Effective Utilization of Data in Inductive Conformal Prediction using Ensembles of Neural Networks

Tuve Löfström, Ulf Johansson and Henrik Boström

Abstract— **Conformal prediction is a new framework producing region predictions with a guaranteed error rate. Inductive conformal prediction (ICP) was designed to significantly reduce the computational cost associated with the original transductive online approach. The drawback of inductive conformal prediction is that it is not possible to use all data for training, since it sets aside some data as a separate calibration set. Recently, cross-conformal prediction (CCP) and bootstrap conformal prediction (BCP) were proposed to overcome that drawback of inductive conformal prediction. Unfortunately, CCP and BCP both need to build several models for the calibration, making them less attractive. In this study, focusing on bagged neural network ensembles as conformal predictors, ICP, CCP and BCP are compared to the very straightforward and cost-effective method of using the out-of-bag estimates for the necessary calibration. Experiments on 34 publicly available data sets conclusively show that the use of out-of-bag estimates produced the most efficient conformal predictors, making it the obvious preferred choice for ensembles in the conformal prediction framework.**

I. Introduction

THE standard classification task is to predict one class label from a set of predefined classes, i.e., a *point prediction*, for each example. One important question about point predictions is how confident we may be that they are correct. Obtaining accurate confidence values for the probability that the prediction is actually correct is important in many real world applications. Examples include risk sensitive applications or situations where the prediction is the basis of potentially costly decisions. In such examples, it is important to be able to deal with high confidence cases using different strategies than the low confidence cases.

Different methods for producing as accurate confidence values as possible have been evaluated in many previous studies. One such approach is the PAC (Probably Approximately Correct) learning [1] that provides bounds on the predictive error. These bounds have, in many practical applications, been noted to be overly loose. Another objection against PAC is that the bounds apply to the overall error rate rather than for predictions on specific cases [2]. Bayesian methods can produce accurate confidence values, but only if the correct priors are known. If the priors are incorrect, the confidence values have no theoretical base [3].

Tuve Löfström and Ulf Johansson are with the School of Business and IT at the University of Borås and Henrik Boström is with the Dept. of Computer and Systems Sciences at Stockholm University, Sweden. (email: {tuve.lofstrom, ulf.johansson}@hb.se, henrik.bostrom@dsv.su.se).

This work was supported by the Swedish Foundation for Strategic Research through the project High-Performance Data Mining for Drug Effect Detection (IIS11-0053) and the Knowledge Foundation through the project Big Data Analytics by Online Ensemble Learning (20120192).

The conformal prediction (CP) framework [4] (which is described in section II) presents a way to output *region predictions* (a set of predicted class labels) with a guaranteed error rate, i.e., the probability of excluding the correct class label is less than a specified threshold (here called the significance level). The property of a guaranteed error rate is called *validity*. The exchangeability assumption[1] on which the framework relies is slightly weaker than the assumption normally assumed in machine learning, i.e., that examples should be independent and identically distributed.

Underlying a conformal predictor is a machine learning algorithm and a conformity measure used to estimate the probability p^c that an example with a selected class label c conforms to the data set, i.e., how similar it is to previously observed examples. For a given significance level $\epsilon \in (0, 1)$, the corresponding region prediction Γ^ϵ for an example is $\{c|c \in Y \wedge p^c > \epsilon\}$, where Y is the set of possible class labels.

The region prediction may include several class labels, when the model does not have sufficient information to make a single prediction but has to safeguard to make sure that the validity holds. It may consist of a single class, which obviously is the optimal case, or it may even be empty, when there is strong evidence against all classes, indicating that the example is atypical. Since the optimal case is when CP produces a single class prediction, one aim when constructing conformal predictors is to find ways to increase the proportion of region predictions containing as few class labels as possible, without obtaining empty prediction regions. In other words, the average size of the region prediction should be as close to one as possible, which is usually referred to as the *efficiency* of a conformal predictor. Efficiency is the most important feature to compare when evaluating different ways of creating conformal predictors, since the validity is guaranteed.

CP was originally defined as an online transductive framework [4] primarily employing lazy learners, like the k-nearest neighbor algorithm. In the online transductive mode, all available data is used to calculate the conformity score for each example which makes it necessary to train one model for every calibration example, as well as for every test example. Many real world applications are, however, offline in nature. Consider for example direct marketing, where the task is to predict which potential respondents that might answer favorably to an offer. Since all the respon-

[1]For the exchangeability assumption to hold, examples must be drawn from the same distribution and the ordering should not matter.

dents will usually get the offer simultaneously they must all be predicted before it becomes possible to measure the actual outcome. Fortunately, CP may also be used within an inductive framework, where one model is built from a training set and then applied to a test set. Inductive CP is hence computationally more efficient than the corresponding transductive conformal predictors. However, the data set has to be divided into a training set and a calibration set. Several different methods for utilizing the available data to both build the model and measure the conformity of an example have been proposed. Vovk presents three straightforward methods in [5]:

- *Inductive conformal prediction* (ICP), which divides the available data using the ordinary holdout method, using one part as the training set and one as the calibration set (see section II-A).
- *Cross-conformal prediction* (CCP), which employs cross-validation to divide the available data so that all data is eventually used as calibration set, one part for every fold. p-values are calculated from each calibration set and essentially averaged (see section II-B).
- *Bootstrap conformal prediction* (BCP), which is similar to CCP, but instead of using cross-validation, generates a bootstrap replicate [6] to train one model for each repetition. The conformity scores for each model are calculated using a calibration set consisting of all the examples not included in the bootstrap. The conformity scores from all repetitions are finally combined and used as the p-values for the BCP (see section II-C).

Both CCP and BCP utilize more examples for generating the predictive models than what ICP does, but at the cost of building multiple models.

When the underlying model is created using a bagging ensemble, there is, however, a fourth option. Instead of somehow dividing the available data as proposed by Vovk, all data can be used to train the ensemble and the out-of-bag (OOB) estimates on the training set can be used as a calibration set. This method will henceforth be referred to as the OOB method, since it utilizes the OOB examples as the calibration set. This has previously been suggested when employing random forests [8] in the conformal prediction framework [7]. Random forest is a special kind of bagging ensemble built with decision trees. In this study, we will investigate whether this allows us to utilize data more effectively than when using the above strategies.

The difference between BCP and OOB is that OOB uses all data as the training set when building a single bagging ensemble whereas BCP uses one bootstrap replicate for each model it builds. The models used by BCP may be bagging ensembles (which in turn use bootstrap replicates when building its members) or some other machine learning algorithm. Each example is used as a calibration example, but instead of making predictions with all models in the ensemble, including the ones that were built using the example, predictions are only made with those models for which the example is out-of-bag. Since the probability for an example being out-of-bag is close to 37%, it means that a little more than a third of the models in the ensemble will take part in the prediction for each calibration example. Note that in contrast to CCP and BCP, OOB will only generate one ensemble model, and hence there is no computational overhead due to generating multiple models.

Until now, most studies on CP have either focused almost exclusively on mathematical aspects of the framework or used a very limited number of data sets, making them more of the proof-of-concept type. In this study, different methods for utilizing the available data in inductive conformal prediction when generating ensembles of neural networks is evaluated on a large number of data sets to allow for statistical inference, thus making it possible to establish best practices. More specifically, the methods proposed by Vovk in [5] will be compared to the OOB method to evaluate how the data can be utilized in an effective way when performing inductive conformal prediction.

In the next section, conformal prediction and bagging ensembles will be described in more detail. The third section presents related work and the fourth section describes the experimental method. Results are presented and analyzed in section five, which is followed by conclusions given in section six. Finally, the findings are discussed and directions for future work are outlined in section seven.

II. Conformal Prediction

In this section, conformal prediction will be introduced and inductive conformal prediction will be explained in more detail. Details on conformal prediction can be found in [4], [9], [5]. This section describes conformal prediction for classification. It has also been defined for regression [4], which however is outside the scope of this paper. The formal presentation of the framework will follow the notation used in [5].

Let us consider a set of training examples $z_i = (x_i, y_i), i = 1, ..., l$, where l is the number of available examples, x_i is the object which consists of a vector of attributes (the independent variables) and $y_i \in Y$ is the class label (the dependent variable). Let x_{l+1} be a new test object. The idea of conformal prediction is to try all possible class labels $c \in Y$ for the test object to see how well that label conforms to the set of training examples. For an object and class label, $z = (x_{l+1}, c)$, the assumption of exchangeability of the sequence of $z_1, ..., z_l, z$ is tested. Since all the other examples $z_i, i \in \{1, ..., l\}$ are from the data set, this is equivalent to determining the likelihood that the label c is the true class label for the object x_{l+1}, or in other words, that z conforms to the data set.

The *conformity measure* is a function A that produces a *conformity score* $A(\{z_1, ..., z_l\}, z)$ measuring how typical the example z is in relation to the previous examples in $\{z_1, ..., z_l\}$. The conformity measure is often defined as

$$A(\{z_1, ..., z_l\}, (x, y)) = S(y, M(x)) \quad (1)$$

where M is a model, trained using $\{z_1, ..., z_l\}$, predicting a class label $c \in Y$ for the object x. S measures the

similarity between the true class label y and the predicted class label $M(x) = c$. The model M is built using an underlying algorithm, which can be any kind of machine learning algorithm, such as neural networks, decision trees, k-nearest neighbor, ensembles etc.

Each class label for an object is assigned a p-value indicating how well that class label for that object conforms to the set of examples. The p-values are defined by equation 2.

$$p^c = \frac{|i \in \{1, ..., l\} : \alpha_i \leq \alpha^c| + 1}{l + 1} \tag{2}$$

where $\alpha_i = A(\{z_1, ..., z_{i-1}, z_{i+1}, ..., z_l, z\}, z_i)$ and $\alpha^c = A(\{z_1, ..., z_l\}, z)$ are the conformity scores for the training and test examples.

Two standard ways of using conformal prediction are:

- Output the *point prediction* arg $\max_{c \in Y} p^c$ together with a measure on the confidence of the prediction. For binary classification tasks, when $Y = \{0, 1\}$, the *confidence* and *credibility* can be defined as $confidence = 1 - \min(p^0, p^1)$ and $credibility = \max(p^0, p^1)$ [5].
- For a given level of significance $\epsilon \in (0, 1)$, output the prediction region $\Gamma^\epsilon = \{c : c \in Y \wedge p^c > \epsilon\}$.

Confidence represents how much we can believe in the point prediction and credibility represents the largest ϵ for which the prediction region is empty. Confidence should thus be as high as possible, whilst credibility should not be too low, since that indicates either that the exchangeability assumption is not valid or that the test instance is atypical.

A. Inductive Conformal Prediction

In the online transductive setting, the underlying algorithm has to be invoked for every example. The underlying algorithm is therefore normally some simple but fast learner, such as the k-nearest neighbor algorithm. If some other kind of underlying algorithm is preferred, or the problem is offline in nature, the computational cost for using the transductive setting soon becomes overwhelming for most data sets. Inductive CP can be used to overcome the computational cost of transductive CP [4].

When using inductive conformal prediction, the available data is divided into a training set z_T, used to train a model using the underlying algorithm, and a calibration set z_C, used to calculate the conformity score. (T, C) is a partition of $\{1, ..., l\}$.

The p-values of the *inductive conformal predictor* (ICP) is defined by:

$$p^c = \frac{|j \in C : \alpha_j \leq \alpha^c| + 1}{|C| + 1} \tag{3}$$

where

$$\alpha_j = A(z_T, z_j)$$

and

$$\alpha^c = A(z_T, (x, c)).$$

One of the drawbacks with ICP is that only part of the available data is used for training the underlying model and for calibrating the conformity scores. How the division is done may affect the results of the ICP in two different ways. Using a small calibration set leads to a high variance of confidence, since the smaller the calibration set is, the less fine-grained the conformity scores will be. The p-values may change dramatically just due high variance in the chosen sample. On the orther hand, the smaller the training set is, the less powerful the predictive model will be.

B. Cross Conformal Prediction

To overcome the drawbacks of having to use only part of the data as training and calibration sets, *cross conformal prediction* (CCP) was proposed in [5]. In CCP, cross-validation is used to ensure that each example is used as part of the calibration set exactly once. $K \in \{2, 3, ...\}$ is a parameter of the method and for every $k \in \{1, ..., K\}$ a model is built. The examples $z_i, i \in \{1, ..., l\}$ are divided into K different sets and one of them is withheld as a calibration set for every fold, whereas the remaining sets are merged into a set used to train the kth model. Using the model and the calibration set from each fold, a total of k p-values for each possible class label $c \in Y$ are calculated and the p-value from the CCP is approximately the average of the k p-values calculated from the folds.

More formally, the examples $z_i, i \in \{1, ..., l\}$ are divided into K different folds. Each fold consists of the examples $z_{F_k}, k = 1, ..., K$, where $(F_1, ..., F_K)$ is a partition of $\{1, ..., l\}$. The p-values from each fold $k \in \{1, ..., K\}$ are defined as

$$p_k^c = \frac{|j \in F_k : \alpha_{j,k} \leq \alpha_k^c| + 1}{|F_k| + 1} \tag{4}$$

and the p-values from the CCP are defined as

$$p^c = \bar{p}^c + \frac{K - 1}{l + 1}(\bar{p}^c - 1) \approx \bar{p}^c \tag{5}$$

where $\bar{p}^c = \frac{1}{K} \sum_{k=1}^{K} p_k^c$.

The conformity scores $\alpha_{j,k}$ and α_k^c are defined for each fold k and each potential class label $c \in Y$ by

$$\alpha_{j,k} = A(z_{F_{-k}}, z_j)$$

and

$$\alpha_k^c = A(z_{F_{-k}}, (x, c))$$

where $F_{-k} = \cup_{f \neq k} F_f$.

According to Vovk, no theoretical results about the validity of CCP exist, which means that CCP has not yet been proved to be valid.

C. Bootstrap Conformal Prediction

Bootstrap conformal prediction (BCP) is similar to CCP. Just like for CCP, $K \in \{2, 3, ...\}$ is a parameter of the method and for every $k \in \{1, ..., K\}$ a model is built. Instead of using cross-validation to separate training and calibration sets, BCP uses bootstrap replicates [6] and uses all

the examples included in the bootstrap to train a model, i.e., approximately 63.2% of all examples, and uses the examples not included in the bootstrap as the calibration set for that model. The bootstrap is a bag, since duplicates are allowed and the examples not included in the bag are often referred to as out-of-bag.

More formally, for each $k \in \{1, ..., K\}$, a training sample z_{b^k} of l examples is drawn (with replacement) from the available examples $z_i, i \in \{1, ..., l\}$. Since instances are drawn with replacement, allowing duplicates to be drawn, b^k denotes a bag of indices for examples used to train a model. The conformity scores $\alpha_{j,k}$ and α_k^c are defined for each fold k and each potential class label $c \in Y$ by

$$\alpha_{j,k} = A(z_{b^k}, z_j), j \in \{1, ..., l\} \backslash b^k$$

and

$$\alpha_k^c = A(z_{b^k}, (x, c))$$

where $\{1, ..., l\} \backslash z_{b^k}$ denotes all out-of-bag examples, i.e. the calibration set, for the kth model.

The p-value of BCP is defined in the following way:

$$p^c = \frac{\sum_{k=1}^{K} |\{j \in \{1, ..., l\} \backslash b^k : \alpha_{j,k} \leq \alpha_k^c\}| + T/l}{T + T/l} \quad (6)$$

where $T = \sum_{k=1}^{K} |\{1, ..., l\} \backslash b^k|$ is the total size of the calibration sets.

Vovk presents BCP in an appendix in [5] and does not mention anything about its validity. Our conclusion is that BCP, just like CCP, also lacks theoretical results proving it to be valid.

D. Using Out-Of-Bag Estimation in Conformity Functions

A *Bagging Ensemble* [10] is an aggregated model combining multiple ensemble members. The ensemble members can be built using any kind of machine learning algorithm. Each ensemble member is trained using a bootstrap replicate drawn with replacement from the available data. For classification tasks, the combination rule that is used to produce the prediction from the ensemble is usually the majority vote of all the ensemble members. When using bootstrapping, approximately 1/3 of all examples will be out-of-bag (OOB) for each ensemble member. Using votes only from ensemble members for which an example is OOB makes it possible to get an unbiased estimate on the training set.

Thus, when the underlying algorithm of CP is a bagging ensemble another option, besides using ICP, CCP or BCP, is also available. Instead of dividing the available data into a proper training set and a calibration set, all data can be used for both purposes by using the OOB examples as a calibration set.

Formally, let $M = \cup_{n=1,...,N} m_n$ be an ensemble of size N, where each m_n is called a member of the ensemble. $M(x) = \cup_{n=1,...,N} m_n(x)$ predicts a class label $c \in Y$ for the object x using majority voting. The probability estimate ρ^c for each class label is the percentage of ensemble members voting for that class[2] . Let a training sample z_{b^n} of size l with examples drawn (with replacement) from the available examples $z_i, i \in \{1, ..., l\}$ be used to train each ensemble member m_n. b^n represents the indices of examples that are in the bag for the nth ensemble member, i.e., these are used for training m_n, and $b^{-n} = \{1, ..., l\} \backslash b^n$ represents the indices of examples that are out-of-bag for m_n. The conformity score α_j for a calibration example (x_j, y_j) is defined by

$$\alpha_j = A(\{z_1, ..., z_l\}, (x_j, y_j)) = S(y_j, M_{b^{-n}}(x_j)) \quad (7)$$

where $M_{b^{-n}}(x_j) = \cup_{n=1,...,N \wedge j \in b^{-n}} m_n(x_j)$.

The conformity score for each class on a test example, α^c, is defined by equation 1, letting $y = c, c \in Y$. In other words, the full ensemble M is used in a normal way for all the test examples.

III. Related Work

Several studies have evaluated conformal prediction in different contexts and with different purposes. Many studies have focused on different (non-)conformity measures. Papadopoulos [11] evaluated two different nonconformity measures defined for neural networks on three different data sets. Bhattacharyya [2] evaluated three different nonconformity measures for random forests on five larger data sets. Devetyarov and Nouretdinov [7] also evaluated three different nonconformity measures for random forests on a total of 10 data sets. They introduced the OOB method used in this study as a nonconformity measure for random forests. There are also several other studies that define some conformity measure or use some special feature of the underlying algorithm, and use a rather small number of data sets for evaluation, such as [12], [13], [14].

To the best of our knowledge, the only study so far to investigate different methods to utilize the available data in the best way is Vovk in his working paper [5]. In that paper, he evaluated ICP, CCP and BCP on two data sets, concluding that CCP seems to be more efficient than ICP and that BCP is similar to ICP. However, using only one or two data sets does not allow any conclusions to be drawn on what to expect in general.

Considering the lack of empirical studies systematically evaluating different aspects of CP on a sufficiently large number of data sets, the main contribution of this study is a systematic investigation of one of these aspects, i.e., effective utilization of data.

IV. Method

As described in the introduction, the overall purpose of this study is to evaluate how different methods for utilizing the available data in inductive CP affect the efficiency and predictive performance of the conformal predictors.

[2]To distinguish between the p-values of the CP, which is calculated from the conformity scores and used to define the region predictions, and the p-values from the underlying algorithm, ρ will be used to denote the p-values of the underlying algorithm and p will be used to denote the p-values of the CP.

The underlying algorithm used with all methods has been bagging ensembles with 100 artificial neural networks (ANNs) as ensemble members. In the first experiment, ordinary bagging has been used, which means that for each ANN, a sample with the same size as the training set was drawn with replacement and used to train the ANN. All ANNs were feed-forward neural networks trained with resilient back-propagation and a maximum of 1000 epochs.

It is well-known that there is no general rule-of-thumb that will always find an optimal, or even acceptable, number of hidden units in an ANN, based on the data set characteristics. However, there are many rules-of-thumb proposed, and most of them suggest that the number of hidden units should be somewhere between the number of input units and the number of output units, thus resulting in pyramid shaped networks. Previous studies [15] have indicated that some variation in the number of hidden nodes among ensemble members is beneficial for ensemble performance. In this study, the number of hidden units h for each ANN were calculated using equation 8.

$$h = \left\lfloor (\eta + 0.5)\frac{|attributes| + |classes|}{2} \right\rfloor \tag{8}$$

where $\eta \in [0, 1]$ is a uniformly distributed random number.

In the experiment, ICP, CCP, BCP and OOB have been compared. How the ICP divides the available data will obviously affect both the performance of the underlying model and the quality of the conformity calibration. The proportion used for ICP in this study was $2:1$, following the recommendation in [5], using $2/3$ of the data as training set and $1/3$ as the calibration set for the ICP. For CCP, the number of folds was set to $K = 5$, again using one of the settings evaluated in [5]. It could be argued that using a larger K would lead to better results, since a larger part would be used as training set in each internal CCP-fold. On the other hand, since one underlying model has to be trained for each internal CCP-fold, a larger K would have a great negative impact on the computation time, especially since the underlying model in this case has been an ensemble of 100 ANNs. $K = 5$ was chosen since it provided a reasonable balance between computational efficiency and sufficiently large training sets. The number of repetitions in BCP does not affect the number of examples used for training a model but will only affect the variance of confidence, which will decrease when increasing the number of repetitions. The same considerations regarding computational cost do, however, apply to BCP as well. The number of repetitions chosen in this study was again $K = 5$ to make the comparison with CCP fair. Since the OOB method used standard bagging, no partitioning parameter had to be chosen since OOB used all available data as training set and used the out-of-bag-examples as a calibration set.

The similarity function S in equation 1 and equation 7 that was used to calculate the conformity scores was defined using the concept of *margin*. Since all data sets used in this study were binary data sets, the definition will assume two classes but may easily be extended to multiple classes. The higher the probability estimate ρ is for the true class y for an example z_i, the more conforming the example is and the higher the probability estimate for the other class, the less conforming the example is. Based on this, the conformity score for a calibration example $z_i = (x_i, y_i)$ with the true class y_i is defined by equation 9.

$$\alpha_i = S(y_i, M(x_i)) = \rho_i^y - \rho_i^{-y} \tag{9}$$

where $\rho_i = M(x_i)$ are the probability estimates for all class labels produced by the model when presented with the ith object, ρ_i^y is the probability for the correct class y_i and ρ_i^{-y} is the probability for the other class. If data sets with multiple classes would have been used, ρ_i^{-y} would have been substituted with $\max(\rho_i^{-y})$, i.e., the highest probability estimate among the remaining classes. For a test example, the model M refers to the entire ensemble, while for examples in the calibration set when using the OOB method, M refers to the ensemble as defined in equation 7.

For a specific test example $z = (x, c)$, for which a conformity score for each possible class $c \in Y$ must be defined, the following equation is used:

$$\alpha^c = S(c, M(x)) = \rho^c - \rho^{-c} \tag{10}$$

where ρ^c is the probability for class c and ρ^{-c} is the probability for the other class.

Two different types of measure were used to evaluate performance and efficiency. Measures of the first type, which we here refer to as model specific, are independent of the significance ϵ used. Confidence was chosen as a model specific measure of efficiency. Furthermore, accuracy and AUC were chosen as model specific performance measures. The point prediction for each example is defined as $\arg\max_{y \in Y} p^y$. It must be noted that accuracy and AUC, as defined in this paper, are based on the p-values from the conformal predictor, i.e., when the conformal predictor is forced to produce point predictions based on the p-values. The second kind of measures was measures that are dependent on the level of significance ϵ used.

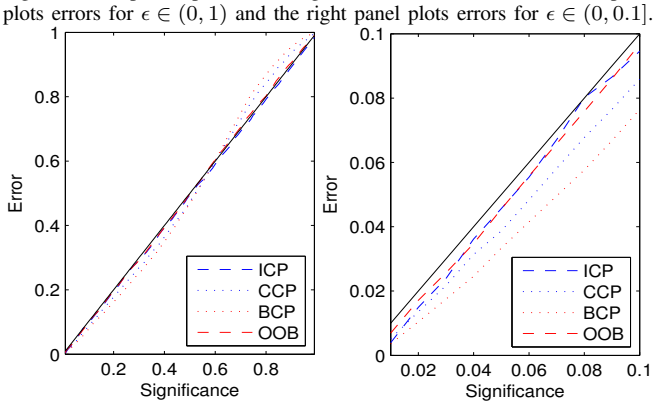
- *MultiC* is the percentage of examples with more than one class in the region prediction.
- *OneC* is the percentage of examples with exactly one class in the region prediction.
- *ZeroC* is the percentage of examples with an empty region prediction.
- *OneAcc* is the percentage of the examples with exactly one class in the region prediction that are also correct.
- The *error* measure is the percentage of region predictions incorrectly excluding the true class label.

During experimentation, cross-validation with 4 folds has been used, with identical folding for all methods. For every fold during experimentation, one fold was used as a test set and the examples in the remaining folds were available for training and calibration. The 34 data sets used are all publicly available from either the UCI repository [16] or the PROMISE Software Engineering Repository [17].

V. Results

Before presenting the results from the main experiment, it is important to verify that the methods and data sets used produce valid conformal predictors. One way to verify validity is to use a so-called calibration plot, showing the error, i.e., the fraction of predictions not including the correct class label, for all $\epsilon \in (0, 1)$. Fig. 1 shows calibration plots for the average error over all data sets. The left panel shows the average error for the whole range of significances, i.e., $\epsilon \in (0, 1)$, whereas the right panel only shows the most important range of significances, i.e., when $\epsilon \in (0, 0.1]$. The black solid line represents the diagonal, when $error = \epsilon$.

Fig. 1. Both panels plot the average error for all data sets. The left panel plots errors for $\epsilon \in (0, 1)$ and the right panel plots errors for $\epsilon \in (0, 0.1]$.



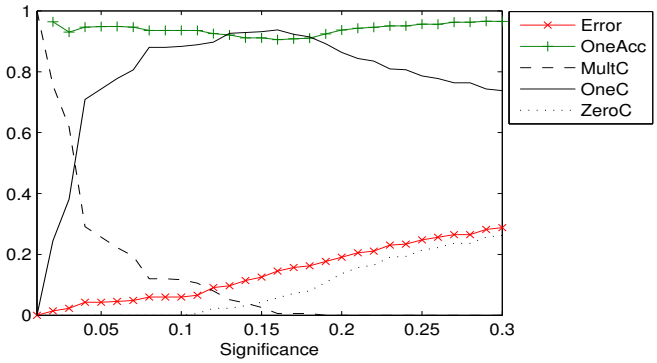
As can be seen in Fig. 1, both ICP and OOB are well calibrated in the sense that the average error over the 34 data sets used follows (and does not exceed) the diagonal of the plots. When looking at the results for CCP and BCP, they seem to be less well calibrated, i.e., the average error rates are higher than the significance thresholds in some cases. This could be seen as an indication that the latter methods may not be valid after all, something which needs to be further investigated. For the time being, we are content by concluding that the proposed OOB method, similar to the ICP method, apparently results in valid conformal predictors.

Fig. 1 only shows the average tendency over all 34 data sets. However, plots for individual data sets have also been inspected and these confirm that all data sets used in this study are reasonably well calibrated, i.e., do not deviate substantially from the general tendency seen in Fig. 1.

Fig. 2 plot the error, the accuracy for the examples with exactly one prediction (OneAcc), and the three measures MultiC, OneC and ZeroC indicating efficiency, for Ionosphere data set. The results using the ICP method are averaged over the 4 folds.

Looking at Fig. 2 and considering the error, it is clear that ICP produces well calibrated conformal predictors for the Ionosphere data set. However, ICP seems to be overly conservative for some levels of significances on this data set. OneAcc is not defined for significance $\epsilon = 0.01$, since all examples have both classes in the region prediction at that level.

Fig. 2. Results for $\epsilon \in (0, 0.3]$ for the Ionosphere data set using ICP



In the main experiment, the three methods (ICP, CCP and BCP) proposed by Vovk in [5] are compared to the OOB method. Table I shows the accuracy, AUC and confidence of the different models. The *Average* denotes the average over all data sets and *Avg. rank* is the average rank among the four methods within each measure over all data sets.

TABLE I
The Confidence, Accuracy, and AUC of the models used

	Confidence				Accuracy				AUC			
Data sets	**ICP**	**CCP**	**BCP**	**OOB**	**ICP**	**CCP**	**BCP**	**OOB**	**ICP**	**CCP**	**BCP**	**OOB**
ar1	.935	.953	.936	.959	.859	.884	.867	.892	.680	.759	.689	.760
ar4	.899	.902	.873	.924	.832	.860	.832	.822	.703	.844	.825	.849
bcancer	.873	.877	.862	.882	.678	.717	.710	.699	.635	.637	.651	.628
breast-w	.980	.991	.989	.992	.951	.963	.969	.964	.981	.991	.990	.991
colic	.937	.934	.924	.946	.774	.815	.813	.815	.846	.870	.882	.881
credit-a	.942	.955	.951	.957	.854	.864	.861	.862	.905	.925	.920	.924
credit-g	.907	.910	.901	.918	.752	.763	.756	.751	.765	.780	.778	.777
diabetes	.915	.922	.915	.923	.776	.775	.780	.768	.832	.838	.837	.838
heart-c	.923	.937	.927	.944	.769	.789	.779	.786	.855	.877	.870	.876
heart-s	.921	.930	.921	.937	.815	.811	.814	.815	.880	.889	.891	.892
hepatitis	.920	.939	.931	.944	.794	.800	.820	.820	.792	.821	.842	.839
ionosphere	.956	.968	.962	.972	.895	.897	.883	.889	.937	.954	.948	.957
jEdit4042	.876	.890	.878	.896	.697	.708	.737	.708	.782	.788	.796	.796
jEdit4243	.827	.817	.787	.825	.618	.629	.648	.631	.663	.687	.684	.685
jm1	.936	.936	.937	.936	.810	.811	.811	.811	.705	.708	.708	.709
kc1	.960	.962	.957	.962	.853	.857	.856	.857	.791	.800	.795	.800
kc2	.945	.945	.946	.949	.831	.829	.835	.829	.819	.830	.832	.827
kc3	.951	.965	.966	.972	.882	.902	.900	.897	.648	.746	.793	.791
kr-vs-kp	.998	.999	.999	.999	.993	.993	.995	.993	.999	1.00	.999	1.00
letter	.997	.999	.999	.999	.993	.994	.996	.996	.998	.999	.999	.999
liver	.876	.875	.847	.879	.713	.728	.731	.731	.745	.760	.743	.756
mozilla4	.969	.968	.968	.968	.878	.878	.878	.878	.930	.930	.930	.930
mw1	.943	.962	.959	.971	.878	.908	.918	.918	.684	.765	.773	.760
pc1_req	.846	.836	.807	.842	.647	.688	.681	.688	.619	.636	.641	.638
pc3	.967	.972	.968	.976	.883	.891	.889	.889	.774	.821	.818	.824
pc4	.983	.983	.982	.984	.908	.908	.913	.910	.904	.921	.915	.917
promoters	.892	.878	.853	.894	.765	.745	.754	.707	.851	.871	.855	.865
sick	.996	.996	.997	.997	.967	.968	.968	.970	.966	.969	.971	.971
sonar	.936	.947	.930	.956	.808	.846	.846	.851	.886	.936	.921	.934
spambase	.990	.991	.990	.991	.942	.948	.946	.945	.980	.983	.983	.983
spect	.910	.915	.913	.934	.816	.805	.804	.808	.772	.755	.793	.774
spectf	.942	.932	.914	.952	.822	.879	.862	.885	.873	.900	.890	.906
tic-tac-toe	.934	.952	.933	.949	.800	.835	.824	.825	.872	.913	.896	.909
vote	.982	.988	.986	.991	.954	.947	.947	.956	.981	.981	.983	.985
Average	**.934**	**.939**	**.930**	**.945**	**.830**	**.842**	**.842**	**.840**	**.825**	**.849**	**.848**	**.852**
Avg. rank	**3.15**	**2.35**	**3.24**	**1.26**	**3.44**	**2.19**	**2.25**	**2.12**	**3.88**	**1.97**	**2.26**	**1.88**

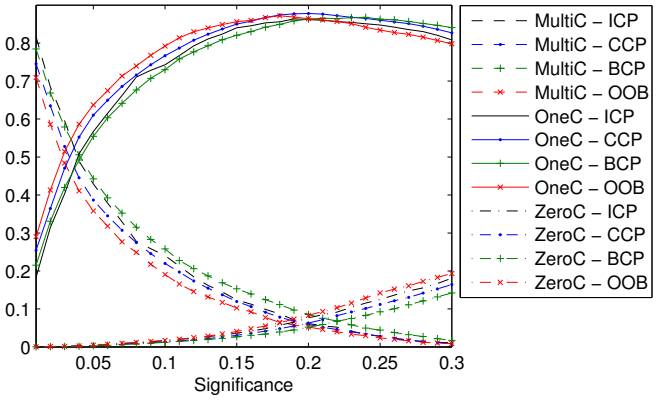
When looking at the confidence measure, it is clear that OOB is the most efficient method. When using statistical tests to compare all the methods, i.e., a Friedman test [18]

followed by a post-hoc Nemenyi test [19], as recommended in [20], the tests show that OOB has significantly higher confidence than the other three methods ($\alpha = 0.05$ gives a critical distance of $CD = 0.782$ when comparing 4 different methods). The test also reveals that CCP has significantly higher confidence than both ICP and BCP.

Furthermore, it is obvious from the results that ICP is clearly performing worse than the other three methods when considering accuracy and AUC. This should be expected, since less information has been utilized both when training the model and when calibrating the p-values. When using the same test as before, this assumption is confirmed; both accuracy and AUC are indeed significantly lower for ICP than for the other three methods.

The measures above do not indicate how the conformal predictors perform for different levels of significance. Fig. 3 shows the averaged results over all data sets for MultiC, OneC, and ZeroC for significance $\epsilon \in (0, 0.3]$. The colors distinguish between methods and the different line styles represents the different measures. In general, OneC should be as high as possible whereas MultiC and ZeroC should be as low as possible. OneC, MultiC and ZeroC sum up to 100%.

Fig. 3. The figure shows the average MultiC, OneC, and ZeroC for significances $\epsilon \in (0, 0.3]$.



As can be seen in the figure, the curves for OOB (the x-marked lines) clearly indicate its superior efficiency, especially in the most relevant interval of significances, with $\epsilon \in (0, 0.15]$. This is most clearly seen by looking at the solid line for OneC, which clearly shows that OOB obtains a higher OneC for the lowest ϵ.

The plot only shows the average trend for all the data sets and does not guarantee that OOB will be better for each specific data set. However, the results show that in general, OOB can be expected to be the most efficient method, at least for the intervals of significances that will be most relevant under normal circumstances.

To further analyze the results in Fig. 3 and enable statistical testing, Table II tabulates OneC for $\epsilon \in \{0.01, 0.05, 0.1\}$.

TABLE II
ONEC FOR $\epsilon \in \{0.1, 0.05, 0.01\}$

	$\epsilon = 0.01$				$\epsilon = 0.05$				$\epsilon = 0.1$			
Data sets	**ICP**	**CCP**	**BCP**	**OOB**	**ICP**	**CCP**	**BCP**	**OOB**	**ICP**	**CCP**	**BCP**	**OOB**
ar1	.000	.000	.000	.000	.459	.686	.445	.742	.784	.884	.876	.983
ar4	.000	.000	.000	.000	.261	.441	.084	.403	.493	.656	.498	.759
bcancer	.000	.028	.000	.066	.273	.277	.172	.263	.469	.469	.375	.441
breast-w	.174	.810	.621	.831	.960	.986	.989	.979	.906	.924	.941	.913
colic	.000	.033	.016	.068	.511	.516	.397	.576	.815	.772	.736	.851
credit-a	.105	.136	.078	.149	.531	.635	.609	.650	.812	.904	.881	.901
credit-g	.026	.076	.039	.104	.408	.396	.366	.432	.628	.622	.571	.645
diabetes	.052	.128	.125	.109	.447	.469	.428	.475	.618	.663	.629	.674
heart-c	.000	.092	.105	.162	.400	.561	.485	.614	.723	.763	.710	.789
heart-s	.000	.078	.059	.111	.511	.515	.396	.522	.693	.741	.663	.789
hepatitis	.000	.032	.006	.039	.206	.554	.510	.554	.652	.755	.755	.787
ionosphere	.000	.185	.063	.228	.743	.798	.761	.823	.883	.943	.926	.963
jEdit4042	.000	.044	.018	.081	.208	.306	.270	.361	.390	.518	.441	.580
jEdit4243	.000	.024	.019	.052	.154	.163	.119	.219	.336	.273	.227	.333
jm1	.131	.137	.144	.139	.483	.470	.484	.471	.749	.752	.761	.751
kc1	.269	.354	.275	.335	.689	.679	.647	.680	.879	.908	.881	.902
kc2	.090	.017	.015	.063	.581	.634	.622	.653	.849	.828	.845	.845
kc3	.011	.022	.033	.136	.668	.817	.810	.852	.945	.972	.952	.987
kr-vs-kp	.992	.997	.997	.996	.964	.961	.961	.952	.911	.91	.918	.90
letter	.992	.996	.994	.997	.955	.968	.972	.956	.905	.939	.948	.905
liver	.000	.052	.032	.075	.200	.217	.156	.223	.417	.432	.342	.458
mozilla4	.409	.390	.376	.375	.779	.774	.767	.774	.959	.957	.955	.957
mw1	.057	.072	.094	.159	.557	.802	.730	.859	.864	.973	.945	.988
pc1_req	.000	.022	.019	.038	.288	.197	.103	.194	.378	.294	.219	.338
pc3	.132	.311	.168	.454	.774	.817	.794	.829	.955	.965	.960	.974
pc4	.545	.641	.589	.669	.912	.888	.885	.883	.969	.979	.982	.972
promoters	.000	.000	.000	.000	.398	.218	.198	.385	.547	.443	.339	.594
sick	.909	.909	.915	.917	.968	.972	.971	.967	.910	.913	.912	.909
sonar	.000	.163	.091	.260	.514	.591	.524	.644	.803	.841	.721	.870
spambase	.661	.742	.729	.766	.980	.981	.977	.985	.931	.938	.943	.931
spect	.000	.049	.087	.180	.427	.323	.338	.511	.616	.654	.609	.688
spectf	.000	.052	.046	.147	.549	.511	.379	.638	.805	.744	.675	.871
tic-tac-toe	.286	.332	.175	.346	.537	.633	.521	.614	.740	.800	.724	.777
vote	.391	.724	.393	.821	.993	.991	.975	.986	.926	.938	.954	.908
Average	**.183**	**.254**	**.215**	**.290**	**.567**	**.610**	**.554**	**.637**	**.743**	**.767**	**.730**	**.792**
Avg. Rank	**3.62**	**2.19**	**2.76**	**1.43**	**2.79**	**2.15**	**3.26**	**1.79**	**3.03**	**2.15**	**2.91**	**1.91**

For all three levels of significance, OOB turns out to be significantly more efficient than both ICP and BCP (using the same test as before with the same CD). For $\epsilon = 0.01$, ICP is significantly less efficient than the three other methods. CCP is also significantly more efficient than BCP for $\epsilon = 0.05$ and ICP for $\epsilon = 0.1$.

The difference between OOB and CCP is very close to being significant using the Nemenyi post-hoc test, which is suitable for testing any difference between multiple methods. However, in a direct comparison, using a pair-wise sign test [21] (with $\alpha = 0.05$) between OOB and CCP, it is seen that OOB is significantly more efficient than CCP for $\epsilon \in \{0.01, 0.05\}$.

Another aspect of the different methods is the computational efficiency. Both CCP and BCP have utilized models that have been approximately five times more costly to build than OOB. The reason is that $K = 5$ and a bagging ensemble has been built for each repetition. This, in combination with comparable model performance regarding accuracy and AUC as well as superior confidence, makes OOB the most natural choice when the underlying algorithm is a bagging ensemble.

VI. CONCLUSIONS

The purpose of this study has been to evaluate how different methods for utilizing the available data when building

a conformal predictor affect the efficiency and performance. To be able to draw general conclusions, a large number of data sets have been used during experimentation. The three methods (ICP, CCP and BCP) proposed by Vovk in [5] have been compared to the OOB method.

The results clearly show that the OOB method is superior to the other three methods. It is the simplest of the four methods in the sense that none of the design choices needed for the other three methods exists for OOB. It simply builds one ensemble model using all available data for both training *and* calibration. The other methods must either divide the data (ICP), or build multiple models and somehow combine their predictions (CCP and BCP). OOB is also the most efficient of all the methods, both in terms of confidence, when producing point predictions, and in terms of OneC, when producing region predictions. Furthermore, the model performance (in terms of accuracy and AUC) when using point predictions is significantly better than both ICP and BCP and comparable to CCP.

VII. Discussion and Future Work

The results in this study are based on bagging ensembles of ANNs and even though the assumption is that the conclusions will hold for bagging ensembles in general, including random forests, this has still to be verified.

The evaluation of the validity of the different methods casts some doubts about whether CCP and BCP are truly valid in general. As Vovk points out in his conclusions in [5], it has not yet been proven theoretically that CCP is truly valid, which is also the case for BCP.

These results may still provide valuable insights even when using inductive conformal prediction with an underlying algorithm not utilizing bagging. The results suggest that using CCP both increases the model performance as well as leads to a more efficient conformal predictor in comparison to ICP. Provided that CCP will eventually be proven valid it should be preferred over ICP when the added computational cost is no problem.

References

[1] Valiant. L.*A theory of the learnable*. Communications of the ACM, 27, 1984.

[2] Bhattacharyya, S. "Confidence in Predictions from Random Tree Ensembles." *In Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pp. 71-80. IEEE, 2011.

[3] Melluish, T., Craig S., Nouretdinov, I., and Vovk, V. "Comparing the Bayes and Typicalness Frameworks." *In Machine Learning: ECML 2001: 12th European Conference on Machine Learning*, Freiburg, Germany, September 5-7, 2001. Proceedings, vol. 12, p. 360. Springer, 2001.

[4] Vovk, V., Gammerman, A., and Shafer, G. *Algorithmic learning in a random world*. Springer, 2005.

[5] Vovk, V. "Cross-conformal predictors," *arXiv preprint arXiv*:1208.0806, 2012.

[6] Efron, B. "Bootstrap methods: another look at the jackknife." *The annals of Statistics*, vol. 7.1, pp. 1-26. 1979.

[7] Devetyarov, D., and Nouretdinov, I. "Prediction with Confidence Based on a Random Forest Classifier." *Artificial Intelligence Applications and Innovations*. pp. 37-44. 2010

[8] Breiman, L. "Random forests." *Machine learning* vol. 45.1, pp. 5-32. 2001.

[9] Shafer, G. and Vovk, V. "A tutorial on conformal prediction." *Journal of Machine Learning Research*, vol 9, pp. 371 - 421. 2008.

[10] Breiman, L. "Bagging Predictors." *Machine learning* vol. 24.2, pp. 123-140. 1996.

[11] Papadopoulos, H. Inductive conformal prediction: Theory and application to neural networks.*Tools in Artificial Intelligence*, Chap. 18, p. 315-330. 2008.

[12] Yang, F., Wang, H. Z., Mi, H., and Cai, W. W. Using random forest for reliable classification and cost-sensitive learning for medical diagnosis. *BMC bioinformatics*, vol 10 (Suppl 1), S22. 2009.

[13] Nouretdinov, I., Gammerman, A., Qi, Y., and Klein-Seetharaman, J. . Determining confidence of predicted interactions between HIV-1 and human proteins using conformal method. In *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*. p. 311. NIH Public Access. 2012.

[14] Lambrou, A., Papadopoulos, H., and Gammerman, A. Reliable confidence measures for medical diagnosis with evolutionary algorithms. *Information Technology in Biomedicine, IEEE Transactions on*, 15(1), pp. 93-99. 2011.

[15] Johansson, U., and Lofstrom, T. "Producing implicit diversity in ANN ensembles." *In Neural Networks (IJCNN), The 2012 International Joint Conference on*, pp. 1-8. IEEE, 2012.

[16] Frank, A. and Asuncion, A. "UCI Machine Learning Repository" [http://archive.ics.uci.edu/ml]. *Irvine, CA: University of California, School of Information and Computer Science*. 2010.

[17] Shirabad, S., and Menzies, T. "The PROMISE repository of software engineering databases." *School of Information Technology and Engineering, University of Ottawa, Canada 24*. 2005.

[18] Friedman, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of American Statistical Association*, 32:675âĂŞ701, 1937.

[19] Nemenyi, P. B.. *Distribution-free multiple comparisons. PhD-thesis*. Princeton University, 1963.

[20] Demšar, J. "Statistical comparisons of classifiers over multiple data sets." *The Journal of Machine Learning Research*. vol 7, pp. 1-30. 2006.

[21] Dixon, W. J., and Mood, A. M. The statistical sign test. *Journal of the American Statistical Association*, 41(236), 557-566. 1946.