Model-Agnostic Nonconformity Functions for Conformal Classification

Ulf Johansson*†, Henrik Linusson†, Tuve Löfström*†, Henrik Boström‡
*Dept. of Computer Science and Informatics, Jönköping University, Sweden
ulf.johansson@ju.se
†Dept. of Information Technology, University of Borås, Sweden
{henrik.linusson, tuve.lofstrom}@hb.se
‡Dept. of Computer and Systems Sciences, Stockholm University, Sweden
henrik.bostrom@dsv.su.se

***Abstract*—A conformal predictor outputs prediction regions, for classification *label sets*. The key property of all conformal predictors is that they are valid, i.e., their error rate on novel data is bounded by a preset significance level. Thus, the key performance metric for evaluating conformal predictors is the size of the output prediction regions, where smaller (more informative) prediction regions are said to be more efficient. All conformal predictions rely on *nonconformity functions*, measuring the strangeness of an input-output pair, and the efficiency depends critically on the quality of the chosen nonconformity function. In this paper, three model-agnostic nonconformity functions, based on well-known loss functions, are evaluated with regard to how they affect efficiency. In the experimentation on 21 publicly available multi-class data sets, both single neural networks and ensembles of neural networks are used as underlying models for conformal classifiers. The results show that the choice of nonconformity function has a major impact on the efficiency, but also that different nonconformity functions should be used depending on the exact efficiency metric. For a high fraction of single-label predictions, a margin-based nonconformity function is the best option, while a nonconformity function based on the hinge loss obtained the smallest label sets on average.**

***Keywords*—Conformal prediction, Classification, Neural networks**

I. Introduction.

Conformal prediction [1] is a framework for producing predictions with guarantees. All conformal predictors are *valid*, i.e., given a significance level $\epsilon \in (0, 1)$, the error rate of a conformal predictor will, in the long run, be exactly ϵ. Conformal predictors output *prediction sets*; in regression a prediction interval and in classification a (possibly empty) subset of the class labels. In conformal prediction, an error is committed when the prediction set does not contain the true target.

Conformal prediction was originally introduced for a transductive setting but most recent studies, including this one, focus on *inductive conformal prediction* (ICP). In ICP, only one model is induced from the training data, and this model is then used for the prediction of all test instances in batch mode. ICP, however, requires an additional labeled data set (called the *calibration set*) that was not used for the training of the model. ICP can be applied on top of any predictive model (the *underlying model*), thus turning it into a conformal predictor. Conformal prediction requires only one assumption; that the data (actually the calibration set and the test set) must be *exchangeable*, a property very similar to, but slightly weaker than, the standard i.i.d.

To generate the valid prediction sets, conformal predictors use *nonconformity functions* that measure the strangeness (the *nonconformity*) of an instance-target pair. Prediction sets in ICP are produced by first applying the nonconformity function to all instances in the calibration set, resulting in a set of nonconformity scores. The fundamental idea of conformal prediction is then to exploit the fact that the calibration set is drawn from the same distribution as the test set in order to provide the guarantees for the test predictions. In predictive modeling, the nonconformity function used is most often based on the underlying model, and instances where the model is wrong or uncertain are deemed to be more nonconforming.

Since all conformal predictors are valid, the key criterion for comparing different conformal predictors is informativeness (or *efficiency*) i.e., how tight the prediction sets are. Efficiency is affected by how accurate the underlying model is, but also by the quality of the nonconformity function. In this paper, we evaluate three well-known loss functions; *hinge loss*, *margin* and *Brier score* as nonconformity functions for predictive classification. Since all three functions operate on probability estimates, they can be applied to any classifier producing probability estimates, i.e., they are *model-agnostic*. In addition, we investigate a couple of different ways to generate the probability estimates when the underlying models are either single neural networks (ANNs) or an ensemble of bagged ANNs.

It must be noted that for a conformal predictor, it is the *ordering* of the nonconformity scores that matter. For two-class problems, all three loss functions will, of course, order the instances identically, i.e., they will be equally efficient. With this in mind, we focus on multi-class problems. When applied to multi-class problems, conformal prediction may result in prediction sets with a single label (a *singleton* prediction), as well as both empty sets and prediction sets with more than one label. Even if singleton predictions are the most informative, the ability to exclude one or several unlikely classes can still be very useful in many real-world applications, especially since the validity of the predicted label subset is guaranteed.

This work was supported by the Swedish Knowledge Foundation through the project Data Analytics for Research and Development (20150185).

978-1-5090-6182-2/17/$31.00 ©2017 IEEE

The conformal prediction framework is under rapid development. While the fundamentals are extremely solid and often mathematically proven, there is an obvious need to establish best practices concerning design choices like underlying models, nonconformity functions and parameter values. We argue that this is an important step to make conformal prediction accessible. Unfortunately there are very few published papers providing a systematic investigation of such questions. In fact, most studies focus on one specific underlying model, and use a very limited number of data sets, making them serve mainly as proofs-of-concept. So, there is an apparent need for larger studies, explicitly evaluating techniques for producing efficient conformal predictors. Such studies should preferably evaluate the effect of key design choices, while using a sufficiently large number of data sets to allow for statistical inference, thus making it possible to establish best practices.

In this study, we investigate both single ANNs and ensembles of bagged ANNs as underlying models. When a bagged ensemble is used as the underlying model, the calibration can be performed on the out-of-bag instances, instead of a separate calibration set, thus making it possible to utilize all the available labeled data for both the training of the model and the calibration of the conformal predictor, see e.g., [2]. All-in-all the overall purpose of this paper is:

1) An outright comparison between different model-agnostic nonconformity functions for single ANNs and ensembles of bagged ANNs
2) Investigating the effect on efficiency of using *early stopping* when training single ANNs
3) Investigating the effect on efficiency of using *averaging* compared to *majority vote* when generating the probability estimates from ensembles of bagged ANNs

Based on this, the goal is to produce recommendations for best practices when using conformal prediction for multi-class prediction.

II. Background.

In this section, we give a formal presentation of the conformal prediction framework and present related work.

A. Conformal prediction.

Conformal predictors [1] output predictions associated with statistically valid measures of confidence. The predictions given are multi-valued, i.e., for a test example x_{k+1}, a conformal predictor outputs a set of labels, Γ^{ϵ}_{k+1}. This *prediction region* contains the true label y_{k+1} with probability $1-\epsilon$, where $\epsilon \in (0,1)$ is the predefined significance level.

To produce these prediction regions, a conformal predictor uses a *nonconformity function*, an arbitrary function $A : X \times Y \rightarrow \mathbb{R}$, that measures the strangeness of an example $(\boldsymbol{x}, y)$. Based on the nonconformity scores of examples with known output labels, and the nonconformity score of a tentatively labeled test pattern $(\boldsymbol{x}_{k+1}, \tilde{y})$, a p-value statistic is calculated in order to attempt to reject the hypothesis that $\tilde{y}$ corresponds with the true label y_{k+1}. All labels $\tilde{y} \subseteq Y$ that are not rejected at the chosen significance level ϵ constitute the final prediction region Γ^{ϵ}_{k+1}, which contains y_{k+1} with a probability of $1-\epsilon$.

In predictive modeling, i.e., classification or regression, nonconformity functions are often based on the prediction error of an underlying classification or regression model

$$A(\boldsymbol{x}_i, y_i, h) = \Delta\left[h\left(\boldsymbol{x}_i\right), y_i\right], \tag{1}$$

where h is a predictive model trained for the problem in question, and Δ is some function that measures the error of a single prediction. This type of nonconformity function is based on the intuition that uncommon or strange examples will often obtain larger prediction errors than common or "normal" examples. For classification problems, the error function Δ normally operates on probability estimates provided by h, e.g., the *hinge loss* function,

$$\Delta\left[h\left(\boldsymbol{x}_i\right), y_i\right] = 1 - \hat{P}_h\left(y_i \mid \boldsymbol{x}_i\right), \tag{2}$$

where $\hat{P}_h\left(y \mid \boldsymbol{x}\right)$ denotes the probability estimate (as given by h) that the pattern $\boldsymbol{x}$ belongs to class y.

An *inductive conformal classifier* [1], [3], [4] is constructed, using some learning algorithm H and some nonconformity measure A, according to the following procedure:

1) Divide the training data Z into two disjoint subsets: a proper training set Z^t and a calibration set Z^c.
2) Apply the learning algorithm H to Z^t, to produce an underlying model h.
3) Use the nonconformity function, e.g., Eq. 1, to measure the nonconformity of the examples in Z^c, obtaining a list of calibration scores $\alpha_1, \ldots, \alpha_q$.

When a new test pattern $\boldsymbol{x}_{k+1}$ arrives, a prediction region is constructed as follows:

1) Select a significance level $\epsilon \in (0,1)$.
2) Obtain a prediction $h(\boldsymbol{x}_{k+1})$.
3) Tentatively assign a label $\tilde{y} \in Y$ as the output label for $\boldsymbol{x}_{k+1}$, and measure the nonconformity of the pattern $(\boldsymbol{x}_{k+1}, \tilde{y})$ using the nonconformity function.
4) Calculate a p-value according to

$$p^{\tilde{y}}_{k+1} = \frac{\left|\left\{z_i \in Z^c : \alpha_i \geq \alpha^{\tilde{y}}_{k+1}\right\}\right| + 1}{q+1}. \tag{3}$$

5) If $p^{\tilde{y}}_{k+1} < \epsilon$, the label $\tilde{y}$ is rejected, otherwise it is included in the prediction set Γ^{ϵ}_{k+1}.
6) Repeat the process from step 3 for each possible label $\tilde{y} \in Y$.

With probability $1-\epsilon$, Γ^{ϵ}_{k+1} contains the true label y_{k+1}.

B. Related work.

Conformal prediction is not the only framework providing guarantees for predictions. PAC-learning [5] returns, based on some confidence level, an upper bound on the error. PAC-learning has, however, two important shortcomings when compared to conformal prediction: firstly, unlike conformal prediction, the PAC bounds are for the overall error and not for the individual predictions; secondly, the bounds tend to be very loose unless the data is exceptionally clean, see [6].

The Bayesian framework, like conformal prediction, can provide probabilistic measures of the quality of individual predictions. However, if the underlying distribution can not be

correctly assumed *a priori*, the framework can not guarantee that the prediction intervals will contain the true target as often as is implied by the confidence level. Consequently, the Bayesian framework relies on a correct knowledge of the priors, which is not necessary in conformal prediction.

Conformal prediction has been applied to many popular machine learning algorithms. For classification, various algorithms have been evaluated, e.g., ANNs [7], [8], k-nearest neighbor (kNN) [9], support vector machines (SVMs) [10], [11], decision trees [12], random forests [10], [13] and evolutionary algorithms [14], [15]. A number of algorithms have, in a similar way, been used in regression studies, like ridge regression [3], ANNs [7] and random forests [16]. A multi-label solution based on kNN was recently proposed in [17].

When considering nonconformity functions for classification, two distinct groups of functions can be identified: *model-dependent functions*, defining nonconformity using features that are extracted from the structure of the model; and *model-agnostic functions*, only relying on information related to the output of the underlying model, i.e., for classification typically a probability estimate. Examples of model-dependent nonconformity functions found in the literature are: the (signed) distance from the separating hyperplane in a SVM [18], [19]; the proximities between instances in a random forest [20], [21] and the distance between instances in a kNN [22]. Examples of model-agnostic nonconformity functions found in the literature are: margin [23] and hinge loss [20]. To the best of our knowledge, no previous study on conformal prediction has used the Brier score as a model-agnostic nonconformity function.

Several studies have evaluated nonconformity functions using different underlying models, e.g., [24], [25], while other studies have evaluated different nonconformity functions for a particular machine learning algorithm, e.g., [20], [21] or how different parameters used when training the underlying model affect the efficiency [23]. Until now, however, no study has evaluated how different model-agnostic functions affect the efficiency of the conformal predictors.

One drawback with ICP compared to transductive conformal prediction is that some data must be set aside as a calibration set. Using less data when training the underlying model is likely to reduce its predictive performance and consequently the efficiency of the ICP. The possibility of using out-of-bag data for calibration was identified in [13]. Since out-of-bag calibration makes it possible to use all training data for both model learning and calibration, it enables the underlying model to be trained on all training data, resulting in more accurate underlying models and eventually in more efficient conformal predictors, see [2], [26], [27].

III. Method.

As described in the introduction, the main purpose of this study is to compare different model-agnostic nonconformity functions with regard to efficiency. In addition, different ways of producing the probability estimates for ANNs and ANN ensembles are investigated and evaluated. In this study, efficiency is measured using two different metrics:

- **OneC** is the proportion of all predictions that are singletons. The motivation for this metric is that singleton predictions are the most informative, and often what is sought for.
- **AvgC** is the average number of class labels in the prediction sets, i.e., a direct measure of how good the model is of rejecting class labels.

All experimentation was performed in MatLab, in particular using the Neural network and the Statistics tool boxes. Default settings were used, except when explicitly described below. For the evaluation, 10x10-fold cross-validation was used, i.e., 90% of the data was used for training of the model and calibrating the corresponding conformal classifier, while the remaining 10% was used as the test set. All results reported below are averaged over the 10x10 folds.

In the first experiment, the underlying models consist of single ANNs. Each ANN is a multi-layer perceptron with one hidden layer. For the number of hidden units, h, a simple rule of thumb was used: $h = \lfloor \frac{2a}{3} + C \rfloor$, where a is the number of attributes and C the number of classes. A localist coding was employed, and cross-entropy was used as the error function. For the calibration, the training data was split $4:1$, i.e., 80% of the training instances were used for the actual training and 20% for the calibration. When early stopping was applied, the validation set consisted of 20% of the instances available for the model training.

In the second experiment, we use bagged ensembles consisting of 150 multi-layer perceptron networks with one hidden layer. The same rule-of-thumb as in Experiment 1 was applied for determining the number of hidden units. In bagging [28], the necessary diversity is introduced by training each model on a *bootstrap replicate*. The result is that approximately 63% of all training instances are present in the training set for a specific model. The instances that are missing in the bootstrap replicate are said to be *out-of-bag* for that model. Consequently, these instances can be used as an independent validation set in order to, for instance, estimate the expected accuracy on novel data. In this experiment, we used the out-of-bag instances of the training set to act as a calibration set, as proposed in [29]. Naturally, since the calibration was performed on the out-of-bag set, all training data was used for both model training and calibration. It must be noted, however, that the out-of-bag error is most often a pessimistic estimation, since only the models that have a certain instance out-of-bag are used for predicting that instance, i.e., the actual ensemble used on out-of-bag instances is much smaller than the original ensemble. As described above, conformal prediction requires that the calibration set and the test set is used identically by the model and the nonconformity function. With this in mind, the actual predictions were for each test instance made by a (random) subset, corresponding to the ensemble size for out-of-bag estimates, of the ensemble members.

In this study, we investigate model agnostic nonconformity functions that are based on probability estimates. The three different non-conformity functions evaluated are described below:

Hinge: This first error measure is based simply on the probability estimate provided for the correct class label, y_i, according to

$$\Delta\left[h\left(\boldsymbol{x}_i\right), y_i\right] = 1 - \hat{P}_h\left(y_i \mid \boldsymbol{x}_i\right), \tag{4}$$

i.e., a pattern is considered to be nonconforming when the probability estimate for the true class is low. The probability estimate provided for any other (incorrect) class is irrelevant for this measure.

Margin: This measure considers two class labels: the true class label and the most likely incorrect class label. The margin of these two probabilities is

$$\Delta\left[h\left(\boldsymbol{x}_i\right), y_i\right] = \max_{y \neq y_i} \hat{P}_h\left(y \mid \boldsymbol{x}_i\right) - \hat{P}_h\left(y_i \mid \boldsymbol{x}_i\right), \quad (5)$$

meaning a nonconforming example is one which has a low probability estimate for the true class label and/or a high probability estimate for any other (incorrect) class label.

Brier score: The Brier score [30] nonconformity measure considers all possible class labels according to

$$\Delta\left[h\left(\boldsymbol{x}_i\right), y_i\right] = \frac{1}{|Y|} \sum_{y}^{Y} \left(P\left[y \mid \boldsymbol{x}_i\right] - \hat{P}_h\left[y \mid \boldsymbol{x}_i\right]\right)^2, \quad (6)$$

where $P\left(y \mid \boldsymbol{x}_i\right) = 1$ if $y = y_i$ and 0 otherwise. Here, the nonconformity of an example is dependent on the probability estimates for all labels, so that even small amounts of confusion regarding the true class label affect the final scoring.

In all 21 publicly available multi-class data sets are used in the experimentation. All data sets are from the UCI [31] repository. The data sets are described in Table I below, where *#class* is the number of classes, *#inst.* is the number of instances and *#attrib*. is the number of input attributes.

TABLE I. DATA SETS

Data set	#class	#inst.	#attrib.	Data set	#class	#inst.	#attrib.
balance	3	625	4	user	5	403	5
cars	4	1728	6	wave	3	5000	40
cmc	3	1473	9	vehicle	4	846	18
cool	3	768	8	whole	3	440	7
ecoli	8	336	7	wine	3	178	13
glass	6	214	9	wineR	6	1599	11
heat	3	768	8	wineW	7	4898	11
image	7	2310	19	vowel	11	990	11
iris	3	150	4	yeast	10	1484	8
steel	7	1941	27	zoo	7	101	16
tae	3	151	5				

Using the experimentation, we aim to answer the three following questions:

1) How will the three model-agnostic nonconformity functions affect OneC and AvgC? We expect the nonconformity functions to perform very differently since they utilize one, two or all class estimates.
2) How will using early stopping when training single ANNs affect the probability estimates, and will this carry over to OneC and AvgC? Typically, using early stopping will result in smoother probability estimates with several classes having positive probability estimates, while employing full training tend to result in probability estimates with one clearly dominating class, while most other classes have probabilities very close to 0.
3) Will using averaging or majority vote when producing the probability estimates from the bagged ensembles affect the probability estimates, and how will this impact OneC and AvgC. Averaging will, most often, produce smoother estimates.

IV. RESULTS.

Before presenting results for the conformal classifiers, we show the underlying model results in Table II below. Looking at the results for the single ANNs, using no early stopping results in slightly more accurate models overall. A Wilcoxon singed-rank test, however, shows that the difference is not statistically significant; the p-value is 0.32. As expected, the bagged ensembles are significantly more accurate than the single nets, but there are very small differences between using averaging or majority vote when forming the ensemble prediction.

TABLE II. MODEL ACCURACIES

	Single MLP		Bagged ensemble	
	Full training	Early stopping	Averaging	Majority vote
balance	.960	.918	.972	.972
cars	.977	.943	.991	.991
cmc	.518	.505	.554	.556
cool	.927	.913	.950	.950
ecoli	.773	.836	.851	.850
glass	.651	.589	.703	.701
heat	.977	.927	.988	.988
image	.966	.954	.978	.978
iris	.943	.959	.953	.953
steel	.676	.713	.771	.770
tae	.534	.465	.597	.597
user	.877	.904	.912	.913
wave	.819	.862	.858	.858
vehicle	.806	.781	.847	.847
whole	.663	.714	.701	.704
wine	.971	.970	.980	.980
wineR	.580	.586	.642	.640
wineW	.552	.532	.586	.585
vowel	.911	.853	.979	.979
yeast	.570	.581	.609	.609
zoo	.946	.915	.961	.961
Mean	**.790**	**.782**	**.828**	**.828**

Next, we take a detailed look at the results for single ANNs and focus on one specific significance level, here $\epsilon = 0.05$. Table III below shows the empirical error rates. Clearly, the error rates are very close to the significance level, even when looking at individual data sets, thus confirming that the models are valid and well-calibrated.

TABLE III. EMPIRICAL ERROR RATES FOR $\epsilon = 0.05$

	Full training			Early stopping		
	brier	hinge	margin	brier	hinge	margin
balance	.053	.053	.053	.050	.049	.049
cars	.050	.050	.050	.053	.053	.054
cmc	.050	.051	.051	.049	.049	.050
cool	.046	.046	.047	.048	.048	.048
ecoli	.045	.045	.043	.047	.050	.048
glass	.047	.054	.049	.054	.048	.051
heat	.048	.048	.047	.046	.045	.046
image	.050	.050	.050	.047	.048	.047
iris	.048	.056	.052	.052	.046	.049
steel	.051	.047	.051	.052	.050	.051
tae	.051	.042	.052	.048	.052	.046
user	.048	.046	.048	.050	.048	.048
wave	.050	.050	.050	.049	.049	.049
vehicle	.051	.051	.052	.051	.050	.050
whole	.053	.048	.053	.048	.046	.048
wine	.050	.050	.049	.050	.049	.049
wineR	.050	.049	.050	.049	.048	.049
wineW	.051	.050	.051	.050	.049	.051
vowel	.048	.049	.048	.049	.050	.050
yeast	.050	.050	.050	.050	.051	.049
zoo	.046	.048	.044	.046	.043	.050
Mean	**.049**	**.049**	**.049**	**.049**	**.049**	**.049**

Table IV shows the efficiency of single ANNs at $\epsilon = 0.05$. Starting with OneC, we see that for this significance level, approximately 50% of all predictions are singletons. Overall there is a small but obvious advantage of using early stopping, regardless of the actual nonconformity function used. For OneC, Wilcoxon signed ranks tests show no statistically significant differences between full training and early stopping. For AvgC, however, using early stopping is significantly more efficient when using the Brier nonconformity function ($p = 0.028$) and when using margin ($p = 0.0457$).

TABLE IV. EFFICIENCY FOR $\epsilon = 0.05$

	OneC						AvgC					
	Full training			Early stopping			Full training			Early stopping		
	br	hin	mar	br	hin	mar	br	hin	mar	br	hing	mar
balance	.965	.965	.966	.914	.906	.918	0.99	0.98	0.99	1.10	1.10	1.11
cars	.962	.961	.962	.962	.961	.963	0.96	0.96	0.96	1.02	1.02	1.02
cmc	.176	.123	.185	.096	.056	.129	2.58	2.56	2.58	2.54	2.54	2.58
cool	.915	.910	.915	.897	.896	.897	1.15	1.09	1.15	1.11	1.10	1.13
ecoli	.337	.136	.335	.605	.469	.620	5.05	3.95	5.06	2.57	1.87	2.81
glass	.142	.043	.136	.127	.058	.144	5.04	4.20	5.03	4.35	3.30	4.69
heat	.963	.963	.964	.881	.882	.882	0.97	0.97	0.97	1.09	1.09	1.12
image	.975	.975	.974	.978	.976	.978	0.98	0.98	0.98	1.00	1.00	1.00
iris	.771	.773	.767	.917	.929	.921	1.30	1.21	1.31	1.06	1.03	1.05
steel	.303	.076	.303	.399	.290	.423	5.07	3.63	5.07	3.07	2.14	3.58
tae	.105	.047	.109	.052	.025	.064	2.75	2.70	2.75	2.75	2.73	2.77
user	.621	.582	.620	.879	.874	.881	2.33	1.58	2.33	1.19	1.13	1.24
wave	.652	.651	.652	.759	.759	.759	1.69	1.35	1.69	1.24	1.24	1.31
vehicle	.552	.475	.552	.590	.559	.601	2.29	1.71	2.29	1.56	1.49	1.75
whole	.099	.047	.101	.049	.018	.068	2.75	2.69	2.76	2.57	2.58	2.58
wine	.958	.958	.959	.961	.961	.960	0.97	0.97	0.98	0.99	0.98	0.99
wineR	.133	.052	.135	.074	.008	.119	5.07	3.38	5.10	3.13	2.63	4.02
wineW	.055	.010	.090	.011	.000	.038	3.62	2.80	4.89	3.09	2.94	4.05
vowel	.882	.855	.881	.684	.617	.696	2.00	1.18	2.02	2.47	1.48	2.90
yeast	.141	.032	.145	.152	.045	.188	8.13	4.47	8.19	5.18	3.32	6.63
zoo	.918	.853	.921	.856	.745	.858	1.25	1.28	1.25	1.57	1.71	1.59
Mean	**.554**	**.499**	**.556**	**.564**	**.525**	**.576**	**2.71**	**2.13**	**2.78**	**2.13**	**1.83**	**2.38**
Rank	**1.60**	**2.79**	**1.62**	**2.05**	**2.62**	**1.33**	**2.21**	**1.10**	**2.69**	**1.90**	**1.24**	**2.86**

Most importantly, there are large differences between the three nonconformity functions, as seen by the average ranks. In particular, it is obvious that *hinge* produces the fewest singletons. In order to determine any statistically significant differences, we used the procedure recommended in [32] and performed a Friedman test [33], followed by Bergmann-Hommel's [34] dynamic procedure to establish all pairwise differences. Using these tests, the nonconformity function based on margin resulted in significantly higher OneC than *Brier* and *hinge*, when early stopping was applied. When using full training, *margin* and *Brier* were both significantly more efficient than *hinge*. Looking at the AvgC metric, there is a clear ordering when early stopping is used; *hinge* is significantly more efficient than *Brier*, which in turn is significantly more efficient than *margin*. With full training, *hinge* resulted in significantly smaller predictions sets compared to both *margin* and *Brier*.

So, interestingly enough, the results show that different nonconformity functions should be used depending on the preferred performance metric. If a large proportion of singleton prediction is important, i.e., OneC is more important than AvgC, the margin-based nonconformity function should be used. If, on the other hand, the overall goal is to exclude as many labels as possible, i.e., AvgC is preferred over OneC, then *hinge* would be the best choice as nonconformity function. Comparing full training and early stopping, the latter is generally beneficial for the efficiency, despite the fact that the underlying models are slightly less accurate.

Table V below shows the average error rates over all data sets for all considered significance levels in Experiment 1, i.e., when using single ANNs as underlying models. The empirical error rates are very close to the significance levels, indicating that all methods produce valid and well-calibrated conformal classifiers.

TABLE V. EXPERIMENT 1: EMPIRICAL ERROR RATES

Significance	Full training			Early stopping		
level	br	hin	mar	br	hin	mar
$\epsilon = 0.01$	.010	.010	.010	.010	.010	.010
$\epsilon = 0.05$	.049	.049	.049	.049	.049	.049
$\epsilon = 0.1$	.099	.099	.099	.099	.099	.099
$\epsilon = 0.2$	.199	.197	.199	.199	.199	.198

Turning to the efficiency, Table VI below summarizes Experiment 1. A first impression is that the results from the other significance levels are quite similar to the results for $\epsilon = 0.05$, which were analyzed in detail above. Specifically, *margin* is most often the best option, in order to ensure a high OneC, while *hinge* is clearly the worst. For a low AvgC, *hinge* is always the best, most often followed by *Brier*.

A direct comparison between using early stopping and full training shows that the use of early stopping will always lead to a lower AvgC. When looking at OneC, however, early stopping only produced higher OneC for $\epsilon = 0.01$ and $\epsilon = 0.05$.

TABLE VI. EXPERIMENT 1: EFFICIENCY

	OneC						AvgC					
	Full training			Early stopping			Full training			Early stopping		
	br	hin	mar	br	hin	mar	br	hin	mar	br	hing	mar
$\epsilon = 0.01$												
Mean	.242	.206	.240	.285	.250	.286	4.05	3.48	4.06	3.70	3.01	3.77
Rank	1.38	2.86	1.76	1.76	2.67	1.57	2.33	1.33	2.33	1.90	1.33	2.76
$\epsilon = 0.05$												
Mean	.554	.499	.556	.564	.525	.576	2.71	2.13	2.78	2.13	1.83	2.38
Rank	1.60	2.79	1.62	2.05	2.62	1.33	2.21	1.10	2.69	1.90	1.24	2.86
$\epsilon = 0.1$												
Mean	.641	.590	.648	.625	.596	.643	2.05	1.61	2.14	1.56	1.51	1.73
Rank	1.83	2.50	1.67	1.86	2.76	1.38	2.33	1.36	2.31	1.98	1.33	2.69
$\epsilon = 0.2$												
Mean	.704	.675	.717	.675	.663	.693	1.32	1.20	1.38	1.19	1.18	1.22
Rank	2.10	2.40	1.50	2.10	2.48	1.43	1.95	1.57	2.48	1.98	1.52	2.50

Table VII below presents statistically significant pairwise differences, as identified by a Friedman test followed by Bergmann-Hommel's dynamic procedure. Here, 'v' indicates that the row setup is significantly more efficient than the column setup, while a '*' shows that it is significantly less efficient.

TABLE VII. EXPERIMENT 1: STATISTICALLY SIGNIFICANT DIFFERENCES BETWEEN THE DIFFERENT NC-FUNCTIONS. $\alpha = 0.05$

OneC	$\epsilon = 0.01$			$\epsilon = 0.05$			$\epsilon = 0.1$			$\epsilon = 0.2$		
Full	br	hin	mar	br	hin	mar	br	hin	mar	br	hin	mar
br	-	v		-	v		-	v		-		
hin	*	-	*	*	-	*	*	-	*		-	*
mar		v	-		v	-		v	-		v	-
ES	br	hin	mar	br	hin	mar	br	hin	mar	br	hin	mar
br	-	v		-		*	-	v		-		*
hin	*	-	*		-	*	*	-	*		-	*
mar		v	-	v	v	-		v	-	v	v	-
AvgC	$\epsilon = 0.01$			$\epsilon = 0.05$			$\epsilon = 0.1$			$\epsilon = 0.2$		
Full	br	hin	mar	br	hin	mar	br	hin	mar	br	hin	mar
br	-	*		-	*		-	*		-		
hin	v	-	v	v	-	v	v	-	v		-	v
mar		*	-		*	-		*	-		*	-
ES	br	hin	mar	br	hin	mar	br	hin	mar	br	hin	mar
br	-		v	-	*	v	-	*	v	-		
hin		-	v	v	-	v	v	-	v		-	v
mar	*	*	-	*	*	-	*	*	-		*	-

Based on this analysis, it is obvious that the best choice overall for a high OneC is *margin*. Similarly, *hinge* is the clear winner for a low AvgC.

Table VIII below shows when the use of early stopping produced significantly more efficient conformal classifiers, as determined by Wilcoxon signed ranks tests. Here, it may be noted that while the use of early stopping was only significantly better for AvgC on some significance levels and for some nonconformity functions, the only situation where using full training appears to be beneficial is when a high OneC is required for $\epsilon = 0.1$ and $\epsilon = 0.2$. It must be noted, however, that OneC is a slightly awkward metric when there are many empty predictions, which is the case for several data sets when $\epsilon = 0.1$ or $\epsilon = 0.2$.

TABLE VIII. EXPERIMENT 1: STATISTICALLY SIGNIFICANT ADVANTAGES USING EARLY STOPPING. $\alpha = 0.05$

	$\epsilon = 0.01$			$\epsilon = 0.05$			$\epsilon = 0.1$			$\epsilon = 0.2$		
	br	hin	mar	br	hin	mar	br	hin	mar	br	hin	mar
OneC												
AvgC	v	v	v	v		v			v			

Based on the results and analyses of Experiment 1, the general recommendations for conformal classification using a single ANN as underlying model are:

- Use early stopping.
- If the goal is to get as many singleton predictions as possible, i.e., to maxmimize OneC, employ the margin-based nonconformity function.
- If the goal is to maximize the number of rejected labels, i.e., to minimize AvgC, employ the nonconformity function based on hinge loss.

Turning to Experiment 2, i.e., using ensembles of 150 bagged ANNs, we see from the empirical error rates in Table IX below that these setups too produce valid and well-calibrated conformal classifiers.

TABLE IX. EXPERIMENT 2: EMPIRICAL ERROR RATES

Significance	Averaging			Majority Vote		
levels	br	hin	mar	br	hin	mar
$\epsilon = 0.01$	.010	.010	.010	.010	.010	.010
$\epsilon = 0.05$	.049	.049	.049	.049	.050	.049
$\epsilon = 0.1$	.100	.099	.100	.100	.100	.099
$\epsilon = 0.2$	.200	.200	.200	.199	.199	.200

Table X below summarizes the efficiency results for the bagged ensembles. Comparing these results to the single models in Table VI above, we see that the much stronger underlying models also generated more efficient conformal classifiers, in particular when a high confidence ($\epsilon = 0.01$ or $\epsilon = 0.05$) was required. Looking at the different nonconformity functions, the results are very similar to Experiment 1; *margin* (and to a lesser degree *Brier*) produced more singleton prediction sets, while *hinge* rejected the most labels. Comparing mean ranks, *margin* was actually always the best choice for high OneC, while *hinge* resulted in the lowest AvgC.

TABLE X. EXPERIMENT 2: EFFICIENCY

	OneC						AvgC					
	Averaging			Majority Vote			Averaging			Majority Vote		
	br	hin	mar	br	hin	mar	br	hin	mar	br	hing	mar
$\epsilon = 0.01$												
Mean	.512	.475	.514	.490	.450	.491	2.81	2.26	2.90	3.00	2.77	3.02
Rank	2.00	2.62	1.38	1.83	2.60	1.57	1.95	1.40	2.64	1.95	1.52	2.52
$\epsilon = 0.05$												
Mean	.634	.603	.650	.649	.601	.656	1.74	1.57	1.98	2.10	1.76	2.18
Rank	2.12	2.38	1.50	2.02	2.48	1.50	1.93	1.57	2.50	1.88	1.60	2.52
$\epsilon = 0.1$												
Mean	.679	.653	.701	.701	.662	.712	1.37	1.34	1.48	1.64	1.37	1.72
Rank	1.95	2.55	1.50	1.79	2.57	1.64	2.07	1.36	2.57	2.17	1.38	2.45
$\epsilon = 0.2$												
Mean	.731	.718	.745	.744	.725	.756	1.08	1.08	1.09	1.12	1.08	1.20
Rank	1.98	2.29	1.74	1.76	2.57	1.67	2.02	1.76	2.21	2.14	1.57	2.29

As seen in Table XI below, many differences are actually statistically significant. Specifically, *hinge* is significantly better for minimizing AvgC than both *Brier* and *margin* on most significance levels, and using both averaging and majority vote. While *margin* is always the best option for maximizing OneC, the difference when compared to *Brier* is most often not significant.

TABLE XI. EXPERIMENT 2: STATISTICALLY SIGNIFICANT DIFFERENCES BETWEEN THE DIFFERENT NC-FUNCTIONS. $\alpha = 0.05$

OneC	$\epsilon = 0.01$			$\epsilon = 0.05$			$\epsilon = 0.1$			$\epsilon = 0.2$		
Averaging	br	hin	mar	br	hin	mar	br	hin	mar	br	hin	mar
br	-	v		-	v		-	v		-		
hin	*	-	*	*	-	*	*	-	*		-	*
mar		v	-		v	-		v	-		v	-
Majority vote	br	hin	mar	br	hin	mar	br	hin	mar	br	hin	mar
br	-	v		-		*	-	v		-		*
hin	*	-	*		-	*	*	-	*		-	*
mar		v	-	v	v	-		v	-	v	v	-
AvgC	$\epsilon = 0.01$			$\epsilon = 0.05$			$\epsilon = 0.1$			$\epsilon = 0.2$		
Averaging	br	hin	mar	br	hin	mar	br	hin	mar	br	hin	mar
br	-	*		-	*		-	*		-		
hin	v	-	v	v	-	v	v	-	v		-	v
mar		*	-		*	-		*	-		*	-
Majority vote	br	hin	mar	br	hin	mar	br	hin	mar	br	hin	mar
br	-		v	-	*	v	-	*	v	-		
hin		-	v	v	-	v	v	-	v		-	v
mar	*	*	-	*	*	-	*	*	-		*	-

Table XII below, finally, confirms that the use of averaging will most often produce more efficient conformal classifiers. Specifically for AvgC, the differences are significant for most significance levels and nonconformity functions. For OneC, averaging is significantly better when $\epsilon = 0.01$ but actually significantly worse for $\epsilon = 0.2$. Again, using OneC when empty predictions are common, should probably be discouraged, so the overall picture is that averaging will often improve efficiency, while almost never deteriorate it.

TABLE XII. EXPERIMENT 2: STATISTICALLY SIGNIFICANT ADVANTAGES USING AVERAGING. $\alpha = 0.05$

	$\epsilon = 0.01$			$\epsilon = 0.05$			$\epsilon = 0.1$			$\epsilon = 0.2$		
	br	hin	mar	br	hin	mar	br	hin	mar	br	hin	mar
OneC	v	v	v							*		*
AvgC	v	v	v	v	v	v	v	v	v			v

Summarizing Experiment 2, the general recommendations for bagged ANN models are:

- Use averaging for producing the probability estimates.
- To maximize OneC, use the margin-based nonconformity function.
- To minimize AvgC, use the nonconformity function based on hinge loss.

Comparing the recommendations for what nonconformity function to use in conjunction with single vs. bagged ANNs, one can see that they are consistent; a nonconformity function based on the margin are in both cases recommended for maximizing OneC, while a nonconformity function based on hinge loss is recommended for minimizing AvgC. Moreover, the recommendation for single ANNs is to use early stopping and for bagged ANNs to employ averaging (in both cases independently of the chosen performance metric). In order to explain these findings, we can take a closer look at the way in which both the nonconformity functions and performance metrics are defined.

In order for a prediction to contribute to a high OneC, all but one of the class labels need to be assigned a high nonconformity score. This is exactly what is promoted by the margin function whenever a single label is assigned a high probability by the underlying model, since this probability will be added to the nonconformity score for all other labels. However, the hinge loss function, which considers each label in isolation, may fail to exclude all but the high-probability labels in such cases, since all the other labels must have a sufficiently low probability of their own, i.e., how much of the remaining probability mass is distributed to the high-probability label does not matter for *hinge*.

When it comes to AvgC, any removal of a label in a prediction will have a positive effect, even if there are more than one label left. If none of the class labels is given a high probability, e.g., there may be a tie between two labels, then the margin function will not result in high nonconformity scores even for the low-probability labels, meaning that it has a tendency to not remove any labels in such cases. In contrast, the hinge loss is more tempted to output a high score for a low-probability label, even in cases in which there are no single high-probability labels, and will hence allow for removing some labels in such cases.

In order to explain the observed benefit of employing early stopping for single ANNs and averaging for bagged ANNs, one should first realize that what is important for the conformal classifier is not the accuracy of the underlying model, but its ranking performance. Hence, although a fully trained ANN may have a higher accuracy than one obtained from early stopping, the tendency of the former to push the entire probability mass to one label risk leading to detrimental ranking performance, since such a model will not be able to distinguish hard from easy instances. Similarly, averaging will allow for taking uncertainty of individual models into account, effectively producing a more fine-grained scale according to which instances are ranked.

V. CONCLUDING REMARKS.

A large-scale empirical investigation has been presented on the generation of conformal classifiers using single ANNs (trained with and without early stopping) and bagged ANNs (combined using averaging and majority voting) together with three nonconformity functions based on standard loss-functions. A thorough statistical analysis of the extensive presented results lead to clear conclusions; a margin-based nonconformity function results in the highest fraction of single-label predictions (OneC), while a nonconformity function based on the hinge loss leads to the lowest average number of labels (AvgC) for all four considered types of underlying model. The investigation further showed that early stopping is beneficial when using a single ANN as the underlying model, while for bagged ANNs, averaging is the recommended option to combine predictions.

The presented findings are of both practical and theoretical importance. The conclusions from the study have lead to clear recommendations (or best practices) for how to construct informative conformal classifiers using ANNs, something which is of importance whenever predictions must come with statistical guarantees. The findings in this study also point to some promising directions for continued research, by indicating the potential of novel nonconformity functions that are tailored for specific performance metrics, including the ones considered in this study as well as the many alternative metrics that have been proposed in the literature.

REFERENCES

[1] V. Vovk, A. Gammerman, and G. Shafer, *Algorithmic Learning in a Random World*. Springer-Verlag New York, Inc., 2005.

[2] T. Löfström, U. Johansson, and H. Boström, "Effective utilization of data in inductive conformal prediction," in *Neural Networks (IJCNN), The 2013 International Joint Conference on*. IEEE, 2013.

[3] H. Papadopoulos, K. Proedrou, V. Vovk, and A. Gammerman, "Inductive confidence machines for regression," in *Machine Learning: ECML 2002*. Springer, 2002, pp. 345–356.

[4] H. Papadopoulos, "Inductive conformal prediction: Theory and application to neural networks," *Tools in Artificial Intelligence*, vol. 18, pp. 315–330, 2008.

[5] L. G. Valiant, "A theory of the learnable," *Communications of the ACM*, vol. 27, no. 11, pp. 1134–1142, 1984.

[6] I. Nouretdinov, V. Vovk, M. Vyugin, and A. Gammerman, "Pattern recognition and density estimation under the general i.i.d. assumption," in *Computational Learning Theory*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2001, vol. 2111, pp. 337–353.

[7] H. Papadopoulos, "Inductive conformal prediction: Theory and application to neural networks," *Tools in artificial intelligence*, vol. 18, no. 315-330, p. 2, 2008.

[8] H. Papadopoulos, E. Kyriacou, and A. Nicolaides, "Unbiased confidence measures for stroke risk estimation based on ultrasound carotid image analysis," *Neural Computing and Applications*, pp. 1–15, 2016.

[9] K. Nguyen and Z. Luo, "Conformal prediction for indoor localisation with fingerprinting method," *Artificial Intelligence Applications and Innovations*, pp. 214–223, 2012.

[10] D. Devetyarov and I. Nouretdinov, "Prediction with confidence based on a random forest classifier," *Artificial Intelligence Applications and Innovations*, pp. 37–44, 2010.

[11] L. Makili, J. Vega, S. Dormido-Canto, I. Pastor, and A. Murari, "Computationally efficient svm multi-class image recognition with confidence measures," *Fusion Engineering and Design*, vol. 86, no. 6, pp. 1213–1216, 2011.

[12] U. Johansson, H. Boström, and T. Löfström, "Conformal prediction using decision trees," in *International Conference Data Mining (ICDM)*. IEEE, 2013.

[13] S. Bhattacharyya, "Confidence in predictions from random tree ensembles," in *Data Mining (ICDM), 2011 IEEE 11th International Conference on*. IEEE, 2011, pp. 71–80.

[14] U. Johansson, R. König, T. Löfström, and H. Boström, "Evolved decision trees as conformal predictors," in *IEEE Congress on Evolutionary Computation*, 2013, pp. 1794–1801.

[15] A. Lambrou, H. Papadopoulos, and A. Gammerman, "Reliable confidence measures for medical diagnosis with evolutionary algorithms," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 15, no. 1, pp. 93–99, 2011.

[16] H. Boström, H. Linusson, T. Löfström, and U. Johansson, "Evaluation of a variance-based nonconformity measure for regression forests," in *Symposium on Conformal and Probabilistic Prediction with Applications*. Springer, 2016, pp. 75–89.

[17] A. Lambrou and H. Papadopoulos, "Binary relevance multi-label conformal predictor," in *Symposium on Conformal and Probabilistic Prediction with Applications*. Springer, 2016, pp. 90–104.

[18] V. N. Balasubramanian, R. Gouripeddi, S. Panchanathan, J. Vermillion, A. Bhaskaran, and R. Siegel, "Support vector machine based conformal predictors for risk of complications following a coronary drug eluting stent procedure," in *2009 36th Annual Computers in Cardiology Conference (CinC)*. IEEE, 2009, pp. 5–8.

[19] P. Toccaceli, I. Nouretdinov, and A. Gammerman, "Conformal predictors for compound activity prediction," in *Symposium on Conformal and Probabilistic Prediction with Applications*. Springer, 2016, pp. 51–66.

[20] D. Devetyarov and I. Nouretdinov, *Prediction with Confidence Based on a Random Forest Classifier*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 37–44. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-16239-8_8

[21] S. Bhattacharyya, "Confidence in predictions from random tree ensembles," *Knowledge and information systems*, vol. 35, no. 2, pp. 391–410, 2013.

[22] K. Proedrou, I. Nouretdinov, V. Vovk, and A. Gammerman, *Transductive Confidence Machines for Pattern Recognition*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 381–390. [Online]. Available: http://dx.doi.org/10.1007/3-540-36755-1_32

[23] U. Johansson, H. Boström, and T. Löfström, "Conformal prediction using decision trees," in *2013 IEEE 13th International Conference on Data Mining*. IEEE, 2013, pp. 330–339.

[24] H. Linusson, U. Johansson, H. Boström, and T. Löfström, "Efficiency comparison of unstable transductive and inductive conformal classifiers," in *IFIP International Conference on Artificial Intelligence Applications and Innovations*. Springer, 2014, pp. 261–270.

[25] T. Löfström, H. Boström, H. Linusson, and U. Johansson, "Bias reduction through conditional conformal prediction," *Intelligent Data Analysis*, vol. 19, no. 6, pp. 1355–1375, 2015.

[26] U. Johansson, H. Boström, T. Löfström, and H. Linusson, "Regression conformal prediction with random forests," *Machine Learning*, vol. 97, no. 1, pp. 155–176, 2014. [Online]. Available: http://dx.doi.org/10.1007/s10994-014-5453-0

[27] H. Boström, H. Linusson, T. Löfström, and U. Johansson, *Evaluation of a Variance-Based Nonconformity Measure for Regression Forests*. Cham: Springer International Publishing, 2016, pp. 75–89. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-33395-3_6

[28] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[29] U. Johansson, H. Boström, T. Löfström, and H. Linusson, "Regression conformal prediction with random forests," *Machine Learning*, vol. 97, no. 1-2, pp. 155–176, 2014.

[30] G. W. Brier, "Verification of forecasts expressed in terms of probability," *Monthly weather review*, vol. 78, no. 1, pp. 1–3, 1950.

[31] K. Bache and M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[32] S. Garcıa and F. Herrera, "An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons," *Journal of Machine Learning Research*, vol. 9, no. 2677-2694, p. 66, 2008.

[33] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of American Statistical Association*, vol. 32, pp. 675–701, 1937.

[34] B. Bergmann and G. Hommel, "Improvements of general multiple test procedures for redundant systems of hypotheses," in *Multiple Hypotheses Testing*. Springer, 1988, pp. 100–115.