_{total} NNs on the basis of their validation performances, i.e. considering their CWC_{vald} values, and then we have selected the n_{best} NNs to be used to estimate the upper and lower bounds of the combined PIs. In method 2, we have filtered the solutions two times consecutively: after we have obtained an overall Pareto front of *m* optimal individual Pareto fronts generated by training *m* NN models, which are diversified through their random initialization in the training stage, we have performed validation for each solution using the validation set. More precisely, to construct such an overall Pareto front, the first (best) front of each of *m* runs is collected and the resulting set of solutions is subjected to the fast non-dominated sorting algorithm [15] with respect to the two objective functions values. Then, the ranked non-dominated fronts $F_1, F_2, \dots, F_k$ are identified, where F_1 is the best front, F_2 is the second best front and F_k is the worst of the *k* fronts. Solutions in the first (best) front F_1 are then retained as overall best front solutions. This procedure gives us the overall best non-dominated Pareto front for the training set. After that, we have performed validation for each optimal solution on the overall front, and then a non-dominated sorting has been applied to these validation solutions. Thus, we have *a posteriori* obtained a set of non-dominated validation solutions forming an optimal Pareto front. This can be viewed as the first ranking

process on the validation. Figure 3 shows the validation solutions before non-dominated sorting and after it (the dominated and non-dominated solutions seem very similar in the plot but they are not identical, especially where 1-PICP is closer to 0, which is the region of interest for high coverage probability solutions). For the sake of clarity of visualization, a zoom on the solutions have CP of 95 % or greater has also been plotted. Note that in Figure 3 the X axis indicates "1-PICP".

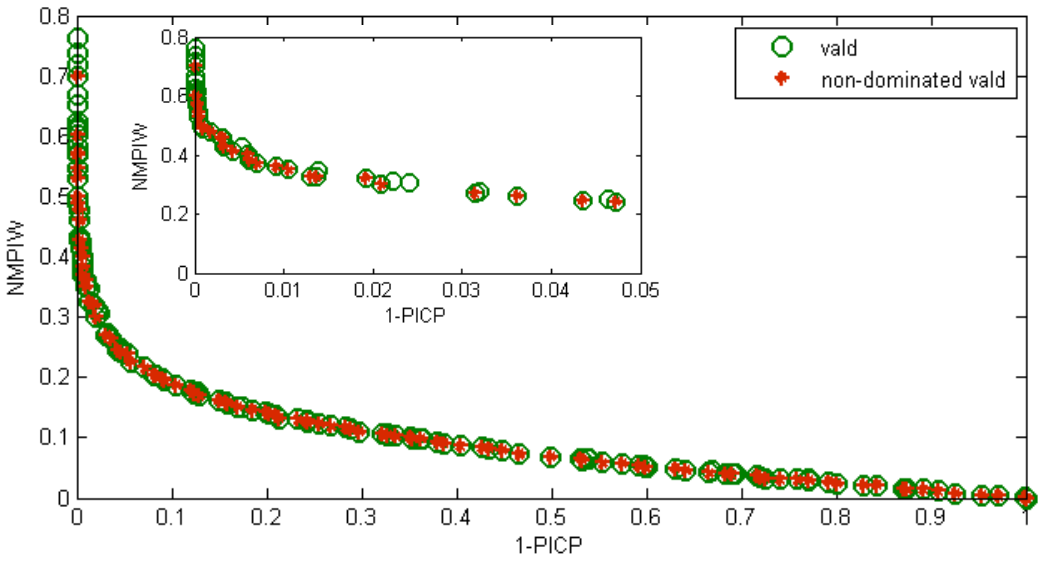


Figure 3. Validation fronts before (marked as circle) and after (marked as star) non-dominated sorting.

After this step, the non-dominated validation solutions are ranked on the basis of their CWC_{vald} results (as in method 1), and then the first n_{best} NNs giving the smallest CWC_{vald} values are kept, while the remaining ones are discarded.

The aggregation phase is identical for both methods: for each pattern i in the testing set, we have found k-nn patterns in the training sets of each selected n_{best} NNs. The k-nearest neighbors of a specific pattern i in the testing set are determined by the Euclidean distance between the input values of the specific pattern and the patterns in the corresponding training set, i.e. given a set X of n samples (training set) and a distance function, a k-nn search lets you find the k closest samples in X for each query sample (pattern) in Y (testing set).

Finally, for each testing pattern i, the upper and lower bounds of PIs estimated from the selected k neighbors are aggregated to generate the combined PIs on the testing set. Note that, herein, the weights used in the "weighted mean" aggregation method correspond to the

reciprocal of the CWC values of the n_{best} selected solutions, i.e. individual NNs, calculated on their corresponding training sets. Hence, a solution which has a smaller CWC value, has a higher weight, and gives thus higher contribution to the combined PIs.

It is worth pointing out that the "weighted average" approach mentioned in *Step* 5 of method 1 is one of the quantitative criteria performed to select a single best compromise Pareto solution. It is a simple strategy used to choose a compromise solution with respect to the particular user-specified weighted average of the objective functions. In our specific two-objective optimization problem, the weight vector $w = (w_1, w_2)$ is set to (0.6, 0.4) by assuming that the first objective (maximizing the CP) is approximately 1.5 times more important than the second one (minimizing the PIW). Note that the weighted average method chooses the Pareto-optimal solution according to the user-specified weights assigned, i.e. another solution can be found with a different weight-vector.

4. CASE STUDY RESULTS

In this Section, the results of the application of the proposed ensemble methods to short-term wind speed forecasting are detailed. The considered wind speed data have been measured in Regina, Saskatchewan, a region of central Canada. The hourly wind speeds measured from 1st of February 2003 to 28th of July 2012 in Regina, Saskatchewan have been downloaded from the website (80000 samples in total) [25]. The entire dataset has been split into three parts, training, validation and testing, including 60000, 10000 and 10000 samples, respectively. For method 1, the training set has been split into n_{total} = 100 sub-sets; thus, each training sub-set has 600 samples. Table 3 reports the parameters used in the experiments. The input dataset has been normalized to have values between 0.1 and 0.9.

In order to select the relevant lagged values of the wind speed ($W_{t-1}, \dots, W_{t-m}$) to be included as input variables in the prediction model for estimating (W_t), the empirical Autocorrelation Function (ACF) and the Partial Autocorrelation Function (PACF) have been inspected [26]. In our case, the ACF and PACF indicate that for predicting W_t in output, the wind speed values at the three previous time steps W_{t-1}, W_{t-2} and W_{t-3} are the most appropriate input variables.

Table 3. Parameters used in the experiments.

Parameter	**Numerical value**
MaxGen	300
Nc	50
P_{m_int}	0.06
M	0.9
H	50
D_{tr}	60000
D_{vald}	10000
D_{test}	10000
n_{total}	100
n_{test}	10

We have calculated the combined results with respect to different *k* values, i.e. for *k* = 1, 3, 5, 7. We have observed that in our specific problem, different values of *k* lead to different results. It is worth nothing that the results obtained with *k* = 5 are quite close to those obtained with *k* = 3. As we have obtained more accurate results with *k* = 3, in Tables 1-2 we report only these ones. Table 2 reports the combined PICP and NMPIW results on the testing set, while in Table 3 the results of the 10 selected best networks are given with respect to methodologies 1 and 2. One can see that in both methods we have obtained quite high CP with very small interval sizes. It is evident that the coverage probabilities with all aggregation methods, i.e. mean, median and weighted mean, are higher than the ones obtained with each of the 10 best individual NNs. For what concerns the interval size, although some of the selected 10 best NNs give slightly smaller interval sizes, their accuracy, i.e. their CPs are lower than the combined ones. For exemplification, the combined PI results (ensemble of NNs) of method 2 via median calculation outperforms all individual NN models (10 best) in terms of the PI accuracy (measured by PICP), while its interval size is slightly larger than 4 out of 10 individual NNs. We remark that our strategy for selecting the *k*-nearest neighbors from the training set for each of the patterns in the testing set has played an important role to obtain high quality PIs, characterized by both high CP and small PIW.

It is worth pointing out that Khosravi et al. [4] provide combined PIs for wind power data by using a single-objective optimization framework, i.e. they provide a framework for

synthesizing PIs generated using an ensemble of NN models trained according to the original LUBE method . If we take as reference their combined PI results estimated for wind power generation [4], we can clearly say that we have obtained higher quality PIs in terms of both CP and PIW on our wind speed case study. For what concerns the comparison of the combined results obtained by methods 1 and 2, although both give high CPs with small PIWs, the combined results are non-dominated: the results obtained by median calculation give 99.62 % PICP and 26.75 % NMPIW for method 1, and 99.18 % PICP and 23.91 % NMPIW for method 2. These two results are non-dominated. Thus, there is an intrinsic trade-off in the selection of any solution, since in making a choice either coverage or interval width has to be favored; hence the decision maker (DM) has to select one final solution according to his/her subjective preferences. On the other hand, the aggregation methods (mean, median and weighted mean) used to generate combined PIs for the testing set do not show significant differences. Both PICP and NMPIW values are quite close for each aggregation method.

Table 2. Combined PI results of methods 1 and 2 on the testing set according to the three aggregation schemes.

***k* = 3**	**Method 1**			**Method 2**		
	Mean	**Median**	**Weighted mean**	**Mean**	**Median**	**Weighted mean**
PICP (%)	99.60	99.62	99.60	99.54	99.18	99.54
NMPIW (%)	26.57	26.75	26.73	24.14	23.91	24.18

Moreover, it can be observed that the validation and corresponding testing results of the 10 selected individual NNs show high consistency in terms of coverage probability and interval size (see Table 3). All solutions result in a CP bigger than 90 % with low interval sizes. This confirms the generalization power of the original MOGA NN approach, which is capable of generating high quality PIs. Figure 4 shows the combined PIs for the testing set estimated by method 2 via median aggregation. For the sake of clarity of visualization, a zoom on the first 300 hours has been plotted.

Table 3. PICP and NMPIW results of the 10 best selected NNs on the validation and testing sets according to methods 1 and 2.

Method 1				**Method 2**			
Validation		**Testing**		**Validation**		**Testing**	
PICP (%)	**NMPIW (%)**	**PICP (%)**	**NMPIW (%)**	**PICP (%)**	**NMPIW (%)**	**PICP (%)**	**NMPIW (%)**
94.57	0.228	93.57	22.85	94.32	22.55	93.61	22.58
94.84	0.232	94.12	23.28	95.27	24.14	94.76	24.18
95.74	0.248	95.06	24.88	95.65	24.54	95.25	24.60
95 .43	0.248	95.09	24.87	94.60	23.85	94.09	23.85
95.26	0.248	94.57	24.84	92.84	21.62	92.10	21.63
96.52	25.78	95.85	25.82	96.38	26.30	96.17	26.35
95.18	24.97	94.50	25.04	96.86	26.97	96.21	27.01
93.09	22.12	92.00	22.14	91.84	20.36	91.10	20.41
95.42	25.33	94.96	25.34	91.75	20.22	90.98	20.25
96.37	25.99	95.77	26.05	97.91	29.97	97.54	30.02

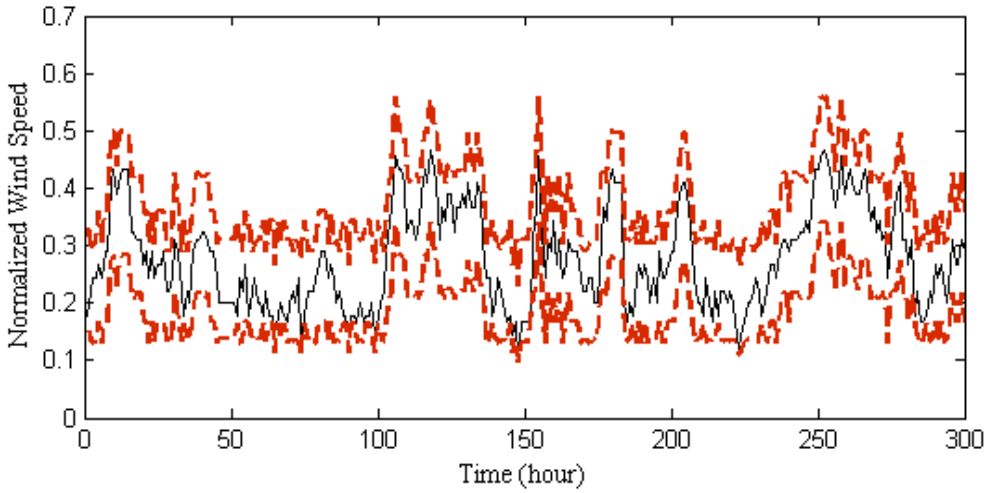


Figure 4. Estimated PIs for 1-h ahead wind speed prediction on the testing set (dashed lines), and wind speed data included in the testing set (solid line).

5. CONCLUSIONS

The goal of this work is to present a framework to construct accurate PIs using ensemble of individual NN models, for short-term wind speed prediction. To this aim, we have introduced two non-parametric methods. On the real data considered as case study, both methods have obtained superior results compared to those yielded from the selected individual networks

selected in the respective ensembles. Compared to literature methods conceptually and methodologically similar to the present ones, the results obtained show a significant improvement in terms of the quality of the predicted PIs. We can, then, conclude that both ensemble modelling frameworks proposed yield a reliable estimation of the PIs, characterized by a high coverage probability and a small interval size. The reported results demonstrate the practically efficient methods proposed for quantification of uncertainties associated with wind speed prediction.

REFERENCES

[1] The European Wind Energy Association, "United in tough times - EWEA annual report 2012".
[2] World Wind Energy Association, "Key Statistics of World Wind Energy Report 2013", Apr. 2014.
[3] Z.-H. Zhou, J. Wu, and W. Tang, "Ensembling neural networks: Many could be better than all", *Artificial Intelligence*, vol. 137, no. 1‑2, pp. 239‑263, May 2002.
[4] A. Khosravi and S. Nahavandi, "Combined Nonparametric Prediction Intervals for Wind Power Generation », *IEEE Transactions on Sustainable Energy*, vol. 4, no. 4, pp. 849‑856, 2013.
[5] D. D. Boos, "Introduction to the Bootstrap World", *Statistical Science*, vol. 18, no. 2, pp. 168‑174, May 2003.
[6] Y. Zhao, J. Gao, and X. Yang, "A survey of neural network ensembles", in *International Conference on Neural Networks and Brain, 2005. ICNN B '05*, 2005, vol. 1, pp. 438‑442.
[7] R. Maclin and D. Opitz, "Popular Ensemble Methods: An Empirical Study", *Journal of Artificial Intelligence Research*, vol. 11, pp. 169‑198, 1999.
[8] D. K. Barrow, S. F. Crone, and N. Kourentzes, "An evaluation of neural network ensembles and model selection for time series prediction", in *The 2010 International Joint Conference on Neural Networks (IJCNN)*, 2010, pp. 1‑8.
[9] P. Büchlmann and B. Yu, "Analyzing Bagging", *The Annals of Statistics*, vol. 30, no. 4, pp. 927‑961, Aug. 2002.
[10] A. Khosravi, S. Nahavandi, D. Creighton, and D. Srinivasan, "Optimizing the quality of bootstrap-based prediction intervals", in *The 2011 International Joint Conference on Neural Networks (IJCNN)*, 2011, pp. 3072‑3078.
[11] R. E. Schapire, "The Boosting Approach to Machine Learning: An Overview", in *Nonlinear Estimation and Classification*, D. D. Denison, M. H. Hansen, C. C. Holmes, B. Mallick, and B. Yu, Ed. Springer New York, 2003, pp. 149‑171.
[12] P. Baraldi, A. Cammi, F. Mangili, and E. E. Zio, "Local Fusion of an Ensemble of Models for the Reconstruction of Faulty Signals", *IEEE Transactions on Nuclear Science*, vol. 57, no. 2, pp. 793‑806, Apr. 2010.
[13] R. Ak, Y. Li, V. Vitelli, E. Zio, E. López Droguett, and C. Magno Couto Jacinto, "NSGA-II-trained neural network approach to the estimation of prediction intervals of scale deposition rate in oil & gas equipment", *Expert Systems with Applications*, vol. 40, no. 4, pp. 1205‑1212, 2013.
[14] R. Ak, Y-F. Li, V. Vitelli, and E. Zio, "Multi-objective Genetic Algorithm Optimization of a Neural Network for Estimating Wind Speed Prediction Intervals", *Applied Soft Computing*, 2014.
[15] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II", *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182‑197, 2002.
[16] S. A. Dudani, "The Distance-Weighted k-Nearest-Neighbor Rule", *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-6, no. 4, pp. 325‑327, Apr. 1976.
[17] P. P. Bonissone, F. Xue, and R. Subbu, "Fast meta-models for local fusion of multiple predictive models", *Applied Soft Computing*, vol. 11, no. 2, pp. 1529‑1539, March 2011.
[18] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators", *Neural Networks*, vol. 2, no. 5, pp. 359‑366, 1989.
[19] G. Zhang, B. Eddy Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks: The state of the art", *International Journal of Forecasting*, vol. 14, no. 1, pp 35‑62, March 1998.

[20] R. Rojas, *Neutral Networks: A Systematic Introduction*. Germany: Springer, 1996.
[21] Bishop, C. M., *Neural networks for pattern recognition*. Oxford university press, 1995.
[22] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya, "Comprehensive Review of Neural Network-Based Prediction Intervals and New Advances", *IEEE Transactions on Neural Networks*, vol. 22, no. 9, pp. 1341‑1356, Sept. 2011.
[23] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya, "Lower Upper Bound Estimation Method for Construction of Neural Network-Based Prediction Intervals", *IEEE Transactions on Neural Networks*, vol. 22, no. 3, pp. 337‑346, 2011.
[24] E. Zio, P. Baraldi, and N. Pedroni, "Optimal power system generation scheduling by multi-objective genetic algorithms with preferences", *Reliability Engineering & System Safety*, vol. 94, no. 2, pp. 432‑444, Feb.. 2009.
[25] "Canadian Weather Office", *Canadian Weather Office*, 2012. [Online]. Available: http://www.weatheroffice.gc.ca/canada_e.html. [Accessed: 08-avr-2013].
[26] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis: Forecasting and Control*, 4e éd. Wiley, 2008.