Report Series / Department of
Computer Systems Sciences No. 15-009

On Effectively Creating Ensembles of Classifiers

Studies on Creation Strategies, Diversity and Predicting with Confidence

Tuwe Löfström

ISSN 1101-8526
ISBN 978-91-7649-179-9

Printed in Sweden by Holmbergs, Malmö 2015

Distributor: Department of Computer and Systems Sciences

This thesis is dedicated to my loving wife Helena for her support throughout this ordeal and to our wonderful children Nathanael, Sam, Kristina, Ingrid, Erik, Signe, Elsa and all future offspring.

Abstract

An ensemble is a composite model, combining the predictions from several other models. Ensembles are known to be more accurate than single models. Diversity has been identified as an important factor in explaining the success of ensembles. In the context of classification, diversity has not been well defined, and several heuristic diversity measures have been proposed. The focus of this thesis is on how to create effective ensembles in the context of classification. Even though several effective ensemble algorithms have been proposed, there are still several open questions regarding the role diversity plays when creating an effective ensemble. Open questions relating to creating effective ensembles that are addressed include: what to optimize when trying to find an ensemble using a subset of models used by the original ensemble that is more effective than the original ensemble; how effective is it to search for such a sub-ensemble; how should the neural networks used in an ensemble be trained for the ensemble to be effective? The contributions of the thesis include several studies evaluating different ways to optimize which sub-ensemble would be most effective, including a novel approach using combinations of performance and diversity measures. The contributions of the initial studies presented in the thesis eventually resulted in an investigation of the underlying assumption motivating the search for more effective sub-ensembles. The evaluation concluded that even if several more effective sub-ensembles exist, it may not be possible to identify which sub-ensembles would be the most effective using any of the evaluated optimization measures. An investigation of the most effective ways to train neural networks to be used in ensembles was also performed. The conclusions are that effective ensembles can be obtained by training neural networks in a number of different ways but that high average individual accuracy or much diversity both would generate effective ensembles. Several findings regarding diversity and effective ensembles presented in the literature in recent years are also discussed and related to the results of the included studies. When creating confidence based predictors using conformal prediction, there are several open questions regarding how data should be utilized effectively when using ensembles. Open questions related to predicting with confidence that are addressed include: how can data be utilized effectively to achieve more efficient confidence based predictions using ensembles; how do problems with class imbalance affect the confidence based predictions when using conformal prediction? Contributions include two studies where it is shown in the first that the use of out-of-bag estimates when using bagging ensembles results in more effective conformal predictors and it is shown in the second that a conformal predictor conditioned on the class labels to avoid a strong bias towards the majority class is more effective on problems with class imbalance. The research method used is mainly inspired by the design science paradigm, which is manifested by the development and evaluation of artifacts.

Sammanfattning

En ensemble är en sammansatt modell som kombinerar prediktionerna från flera olika modeller. Det är välkänt att ensembler är mer träffsäkra än enskilda modeller. Diversitet har identifierats som en viktig faktor för att förklara varför ensembler är så framgångsrika. Diversitet hade fram tills nyligen inte definierats entydigt för klassificering vilket resulterade i att många heuristiska diverstitetsmått har föreslagits. Den här avhandlingen fokuserar på hur klassificeringsensembler kan skapas på ett ändamålsenligt (eng. effective) sätt. Den vetenskapliga metoden är huvudsakligen inspirerad av design science-paradigmet vilket lämpar sig väl för utveckling och evaluering av IT-artefakter. Det finns sedan tidigare många framgångsrika ensemblealgoritmer men trots det så finns det fortfarande vissa frågetecken kring vilken roll diversitet spelar vid skapande av välpresterande (eng. effective) ensemblemodeller. Några av de frågor som berör diversitet som behandlas i avhandlingen inkluderar: Vad skall optimeras när man söker efter en delmängd av de tillgängliga modellerna för att försöka skapa en ensemble som är bättre än ensemblen bestående av samtliga modeller; Hur väl fungerar strategin att söka efter sådana delensembler; Hur skall neurala nätverk tränas för att fungera så bra som möjligt i en ensemble? Bidraget i avhandlingen inkluderar flera studier som utvärderar flera olika sätt att finna delensembler som är bättre än att använda hela ensemblen, inklusive ett nytt tillvägagångssätt som utnyttjar en kombination av både diversitets- och prestandamått. Resultaten i de första studierna ledde fram till att det underliggande antagandet som motiverar att söka efter delensembler undersöktes. Slutsatsen blev, trots att det fanns flera delensembler som var bättre än hela ensemblen, att det inte fanns något sätt att identifiera med tillgänglig data vilka de bättre delensemblerna var. Vidare undersöktes hur neurala nätverk bör tränas för att tillsammans samverka så väl som möjligt när de används i en ensemble. Slutsatserna från den undersökningen är att det är möjligt att skapa välpresterande ensembler både genom att ha många modeller som är antingen bra i genomsnitt eller olika varandra (dvs diversa). Insikter som har presenterats i litteraturen under de senaste åren diskuteras och relateras till resultaten i de inkluderade studierna. När man skapar konfidensbaserade modeller med hjälp av ett ramverk som kallas för conformal prediction så finns det flera frågor kring hur data bör utnyttjas på bästa sätt när man använder ensembler som behöver belysas. De frågor som relaterar till konfidensbaserad predicering inkluderar:

Hur kan data utnyttjas på bästa sätt för att åstadkomma mer effektiva konfidensbaserade prediktioner med ensembler; Hur påverkar obalanserad datade konfidensbaserade prediktionerna när man använder conformal perdiction? Bidragen inkluderar två studier där resultaten i den första visar att det mest effektiva sättet att använda data när man har en baggingensemble är att använda sk out-of-bag estimeringar. Resultaten i den andra studien visar att obalanserad data behöver hanteras med hjälp av en klassvillkorad konfidensbaserad modell för att undvika en stark tendens att favorisera majoritetsklassen.

Acknowledgements

Obviously, after spending almost a decade as a PhD student, there have been plenty of people who have contributed in the process of finalizing the thesis and to whom I am truly grateful.

First of all, a very special thanks to Henrik Boström, who has been my main supervisor for most of the time and who guided me towards, hopefully, becoming more and more independent as a researcher. He has many commendable qualities, but the ones I feel most grateful for are his wisdom and patience. On many occasions when I have felt stuck, he has been able to point me in the right direction.

The one person with whom I have been working most closely together throughout these years was undoubtedly my supervisor Ulf Johansson. He has been both a good friend, co-worker, and, when needed, supervisor. He is the one most instrumental in getting me to this point.

I am also very grateful to my good friends in our research group in Borås: Cecilia Sönströd, Rikard König, Håkan Sundell, Henrik Linusson, Anders Gidenstam, Karl Jansson, Patrick Gustafsson and Shirin Tavara. The discussions over lunch or at more or less spontaneous meetings have been very valuable. A special thanks goes to those of you who have had to share office with me and with whom I have had many constructive discussions and learned a lot from.

I am grateful to the University of Borås for giving me this opportunity. This gratefulness extends to the Universtiy of Skövde and the Knowledge Foundation, since my PhD studies were a shared investment for a couple of years. I hardly remember all the people who were making all the necessary decisions along the route, but Lars Niklasson and Rolf Appelqvist deserve a special thanks.

I want to thank my parents for always encouraging me to pursue whatever interest I might have fancied at the moment. I am grateful that I have had the opportunity to find my own path in life and that I have always felt your support. I am also grateful for the support I have got from my parents in law, not least in this project.

The gratitude I feel towards my wife Helena is beyond expression. You are the best! Without you, I would probably never even started on this journey and without you, I would not have finished it. Thank you for all the joy and support you have given me! And thank you for all our children, who are our

greatest achievement and who have, in their different ways, tried to support and encourage me, even when I have not given them the time they deserved!

Finally, there are also a whole lot of people who have helped me along the route towards this day: Matilda and Chikezie Okafor; Mia Löfström; Anna and Olle Kristell; Lars Cavallin; Kristoffer and Katarina Holt; Clemens and Nathalie Cavallin; Jacob and Eva Cavallin; Samuel Johan and Maria Cavallin; Pålle (Christian) and Marianne Cavallin; Marta/Sr Maria; Maria; Josef; the Borås Karate club; our neighbors; and many others.

List of Included Papers

The following papers, referred to in the text by their Roman numerals, are included in this thesis.

PAPER I:

Tuve Löfström, Ulf Johansson and Lars Niklasson, *Empirically investigating the importance of diversity*, **Information Fusion, 2007 10th International Conference on**, 1-8 (2007).

PAPER II:

Tuve Löfström, Ulf Johansson and Henrik Boström, *On the use of accuracy and diversity measures for evaluating and selecting ensembles of classifiers*, **Machine Learning and Applications, 2008. ICMLA'08. Seventh International Conference on**, 127-132 (2008).

PAPER III:

Tuve Löfström, Ulf Johansson and Henrik Boström, *Ensemble member selection using multi-objective optimization*, **Computational Intelligence and Data Mining, 2009. CIDM'09. IEEE Symposium on**, 245-251 (2009).

PAPER IV:

Tuve Löfström, Ulf Johansson and Henrik Boström, *Comparing methods for generating diverse ensembles of artificial neural networks*, **Neural Networks (IJCNN), The 2010 International Joint Conference on**, 1-6 (2010).

PAPER V:

Ulf Johansson, Tuve Löfström and Henrik Boström, *Overproduce-and-select: The grim reality*, **Computational Intelligence and Ensemble Learning (CIEL), 2013 IEEE Symposium on**, 52-59 (2013).

PAPER VI:

Ulf Johansson, Tuve Löfström and Henrik Boström, *Producing* *implicit diversity in ANN ensembles*, **Neural Networks (IJCNN), The 2012 International Joint Conference on**, 1-8 (2012).

PAPER VII:

Tuve Löfström, Ulf Johansson and Henrik Boström, *Effective utilization of data in inductive conformal prediction using ensembles of neural networks*, **Neural Networks (IJCNN), The 2013 International Joint Conference on**, 1-8 (2013).

PAPER VIII:

Tuve Löfström, Henrik Boström, Henrik Linusson and Ulf Johansson, *Bias Reduction through Conditional Conformal Prediction*, **Intelligent Data Analysis Journal**, vol 19(6), 2015 (in press).

Related Papers

The following papers also contribute to the thesis work but are not included. * indicates equal contribution.

PAPER IX:

Ulf Johansson, Tuve Löfström and Lars Niklasson, *Obtaining accurate neural network ensembles*, **Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on**, 103-108 (2005).

PAPER X:

Ulf Johansson, Tuve Löfström, Rikard König and Lars Niklasson, *Introducing GEMS-A Novel Technique for Ensemble Creation*, **FLAIRS Conference**, 700-705 (2006).

PAPER XI:

Ulf Johansson, Tuve Löfström, Rikard König and Lars Niklasson, *Genetically evolved trees representing ensembles*, **Artificial Intelligence and Soft Computing, International Conference on**, 613-622 (2006).

PAPER XII:

Ulf Johansson, Tuve Löfström, Rikard König and Lars Niklasson, *Accurate Neural Network Ensembles Using Genetic Programming*, **The 23rd Annual Workshop of the Swedish Artificial Intelligence Society**, (2006).

PAPER XIII:

Ulf Johansson, Tuve Löfström, Rikard König and Lars Niklasson, *Building neural network ensembles using genetic programming*, **Neural Networks, 2006. IJCNN 2006. International Joint Conference on**, 1260-1265 (2006).

PAPER XIV:

Ulf Johansson, Tuve Löfström and Lars Niklasson, *Accuracy on a Hold-out Set: The Red Herring of Data Mining*, **The 23rd Annual Workshop of the Swedish Artificial Intelligence Society**, (2006).

PAPER XV:

Ulf Johansson, Tuve Löfström and Lars Niklasson, *The importance of diversity in neural network ensembles-an empirical investigation*, **Neural Networks, 2007. IJCNN 2007. International Joint Conference on**, 661-666 (2007).

PAPER XVI:

Tuve Löfström*, Ulf Johansson* and Lars Niklasson, *Evaluating standard techniques for implicit diversity*, **Advances in Knowledge Discovery and Data Mining**, 592-599 (2008).

PAPER XVII:

Tuve Löfström*, Ulf Johansson* and Henrik Boström, *The Problem with Ranking Ensembles Based on Training or Validation Performance*, **Neural Networks, 2008. IJCNN 2008. International Joint Conference on**, 3222-3228 (2008).

PAPER XVIII:

Tuve Löfström, *Utilizing Diversity and Performance Measures for Ensemble Creation*, **Licentiate Thesis, Örebro University**, (2009).

PAPER XIX:

Ulf Johansson, Tuve Löfström and Ulf Norinder, *Evaluating ensembles on QSAR classification*, **Skövde Workshop on Information Fusion Technologies, SWIFT 2009**, (2009).

PAPER XX:

Tuve Löfström, Ulf Johansson and Henrik Boström, *Using Optimized Optimization Criteria in Ensemble Member Selection*, **Skövde Workshop on Information Fusion Technologies, SWIFT 2009**, (2009).

PAPER XXI:

Ulf Johansson, Cecilia Sönströd, Tuve Löfström and Rikard König, *Using genetic programming to obtain implicit diversity*, **Evolutionary Computation, 2009. CEC'09. IEEE Congress on**, 2454-2459 (2009).

PAPER XXII:

Ulf Johansson, Tuve Löfström and Henrik Boström *Random Brains*, **Neural Networks (IJCNN), The 2013 International Joint Conference on**, 1-8 (2013).

PAPER XXIII:

Ulf Johansson, Rikard König, Tuve Löfström and Henrik Boström *Evolved decision trees as conformal predictors*, **Evolutionary Computation, 2013. CEC'13. IEEE Congress on**, 1794-1801 (2013).

PAPER XXIV:

Ulf Johansson, Henrik Boström and Tuve Löfström, *Conformal Prediction Using Decision Trees*, **International Conference on Data Mining - ICDM**, 330-339 (2013).

PAPER XXV:

Ulf Johansson, Henrik Boström, Tuve Löfström and Henrik Linusson, *Regression conformal prediction with random forests*, **Machine Learning**, 1-22 (2014).

PAPER XXVI:

Henrik Linusson, Ulf Johansson, Henrik Boström, and Tuve Löfström, *Efficiency Comparison of Unstable Transductive and Inductive Conformal Prediction*, **Artificial Intelligence Applications and Innovations Conference - AIAI**, (2014).

Reprints were made with permission from the publishers.

Author's Contribution

In the articles where I am the leading author the main contribution is mine. I have in these cases been in charge of the study, performed most of the experiments and authored the main part of the article. In the articles with equal contribution, all parts of the studies have in a similar way been performed in a close collaboration and with a shared responsibility. In the remaining studies my contribution varies but I have always contributed in a significant way towards the quality of the resulting article. More specifically, I have, in many of the studies I have co-authored, been responsible for the implementation of the algorithms and the execution of the experiments.

Regarding papers V and VI, where I am not the leading author, Ulf Johansson was in charge and decided on the scope of the studies. However, a long series of discussions, primarily between Johansson and me, preceded the studies, in which we discussed different ideas for new studies based on feedback and experience from previous studies. The implementation and execution of the experiments was largely my responsibility. The analysis of the results was done partly in collaboration. So, even if Ulf Johansson was in charge and did much of the writing, both these papers were produced in a rather close collaboration. Papers I-IV and VII are also in a similar way the results of this process, with the difference that I was in charge for these studies.

Contents

1. Introduction

1.1 Background

In many fields of research and in society in general, synergistic advantages may appear when heterogeneous parts are combined, making the whole greater than the sum of the parts. An example of such an advantage is the much more impressive music produced by a musical ensemble, compared to the music each of the musicians could have produced by themselves or when performing in sequence.

In predictive modeling, historical data is used to train models using machine learning algorithms. The data is composed of a number of instances, representing individual entities and each instance is in turn composed of two parts, the object and the target value. A model is trained using data with known target values and the trained model is used to predict the target values for new and unseen instances from the same domain for which only the objects are known. In classification, the target value belongs to a predefined set of class values, but in a regression the target value is a real number.

In general terms, the goal when training a model using a machine learning algorithm is that the model shall perform well when it is applied to new data. Naturally, performance can be measured differently depending on the circumstances. Furthermore, combining the predictions from several models has proven, both theoretically and empirically, to be a successful approach to increasing the performance. Several different terms have been used to denote models that make predictions by combining the predictions from several models, but in this thesis this is referred to as an ensemble or an ensemble model.

To explain why ensembles work better on average than the individual models used by the ensemble, a theorem formulated in the field of political science can be useful. In 1785, the Marquis de Condorcet published an essay where he showed that if the probability p of each of the voters being correct is above 0.5 and the voters are independent, then adding more voters increases the probability that the majority vote will be correct until the probability approaches 1 [1]. The Marquis de Condorcet obviously did not have machine learning in mind when he studied these questions but it is still a similar mechanism as the one he studied that makes ensembles perform well. In fact, Hansen and

Salamon proved that the assumption of de Condorcet's theorem also holds for ensembles [2] (without actually referring directly to de Condorcet's theorem).

It is obvious that the benefits of using ensembles cannot be achieved by simply copying an individual model and combining the copies. For the ensembles to increase the performance over that of the individual models, the individual models must be accurate individually and they need to be sufficiently diverse, in other words, they need to be sufficiently different from each other in terms of which errors are made. The *diversity* requirement reflects the need for independence. However, since the individual models are trained to perform well on the same dataset, it is unrealistic to assume any real independence between the models.

The importance of diversity has been investigated in several studies (see Section 2.2.1 for further details on diversity). Krogh and Vedelsby [3] derived that the error of an ensemble that is used to solve a regression problem is determined by the average performance of the individual models and the average diversity among the models in the ensemble. More specifically, the ensemble error, E, can be derived from

$$E = \overline{E} - \overline{D} \tag{1.1}$$

where the first term, $\overline{E}$, is the average error of the individual models and the second, $\overline{D}$, is the diversity term, i.e., the amount of variability among the ensemble members. From Equation (1.1) it is obvious that the error of the ensemble is guaranteed to be less than or equal to the average error of the individual models. Since the second term (which is never negative) is subtracted from the first to obtain the ensemble error, this decomposition proves that the ensemble will always have at least as high an accuracy as the average accuracy obtained by the individual models.

Naturally, this is a very encouraging result for ensemble approaches. The problem is, however, that the two terms are highly correlated, making it necessary to balance them rather than just maximizing the diversity. When looking at the concept of classification and the situation where the classifier only predicts class labels, the situation is even more complex. For a long time, no equivalent to the natural way of defining diversity used for regression in Equation (1.1) was available. Instead, a number of heuristic diversity measures have been proposed in the literature, cf. [4]. Unfortunately, none of the proposed measures are, by themselves, well correlated with ensemble performance.

Brown et al. introduces a taxonomy of methods for building ensembles of neural networks [5] (see Section 2.3 for further details on ensembles of neural networks). They identify two different approaches used to handle the tradeoff between accurate individual models and diversity between models. The tradeoff can be handled either explicitly, by somehow explicitly optimizing the balance between accuracy and diversity, or implicitly, by simply training the individual models in such a way that they are likely to be both accurate and diverse without explicitly targeting this. Many explicit methods have also been proposed that search among the available classifiers to find an ensemble, consisting of a subset of the classifiers, that is performing better than the ensemble consisting of all available classifiers. The concrete type of explicit learning strategy used when building ensembles in this way is sometimes called the overproduce-and-select paradigm. These techniques often use diversity measures, sometimes together with performance measures, when trying to find the subset of classifiers resulting in an ensemble with optimal performance [6–15]. The overproduce-and-select paradigm can be employed either statically, by identifying a single ensemble to be used for all test instances, or dynamically, by searching for an optimal ensemble for each test instance.

There are also many ensemble techniques that are implicit in nature, usually using bagging (see Section 2.3.4 for further details on bagging). Bagging is very robust and generally produces good results [16]. Random forest [17] is an ensemble technique utilizing bagging together with decision trees. The diversity produced by the use of bagging is complemented by a technique where only a random subset of the features are used when deciding on the best split when building each decision tree. Since individual neural networks are often more accurate than individual decision trees, it lies close at hand to think that ensembles of neural networks would also perform better than ensembles of decision trees. In reality, though, neural networks have a number of training parameters that might affect the performance of an ensemble of neural networks by altering the degree of diversity among the neural networks [5; 18].

By using different statistical evaluation methods it is possible to get an estimate of how well the model will perform when applied to new data (see Section 4.3 for further details on evaluation). The estimation of performance has to be made using some portion of data not used during training, making it necessary to use only a subset of the available information when building the model. However, when using a bagging ensemble, another option presents itself. An estimate for the ensemble can be achieved by taking advantage of the fact that each model is trained using a bag of the instances, leaving a subset of the instances unused when training each model. Since different bags are used to train the different models, it is possible to use each instance as an unbiased evaluation instance for all models for which it was not part of the training set. This is usually referred to as an out-of-bag estimate of the ensemble performance, since the performance of each instance is measured on an ensemble consisting only of the subset of models for which that particular instance was not in the bag used to train the model.

In general, only the overall performance of predictive models is measured, making it possible to know how well the model is likely to perform on average on unseen data. However, in many domains, it is crucial to know with some degree of confidence whether the prediction is correct on an individual instance. Conformal prediction [19] is a relatively new framework for associating classification and regression predictions with reliable confidence estimates (see Chapter 3 for further details on conformal prediction). The user is guaranteed that each prediction is correct with a user specified degree of confidence. The conformal predictions are valid, since the decided level of confidence is guaranteed in the long run. To achieve validity, a tradeoff between certainty regarding the correctness and the amount of information provided by the prediction is introduced. Instead of always making a point prediction, which is usually the case in predictive modeling, the conformal predictor outputs a prediction region. For regression problems, the prediction regions are represented as prediction intervals; for classification, the prediction regions produced are in the form of class label sets, i.e., the set of class labels that are not unlikely to be correct. The conformal prediction framework is applied on top of an ordinary machine learning model, such as an ensemble, and transforms its predictions into prediction regions. The prediction regions produced by conformal predictors are proven to be valid, which means that the probability of making an erroneous prediction is guaranteed to be less than or equal to a predefined significance level ε in the long run: the confidence in such a prediction is $1-\varepsilon$.

One major benefit of the framework is that it is theoretically well grounded. However, there are still many open questions regarding the best practices when applying the framework to different kinds of underlying models and predictive situations. Open questions include how to best take advantage of the available data and how to tune the parameters of different algorithms to achieve as good a performance as possible. Another research direction in which there are many open questions is related to how the prediction regions are affected when the class distribution is more or less imbalanced between classes.

1.2 Problem

As pointed out in the background, algorithms used for predictive modeling must produce models that perform sufficiently well. Both theoretical and empirical research affirms that ensembles will generally be more accurate than single models [2; 3; 16]. Even though it has been shown that diversity is an important factor in explaining why ensembles perform so well, it is still an open question how the tradeoff between the accuracy of the individual models and the diversity among the models should be handled. The tradeoff between performance and diversity can be handled either implicitly or explicitly. Diversity was originally thought to be an important criterion when trying to optimize the ensemble using the explicit learning strategy. For classification where the models predict a single class label, several heuristic diversity measures have been proposed. It has, however, been shown that none of the proposed diversity measures are, by themselves, suitable as optimization criteria, since none of them are well correlated with ensemble performance [4; 20–22]. Even so, several static overproduce-and-select algorithms have been proposed that use diversity and/or performance measures when searching for a smaller ensemble.

When considering algorithms using the implicit learning strategy, there is no need for an optimization criterion. Instead, it is primarily the way the models in the ensemble are trained that will determine the success of the algorithm. Despite the fact that individual neural networks are generally more accurate than individual decision trees, ensembles of neural networks are not guaranteed to outperform ensembles of decision trees. There are a number of parameters that can be varied to create a set of neural networks that, when combined, will result in well performing ensembles and there is still no best practice on how to train neural networks in order to obtain a maximal ensemble performance. Among ensembles of decision trees, random forests [17] have emerged as a de facto standard due to their robustness and good performance. The main reason why random forests perform so well is often attributed to the combination of diversity creating methods, combining randomization of both the instances and the features when training the decision trees. Using bagging, as is done in random forests, also enables using out-of-bag estimates when evaluating or optimizing the ensemble performance.

Conformal prediction [19] is suitable for situations where it is important to know with some degree of certainty if the model is correct on a specific instance. Conformal prediction makes it possible to estimate with some user-defined level of certainty how an ordinary machine learning model will predict a specific instance. Since the level of accuracy is decided by the user, performance is instead measured as efficiency, indicating how informative the predictions are in general. The conformal framework can be used either transductively, making it necessary to train one model for each instance and class label, or inductively: only one model has to be trained when using inductive conformal prediction. However, to ensure its validity, the data available for training must be divided into a proper training set and a set used to calibrate the prediction regions when using inductive conformal prediction. As the framework is relatively new, there are many open questions regarding how to use the framework to make it as efficient as possible. A general question is how to utilize the available data as effectively as possible to maximize its efficiency. A specific question related to bagging ensembles is whether it is possible to use the out-of-bag estimates as a calibration set, making it possible to use all the available data for both training and calibration.

A common issue in many classification problems is that the classes are imbalanced. In most cases it is the minority class which is most important to be able to predict correctly. At the same time, most machine learning techniques will be better at predicting the majority class, making them biased towards that class [23]. When using conformal prediction, the guarantees are for the prediction regions, including all classes. The proportion of errors made on the different classes are not guaranteed to have the same distribution as the prior class distribution. The degree to which conformal predictors are affected by the problem of imbalanced data has not been studied. Conditional conformal prediction is an extension of the conformal prediction framework which makes it possible to condition the guarantees on, e.g., the class labels. When using class label conditional conformal prediction, the conformal predictor is guaranteed to make its errors proportional to the prior class distribution. No comparison between conditional conformal prediction and ordinary conformal prediction regarding efficiency has been conducted. The conditional conformal predictors differ from ordinary conformal predictors in how the data is utilized.

To summarize, there are many different aspects that can be considered in order to achieve good performance when creating ensembles, including such aspects as: whether to optimize the composition of the ensemble; if optimizing, what criterion to optimize; how to train the models to make the ensemble perform as well as possible; how to handle the data in different predictive situations when predicting with confidence.

1.3 Research Question

Based on the problem discussion presented above, the research question of this thesis is: *How can ensembles be created effectively in the context of classification?* The context of classification is vast and there are a large number of possible aspects to take into account when considering how to effectively create ensembles in this context. Since the studies presented in this thesis have focused on some aspects, the main research question is addressed by answering two sub-questions covering the aspects covered by the presented studies:

1. *Which strategy is most effective to use when creating ensembles: the implicit or the explicit learning strategy?*

2. *How should data be utilized effectively in confidence-based predictions using ensembles?*

In this thesis, the main focus regarding the explicit learning strategy is on static overproduce-and-select algorithms. Furthermore, the main focus regarding ensembles in general and the implicit learning strategy in particular is on ensembles of neural networks, even if some studies also involve ensembles of decision trees. Creating ensembles effectively means that the resulting ensembles should be capable of performing well in the tasks they are intended for.

1.4 Contributions

Below, a summary of the content and the contribution of each paper is given.

PAPER I: **Empirically investigating the importance of diversity**
The paper studies the relationship between diversity and accuracy and the use of combinations of diversity and/or performance measures.

This paper contributes to the first sub-question by evaluating 10 diversity measures previously studied using a realistic static overproduce-and-select setup where ensembles of varying sizes were evaluated together. The conclusions support the claim made in previous studies that no individual diversity measure is well correlated with ensemble accuracy. A novel contribution of the paper is that it shows that the correlation between accuracy measured on training or validation data and ensemble accuracy measured on test data is comparable to the correlation measured between the most correlated diversity measures and ensemble accuracy. Furthermore, the idea of using combinations of measures was introduced in this paper. The results achieved when evaluating combined measures indicate that it might be a promising solution, potentially leading to more accurate ensembles.

PAPER II: **On the use of accuracy and diversity measures for evaluating and selecting ensembles of classifiers**
The paper studies the use of combinations of diversity and/or performance measures.

This paper contributes to the first sub-question by empirically evaluating the idea of combining diversity and performance measures. The results could not confirm that combined measures were clearly better than using individual measures. Using only performance measures, either individually or in combination, did not turn out to be better than using diversity measures or combinations of performance or diversity measures. When evaluating ensembles of different sizes, including small ensembles, the average accuracy of the classifiers and the double fault measure turned out to be quite useless since they, by design, always prefer smaller ensembles.

PAPER III: **Ensemble member selection using multiobjective optimization**
The paper studies the use of combinations of diversity and/or performance measures; selection of ensemble members; multi-objective optimization; explicit learning schemes.

This paper presents a novel algorithm for constructing ensembles using a static overproduce-and-select algorithm, which makes this paper contribute to the first sub-question. The technique uses a genetic algorithm to find a combination of measures most suitable to be used as an optimization criterion for the individual dataset. It is shown that it is sometimes better to use the identified optimization criterion than to use ensemble accuracy as the selection criterion. The algorithm worked better for ensembles of neural networks than for ensembles of decision trees.

PAPER IV: **Comparing methods for generating diverse ensembles of artificial neural networks**
The paper studies the selection of ensemble members; implicit and explicit learning schemes.

This paper contributes to the first sub-question by comparing implicit and explicit ensemble creation strategies. The implicit approaches simply trained a number of neural networks, with or without bagging, and included all networks in the ensemble. Two ensemble algorithms using explicit strategies were used as comparison. The empirical study provided strong evidence in favor of the implicit approaches. The two implicit approaches performed equally well. An analysis of the diversity and performance measures of the two implicit approaches revealed that the lower average accuracy achieved when using bagging was compensated for by more diversity and vice versa.

PAPER V: **Overproduce-and-select: The grim reality**
The paper studies the selection of ensemble members; explicit learning schemes.

This paper evaluates the static overproduce-and-select paradigm and consequently contributes to the first sub-question. The main result is that there is absolutely nothing to gain by selecting an ensemble based on any of the metrics evaluated, including ensemble accuracy, average accuracy among the ensemble members, and diversity. Since the ensembles were trained using bagging, out-of-bag estimates were also used in the evaluation. Even though there were many smaller ensembles that were better than using all trained neural networks as an ensemble, there was no way of identifying them by measuring either the training, validation, or out-of-bag data.

PAPER VI: **Producing implicit diversity in ANN ensembles**
The paper studies diversity creation strategies; implicit learning schemes.

This paper contributes to the first research question by further evaluating the implicit approach of simply training neural networks in different ways and then combining all the trained networks. The purpose of the paper was to evaluate how to train neural networks to achieve ensembles performing as well as possible.

PAPER VII: **Effective utilization of data in inductive conformal prediction using ensembles of neural networks**
The paper studies utilization of data for confidence based prediction using ensembles.

In this paper different strategies for how to utilize all available data when using an inductive conformal predictor were compared, making it contribute to the second sub-question. The solution promoted in the paper is to train a bagging ensemble using all the data and to use the out-of-bag estimates as the calibration set. The promoted solution turns out to outperform the other evaluated solutions.

PAPER VIII: **Bias Reduction through Conditional Conformal Prediction**
The paper studies the use of data for confidence based prediction using ensembles.

The contribution of this paper is to the second sub-question since it evaluates the effects of applying conformal prediction and conditional conformal prediction on datasets with varying degrees of class imbalance. The results show that the way the data is used when creating the conformal predictor strongly affects the tendency to be biased towards the majority class. By using the data in such a way as to condition the conformal predictor for each class, the efficiency, measured in terms of the ability to avoid a bias towards the majority class, is far superior compared to conformal predictors using data in the ordinary way.

1.5 Outline of the Thesis

Chapter 2 introduces ensemble learning and presents related work. The chapter starts with an introduction of machine learning techniques that have been used as building blocks in the ensembles evaluated in the included papers. The remainder of the chapter introduces various aspects of ensemble learning and ends with a section presenting work related to the included papers. Chapter 3 introduces the conformal prediction framework and presents related work. The framework is first described in general, followed by details on how it works for classification. After the framework is introduced, the conditional version is presented followed by work related to the included papers. Chapter 4 presents the research approach. First, a theoretical framework is established, followed by a motivation for the kind of experimental setups used in all included papers as well as how to evaluate the results achieved. Chapter 5 presents summaries of the included papers, and Chapter 6, finally, presents a discussion, the conclusions, and some ideas for future research.

2. Ensemble Learning

When performing predictive classification, the primary goal is to obtain good performance. Performance can be measured using a variety of measures. The most common performance measure in predictive classification is accuracy; i.e. the proportion of misclassifications when the model is applied to novel data. Within the machine learning research community, it is well known that it is possible to obtain even higher accuracy by combining several individual models into ensembles; see, e.g., [16; 24]. An ensemble is thus a composite model aggregating multiple base models, and the ensemble prediction, when applied to a novel instance, is therefore a function of all included base models. Ensemble learning, consequently, refers to a large collection of methods that learn a target function by training a number of individual learners and combine their predictions.

This chapter presents basic concepts regarding ensembles. It is to a large degree taken from [25]. Many different terms have been used as synonyms for ensembles; combinations of multiple classifiers [26–29], committees or committee machines [30; 31], mixture of experts [32; 33], composite classifiers [34] and classifier fusion [35; 36] are some of the more frequently used terms. The term employed throughout this thesis is ‘ensemble’ [2; 37].

2.1 Introducing Techniques Related to Ensembles

This chapter will start with presenting the data mining techniques used in the empirical studies. The three sets of techniques presented in this introductory section will be neural networks, decision trees, and evolutionary algorithms.

2.1.1 Neural Networks

The area of (artificial) neural networks (ANNs) has been inspired by the knowledge of how the brain works, i.e., how biological neural networks work. ANNs have become one of the most popular data mining techniques, since this technique is quite powerful and can be used in several different problem domains. We will only describe one sort of ANN, i.e., the multi-layered feed-forward ANN [38], since it is the most suitable architecture for most classification tasks.

A neural network is a collection of connected neurons. Each neuron has three basic elements.

1. A set of *connected links*, each of which has a *weight* of its own. A signal x_j at the input of the link j connected to neuron k is multiplied by the link weight w_{jk}. The weight in a neural network link may lie in a range that includes negative as well as positive values, while the corresponding element in the brain, a synapse, outputs a signal of varying strength.

2. An *adder* for summing the input signals, weighted by their respective links to the neuron.

3. An *activation function* for limiting the amplitude of the output of a neuron. Typically, the normalized amplitude range of the output of a neuron is the closed unit interval [0, 1] or alternatively [-1, 1].

Often a *bias* is also applied to the neuron. The bias b_k has the effect of increasing or lowering the net input of the activation function, depending on whether it is positive or negative.

A neuron may mathematically be described by the following pair of equations:

$$u_k = \sum_{j=1}^{m} w_{jk} x_j \tag{2.1}$$

$$y_k = f(u_k + b_k) \tag{2.2}$$

where $x_1, x_2, ..., x_m$ are the input signals; $w_{k1}, w_{k2}, ..., w_{km}$ are the synaptic weights of neuron k; u_k is the linear combiner output for the input signals; b_k is the bias; $f(\cdot)$ is the activation function; and y_k is the output signal of the neuron.

Feed-Forward Neural Networks

In a layered ANN, the neurons are organized in the form of layers. The input signals propagate through the network in a forward direction, on a layer-by-layer basis. In the simplest form there is only an input layer of source nodes that projects onto an output layer of neurons, but not vice versa. These simple networks are called single-layered feed-forward neural networks, or single-layered perceptrons. The input layer is typically not counted, since no computation is performed there. Single-layered networks can only represent linear functions. Multilayered feed-forward ANNs, or multilayered perceptrons (MLPs), are ANNs with at least one layer of neurons between the input and the output layer. The layer(s) between the input and the output layer is called a

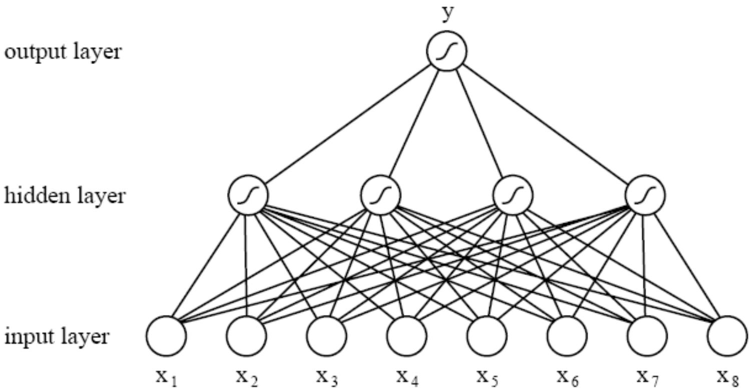

Figure 2.1: A neural network

hidden layer(s). By adding hidden layers, the network is enabled to represent higher-order (nonlinear) functions.

As can be seen in Figure 2.1, each layer has one or more neurons. The input layer in a feed-forward ANN has as many input nodes as input variables. The number of neurons in the hidden layer(s) varies. This number affects the network's ability to adjust its interior state to match the patterns in the training data. More hidden neurons make the network more capable of learning details. This might result in overfitting of the training data, i.e., learning details about the training data that are not general patterns of the problem, and as a consequence it might not generalize well to unseen data. General patterns are general for the entire problem space and not specific only to the training set. Too few hidden neurons might lead to a network's being unable to learn all the general patterns in the training data, and thus the network will not be powerful enough.

The activation function in an MLP should be nonlinear and smooth, i.e., differentiable everywhere. A commonly used form of nonlinearity that satisfies this property is a sigmoidal nonlinearity defined by the logistic function. The MLP is usually trained with an algorithm called error back-propagation [39; 40]. The training of a network is done by iterating through the training data many times and adjusting the weights a little bit on each iteration. The back-propagation algorithm consists of two passes through the different layers of the network on each iteration: a forward and a backward pass. In the forward pass, an input pattern is propagated through the network. An output is produced as the actual response of the network. During the forward pass the weights of the connected links are all fixed. During the backward pass, on the other hand, the weights of the connected links are all adjusted in accordance with an error-correction rule. The response of the network is subtracted from the desired response, i.e. the actual target response, to produce an error signal. This error signal is then propagated back through the network, reversing the forward pass. The weights of the connected links are adjusted to make the response of the network closer to the desired response in a statistical sense. This procedure is repeated multiple times. Each repetition is often referred to as an epoch.

The back-propagation algorithm has no well-defined criteria for stopping the training. Different stopping criteria can be used. They all have the drawback that the algorithm might stop at a local minimum of the error surface, i.e., the resulting model might not be the best possible. For further details on ANNs see any introductory book on Neural Networks, e.g., [41].

2.1.2 Decision Trees

Decision tree learning is a predictive modeling technique most often used for classification. Decision trees partition the input space into cells, where each cell belongs to one class. The partitioning is represented as a sequence of tests. Each interior node in the decision tree corresponds to one test of the value of some input variable, and the branches from the node are labeled with the possible results of the test. The leaf nodes represent the cells and specify the class to return if that leaf node is reached. The classification of a specific instance is thus performed by starting at the root node and, depending on the results of the test, following the appropriate branches until a leaf node is reached.

The decision tree is created from examples (the training set) with the obvious requirement that it should agree with the training set. The basic strategy for building the tree is to recursively split the cells of the input space. To choose the variable and threshold at which to split, a search over possible input variables and thresholds is performed to find the split that leads to the greatest improvement of a specified score function. Typically this score function is based on some information theory measurement, like information gain or entropy. The overall idea is to minimize the size of the final tree by always choosing splits that make the most difference to the classification of an instance. The splitting procedure could in principle be repeated until each cell contains instances from one class only. At the same time the decision tree must not simply memorize the training set, but should be capable of generalizing to unseen data; i.e. the decision tree should not overfit. The goal is thus to have a decision tree as simple (small) as possible, but still representing the training set well.

Two basic strategies for avoiding overfitting are to stop the growth of the tree when some criterion has been met, or to afterwards reduce (prune) a large tree by iteratively merging leaf nodes.

Classification and regression trees (CART) [42] is a technique that generates binary decision trees. Each internal node in the tree specifies a binary test on a single variable, using thresholds on real and integer-valued variables and subset membership for categorical variables. The Gini coefficient is used as a measure for choosing the best splitting attribute and criterion. The Gini coefficient is a measure of statistical dispersion. It is defined as a ratio with values between 0 and 1; A low Gini coefficient indicates a more equal distribution, while a high Gini coefficient indicates a more unequal distribution. The splitting is performed around what is determined to be the best split point. At each step, an exhaustive search is used to determine the best split. For details about the function used to determine the best split, see the book introducing the algorithm. The score function used by CART is the misclassification rate on an internal validation set. CART handles missing data by ignoring the missing value when calculating the goodness of a split on that attribute. The tree stops growing when no split will improve the performance.

2.1.3 Evolutionary Techniques

Like ANNs, evolutionary techniques are based on an analogy to biological processes. The theory of evolution stands as the model for evolutionary techniques such as genetic algorithms (GA) [43]. Evolution optimizes the fitness of individuals over succeeding generations by propagating the genetic material in the fittest individuals of one generation to the next generation. The core in evolutionary techniques consists of three stages:

1. A population of potential problem solutions (individuals) is encoded into a representation that is specific to the type of evolutionary technique used.

2. The fitness of each individual solution is measured to rank the solutions. The highest ranked individuals are favored in the shaping of the next generation of solutions.

3. A new population for the next generation is formed by reproduction and survival of individual solutions. Mating of individuals (called *crossover*) recombines the individuals from the parent generation to form the individuals of the next generation. *Mutation* is also used to introduce new genetic material into the population by randomly changing an individual.

The representation of solutions differs between evolutionary techniques, making it necessary to have distinct mating and mutation operations adjusted to the particular representation. On the other hand, the strategies used for selecting individuals to whom to apply these operations are the same. In the *roulette* *wheel selection* strategy, an individual's probability to be selected for mating is proportional to its fitness. In *tournament selection*, a number of individuals are drawn at random and the best among them is selected. This is repeated until the required number of individuals has been selected for reproduction and survival. The tournament selection strategy only considers whether a solution is better than another, not how much better. This prohibits an extraordinarily good individual from swamping the next generation with its children, which would lead to a disastrous reduction of diversity in the population.

The fitness function is the measure that should be optimized, and is sometimes referred to as the objective of the optimization. Any measure that can be used to score individual solutions based on performance could be a fitness function. More than one fitness function can be used simultaneously, this is often referred to as multi-objective evolutionary optimization. When more than one fitness function is used, the result is not a single best solution but rather a set of best solutions.

For further details, see any introductory book on evolutionary techniques, e.g., [44].

Genetic Algorithms

The representation used in a GA employs character strings, most often bit strings. The crossover and mutation operations are used to produce new individuals by using parts of their parents.

In crossover, two parent individuals are selected, and they are divided at one or many randomly chosen point(s). When only one division point is used, one part of each parent is kept, and is joined with the remaining part of the other parent. If multiple division points are used, then some parts from each parent are kept, while the remaining parts are switched. If crossover does not take place, then the parents are cloned to the next generation, i.e. they are transferred intact to the next generation.

To avoid having important parts eliminated from the entire population for good, or the search stagnating in a local minimum, mutation is used as a means of reintroducing randomly generated parts to the population. Mutation takes one parent and changes some part randomly. For bit string representations, the mutation most often means flipping any bit from 0 to 1, or vice versa. The theoretical foundation for GA is the schema theorem, formulated by Holland [43]. In short, the theorem states that more important parts of the individuals, i.e. parts contributing positively to the fitness function, are more likely to survive to the next generation. The search is thus a search through schemes (parts of individuals), rather than through complete solutions. By searching for good parts, rather than good solutions, the search becomes exponentially more efficient.

Multi-Objective Optimization

Evolutionary techniques can optimize single objectives or multiple objectives. The goal of multi-objective optimization (MOO) is to find solutions that are optimal, or at least acceptable, according to all criteria simultaneously. MOO can be performed in all kinds of evolutionary techniques.

Combining multiple objectives into a scalar fitness function is the most primitive form of MOO. The simplest form of this combination is a (weighted) linear combination of the different objectives.

The obvious alternative to combining the different objectives into a scalar fitness function is keeping the objectives apart. The main motivation for keeping the objectives apart is to encourage diversity among the solutions. When the objectives are kept apart, the selection strategies are affected. The main idea in MOO is the notion of *Pareto dominance*. A solution a_i is non-dominated iff there is no other alternative $a_j \in S, j \neq i$ such that a_j is better than a_i on all criteria. Or, expressing the opposite relation less formally, a solution is said to Pareto dominate another if it is as good as the second on all objectives and better on at least one objective. This results in a partial ordering, where several solutions can be non-dominated, and thus constitute the set of best solutions for the particular set of objectives. The set of *all* non-dominated solutions in the search space is called the *Pareto front*, or the *Pareto optimal set*. It is often unrealistic to expect to find the complete Pareto front, since its size is often limited only by the precision of the problem representation. [44]

2.2 Basic Ensemble Concepts

An ensemble is basically constructed by training a set of L models, henceforth called base classifiers, on L data sets and combining these models. The data sets are often either identical or highly overlapping subsets drawn from a single data source, but they can just as well be entirely different data sets gathered from different data sources, capturing different aspects of the problem. To predict the target value for a new instance, the target value of the combined model is calculated, often by applying each base classifier in turn and combining their outputs.

The most intuitive explanation for why ensembles work is probably given by Condorcet's jury theorem [1]. The assumption of the theorem is that a group wishes to reach a decision by majority vote. The outcome of the vote could be either correct or incorrect, and each voter has an independent probability p of voting for the correct decision. The number of voters to include depends on whether p is greater than or less than 0.5. If $p > 0.5$ (each voter is more likely than not to vote correctly), then adding more voters increases the probability that the majority decision is correct. In the limit, the probability that the majority votes correctly approaches 1 as the number of voters increases. The output type from each classifier in the ensemble could be distinguished in four different ways [45]:

- *The oracle level*: The only information considered is whether the classifier is correct or incorrect in its prediction for each instance. This is the type of output containing the least information. There is no information on the actual prediction made by the classifier, only if it is right or wrong. This level is useful primarily for analytical purposes. Most of the diversity measures presented below can be defined using the oracle level.

- *The abstract level*: The classifier outputs the label of the predicted class for each instance. There is no information on the certainty of the prediction.

- *The rank level*: The possible classes are ranked in order of plausibility. This kind of output is especially suitable for problems with a large number of classes.

- *The measurement level*: The output containing most information is when the classifier outputs a measure of certainty about its prediction for each class. For each instance, the classifier will produce a vector of measures of certainties, one measure for each class.

It must be noted that outputs at each level can always be reduced to fit the preceding levels, apart from the oracle level, i.e. any model producing measurements can also produce ranked and labeled output, and so on. However, to produce the oracle level output, the ground truth, i.e. the correct labels, must be known.

2.2.1 Diversity

Naturally, there is nothing to gain by combining identical models, doing exactly the same things. Consequently, the base classifiers must commit their errors on different instances, which is the informal meaning of the key term diversity. Krogh and Vedelsby [3] derived the result that ensemble error depends not only on the average error of the base models[1], but also on their diversity[2]. More formally, the ensemble error, E, is

$$E = \overline{E} - \overline{A} \tag{2.3}$$

where $\overline{E}$ is the average error of the base models and $\overline{A}$ is the ensemble diversity (or ambiguity), measured as the weighted average of the squared differences in the predictions of the base models and the ensemble. In a regression context and using averaging, this is equivalent to

$$E = (\hat{Y}_{ens} - Y)^2 = \frac{1}{L}\sum_i (\hat{Y}_i - Y)^2 - \frac{1}{L}\sum_i (\hat{Y}_i - \hat{Y}_{ens})^2 \tag{2.4}$$

where the first term is the (possibly weighted) average of the individual models and the second is the diversity term; i.e. the amount of variability among ensemble members. The diversity term is always positive, proving that the ensemble will always have higher accuracy than the average accuracy obtained by the individual models. Based on this, the overall goal of getting low ensemble error could be divided into the two sub-goals of combining models that commit few errors, but at the same time differ in their predictions. The two terms are, however, normally highly correlated, making it necessary to balance them instead of just maximizing the diversity term.

By relating this to the bias–variance decomposition and assuming that the ensemble is a convex combined ensemble (e.g. using averaging), a bias–variance–covariance decomposition can be obtained for the ensemble MSE; see 2.5 below.

$$E = (\hat{Y}_{ens} - Y)^2 = \overline{bias}^2 + \frac{1}{L}\overline{var} + \left(1 - \frac{1}{L}\right)\overline{covar} \tag{2.5}$$

From this it is evident that the error of the ensemble depends critically on the amount of correlation between models, quantified in the covariance term. Ideally, the covariance should be minimized, without causing negative changes that result in increases of the bias or variance terms.

However, unless classification is handled like an instance of regression (i.e. the outputs are at the measurement level) the framework described above does not apply for ensembles of classifiers. When predictors are only able to output a class label, the outputs have no intrinsic ordinality between them, thus making the concept of covariance undefined. Using a zero–one loss function, there is no clear analogy to the bias–variance–covariance decomposition.

[1] The theory was formulated for regression problems. Consequently, the term base models is more appropriate than the term base classifiers in this case.

[2] Krogh and Vedelsby used the term ambiguity instead of diversity in their paper. In this thesis, the more common term diversity is, however, used exclusively.

Brown and Kuncheva [46] have made a decomposition of the ensemble error when majority vote is used. Instead of achieving simply one diversity term, as is the case when dealing with measurement output, the decomposition results in two diversity terms. This decomposition will be presented in Section 2.2.1

Before Brown and Kuncheva provided the decomposition, obtaining an expression where the classification error is decomposed into error rates of the individual classifiers and a diversity term was beyond the state of the art. Instead, methods typically used heuristic expressions that tried to approximate the unknown diversity term. Naturally, the goal was to find a diversity measure correlating well with majority vote accuracy.

Because of this, there exist several suggested diversity measures for a classification context. The presentation of the different diversity measures follows Kuncheva and Whitaker [4] and Stapenhurst [47] closely. Most of the diversity measures presented below are defined using the oracle output, and can thus be applied to any type of base classifier.

The set of instances is denoted $\boldsymbol{Z}$, which is the Cartesian product $\boldsymbol{X} \times \boldsymbol{Y}$ of the independent variables $\boldsymbol{X}$, henceforth called the object space, and the dependent variable $\boldsymbol{Y}$. Consequently, each example $z \in \boldsymbol{Z}$ consists of two parts: $z = (x, y)$, where $x \in \boldsymbol{X}$ is the object and $y \in \boldsymbol{Y}$ is the dependent variable. In classification, $\boldsymbol{Y}$ is a finite set usually referred to as the class variable, and in regression, $\boldsymbol{Y}$ is the real line $\mathbb{R}$. If nothing else is mentioned, we can assume that $\boldsymbol{Y} = \{1, -1\}$ in the presentation below.

Let us consider a set $z_1, ..., z_N$ of training instances, where N is the number of available instances and $z_i = (x_i, y_i) \in \boldsymbol{Z}$. The ensemble has L base models, $h_1, ..., h_L$, which produces predictions $h_l(x) \in \boldsymbol{Y}$. The ensemble prediction is an unweighted majority vote,

$$H(x) = \max_{c \in \boldsymbol{Y}} \left(\sum_{l=1}^{L} I\left[h_l(x) = c\right] \right) = \text{sign}\left(\frac{1}{L} \sum_{l=1}^{L} h_l(x) \right)$$

where I is the indicator function. The number of models that correctly classifies an instance is $c_i = \frac{1}{L} \sum_{l=1}^{L} I\left[h_l(x_i) = y_i\right]$. The proportion of models that correctly classifies an instance is $p_i = \frac{1}{L} c_i$ and its average is $\overline{p} = \frac{1}{N} \sum_{i=1}^{N} p_i$.

For many of the pairwise measures, the following notation is used.

$$\begin{aligned}
N^{11} &= \sum_{i=1}^{N} I\left[h_j(x_i) = y_i \wedge h_k(x_i) = y_i\right] \\
N^{10} &= \sum_{i=1}^{N} I\left[h_j(x_i) = y_i \wedge h_k(x_i) \neq y_i\right] \\
N^{01} &= \sum_{i=1}^{N} I\left[h_j(x_i) \neq y_i \wedge h_k(x_i) = y_i\right] \\
N^{00} &= \sum_{i=1}^{N} I\left[h_j(x_i) \neq y_i \wedge h_k(x_i) \neq y_i\right] \qquad (2.6)
\end{aligned}$$

Here, N^{11} measures the number of instances on which the two models h_j and h_k are correct, N^{10} those where h_j is correct and h_k is incorrect, N^{01} those where h_j is incorrect and h_k is correct, and N^{00} those where both h_j and h_k are incorrect, respectively. In cases where more than two classes exist, two more measures can be defined, counting the cases when both models are incorrect but predicting either the same or different classes.

$$\begin{aligned}
N^{00}_{\text{same}} &= \sum_{i=1}^{N} I\left[h_j(x_i) \neq y_i \wedge h_k(x_i) \neq y_i \wedge h_j(x_i) = h_k(x_i)\right] \\
N^{00}_{\text{different}} &= \sum_{i=1}^{N} I\left[h_j(x_i) \neq y_i \wedge h_k(x_i) \neq y_i \wedge h_j(x_i) \neq h_k(x_i)\right] \qquad (2.7)
\end{aligned}$$

For a pairwise measure $diversity_{j,k}$ between the models h_j and h_k, the overall diversity of the ensemble is defined as the average over all pairs:

$$diversity = \frac{2}{L(L-1)} \sum_{j=1}^{L-1} \sum_{k=j+1}^{L} diversity_{j,k} \qquad (2.8)$$

There are, in a similar way, some diversity measures that are defined in a meaningful way for individual instances. In these cases, the $diversity_i$ for the ith instance can be averaged over all instances:

$$diversity = \frac{1}{N} \sum_{i=1}^{N} diversity_i \qquad (2.9)$$

Pairwise Measures

The first measure, Yule's Q statistic [48] varies between -1 and 1. If the classifiers commit their errors independently, Q will be negative. Q is, for two classifiers, h_j and h_k,

$$\mathrm{Q}_{j,k} = \frac{N^{11}N^{00} - N^{10}N^{01}}{N^{11}N^{00} + N^{10}N^{01}} \tag{2.10}$$

Pearson's correlation coefficient (ρ) between h_j and h_k is

$$\rho_{j,k} = \frac{N^{11}N^{00} - N^{10}N^{01}}{\sqrt{(N^{11}+N^{10})(N^{11}+N^{01})(N^{00}+N^{10})(N^{00}+N^{01})}} \tag{2.11}$$

For any two classifiers, Q and ρ have the same sign. The disagreement measure [49] is the ratio between the number of instances for which one classifier is correct and the other incorrect and the total number of instances:

$$\mathrm{D}_{j,k} = \frac{N^{01}+N^{10}}{N^{11}+N^{00}+N^{10}+N^{01}} = \frac{N^{01}+N^{10}}{N} \tag{2.12}$$

The *double-fault* measure [8; 50] is the proportion of instances misclassified by both: classifiers

$$\mathrm{DF}_{j,k} = \frac{N^{00}}{N^{11}+N^{00}+N^{10}+N^{01}} = \frac{N^{00}}{N} \tag{2.13}$$

The pairwise inter-rater agreement (the κ coefficient) [51] is defined as

$$\kappa_{j,k} = \frac{2(N^{11}N^{00} - N^{10}N^{01})}{(N^{11}+N^{10})(N^{11}+N^{01})(N^{00}+N^{10})(N^{00}+N^{01})} \tag{2.14}$$

For all pairwise measures, the averaged value over the diversity matrix is calculated using 2.8.

Non-Pairwise Measures

The *entropy* measure E defined by Kuncheva and Whitaker [4], varying between 0 and 1 (highest possible diversity), is

$$E = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{(L-\frac{L+1}{2})}\min\{c_i, L-c_i\} \tag{2.15}$$

Another *entropy* measure H is defined by Cunningham [52]:

$$H = -\frac{1}{N}\sum_{i=1}^{N} P(y_i=1|x_i)\log P(y_i=1|x_i) + P(y_i=-1|x_i)\log P(y_i=-1|x_i) \tag{2.16}$$

The *Kohavi–Wolpert* variance [53] can be used to obtain another diversity measure KW, which turns out to differ from the averaged disagreement measure only by a coefficient: for details see [4].

$$\mathrm{KW} = \frac{1}{NL^2}\sum_{i=1}^{N} l(z_i)\,(L - l(z_i)) \tag{2.17}$$

The *inter-rater agreement* (the κ coefficient) [37; 51] is

$$\kappa = 1 - \frac{c_i(L-c_i)}{L(L-1)\overline{p}(1-\overline{p})} \tag{2.18}$$

The *difficulty* measure was used by Hansen and Salomon [2]. Let X be a random variable taking values in $0/L, 1/L, ..., 1$. Then X is defined as the proportion of classifiers that correctly classify an instance x drawn randomly from the data set. To estimate X, all L classifiers are run on the data set. The difficulty is then defined as the variance of X. Using the formalization used in [47] the measure is

$$\mathrm{diff}_i = \frac{c_i^2 - L(1-\overline{p})^2}{L} \tag{2.19}$$

The *generalized diversity* measure was proposed by Partridge and Krzanowski [54].

$$\mathrm{GD} = 1 - \frac{p(2)}{p(1)} \tag{2.20}$$

where $p(1) = \sum_{l=1}^{L} \frac{l}{L} p_l$ and $p(2) = \sum_{l=1}^{L} \frac{l}{L} \frac{(l-1)}{L(L-1)} p_l$

Here, B is a random variable expressing the proportion of classifiers that are incorrect on a randomly drawn instance, p_l is the probability that $B = l/L$, and $p(l)$ is the probability that l randomly chosen classifiers will fail on a randomly chosen instance. GD varies between 0 (minimum diversity) and 1.

The *coincident failure diversity* is a modification of GD, also presented in Partridge and Krzanowski [54].

$$CFD = \begin{cases} 0, & p_0 = 1.0 \\ \frac{1}{1-P_0}\sum_{l=1}^{L} \frac{L-l}{L-1} p_l, & p_0 < 1.0 \end{cases} \tag{2.21}$$

Tsymbal et al. [55] presents a diversity measure which they refer to as ambiguity:

$$amb_i = \frac{1}{L}\sum_{l=1}^{L}\left(I\left[h_l(x_i) = 1\right] - \frac{1}{2}(1 + y_i m_i)\right)^2 + \left(I\left[h_l(x_i) = -1\right] - \frac{1}{2}(1 - y_i m_i)\right)^2 \tag{2.22}$$

Another measure was used in [7; 11] and is defined as

$$d = \frac{1}{L}\sum_{l=1}^{L}\frac{1}{N}\sum_{i=1}^{N} I\left[h_l(x_i) \neq H(x_i)\right] \tag{2.23}$$

Finally, a diversity measure was defined in the dissertation by Chen [56]:

$$A = \frac{1}{2N}\sum_{i=1}^{N}\sum_{l=1}^{L}\left(\frac{1}{L}H(x_i) - w_l h_l(x_i)\right) y_i \tag{2.24}$$

where w_l is a model specific weight that can be used when the different models are weighted differently. Chen showed that this diversity measure was more correlated with test set accuracy than other diversity measures. He also used this measure to propose several regularized negative correlation learning algorithms suitable for classification.

Diversity of Errors

The diversity measures defined above do not distinguish between a diversity achieved on instances where the ensemble is correct and a diversity achieved on instances where the ensemble is wrong. A special situation arises when trying to predict multiclass problems, since diversity can also be measured among the different classes. Based on this, three additional diversity measures, referred to as measures for diversity of errors, are presented in [57], which measure the diversity among classes when the ensemble is incorrect.

The distinct failure measure (DFD) [58] focuses on cases where incorrect predictions are coincident but distinct, i.e. resulting in different erroneous outputs. Let

$$t_n = \frac{\text{number of times that } n \text{ classifiers fail identically}}{\text{total number of times a classifier fails}}.$$

The DFD is defined as

$$DFD = \sum_{n=1}^{N}\frac{N-n}{N-1}t_n \tag{2.25}$$

When no errors are made, DFD is defined to be 1. A higher DFD indicates more diversity.

The same fault (SF) measure is a variant of the double fault measure. Instead of only calculating the proportion of instances misclassified by both classifiers, the same fault measure calculates the proportion of instances misclassified as the same class by both classifiers. For a pair of classifiers, *i* and *k*, the measure is defined as

$$SF_{i,k} = \frac{N_{same}^{00}}{N} \tag{2.26}$$

A lower SF indicates more diversity.

The weighted count of errors and correct results measure (WCEC) is also a pairwise measure. The measure includes both correct and incorrect predictions and uses a weight to punish pairs that make the same mistake more than pairs that make different mistakes. For a pair of classifiers, *i* and *k*, the measure is defined as

$$WCEC_{i,k} = N^{11} + \frac{1}{2}\left(N^{01} + N^{10}\right) - N^{00}_{different} - 5N^{00}_{same}. \tag{2.27}$$

The weighting is arbitrary and is based on the idea of especially penalizing identical errors.

Majority Vote Decomposition of Ensemble Error

The same measure (without the weight) as defined in Equation (2.24) was used by Brown and Kuncheva in [46], where it was used in a decomposition of the ensemble error into the individual error and this measure. The decomposition further divides this measure into a constructive and destructive part. Brown and Kuncheva refer to these parts as ‘good’ and ‘bad’ diversity.

The decomposition as given in [46] is provided below. In the case of a binary problem, where $\boldsymbol{Y} = \{1, -1\}$, the zero–one error of the individual model on an instance $z = (x, y)$ is

$$e_l(x) = \begin{cases} 0, y = h_l(x) \\ 1, y \neq h_l(x) \end{cases} = \frac{1}{2}(1 - yh_l(x)) \tag{2.28}$$

The zero–one error of the ensemble using majority vote is

$$e_{\text{maj}}(x) = \frac{1}{2}(1 - yH(x)) \tag{2.29}$$

The disagreement between model h_l and the ensemble H is defined as

$$d_l(x) = \frac{1}{2}(1 - h_l(x)H(x)) \tag{2.30}$$

Since $H(x) \in \{1, -1\}$, making it possible to write $\frac{h_l(x)}{H(x)} = h_l(x)H(x)$, the difference between the ensemble error and the average individual error, e_{ind}, is

$$\begin{aligned}
\Delta &= e_{\text{maj}}(x) - e_{\text{ind}}(x) \\
&= \frac{1}{2}(1 - yH(x)) - \frac{1}{L}\sum_{l=1}^{L}\frac{1}{2}(1 - yh_l(x)) && (2.31) \\
&= \frac{1}{2} - \frac{1}{2}yH(x) - \frac{1}{2} + \frac{1}{2L}\sum_{l=1}^{L} yh_l(x) && (2.32) \\
&= -yH(x)\frac{1}{L}\sum_{l=1}^{L}\frac{1}{2}\left(1 - \frac{h_l(x)}{H(x)}\right) && (2.33) \\
&= -yH(x)\frac{1}{L}\sum_{l=1}^{L}\frac{1}{2}\left(1 - h_l(x)H(x)\right) && (2.34) \\
&= -yH(x)\frac{1}{L}\sum_{l=1}^{L} d_l(x) && (2.35)
\end{aligned}$$

Consequently, the ensemble error can be shown to be composed of the average individual error and Δ:

$$e_{\text{maj}}(x) = e_{\text{ind}}(x) - yH(x)\frac{1}{L}\sum_{l=1}^{L} d_l(x) \qquad (2.36)$$

The decomposition essentially shows that a lower average accuracy of individual models can be compensated for by a higher disagreement with the ensemble as long as the ensemble is correct [59].

One important difference between the decomposition of ensemble error for regression and classification is that the diversity term in the classification includes the class label of the instance. The above equations calculated the zero–one error of a single instance. To calculate the majority vote error over all the instances, E_{maj}, taking advantage of the fact that $yH(x) = 1$ when correct and $yH(x) = -1$ when incorrect, the integration with respect to the probability density function becomes

$$\begin{aligned}
E_{\text{maj}} &= \int_x e_{\text{ind}}(x) - \int_x yH(x)\frac{1}{L}\sum_{l=1}^{L} d_l(x) && (2.37) \\
&= \int_x e_{\text{ind}}(x) - \underbrace{\int_{x+}\frac{1}{L}\sum_{l=1}^{L} d_l(x)}_{\text{good diversity}} + \underbrace{\int_{x-}\frac{1}{L}\sum_{l=1}^{L} d_l(x)}_{\text{bad diversity}} && (2.38)
\end{aligned}$$

where $x+$ refers to instances on which the ensemble is correct and $x-$ refers to instances on which the ensemble is wrong.

Consequently, increasing the disagreement is beneficial for instances where the ensemble is correct and detrimental for instances where the ensemble is wrong, hence the labels ‘good’ and ‘bad’ diversity.

In [59], the decomposition in Equation (2.36) is generalized to more than two classes. The generalized decomposition is

$$e_{\text{maj}}(x) = e_{\text{ind}}(x) - \frac{1}{L}\sum_{l=1}^{L}\left(I\left[H(x)=y\right] - I\left[h_l(x)=y\right]\right) \tag{2.39}$$

For binary problems, the right hand side of Equation (2.36) can be considered to be a diversity measure. In the general case, when $|\boldsymbol{Y}| > 2$, the right hand side of Equation (2.39) must be used instead. For $|\boldsymbol{Y}| > 2$, the disagreement is no longer expressed in terms of class labels but in terms of correctness (which coincides for $|\boldsymbol{Y}| = 2$, as shown above).

2.2.2 Diversity and Margins

The connection between diversity and the concept of margin has been discussed in several papers [22; 47; 60]. In his thesis, Stapenhurst analyzes diversity in terms of margin theory [60]. He shows that many of the diversity measures presented above can be defined using the margin as well. The margin of an ensemble where $\boldsymbol{Y} = \{1, -1\}$ is

$$m(x,y) = \frac{1}{L}\sum_{l=1}^{L} y h_l(x) \tag{2.40}$$

where the following shorthand, $m_i = m(x_i, y_i)$, is used when referring to a specific instance. The averaged margin over all training data is $\overline{m} = \frac{1}{N}\sum_{i=1}^{N} m_i$. The margin equivalent of the diversity measures are presented in Table 2.1.

Translating the diversity into the terminology of a margin makes it possible to analyze diversity from a perspective that has been studied extensively in other contexts. A discussion of the findings by Stapenhurst is provided in Section 2.4 on related work.

2.3 Ensemble Creation

The process of building an ensemble could be said to consist of three stages [22]. In the first stage, a set of base classifiers is generated. The selection of base classifiers to include is performed in the second stage. The third stage consists of combining the selected base classifiers using an appropriate combination strategy. These stages do not have to be performed sequentially. In fact,

Measure	Margin Interpretation
Q statistics	None
Pearson's correlation coefficient (ρ)	None
Disagreement	$D_i = \frac{L}{2(L-1)}(1 - m_i^2)$
Double Fault	$DF_i = \frac{1}{2}(1 - m_i) - \frac{L}{4(L-1)}(1 - m_i^2)$
Pairwise inter-rater agreement (κ coefficient)	None
Entropy E (Kuncheva)	$E_i = \frac{L}{L-1}(1 - \lvert m_i \rvert)$
Entropy H (Cunningham)	$H_i \approx \frac{5}{8}(1 - m_i^2)$
Kohavi–Wolpert variance	$KW_i = \frac{1}{4}(1 - m_i^2)$
Inter-rater agreement (κ coefficient)	$\kappa_i = 1 - \frac{L}{L-1}\left(\frac{1-m_i^2}{1-\overline{m}^2}\right)$
Difficulty	$diff_i = \frac{L}{4}(m_i^2 - \overline{m}^2)$
Generalized Diversity	$GD_i = \frac{L}{L-1}\left(\frac{1-m_i^2}{2(1-\overline{m})}\right)$
Coincident failure diversity	$CFD_i = \frac{L}{L-1}\left(1 - \frac{1-m_i}{2(1-p_0)}\right)$
Ambiguity (Tsymbal)	$Amb_i = 1 - m_i^2$
Ambiguity (Zenobi/Melville)	$d_i = \frac{1}{2}(1 - \lvert m_i \rvert)$
Ambiguity (Chen/Brown)	$A_i = -y_i H(x_i)\frac{1}{2}(1 - \lvert m_i \rvert)$

Table 2.1: Diversity Measures Interpreted using Margins

many ensemble creation algorithms execute these stages iteratively or even in parallel.

The focus of the first stage is to generate a diverse set of base classifiers. Diversity can be achieved either by explicitly seeking to maximize diversity or in an implicit way. A method is said to use implicit diversity whenever diversity is not achieved by explicitly altering the parameters of the algorithm based on the intermediate results.

Brown et al. [5] introduced a taxonomy of methods for creating diversity. The first obvious distinction made is between explicit methods, where some metric of diversity is directly optimized, and implicit methods, where the method is likely to produce diversity without actually targeting it. The different methods for producing and using diversity were divided into three categories: *starting point in hypothesis space*, *set of accessible hypotheses*, and *traversal of hypothesis space*. These categories, and how they apply to ANN ensembles, are further described below.

2.3.1 Starting Point in Hypothesis Space

For ANNs, the most obvious *starting point in hypothesis space* method is to simply randomize the starting weights; something that must be considered a standard procedure for all ANN training. Alternatively, the weights could be placed in different parts of the hypothesis space. Unfortunately, experimentation has found that ANNs often converge to the same, or very similar optima, in spite of starting in different parts of the space; see e.g. [61]. Thus, according to Brown et al. [5], varying the initial weights of ANNs does not seem to be an effective stand-alone method for generating diversity.

2.3.2 Set of Accessible Hypotheses

The two main principles regarding *set of accessible hypotheses* are to manipulate either the training data or the architecture. Several methods attempt to produce diversity by supplying each classifier with a slightly different training set. Regarding resampling, the view is that it is more effective to divide the training data by feature than by instance; see [62]. All standard resampling techniques are by nature implicit.

According to Brown et al. [5], the effect of only differentiating the number of units in each layer is very limited. Hybrid ensembles where, for instance, MLPs and RBF networks are combined, are sometimes considered to be a "productive route"; see [63]. Regarding hybrid ensembles, Brown et al. [5] argue that two techniques that search the problem space in very different ways will probably result in models that specialize in different parts of the problem space. This implies that when using hybrid ensembles, it would most likely be better to select one specific classifier instead of combining the outputs from the different models.

2.3.3 Traversal of Hypothesis Space

A common solution for *traversal of hypothesis space* is to use a penalty term enforcing diversity in the error function when training ANNs. A specific and very interesting example is negative correlation learning [64], where the covariance between networks is explicitly minimized. For regression problems it has been shown that NC directly controls the covariance term in the bias–variance–covariance trade-off; see [65]. However, it does not work for classification problems when the models used can only output results at the abstract level, i.e., the class labels.

2.3.4 General Ensemble Creation Strategies

Some approaches for ensemble creation are better characterized as strategies rather than algorithms. The best known and most acknowledged among these are bagging and boosting. Many ensemble creation algorithms are variations of these strategies.

In bagging [66], diversity is achieved by training each base model using different emulated training sets obtained using resampling. Each training set (called *bootstrap*) consists of the same number of instances as the entire set of data available for training. Every bootstrap is created using sampling according to a uniform distribution. Instances may appear more than once in a bootstrap, since instances are drawn with replacement, with the result that approximately 63.2 % of available instances are included in each bootstrap. After the models have been trained, test instances are predicted using either voting or averaging. Bagging is most often used in classification, but may also be used in regression. When used for regression, the median may be used instead of averaging to achieve more robustness. Bagging almost never leads to increased error rates.

Since only a subset of the instances are used to train each model, there is always a portion of the available instances that has not been used to train a model. These instances are called out-of-bag, since they are not included in the bag of instances training the model. The out-of-bag instances can be used as an unbiased estimator of model performance, since they have not been used when training the model. It is also possible to get an out-of-bag estimate of the performance of an entire bagging ensemble, even though every instance has been used when training a subset of the models. By forming an ensemble composed of only the models for which an instance is out-of-bag, it is possible to get an unbiased estimate of the bagging ensemble while still using all instances for training. Since the out-of-bag ensemble is approximately $1/3$ the size of the entire bagging ensemble, the out-of-bag error estimate tend to overestimate the actual error made by an ensemble, simply because a larger ensemble is normally a stronger model.

Targeting diversity is inherent in random forest models [17], a technique utilizing bagging as a foundation for how the algorithm works, even if no diversity measure is explicitly maximized. A single tree in a random forest is very similar to a standard decision tree like CART. The basic idea is, however, to directly create an accurate decision tree ensemble, by introducing randomness in both the instance selection and in the feature selection. The feature selection is performed by randomly selecting a subset of the features to use at each new node during the creation process.

A novel approach, called random brains, to create accurate but diverse ensembles of ANNs was presented and evaluated in papers VI and XXII [67; 68]. The algorithm was inspired by how random forests manage to create both accurate and diverse ensembles without explicitly targeting diversity. Apart from using bagging, each ANN had a slightly randomized architecture, with randomly removed links between layers. The idea is that the randomly removed links will increase diversity by making each neuron affected by only a subset of the input attributes. The inspiration came from the procedure used when training the decision trees in random forests, where each split is based on only a randomized subset of the attributes.

In boosting, introduced by Schapire [69], the models are trained on data sets with entirely different distributions. It can be used to improve the performance of any learning algorithm. Unlike bagging, where training could be done independently and even in parallel, boosting is an inherently sequential procedure. The basic idea in boosting is to assign a higher weight to more difficult instances, thus making the learning procedure focus on these instances. Initially, all instances have equal weights. After the first model has been trained, all instances it fails on get an increased weight, while all instances it succeeds on get a weight reduction. The weight of an instance can be used either as part of the score function or as the probability that that instance will be drawn when bootstrapping is used. Most often, the ensemble prediction is based on a weighted majority vote, where the weights are based on the performance on some validation set.

Three fundamentally different types of boosting exist, according to Haykin [41].

- Boosting by filtering involves filtering the training examples by different models. The procedure is that a first model is trained on a subset of N instances. Then a new set of N instances is built by evenly selecting instances correctly and incorrectly classified by the first model. Thus, the first model will be approximately 50 % correct on the second data set. A second model is then trained using the second data set. Finally, a third set of instances is formed by adding N instances that the first two models disagree about. The third set is again used to train a final model. This form of boosting is rather uncommon.

- Boosting by subsampling works with a training sample of fixed size. The training instances are resampled according to a given probability distribution during training.
- Boosting by reweighting also works with a fixed sample size. It assumes that the learning algorithm can receive weighted training instances.

Many different boosting algorithms have been proposed. The typical variations are in how the weights are updated and what kind of combiner to use. The most well-known and used boosting algorithm is called AdaBoost (Adaptive Boosting) [70]. AdaBoost can use either subsampling or reweighting.

2.3.5 Combination Strategies

Kuncheva [45] gives a detailed review of the different kinds of combiners for the two most common types of outputs, i.e., labeled and measurement output. Since combiners are not the focus of this thesis, only the most common combiners will be briefly explained.

Combination of labeled outputs

The most straightforward combination strategy for labeled output is the majority vote. The class that most base classifiers vote for will be the output of the ensemble. In case of a tie, the class can either be randomly selected from the tying classes, or the decision can be guided by other information, such as the prior probabilities of the classes.

When the base classifiers are not equally accurate, it makes sense to let the vote of each base classifier be weighted by how competent it is. In practice, the weighted majority vote might be better than the simple majority vote, but it is susceptible to overfitting if the weights are based only on the performance on the training set.

There are a number of other combiners for labeled output, some of which might outperform the simple majority vote under certain conditions. Refer to [45] for a theoretical examination and comparison of several different combiners for labeled output.

Combination of measurement outputs

When considering measurement outputs, there are several different approaches proposed in the literature. They can be divided into class-conscious combiners and class-indifferent combiners. The class-conscious combiners derive the overall support for a particular class from all the measures for that class, one class at a time. The class-indifferent combiners, on the other hand, treat the combination task as a new problem to be learned, where the measurements from the ensemble members are used as input. Though class-indifferent combiners represent interesting alternatives, they have not been used in this thesis and the reader is referred to, e.g., [45] for further details. The class-conscious combiners that are described here can be applied as soon as the base classifiers are trained. Let $d_{i,j}(\mathbf{x})$ be the output of base classifier i for class j on instance $\mathbf{x}$ and let L be the number of base classifiers in the ensemble. The support S for each class is calculated using 2.41

$$S_j(\mathbf{x}) = f\left(d_{1,j}(\mathbf{x}), ..., d_{L,j}(\mathbf{x})\right) \tag{2.41}$$

where f is the combination function. The class label of instance $\mathbf{x}$ is found as the index of the maximum $S_j(\mathbf{x})$. The most popular choices for f include:

- *Simple mean* (*average*) ($f =$ average).
- *Minimum/maximum/median* (f = minimum/maximum/median).
- *Product* (f = product).
- *Trimmed mean* (*competition jury*). For a K percent trimmed mean, the individual support of the L base classifiers are sorted and K percent of the values are dropped on each side. The overall support is found as the mean of the remaining degrees of support.

The most common and most intensively studied combiners in this group are the product and the average combiners. There is no guideline as to which of these is best for a specific type of problem. The average, on the one hand, might in general be less accurate for some problems, but on the other hand, it is the more stable of the two [62; 71–73]. There are other class-conscious combiners not covered here. Again, the interested reader is referred to, e.g., [45] or [74] for further reading.

2.4 Related Work

2.4.1 Analysis of Diversity

Kuncheva and Whitaker [4] analyzed 10 diversity measures and from their experiments, diversity did not appear to be very useful as a selection criterion when constructing ensembles. Even though several different experiments were performed, some of them were rather artificial while the others were run only on a very limited set of problems.

The conclusions drawn by Kuncheva and Whitaker were further strengthened in a number of studies [21; 22]. Tang, Suganthan and Yao [22] analyzed some of the diversity measures and argued that since diversity is not precise, in the sense that some ensembles might have the same average base classifier accuracy and diversity on the training data while still achieving different performance on the test data, it should not be used as a selection criterion. Saitta [21] supported the negative view of Kuncheva [20] and also showed that not only does no working diversity measure exist, but no diversity measure is likely to ever exist. The reason is that the relationship between performance and diversity is not monotonic, i.e., the greatest diversity does not correspond to the best performance. Furthermore, the level of diversity necessary to achieve optimal performance for an ensemble of size L also depends in a non-monotonic way on L itself.

Another approach to diversity could be described as localized diversity [57; 75]. Sun and Zhang [75] proposed using region partitioning and region weighting by neighborhood accuracy to implement effective subspace ensembles where the performance of *k-nearest neighbor* (k-NN) is used to adjust the weights of the classifier for a local region. k-NN is a predictive classification technique where the classification is based on a majority vote by the neighbors of the instance to be classified. Aksela and Laaksonen [57] examined diversity of error measures. They argue that the final objective in classifier combination is not to produce a set of classifiers that has a maximal level of diversity regardless of what the correct classification would be. Instead, situations where the classifiers agree on the correct result should be rewarded rather than penalized when selecting the member classifiers, even though this is contradictory to the naive diversity maximization principle. Aksela and Laaksonen in particular argue against the use of pairwise measures, saying that

> [w]hen having found the two most diverse classifiers, adding a third to the set always decreases the set's overall diversity as the measure value for the larger set is an average of the pairwise values. Naturally in most cases selecting just two member classifiers from a large pool will not be the optimal solution for classifier

combining purposes. [57]

Breiman analyzed the performance of random forests in terms of strength and correlation [17]. He showed that the two ingredients involved in the generalization error are the strength of the individual classifiers and the correlation between them. The empirical evaluation indicate that better ensembles have lower correlation between classifiers and higher strength.

Stapenhurst used the connection with margin theory previously identified by Tang et al. [22] to carry out an analysis of diversity in his thesis [60]. All but a few of the proposed diversity measures can be expressed using a margin, i.e., they can be said to be margin measures. He shows that there are situations where the margin distributions are identical while the diversities (using the ρ correlation diversity measure) differ. Furthermore, he showed that when transforming a weighted ensemble into an unweighted ensemble (using duplication of more strongly weighted models), the margin is unaffected while the diversity can change. The conclusion drawn by Stapenhurst is that in most cases, it makes more sense to discuss ensemble performance in terms of margin theory, since it is better understood. However, he also acknowledges that the Q-statistics and the ρ correlation measures cannot be expressed using margin theory. He showed experimentally that a high diversity can be detrimental to the test set accuracy when using some algorithms and datasets. However, diverse bagging ensembles seemed to generalize so that the ensembles still achieved high test set accuracy.

Diversity in boosting ensembles has been studied in a number of papers [60; 76; 77]. Since AdaBoost introduces diversity by example reweighting, it should be seen as an ensemble algorithm that explicitly optimizes diversity. However, no diversity measure is explicitly optimized, but the optimization is inherent in the weighting scheme. Shipp and Kuncheva [77] showed that the diversity initially increases but later gradually returns to its starting level for almost all evaluated diversity measures. Stapenhurst [60] is able to explain the connection between performance and diversity in terms of a margin, showing, e.g., that quadratic loss represents a tradeoff between squared margin diversity (e.g., disagreement diversity) and average margin (i.e., average individual accuracy).

Kuncheva recently published a paper [78] evaluating κ-error plots. The experiments show that for smaller ensembles, the most important factor explaining ensemble accuracy is the accuracy of the individual models. It is also evident that it is possible to achieve good performance either by having very accurate individual models that are less diverse or by having slightly less accurate individual models that are more diverse. Furthermore, Kuncheva demonstrates that the analysis of diversity using κ-error plots is a fruitful way to increase the understanding of the relationship between ensemble error, individual error, and the diversity of the ensembles.

2.4.2 The Static Overproduce-and-Select Paradigm

Kuncheva presents a review of previous work where diversity in some way has been utilized to select the final ensemble [45]. Giacinto and Roli [8] form a pairwise diversity matrix using the double fault measure and the Q statistic [79] to select the least related classifiers. They search through the set of pairs of classifiers until the desired number of ensemble members is reached. The algorithm was evaluated on one dataset and was compared to the ensemble formed by using the complete pool of models. It is impossible to tell whether the proposed algorithm is significantly different from using the entire pool of models as the ensemble. Giacinto and Roli [10] also applied a hierarchical clustering approach where the ensembles are clustered based on pairwise diversity. The ensemble was formed by picking a classifier from each cluster and step-wise joining the two least diverse classifiers until all classifiers belong to the same cluster. The ensemble used in the end was the ensemble with the highest accuracy on a validation set. Margineantu and Dietterich [9] also search for the most diverse pairs of classifiers from a set of classifiers produced by AdaBoost. They call this approach “ensemble pruning”.

Banfield et al. [80] used an approach where only the uncertain data points were considered and used to exclude classifiers failing on a larger proportion of these instances, compared to other classifiers. No significant difference between the solutions could be detected.

It should be noted that all these approaches select ensembles based on the diversity between pairs of classifiers, rather than on ensemble diversity.

Chandra and Yao [81] proposed an algorithm, called the diverse and accurate ensemble learning algorithm (DIVACE), that uses a multi-objective evolutionary approach to ensemble learning. DIVACE tries to find an optimal trade-off between diversity and accuracy by treating them explicitly as two separate objectives. The diversity measure used in DIVACE is the correlation measure. The DIVACE algorithm continuously produces neural networks and tests them against other networks. Neural networks that are non-dominated are kept and dominated networks are discarded. The algorithm is evaluated on two datasets and compared to another algorithm using a similar produce-and-discard strategy.

Chen evaluates different pruning (i.e., static overproduce-and-select) algorithms in his dissertation and compares them to the full ensemble formed using all the available models. The proposed static overproduce-and-select algorithm is based on expectation propagation (EP) and works for both classification and regression problems. It seems to work for regression problems, winning over the full ensemble on 6 out of 7 datasets. On classification problems, the pruned ensembles are not significantly better than using the full ensemble. In fact, on most datasets, the pruned ensemble and the original ensemble are equally effective, i.e., they tie.

In a more recent paper [82], an information theoretic link between accuracy and diversity is proposed and used as a selection criterion. The selected subensembles were compared to the single best individual model.

3. Conformal Prediction

Conformal prediction (CP) was introduced as an approach for associating classification or regression predictions with reliable confidence estimates [83; 84]. Vovk, Gammerman and Shafer provided a comprehensive introduction to conformal classification in [85] and presented a tutorial on CP in [86]. This chapter introduces the elements of conformal prediction relevant to this thesis. Much of the material is adapted from [87–89]. Since the focus of this thesis is classification, the presentation of CP will also be focused on how the framework can be applied in a classification context.

3.1 Introduction

In essence, the conformal prediction framework makes it possible to answer the question: how confident can we be that a prediction is actually correct? This question is not new, and several approaches have been proposed for answering it; most well-known are the Bayesian framework and the theory of Probably Approximately Correct learning (PAC theory) [90]. Bayesian learning requires that the distribution that generates the data is known beforehand, resulting in misleading confidence estimates if the correct prior is not known [91; 92]. PAC theory, on the other hand, can only provide bounds for the overall error of the model and not for individual test examples [91]. In cases when the data is not very clean, the bounds of the confidence estimates produced by PAC theory also tend to be too wide to be useful in practice [93]. CP, on the other hand, does not rely on any knowledge of the prior distribution, and provides separate confidence estimates for each predicted example.

An ordinary machine learning model is used as an *underlying model* to CP, and the conformal framework transforms the predictions from the underlying model into valid prediction sets.

When applied to regression problems, CP produces predictions in the form of prediction intervals for a specified confidence level; for classification problems, CP produces predictions in the form of class label sets. Typically, the higher the confidence expected from a conformal predictor, the larger the prediction intervals (or class label sets) will be. A prediction is erroneous if the true target is not included in the prediction interval (or class label set) for that instance.

The confidence predictions provided by CP are said to be *valid*, meaning that the probability of making an erroneous prediction is guaranteed to be less than or equal to a predefined significance level ε in the long run; the confidence in such a prediction is thus $1-\varepsilon$. The practical meaning of validity is that when predicting a set of instances with for example a confidence of 95 %, i.e. $\varepsilon = 0.05$, the percentage of erroneous predictions in that set is guaranteed not to exceed 5 % in the long run.

Validity is guaranteed under the assumption that the data is exchangeable, which is a slightly weaker assumption than the assumption that the data is independent and identically distributed (i.i.d). Most machine learning algorithms work under the assumption of i.i.d, i.e., that the instances are selected randomly and independently according to an identical probability distribution [94]. For the assumption of exchangeability to hold, any ordering of instances must be equally likely. When data is evaluated offline with access to the complete dataset, exchangeability can easily be achieved by randomizing the ordering of instances. For streaming data, where the ordering of instances is fixed, the framework will work as long as no concept drift occurs. However, the framework can also be used to identify when a concept drift has indeed occured [95].

The central component of the CP framework is the conformity function. The conformity function assigns a conformity score to each instance–label pair. When predicting a specific test instance, a conformity score is assigned to each possible class label and the scores are compared to the scores obtained from instances with known class labels. The instances with known class labels are only assigned a conformity score for the true class. The labels that are found to be nonconforming compared to the scores of the labeled instances are excluded. A label is considered nonconforming if the conformity score for that label is lower than a predefined fraction (the significance level ε) of the scores assigned to the labeled instances. The prediction for the test instance is the set of class labels that was not excluded.

CP has a resemblance with hypothesis testing since each possible class could be considered as a null hypothesis which may be disproved if it is significantly different from the labeled data.

CP was originally formulated in a transductive setting, which means that the conformity scores had to be recalculated for each new test instance and class label. The implication of that is that a new model has to be trained for each test instance and each class. For most machine learning techniques, this is clearly not a feasible approach. To overcome this, an inductive version was introduced for classification by Papadopolous in [91]. When using Inductive Conformal Prediction (ICP), the training data is split into a proper training set, used to train a single model, and a calibration set, used to calibrate the conformity scores for a test example to identify whether it is conforming or not.

In the following more formal description of ICP, a similar notation is used as in [96]. The set of instances is denoted by $\boldsymbol{Z}$, which is the Cartesian product $\boldsymbol{X} \times \boldsymbol{Y}$ of the independent variables $\boldsymbol{X}$, henceforth called the object space, and the dependent variable $\boldsymbol{Y}$. Consequently, each example $z \in \boldsymbol{Z}$ consists of two parts: $z = (x, y)$, where $x \in \boldsymbol{X}$ is the object and $y \in \boldsymbol{Y}$ is the dependent variable. In classification, $\boldsymbol{Y}$ is a finite set, usually referred to as the class variable, and in regression, $\boldsymbol{Y}$ is the real line $\mathbb{R}$.

Let us consider a set $z_1, ..., z_N$ of training instances, where N is the number of available instances and $z_i = (x_i, y_i) \in \boldsymbol{Z}$. We split the set into a proper training set $(z_1, ..., z_m)$ of size $m < N$ and a calibration set of size $l := N - m$.

Let x_{N+1} (or x for short) be a new test object. The idea of conformal prediction is to try all possible class labels $c \in \boldsymbol{Y}$ for the test object to measure how well each label conforms to the proper training set. In other words, for an object and class label, $z = (x, c)$, the aim is to determine if it is possible that the label c can be the true class label for the object x. To determine if that is possible, a *conformity score* $A((z_1, ..., z_m), z)$ needs to be calculated using the *inductive conformity function* $A : \boldsymbol{Z}^m \times \boldsymbol{Z} \to \mathbb{R}$. The conformity function is often defined by

$$A((z_1, ..., z_m), (x, c)) := \Delta(c, f(x)), \tag{3.1}$$

where $f : \boldsymbol{X} \to \boldsymbol{Y}'$ is a predictive model, trained using the proper training set $\{z_1, ..., z_m\}$, predicting $f(x) \in \boldsymbol{Y}'$ for the object x. $\Delta : \boldsymbol{Y} \times \boldsymbol{Y}' \to \mathbb{R}$ measures the similarity between the class label $c \in \boldsymbol{Y}$ and the prediction $f(x)$ of the underlying model. The reason why the model is allowed to produce a prediction $\boldsymbol{Y}'$ different from the set of available classes $\boldsymbol{Y}$ is that the model may output additional information, such as a probability estimate for each class, that can be used by the similarity measure Δ. The model f is trained using a machine learning algorithm, such as neural networks, decision trees, k-nearest neighbor, ensembles, etc.

A randomized ('smoothed') *inductive conformal predictor* (ICP) using the conformity function A is defined as the set predictor

$$\Gamma^{\varepsilon}(z_1, ..., z_N, x) := \{c | p^c > \varepsilon\}, \tag{3.2}$$

where $\varepsilon \in (0, 1)$ is the chosen significance level and $p^c, c \in \boldsymbol{Y}$ is defined by

$$p^c = \frac{|\{i = m+1, ..., N | \alpha_i < \alpha^c\}| + \theta |\{i = m+1, ..., N | \alpha_i = \alpha^c\}| + 1}{N - m + 1}, \tag{3.3}$$

where θ is a uniform random number in the interval $[0,1]$ and

$$\begin{aligned} \alpha_i &= A((z_1,...,z_m),z_i), i = m+1,...,N, & (3.4) \\ \alpha^c &= A((z_1,...,z_m),(x,c)) & (3.5) \end{aligned}$$

are the conformity scores for the calibration set and the test example, respectively. Conformity scores are only calculated for the true target on the calibration instances, while one conformity score is calculated for each class label $c \in \boldsymbol{Y}$ for the test object x.

Obviously, it is always possible to get a point prediction from CP by simply selecting the class with the highest p^c as the predicted class.

3.1.1 Measuring Efficiency

Since the error level is directly controlled by the user specified significance level, accuracy is not a very useful measure when evaluating and comparing conformal predictors. Instead, CP is evaluated in terms of *efficiency*—the size of the prediction intervals or prediction sets. In other words, the mechanism enabling CP to produce valid predictions is a trade-off between the accepted error level and the crispness of the predictions. A prediction set may contain all, some, or even no class labels in classification. For regression, efficiency can be measured as the average or median width of the prediction intervals [97].

For classification, two general and often used sets of criteria of efficiency are:

- The *confidence* and *credibility* of the prediction $p^c, c \in \boldsymbol{Y}$. Confidence is $1 - min_c(p^c)$ and credibility is $max_c(p^c)$. These criteria do not depend on the significance level ε.
- Whether the prediction set contains a single class (the ideal case), multiple classes (an inefficient prediction), or no classes (a super efficient prediction) for a certain significance level ε. The average number of singleton predictions for a certain significance level ε has been used in some publications.

In [98], a more systematic discussion of criteria for measuring efficiency in the classification context was given. Two kinds of criteria were identified: those applicable to prediction sets Γ^{ε} and consequently dependent on the significance level ε, and those applicable directly to the sets of p-values $(p^c|c \in \boldsymbol{Y})$ and consequently independent of ε. A further distinction that was made is between prior and observed criteria, where prior criteria are ignorant of the true, or observed, class label of a test instance and the observed criteria are based on knowledge of the true class label. Ten different criteria of efficiency was discussed and divided into prior criteria and observed criteria.

Prior Efficiency Criteria

The prior criteria presented in [98] are divided into criteria independent and dependent of the significance level ε. For measures independent of ε, smaller values are preferable.

- The *Sum criterion* [99] measures efficiency by the average sum of the p-values over all t test instances.

$$\frac{1}{t}\sum_{i=N+1}^{N+t}\sum_{c} p_i^c, c \in \boldsymbol{Y}$$

- The *Unconfidence criterion* is the average unconfidence, which is the second largest p-value.

$$\frac{1}{t}\sum_{i=N+1}^{N+t} \min_c \max_{c' \neq c} p_i^{c'}, c \in \boldsymbol{Y}$$

  This is equivalent to measuring the average confidence $(1 - \text{unconfidence})$.

- The *Fuzziness criterion* is the average fuzziness, where fuzziness is defined as the sum of all but the largest p-value.

$$\frac{1}{t}\sum_{i=N+1}^{N+t}\sum_{c} p_i^c - \max_c p_i^c, c \in \boldsymbol{Y}$$

  If two conformal predictors are compared and fare equally well using either the unconfidence or fuzziness criterion, then the average credibility is used instead.

Criteria that are dependent of ε.

- The *Number criterion* [99; 100] measures efficiency as the average size of the prediction sets. The size of a prediction set is the number of labels in the set.

$$\frac{1}{t}\sum_{i=N+1}^{N+t} |\Gamma_i^{\varepsilon}|$$

- The *Multiple criterion* measures the percentage of prediction sets containing multiple class labels at a specific significance level ε. Once the multiple criterion reaches zero, the percentage of empty prediction sets is measured instead. As an alternative, the percentage of singleton predictions can be measured instead. For the multiple criterion smaller values are preferable, while larger values are preferable when measuring the percentage of empty prediction sets.

- The *Excess criterion* is similar to the number criterion, except that it is the number of class labels exceeding 1 that is measured, i.e. the average number of excess labels.

  $$\frac{1}{t}\sum_{i=N+1}^{N+t}(|\Gamma_i^{\varepsilon}|-1)$$

  Smaller values are preferable.

For binary problems, i.e. when $|\boldsymbol{Y}| = 2$, the unconfidence and fuzziness, as well as the multiple and excess criteria coincide. Both paper VII and VIII use a criterion similar to the Number and Excess criteria, called OneC, which measures the number of instances with exactly one class label in the prediction region.

Observed Efficiency Criteria

The four observed criteria taking advantage of the knowledge of the true class label are presented below, divided into criteria that are independent and dependent of the significance level ε. For the independent observed criteria, smaller values are preferable.

- The *Observed Unconfidence criterion* measures the average observed unconfidence, where the observed unconfidence is the the largest p-value among the false labels.

  $$\frac{1}{t}\sum_{i=N+1}^{N+t}\max_{c\neq y_i} p_i^c, c \in \boldsymbol{Y}$$

  where y_i is the true label for instance i.

- The *Observed Fuzziness criterion* uses the average p-values among the false class labels.

  $$\frac{1}{t}\sum_{i=N+1}^{N+t}\sum_{c\neq y_i} p_i^c, c \in \boldsymbol{Y}$$

For the dependent observed criteria, smaller values are better.

- The *Observed Multiple criterion* measures the average number of prediction sets containing any false labels.
- The *Observed Excess criterion* measures the average number of false labels included in the prediction sets.

The Sum and Number criteria have no equivalents among the observed criteria since they are calculated using all classes.

For binary problems, i.e., when $|\boldsymbol{Y}| = 2$, just as with the previously mentioned criteria, the observed unconfidence and observed fuzziness coincide, as do the observed multiple and observed excess criteria.

In paper VIII, which investigates how conformal predictors are affected by imbalanced data, efficiency is measured using two observed criteria. The first criterion is called majority error and is defined as the number of majority class instances from which the true class has been excluded from the prediction set divided by the total number of instances where the true class has been excluded from the prediction set. It shows how biased a predictor is in making its errors, since an unbiased predictor will have errors distributed approximately according to the prior probability distribution. The majority error criterion only considers the instances that are incorrectly predicted (or not predicted at all, if the prediction set is empty). The second criterion used is a variation of the observed excess criterion, measuring the number of times the majority class is excluded from prediction sets divided by the total number of class labels excluded from prediction sets.

3.2 Handling Computational Efficiency of Conformal Predictors

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 made may affect the results of the ICP in two different ways. Using a small calibration set leads to a high variance of the 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 to a high variance in the chosen sample. On the other hand, the smaller the training set is, the less powerful the predictive model will be. In a study aimed at comparing the efficiency of transductive and inductive conformal prediction, part of the empirical evaluation included an analysis of the effects of the calibration set size [101]. The empirical results showed that the best performing ICP classifiers used $15-30$ % of the full training set as calibration set. Furthermore, to get good performance and high confidence ($\varepsilon = 0.01$), the calibration set should contain at least 500 instances. On the other hand, having a large proper training set was clearly the most important factor for maximizing efficiency.

3.2.1 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 [**?**]. 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 a model is built for every fold $k \in \{1,...,K\}$. The examples $z_1,...,z_n$ are divided into K different sets. One model is trained for each of the K folds and for each fold one of the sets is withheld as a calibration set, whereas the remaining sets are merged into a proper training 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 $z_1,...,z_N$ 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 $\{1,...,N\}$. The p-values from each fold $k \in \{1,...,K\}$ are defined as

$$p_k^c = \frac{|i \in F_k : \alpha_{i,k} \leq \alpha_k^c| + 1}{|F_k| + 1} \tag{3.6}$$

and the p-values from the CCP are defined as

$$\begin{aligned} p^c &= \frac{\sum_{k=1}^K |i \in F_k : \alpha_{i,k} \leq \alpha_k^c| + 1}{N+1} && (3.7) \\ &= \bar{p}^c + \frac{K-1}{n+1}(\bar{p}^c - 1) && (3.8) \\ &\approx \bar{p}^c && (3.9) \end{aligned}$$

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

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

$$\begin{aligned} \alpha_{i,k} &= A(z_{F_{-k}}, z_i), i \in F_k && (3.10) \\ \alpha_k^c &= A(z_{F_{-k}}, (x,c)) && (3.11) \end{aligned}$$

where $F_{-k} = \cup_{j \neq k} F_j$.

3.2.2 Bootstrap Conformal Prediction

Bootstrap conformal prediction (BCP) is similar to CCP. Just as with 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 the training and calibration sets, BCP uses bootstrap replicates [102] 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 $z_1,...,z_l$. Since instances are drawn with replacement, allowing duplicates to be drawn, B_k denotes a bag of indices for the examples used to train the kth model. The conformity scores $\alpha_{i,k}$ and α_k^c are defined for each fold k and each potential class label $c \in \boldsymbol{Y}$ by

$$\alpha_{i,k} = A(z_{B_k}, z_i), i \in B_{-k} \quad (3.12)$$

$$\alpha_k^c = A(z_{B_k}, (x,c)) \quad (3.13)$$

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

The p-value of BCP is defined by

$$p^c = \frac{\sum_{k=1}^{K} |\{i \in B_{-k} : \alpha_{i,k} \leq \alpha_k^c\}| + T/l}{T + T/l} \quad (3.14)$$

where $T = \sum_{k=1}^{K} |B_{-k}|$ is the total size of the calibration sets.

3.2.3 Using Out-Of-Bag Estimation in Conformity Functions

Bagging Ensembles were introduced in Section 2.3.4. Just to briefly recapitulate, a bagging ensemble [66] is an aggregated model combining several 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 one-third 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 $H = \cup_{l=1,...,L} h_l$ be an ensemble of size L, where each h_l is called a member of the ensemble. $H(x)$ predicts a class label $c \in \boldsymbol{Y}$ for the object x using majority voting. Let a training sample z_{E_l} of size N with examples drawn (with replacement) from the available examples $z_1,...,z_N$ be used to train each ensemble member h_l. E_l represents the indices of the examples that are in the bag for the lth ensemble member, i.e. these are used for training h_l, and $E_{-l} = \{1,...,N\} \backslash E_l$ represents the indices of the examples that are out-of-bag for h_l. The conformity score α_i for a calibration example $z_i = (x_i, y_i)$ is defined by

$$\alpha_i = A(\{z_1,...,z_N\}, z_i) = \Delta(y_i, M_{E_{-l}}(x_i)), i = 1,...,N \qquad (3.15)$$

where $H_{E_{-l}}(x_i) = \cup_{l=1,...,L \wedge i \in E_{-l}} h_l(x_i)$, i.e, only the models for which instance i is out-of-bag are combined into an ensemble used to predict the calibration instances.

The conformity score for each class on a test example, α^c, is defined by Equation (3.1), letting $y = c, c \in \boldsymbol{Y}$ and $f = M$. In other words, the full ensemble M is used normally for all the test examples.

The following argument is taken from [89], where it was given for random forests and regression. However, the argument is applicable to bagging in general, including classification tasks. The text presented here is very similar to the original text in [89] but is slightly adapted to fit the context of this chapter.

When using out-of-bag instances instead of a separate calibration set, the actual underlying model, i.e., the bagging ensemble, is no longer used when calculating the nonconformity scores and p-values. As shown above, various subsets of the ensemble are used for the out-of-bag-instances, but the entire ensemble is used for the test instances. In other words, the nonconformity functions applied to the calibration and test instances are defined differently:

$$\begin{aligned} Calibration: \alpha_i &= \Delta(y_i, H_{E_{-l}}(x_i)), i = 1,...,N && (3.16) \\ Test: \alpha^c &= \Delta(c, H(x)), && (3.17) \end{aligned}$$

where E_{-l} is a random factor determining the subset of models for which instance i is out-of-bag. In general, the use of different nonconformity functions could clearly cause the resulting conformal predictor to become invalid, i.e., the probability of excluding the true target value would no longer be bounded by the provided confidence level.

In principle, the same random component as used in Equation (3.16) may also be used when predicting the target value for the test instance (by only considering a random subset of the forest when predicting the target of the test instance), and in that case the same nonconformity function, Equation (3.16), would obviously be used for all instances, hence not violating the assumptions underlying the ICP framework.

However, when using the whole ensemble for the test instance, as proposed here, one would expect the predicted values to be closer to the true target than when using a random subset of the models. The simple reason for this is that a larger ensemble is normally a stronger model. In fact, it is well-known that out-of-bag error estimates tend to overestimate the actual error made by an ensemble for the same reason. Not until the ensemble is so large that the randomized sub-ensembles will be as accurate as the entire ensemble, is this bias eliminated. For random forests, empirical results indicate that this might happen when the entire ensemble contains somewhere between $1000-3000$ trees [103]. Consequently, the expected nonconformity of a test instance is less than (or for a very large ensemble equal to) the expected nonconformity of a calibration instance, i.e. the probability of including nonconforming targets in the prediction region is unchanged or increased when using the whole forest. Hence, rather than increasing the risk for generating an invalid conformal predictor, one would expect the conformal predictor using out-of-bag instances to be conservative. Therefore, the proposed setup should be, if anything, less efficient than if the whole ensemble was used together with additional calibration instances.

3.3 Class Label Conditional Conformal Prediction

An often encountered challenge in real-world scenarios is that classes are imbalanced. The problem of getting good performance in situations where some data is underrepresented or the class distribution is severely skewed is called *the imbalanced learning problem* [23; 104]. Chawla et al. wrote in [104] that “[t]he class imbalance problem is pervasive and ubiquitous, causing trouble to a large segment of the data mining community.”

According to He, one important reason for imbalanced data being problematic is that most machine learning techniques assume a balanced class distribution [23]. When such models are trained on imbalanced data, the trained model often ends up being biased towards the majority class [104; 105]. A model that is biased towards the majority class will make a disproportionally large number of errors on the minority class; this is called *the class bias problem* [106; 107]. In the extreme case, all instances will be predicted as the majority class, making the model practically useless.

When ordinary CP makes a prediction, the error rate is guaranteed for the prediction set. However, for an individual class, nothing can be guaranteed, since it is possible that all errors are made on only one of the classes. Luckily, CP can be transformed into a class label conditional CP (LCCP) with only minor alterations. An LCCP is guaranteed to be valid for each individual class [96].

Transforming CP into LCCP is a very straightforward operation. The main difference is that instead of considering all the instances in the calibration set when calculating the p-value for an object and class label, $z = (x_{n+1}, c)$, only calibration instances with the class c are considered. For a label-conditional ICP, this means that Equation (3.3) for ICP needs to be changed into

$$p^c = \frac{|\{i = m+1, ..., N : y_i = c \wedge \alpha_i \leq \alpha^c\}| + 1}{|\{i = m+1, ..., N : y_i = c\}| + 1}, \tag{3.18}$$

where y_i is the true class of calibration instance z_i.

CP produces, for each new test object, a prediction region which may contain all possible class labels. When considering LCCP, in theory one conditional conformal predictor is created for each class label:

$$\Gamma_c^\varepsilon(\{z_1, ..., z_N\}, x_{N+1}) = \{c | p^c > \varepsilon\}, \tag{3.19}$$

where p^c is calculated using Equation (3.18).

The label-conditional ICP Γ_c^ε will only produce predictions indicating whether object x is likely to have the class label c or not, with certainty $1 - \varepsilon$. Consequently, since one conditional ICP is defined for each class $c \in \boldsymbol{Y}$, different values of ε can be used for each class, allowing the prediction of different classes with different levels of certainty. However, since one conformity score is still calculated for each class, several label-conditional ICPs can be combined into a conditioned prediction set over all classes, potentially with different significance levels for each class, ε_c:

$$\Gamma(\{z_1, ..., z_N\}, x_{N+1}) = \{c | p^c > \varepsilon_c\}. \tag{3.20}$$

If the significance level is the same for all classes, $\forall c \in \boldsymbol{Y} : \varepsilon_c = \varepsilon$, a label-conditional ICP can be defined using Equation (3.2) and it will be valid, both for each class individually, and for the prediction set as a whole.

4. Research Approach

This chapter presents the method adopted to address the research question. It starts with a presentation of the research methodology adopted, followed by a presentation of the datasets used. The chapter ends with a presentation of how an evaluation should be performed when evaluating machine learning results.

4.1 Research Methodology

Research is “the systematic investigation into and study of materials and sources in order to establish facts and reach new conclusions” [108]. Research is normally performed in the context of a paradigm, defining our ontology, epistemology and methodology. Ontology helps us define the form and nature of the world; epistemology helps us define what can be known about the world; and methodology gives us the means to obtain knowledge about the world. When developing IT artifacts, Hevner et al. [109] argues that both the behavioral science and design science paradigms are suitable alternatives. The behavioral science paradigm seeks to develop theories that can explain or predict how people will behave in relation to the analysis, design, implementation, management and use of information systems. Design science, on the other hand, focuses on problem solving by creating artifacts. Artifacts in this context can be broadly defined as constructs (vocabulary and symbols), models (abstractions and representations), methods (algorithms and practices), and instantiations (implemented and prototype system). Design is considered to be both the process of creation and the artifact designed within the design science paradigm.

Machine learning research can be defined as the field of scientific study that focuses on algorithms that can learn from data [110]. The underlying assumption within the machine learning field is closely related to the positivist paradigm. The underlying epistemological assumption is that of objectivism, i.e. that objective knowledge exists and can be obtained through observation and experiment. Methodologically, defining how the research question can be answered, the positivist uses an approach that places the point of decision with Nature rather than with the inquirer’s bias [111]. As a consequence, experimental research is heavily relied upon, since it allows hypotheses to be formulated and verified (or falsified) using statistical tests applied to empirical data.

When doing research in the field of machine learning, the artifacts are either theoretical, like algorithms, practical, like implementations or systems, or a combination of both. When using design science, machine learning algorithms are often implemented as software solutions whose performance can be empirically evaluated and compared to other solutions [112]. For theoretical artifacts, an alternative method of validation is to use mathematical proofs. Mathematical proofs uses axioms, i.e., self-evident or assumed facts, and theorems, which are statements that have been proven using axioms and other theorems. A mathematical proof is deductive and must demonstrate that a statement is always true. When it is possible to prove something mathematically, it is always preferable to empirical evidence. If mathematical proofs can be derived, empirical evaluation becomes less important and may serve primarily as an illustrative example or be used for pedagogical purposes.

Since the research question of this thesis regards how ensembles can be created effectively in the context of classification, the experimental approach is suitable. It makes it possible to use measures suitable for measuring effectiveness when comparing different methods in controlled experiments. The research presented in this thesis includes the design, implementation and comparison of artifacts of different kinds, and as a consequence it makes sense to relate the research to the seven guidelines given by Hevner and Chatterje [113]:

- **Design as an Artifact:** To be counted as design science, a viable artifact in the form a construct, a model, a method or an instantiation must be produced. The results of the research presented in this thesis are artifacts in the form of methods and instantiations.

- **Problem relevance:** In Chapter 1, the problem motivating the research question was presented. By addressing the research question, new insights into how to effectively create ensembles in the context of classification have been provided.

- **Design evaluation:** The artifacts must be rigorously evaluated regarding their utility, quality and efficacy. The artifacts are compared with each other as well as with standard artifacts in terms of effectiveness as defined in Section 4.1.1.

- **Research contributions:** The contributions of this thesis are artifacts evaluated in order to answer the research question. The outcome of the studies are summarized and the concluding chapter contains a detailed discussion of how the studies relate to each other and together help to answer the research question.

- **Research rigor:** The proposed methods have been rigorously evaluated following the recommendations for algorithmic comparisons within the field of machine learning research presented in Section 4.3. The evaluation has been carried out primarily on benchmarking datasets and the motivation for using such datasets is given in Section 4.2.

- **Design as a search process:** The process of reaching the conclusions of this thesis have involved investigating and comparing different ways of creating effective ensembles in the context of classification. The investigation included surveying the literature for relevant related work.

- **Communication of the research:** The results of the research have been presented to the research community in the form of the eight peer reviewed publications included in this thesis.

4.1.1 Measuring Effectiveness

To be able to address the research question, it is necessary to define what is meant, in the context of this thesis, by ‘effectiveness’. The purpose of such a definition is both to clarify to the reader what is meant when the term is used in different contexts but also to make it possible to ensure that effectiveness can be measured adequately. There are a number of different ways to define effectiveness. Which definition is used can be expected to greatly affect both the theoretical background, the results achieved, and the kind of conclusions that can be drawn. Effectiveness is defined differently depending on whether the context is related to the first or second sub-question.

Effectiveness in the context of the first sub-question is defined as the ensemble accuracy (or ensemble error) on test data. An effective ensemble is accurate on test data and the more accurate, the more effective. The main motivation for this definition is that high accuracy (or low error) is in most cases the ultimate objective when creating predictive models like ensembles. Another motivation for using accuracy as the definition of effectiveness is that several studies have evaluated to what extent diversity measures can be used as optimization criteria, and most diversity measures are defined in terms of accuracy. However, other definitions of effectiveness could also be used to determine which of the implicit and explicit learning strategies is most effective. One example of an alternative focus would be to define effectiveness in terms of how well ensembles can rank correctly predicted instances ahead of incorrectly predicted. The area under the ROC curve could have been used as a definition of effectiveness if ranking ability had been the focus. Another example of an alternative focus would be to define effectiveness in terms of how well the created ensembles can correctly estimate the class probabilities. Using this alternative focus, the Brier Score, measuring the accuracy of probabilistic predictions, would have been a suitable definition of effectiveness.

In the context of the second sub-question, effectiveness is defined as efficiency, as the term is used within the conformal prediction framework. As was demonstrated in Section 3.1.1, efficiency can be measured in several different ways. Both prior and observed efficiency criteria are used in the papers. The conformal predictor that is the most efficient, using an appropriate efficiency criterion, will be considered the most effective.

4.2 Datasets

It is more or less standard procedure in machine learning research to conduct experiments on a large number of benchmark datasets. However, as noted in [114; 115], performing experiments with such datasets has both drawbacks and benefits. Repositories like UCI [116] and PROMISE [117] provide data from several different problem domains with a lot of different characteristics. This allows researchers to perform experiments on collections of datasets that are large enough to make it possible to test for statistical difference in performance between different ways of creating models. Obviously, when using a large collection of datasets, it is not the individual models that are compared but rather the algorithm and a specific parameter setup used to train each model on each of the datasets that is compared. A repository is also useful when evaluating new algorithms to identify the characteristics of the problem domains on which the algorithm performs well or poorly.

Among the drawbacks there is the fact that many of the datasets have been extensively used in experiments and many algorithms might have been sub-optimized in order to appear to be performing extremely well on specific datasets. This is problematic when comparing results between papers but when primarily comparing results achieved within a controlled experiment using many randomly selected datasets, it is reasonable to assume that such sub-optimizations may only affect the results marginally. Another aspect is that the benchmark datasets do not represent all possible problem domains, something which makes generalizations beyond the datasets used in the evaluation not necessarily valid.

A final point made in [115] is that repeated tests for significant differences may result in significant differences appearing by chance, as a consequence of repeating a test often enough. This problem can be handled by a proper evaluation procedure, something which is further discussed in Section 4.3 below.

In the papers included in the thesis, datasets from the UCI and PROMISE data repositories have been used. Explanations of what characterizes the problem domains of the datasets can be found on the web pages of those repositories. A number of chemoinformatics datasets used and described in [118; 119] were also used in paper IV. The two feature sets used in [118] were used in the paper.

4.3 Evaluation of Research in Machine Learning

Several different aspects have to be considered when evaluating classifier models. This section is to a large degree taken from [25]. The topics covered are:

- Which measure(s) should be used to evaluate classifier performance?
- How should future performance (i.e., on novel data) of a single classifier be evaluated?
- How should the performance of two different classifiers be compared on a single data set?
- How should the performance of two or more classifiers be compared on multiple data sets?

The rest of this chapter describes the common approaches used when evaluating and comparing different methods for generating classifiers. The following section, on performance measures, discusses both measures used in the studies as well as commonly used alternative performance measures that could have been relevant if the research had had a different focus (see Section 4.1.1).

4.3.1 Performance Measures

The most frequently used measurement for classification problems is accuracy (or, conversely, the error rate), i.e., the percentage of correct (or incorrect) predictions for the test set. In most cases, accuracy is a good choice for evaluating classifiers. However, if the classes are imbalanced, accuracy is often not very informative, since the minority class(es) will be less important for the overall measure. For example, if 99 % of the instances belong to one class, any classifier predicting only that class will always be 99 % accurate, but will completely fail to identify the minority class.

With imbalanced problems, alternative measures to accuracy are often used. For binary classification problems, a series of measures can be used. In this context, one of the classes is referred to as the positive class. Which class that might be is problem specific and depends on the circumstances.

- True positive (TP) is the number of positive instances correctly classified as positive.
- False negative (FN) is the number of positive instances wrongly classified as negative.
- False positive (FP) is the number of negative instances wrongly classified as positive.
- True negative (TN) is the number of negative instances correctly classified as negative.

Recall for the positive class, which is sometimes referred to as *sensitivity*, is the proportion of positive instances correctly classified as positive, i.e., $TP/(TP+FN)$. Recall for the negative class is sometimes referred to as *specificity*, i.e., TN / (TN + FP). *Precision* for the positive class is the proportion of instances classified as positive that actually are positive, i.e., $TP/(TP+FP)$.

A receiver operating characteristic (ROC) curve [120] is a graphical representation of the trade-off between the true positive rate and the false positive rate. The area under the ROC curve (AUC) is often used to evaluate models that are able to output class probabilities that can be used for ranking. Models that perfectly manage to rank all positive instances correctly ahead of any false positives will have an area equal to 1. A model that ranks examples randomly has an expected area under ROC curve of 0.5.

4.3.2 Evaluating Classifier Performance

When evaluating a classifier model, the error rate on the set used for training is almost guaranteed to underestimate the future error rate, since the model has been built to fit the training set. By evaluating the classifier on a data set not used during training (often referred to as the test set) it is possible to get an unbiased estimate of the classifier performance. The performance on the test set can also be used to compare the relative performance of different classifiers on the same domain. Obviously, the class labels on the test set must be known in order to compute the test set performance. This approach is called the *holdout method* and the set used for evaluation is sometimes called the holdout set. The proportion of data used in each set must be decided by the analysts.

The holdout method has several limitations. First of all, by setting aside part of the data set for evaluation, fewer instances are available for training the model. This might lead to reduced performance compared to what could have been achieved if all the data were used for training. Furthermore, the model will be influenced by how the data set was split. If the training set is too small, the variance of the model will increase, while if the training set is too large, this will result in a less reliable performance measure, with increased confidence intervals. Finally, the training and test sets are not independent of each other. If one class is over-represented in the training set, it follows that it must be comparably under-represented in the test set, and vice versa. However, the last limitation can be avoided if the data is sampled with stratification, i.e. if it is made sure that the classes are evenly distributed in all sets.

A repeated use of the holdout method, using a randomly selected part of the data as a test set each time, is called *random subsampling*. The estimated performance is the average performance on the test sets over all repetitions. Random subsampling improves the estimation of the classifier by reducing the variance, but still encounters some of the problems associated with the holdout method since it does not use all the instances for training. For further details on why random subsampling is a dubious choice, see e.g. [121].

An alternative to random subsampling is *cross validation*. In this approach each instance is used the same number of times for training and exactly once for testing. When using cross-validation, the data set of N instances is divided into k subsets (usually called *folds*). The model is trained with all the subsets except one and the validation error is measured by testing the subset left out. This procedure is repeated for a total of k trials, where $1 < k \leq N$, and is referred to as *k-fold cross-validation*. The evaluation of the model is assessed by averaging the performance on the test sets over all the k trials. When there are few instances in the data set, the *leave-one-out* approach could be used, which means that $k = N$. An obvious drawback of the leave-one-out approach is that it is very computationally intensive. A more common approach is to let $k = 10$, which is referred to as *10-fold cross-validation*. k-fold cross-validation is a very common procedure when comparing classifiers and reporting results, but it is susceptible to some problems. First of all, the test sets are usually rather small, leading to large confidence intervals. Secondly, there is much overlap in the training sets, leading to some risk that the classifiers will depend on the distribution of classes in the fold used. To minimize this risk, it is sometimes suggested that the crossvalidation procedure should be repeated several times. One typical example sometimes seen is to run 10-fold cross-validation ten times, an approach called *10x10 fold cross-validation*. Obviously, this approach is rather computationally intensive and does not allow differences in performance to be tested for significance.

Stratification can be used with any validation method and the purpose is to make sure that the values of the critical variable, usually the target variable, are evenly distributed over all folds.

All approaches presented so far assume that the instances are sampled without replacement. But an alternative is to draw instances used for training with replacement, possibly leading to a certain degree of duplicates, as, e.g., in bootstrapping. The standard bootstrapping approach is called 0.632 bootstrap, since each bootstrap contains approximately 63.2% of the instances, and is normally repeated a number of times where the overall result is obtained by combining the performance on each bootstrap sample. Specifically, the accuracy is calculated using Equation (4.1).

$$accuracy_{.632bootstrap} = \frac{1}{b}\sum_{i=1}^{b} 0.632acc_i + 0.368acc_{tot} \tag{4.1}$$

where b is the number of bootstraps, acc_i is the accuracy on bootstrap i and acc_tot is the accuracy achieved by a model trained with the whole data set, and measured on the same set.

4.3.3 Methods for Comparing Classifiers on a Single Data set

An important distinction that has to be made when discussing the accuracy or error rate of a classifier is between the sample error and the true error. The sample error of a classifier, with respect to some sample of instances S, drawn from the space of possible instances X, is the fraction of S that the classifier misclassifies. The true error p is the probability that the classifier misclassifies an instance drawn at random from the distribution D. The distribution D specifies for each possible instance $x \in X$ the probability that, if just one instance is drawn randomly from X, this instance is x.

4.3.4 Methods for Comparing Classifiers on Multiple Data sets

One important question when presenting results where algorithms are evaluated on different data sets is how to validate that one algorithm is better than others. Different kinds of statistical tests have to be used depending on the experiments performed.

In the paper by Demšar [122], a thorough theoretical and empirical evaluation is given of which statistical tests are most suitable when comparing two or more algorithms on many data sets. After analyzing contributions to major conferences in machine learning in recent years, he concludes that many researchers in the field are unsure of which tests are appropriate for evaluating the differences between algorithms.

Regarding the comparison of two algorithms on many data sets, the preferable statistical test to use, according to Demšar, is the non-parametric *Wilcoxon signed-ranks test*. The Wilcoxon test is preferred over paired t-tests because the assumptions of t-tests might be violated when evaluating using real-world problems. Demšar concludes that since the sample size (i.e., the number of data sets) is usually small (around 30 data sets), the t-test requires that the differences between the two random variables compared (i.e., the algorithms) are distributed normally. He concludes his reasoning about the paired t-tests with the following statement:

> For using the t-test we need normal distributions because we have small samples, but the small samples also prohibit us from checking the distribution shape. [122]

In the experiments reported in Demšar's paper, it is shown that the Wilcoxon test is the most powerful and also the most reliable when considering Type I errors, i.e., rejecting a null hypothesis when it should have been accepted.

The main principle in the Wilcoxon test is that the differences in performance between the two classifiers for each data set are ranked, ignoring signs (i.e., based on absolute values). The smallest difference gets rank 1 etc. In case of ties, the average ranks are used. Then the sums of the ranks for positive and negative differences are compared. Let $R+$ be the sum of the ranks for the data sets on which the second algorithm outperformed the first, and $R-$ the sum of the ranks for the opposite. Equal ranks are split evenly between the sums; if there is an odd number of them, one is ignored. The test statistic is $T = min(R+, R-)$. Most books on general statistics include a table of exact critical values for T for N up to 25 (or sometimes more). For a larger number of data sets, the statistic

$$z = \frac{T - \frac{1}{4}N(N+1)}{\sqrt{\frac{1}{4}N(N+1)(2N+1)}} \tag{4.2}$$

is distributed approximately according to the normal distribution.

When comparing the Wilcoxon signed ranks test to the t-test, Demšar points out that it assumes commensurability but only qualitatively: greater differences still count more, but absolute magnitudes are ignored. The Wilcoxon signed ranks test is also safer, since it does not assume normal distributions. Furthermore, the outliers have less effect on the Wilcoxon than on the t-test. When normal distributions can be assumed, Demšar recommends using the t-test. Finally, he argues that another commonly used test, the sign test, is weaker than the other tests when considering Type I errors.

When considering comparisons of many algorithms over many data sets, there are basically two different approaches. In the first approach, all algorithms are compared to all other algorithms, while in the second approach only one algorithm is compared to all the others. Depending on which approach is used, different statistical tests are appropriate.

When comparing many algorithms, the test procedure is usually performed in two steps. First, a test is used to determine if the null-hypothesis that no significant differences exist between any algorithms can be rejected. If the null-hypothesis is rejected, a post-hoc test is used to identify which algorithms actually differ. Depending on which test is used, different post-hoc tests ought to be used.

Demšar recommends the non-parametric Friedman test as the first step. The parametric ANOVA test turns out to be a dubious choice, because the assumptions of the ANOVA test cannot be guaranteed to be met when analyzing the performance of machine learning algorithms. When the assumptions of the ANOVA test are met, the ANOVA test is often more powerful than the Friedman test. If the assumptions of ANOVA are violated, the Friedman test can be more powerful. However, it is not trivial to prove that the assumptions are met for any given set of algorithms and/or problems.

When using the Friedman test to compare k algorithms over N data sets, the algorithms are ranked for each data set separately, with the best performing algorithm getting rank 1 etc. In case of ties, average ranks are assigned. Under the null-hypothesis, which states that all the algorithms are equivalent and so their ranks should be equal, the Friedman statistic

$$\chi_F^2 = \frac{12N}{k(k+1)}\left(\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right) \tag{4.3}$$

is distributed according to χ_F^2 with $k-1$ degrees of freedom, when N and k are big enough (as a rule of a thumb, $N > 10$ and $k > 5$). For a smaller number of algorithms and data sets, exact critical values have been computed.

Iman and Davenport [123] showed that Friedman's χ_F^2 is undesirably conservative and derived a better statistic

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2} \tag{4.4}$$

which is distributed according to the F-distribution with $k-1$ and $(k-1)(N-1)$ degrees of freedom. The table of critical values can be found in any statistics book.

Even if the Friedman test indicates a significant difference, the test does not indicate which algorithms might differ. To determine which algorithms have statistically different performances, post-hoc tests are used. The *Nemenyi test* is used when all algorithms are compared to each other. The performance of two algorithms is significantly different if the corresponding average ranks differ by at least the critical difference given below

$$CD = q_a\sqrt{\frac{k(k+1)}{6N}} \tag{4.5}$$

The table of critical values can be found in any statistics book.

When, instead, one specific algorithm is compared to all the others, it is better to use one of the general procedures for controlling the family-wise error, i.e., the probability of making at least one Type I error in any of the comparisons. The test statistic for comparing algorithms i and j using these methods is presented below.

$$z = \frac{R_i - R_j}{\sqrt{\frac{k(k+1)}{6N}}} \tag{4.6}$$

The z value is used to find the corresponding probability from the table of the normal distribution, which is then compared to an appropriate α. The tests differ in the way they adjust the value of α to compensate for multiple comparisons. When using the *Bonferroni–Dunn test*, the same equation as for the Nemenyi test is used to calculate the CD. Again, the table of critical values can be found in any statistics book. The power of the post-hoc test is much greater when comparing classifiers with only one control classifier (which is the case when the Bonferroni–Dunn is used) instead of when comparing all classifiers to all other [124].

5. Summary of Papers

This chapter summarizes the individual contributions for each paper.

5.1 Papers Related to the Question: Which Strategy Is Most Effective to Use When Creating Ensembles: The Implicit or the Explicit Learning Strategy?

The first three papers are all focused on identifying the best way to apply the static overproduce-and-select paradigm by evaluating different optimization criteria based on performance and/or diversity measures. The results from papers I, II and III, as well as results from several other studies not part of this thesis, had all suggested that it might not be viable to try to find an optimization criterion based on performance and/or diversity measures and using it to find a sub-ensemble that is better than using the entire pool as an ensemble. The lessons learned from these earlier studies were further discussed in paper V. The insights also led to a shift of focus, from the explicit learning strategy, towards using an implicit learning strategy and identifying how to train the base classifiers to obtain ensembles that would perform as well as possible. Papers IV and VI both elaborate on how, using an implicit learning strategy, to train the base classifiers to obtain ensembles with good predictive performance.

5.1.1 Paper I: Empirically Investigating the Importance of Diversity

Paper I contributes to the first sub-question and indirectly to the main research question by investigating the best criterion to use in the overproduce-and-select paradigm.

The study introduces diversity as an optimization criterion when adopting the static overproduce-and-select paradigm. In the paper, ten diversity measures were evaluated as selection criteria when selecting a sub-ensemble from a pool of ensemble members. The general conclusion was that no diversity measure is better than accuracy as a selection criterion. A few diversity measures were singled out as more promising than the others: namely, difficulty, but also to a lesser degree Kohavi–Wolpert, coincident failure diversity, and double fault.

A problem identified in the study was that very small ensembles were allowed. This will lead to ensembles with very poor accuracy, allowing measures that are size dependent, such as double fault, to appear better than they are.

The study introduces the concept of using a combination of diversity and performance measures as selection criteria. It is shown that combinations of accuracy and one or more diversity measures as a selection criterion seems to be an approach worth further investigation. This connects to papers II and III, where this idea is developed further.

5.1.2 Paper II: On the Use of Accuracy and Diversity Measures for Evaluating and Selecting Ensembles of Classifiers

In paper II[1] , the idea of combining performance and diversity measures was further evaluated. In the first experiment it was shown that combinations of performance and diversity measures were, as selection criteria, at least as good as using only performance measures. In the second and third experiments, multi-objective GA was used to search for ensembles that were both accurate and diverse. When having a Pareto optimal set of best solutions, it was generally best to select ensembles that balanced accuracy and diversity, or to select the most diverse ensembles in the set.

Consequently, this paper also contributes to the first sub-question and indirectly to the main research question by investigating the use of combined measures as optimization criterion in the static overproduce-and-select paradigm.

This paper ends by suggesting the study performed in Paper III.

5.1.3 Paper III: Ensemble Member Selection Using Multi-Objective Optimization

Paper III contributes to the main research question and to the first sub-question by introducing and evaluating a novel algorithm that, for each dataset, searches for the most effective combination of performance and diversity measures to use as optimization criteria when adapting the static overproduce-and-select paradigm. The algorithm was evaluated on both ensembles of neural networks and ensembles of decision trees. The algorithm works in two steps: in the first step, a pool of randomly selected sub-ensembles are used to search for the most effective weighted combination of measures, and in the second step, the combination found is used to search among all possible sub-ensembles. In the first step, a multi-objective GA was used that optimized two different fitness

[1] There is an error in the list of used datasets (Table 2). The list contains 30 datasets but only results for 27 datasets are reported. It has not been possible to identify with certainty which datasets should be excluded.

functions. The first fitness function was the ranking capability over the entire pool, measured using the correlation between the combination of measures and ensemble accuracy on a hold-out set. The second fitness function was the average ensemble accuracy among the 5 % highest ranked ensembles in the pool.

The proposed algorithm worked fairly well for ensembles of neural networks. For decision trees, however, it was just as effective as always using only ensemble accuracy as the selection criterion. The main contribution of these first three papers is, as mentioned in the Introduction (Section 5.1), that the underlying assumption of the overproduce-and-select paradigm appears more and more questionable.

5.1.4 Paper IV: Comparing Methods for Generating Diverse Ensembles of Artificial Neural Networks

In paper IV, a number of both implicit and explicit techniques for creating ensembles were compared. The evaluated algorithms were ensembles of neural networks, with and without bagging as well as randomized architecture of the neural networks, an overproduce-and-select algorithm called GASEN, searching for a better sub-ensemble from the previously mentioned ensembles of neural networks, and an algorithm called NegBagg that trains neural networks in parallel using negative correlation learning and a constructive approach which automatically determined the architecture of the neural networks. The paper also made an attempt to explain the relative performance in terms of accuracy and diversity.

The results were very clear and showed that the simplest approach, using ensembles of neural networks with randomized architecture, outperformed the more complex explicit algorithms, regardless of whether bagging was used or not. GASEN was significantly less effective than the ensembles using all the models GASEN selected from. The NegBagg algorithm was used both with and without negative correlation learning, but without any significant difference in performance. Both variants of Negbagg achieved comparably low average individual accuracies compared to the implicit ensembles, perhaps as a consequence of the constructive approach used to decide the architecture of the neural networks. Both the NegBagg algorithms were significantly less effective than the implicit ensembles used as comparison.

When analyzing base classifier accuracy and diversity (double fault), the expected result was confirmed: the ensembles trained without bagging had a much higher base classifier accuracy and a much lower diversity, and ensembles trained with bagging were more diverse and less accurate, on average. Thus, it was clearly shown that the ensemble accuracy could not be explained only by considering average accuracy or diversity. Instead, ensembles could be successful either by having accurate base classifiers or by having base classifiers that are sufficiently diverse.

Since explicit and implicit ensemble techniques are compared in the empirical evaluation in that paper, it directly contributes to the first sub-question by strongly indicating that the implicitly trained ensembles were superior to the evaluated explicitly created ensembles. The results further strengthen the doubt regarding the underlying assumptions of the overproduce-and-select paradigm.

In the discussion section of paper IV, the ideas that led to the study performed in paper VI were discussed.

5.1.5 Paper V: Overproduce-and-Select: The Grim Reality

In paper V, the basic assumption, that it is possible to find a sub-ensemble that is better than the ensemble formed by the entire pool of base classifiers (the all-ensemble), which is what motivates the static overproduce-and-select paradigm, was thoroughly evaluated. The study first established that there were a number of sub-ensembles that were actually better than the all-ensemble and it was shown that a substantial subset of the evaluated sub-ensembles were indeed better. The proportion that was better seemed to be dependent on the size of the sub-ensemble. Once the prerequisite, that sub-ensembles better than the all-ensemble existed, had been confirmed, a number of comparisons with the aim of determining if it was possible to identify any of the better sub-ensembles using any of the evaluated selection criteria was carried out.

The main result was that there is absolutely nothing to gain by selecting an ensemble based on the very natural metrics evaluated. When using larger ensembles (i.e. picking 51 neural networks from the 100 pool), each ensemble was quite accurate, actually comparable to using the entire pool as the ensemble, but there was no way to detect the best ensembles. On the other hand, when using smaller ensembles (7 ANNs) it was possible to distinguish poor ensembles from better ones by comparing either ensemble accuracy or mean base classifier accuracy on out-of-bag. Unfortunately, the results also showed that in this scenario, even the most promising ensembles could not compete with the strategy of using all available models as the ensemble.

The main explanation identified in this study is the robustness inherent in ensembles. At least for ANN models and these fairly small data sets, the often marginal differences in performance on training, validation or OOB data, will simply not carry over to test data.

By thoroughly investigating the underlying assumption of the static overproduce-and-select paradigm, this paper contributes directly to both the main research question and to the first sub-question by its clear demonstration that the previously mentioned doubts were justified.

5.1.6 Paper VI: Producing Implicit Diversity in ANN Ensembles

Paper VI evaluated a number of straightforward techniques for introducing implicit diversity in ensembles of neural networks. The study also briefly introduced a novel algorithm later presented in paper XXII. Four different parameters were altered: the number of epochs for which each neural network was trained; using bagging or not; the architecture of the neural networks; and the number of features used when training each neural network. Altogether 54 different setups were evaluated. Two typical baseline setups were identified. In both setups, all features were used when training, all networks had the same architecture and were trained for a fixed maximum number of epochs. The difference between the two baseline setups were whether ordinary bagging was used or not.

The results showed that most of the evaluated setups outperformed the baseline setup without bagging. More specifically, when using larger ensembles, with 51 members instead 15, a majority of the setups were significantly better. Furthermore, a majority of the setups were also at least as good as the baseline setup using bagging and some of the setups were significantly better when using larger ensembles. The levels of increased diversity produced by the methods evaluated normally resulted in increased ensemble accuracy, i.e. diversity was produced without lowering the base classifier accuracy to such an extent that it affected ensemble accuracy. Furthermore, the results also suggested that diversity is more important for larger ensembles.

This paper could be seen as a continuation of paper IV by following up the success of the implicitly trained ensembles of neural networks. It contributes directly to the main research question and to the first sub-question by its investigation of how to make the implicitly trained ensembles even better.

5.2 Papers Related to the Question: How Should Data Be Utilized Effectively in Confidence-Based Predictions Using Ensembles?

In all the previous papers, ensemble creation using either implicit or explicit learning strategies was the focus. In papers VII and VIII, the focus was instead directed towards how to effectively create confidence based predictors. Both these papers focus on the conformal prediction framework and both of them use ensembles to create confidence based predictors.

5.2.1 Paper VII: Effective Utilization of Data in Inductive Conformal Prediction Using Ensembles of Neural Networks

Paper VII focused on how to maximize the efficiency when using bagging ensembles as the underlying models in conformal prediction. The study compared four different setups. The baseline setup, inductive conformal prediction (ICP, see Section 3.1), divided the available data into a proper training set and a calibration set, whereas the the other three setups in different ways used all available data both for training the models and for calibration.

In two of the evaluated setups, different re-sampling schemes were used to make sure every training instance could be used as a calibration set at least once. This was achieved by training multiple models and using different data as the calibration set every time, and then combining all the models into one conformal prediction. In the first of these setups, cross conformal prediction (CCP, see Section 3.2.1), a cross validation strategy, was used to divide the data, and in the other of these two setups, bootstrap conformal prediction (BCP, see Section 3.2.2), a bootstrap strategy, was used. In the final setup, standard bagging was used, where all the data was used to train the model. In this setup, the out-of-bag results were used as the calibration set, thus it is called out-of-bag (OOB, see Section 3.2.3). The solution that turned out to be most efficient was to use a bagging ensemble and use the out-of-bag results as calibration set. The bagging ensemble also had the advantage of having no additional parameters to consider. When using the other methods, the available data have to be divided somehow (ICP) or multiple models need to be built and combined (CCP and BCP). A longer discussion and motivation for the use of out-of-bag estimates as the calibration set is given in paper XXV.

This paper contributes indirectly to the main research question and directly to the second sub-question by its investigation of how to use the inherent benefit of bagging ensembles when predicting with confidence.

5.2.2 Paper VIII: Bias Reduction through Conditional Conformal Prediction

Also in this final paper the focus is on conformal predictions. But unlike the previous papers, where the focus was directly or indirectly on ensembles, this study focused on to what extent class imbalance affects the outcome of conformal prediction and to what extent conditional conformal prediction can be used to counter these effects. However, ensembles were still used, together with decision trees, as the underlying models in the evaluation.

In a number of experiments it was shown that conformal prediction was highly biased towards the majority class. We also showed that this bias meant that we had no guarantees that the errors were distributed according to the prior class distribution when using conformal prediction. Instead, conformal prediction could be expected to make a majority of its errors on the minority class. The greater efficiency, measured using the prior efficiency criterion OneC, achieved by conformal prediction was an effect of this bias towards making errors on the minority class, making it possible to be very efficient on the majority class and consequently also in general. Furthermore, the presented results showed that class conditional conformal prediction was able to remedy all the drawbacks of conformal prediction, even if the former still had a lower OneC. Class conditional conformal prediction was valid, both in general, like conformal prediction, and for each class individually, in contrast to conformal prediction. Class conditional conformal prediction was generally not biased.

Class conditional conformal prediction did not achieve its efficiency (using the OneC criterion) by making all errors on the class that was easiest to get rid of, i.e. the minority class. Instead, it was generally almost equally efficient on both classes. In fact, it was equally or more efficient than conformal prediction when only considering the minority class, but without making as many mistakes on that class. Furthermore, class conditional conformal prediction was much more efficient when the two observed criteria only considering one class at a time described last in Section 3.1.1 was used to measure the efficiency.

Regarding ensembles, the results imply that even though the ensembles used as the underlying models were slightly more biased than the decision trees used as a comparison, they were still more effective as underlying models, resulting in more efficient conformal predictors.

Since the difference between ordinary conformal prediction and class label conditional conformal prediction lies in what data is used as the calibration set, this paper contributes to the second sub-question. Since it results in recommendations regarding when to use class label conditional conformal prediction when predicting with confidence, it also contributes directly to the main research question, even if these recommendations are not limited to ensembles.

6. Concluding Remarks

This chapter starts with the presentation of the conclusions regarding the sub-questions before presenting the conclusions regarding the research question. The chapter ends with a presentation of future work.

6.1 Discussion

In this discussion section, the results and conclusions presented in the included articles will be discussed in relation to related research as presented in Chapters 2 and 3. The discussion is structured around the sub-questions presented in Section 1.3.

6.1.1 Discussion of Whether an Explicit or an Implicit Learning Strategy is Preferable for Creating Effective Ensembles

The first six studies all address the first sub-question in different ways. The topics covered in these papers include: using combinations of diversity and/or performance measures; selection of ensemble members; diversity creation strategies; the explicit learning strategy; and the implicit learning strategy. Furthermore, there have also been several studies published by other authors on this topic since the first article included in the thesis was published. In the following discussion, the contributions of the six articles will be discussed from different perspectives and also put in relation to the findings presented by other researchers. The different perspectives are, in order of discussion: the explicit learning strategy; using combinations of diversity and/or performance measures; selection of ensemble members; the implicit learning strategy; diversity creation strategies; and the relationship between diversity and accuracy.

A major distinction exists between ensembles created using an implicit or an explicit learning strategy. The role of diversity is to a large degree different for the two learning strategies. When creating an ensemble using an implicit learning strategy, understanding the role of diversity can be used to better understand how the individual models should be trained in order to maximize the predictive performance of the ensemble and in this way achieve effectiveness. When instead using an explicit technique to create an ensemble, understanding the role of diversity can be used both to guide how to train the individual models but also to indicate to what extent it is useful to use diversity when selecting which models to add or remove from the ensemble. Consequently, the concept of diversity plays an important role when trying to answer the first sub-question.

Many of the papers presented in this thesis are focused on an explicit strategy, evaluating to what extent diversity can be used as an optimization criterion when trying to optimize which models to include in the ensemble (papers I, II, III, V). Paper IV compares both explicit and implicit techniques in the experiments. The kind of explicit techniques that have been the focus of these papers is the static overproduce-and-select paradigm. The basic assumption when using the static overproduce-and-select paradigm is that it is somehow possible to use information possible to measure on the available data to make a selection from among all the available models to create a smaller ensemble that will be effective when applied to yet unseen data. When considering the role of diversity when creating ensembles using an explicit strategy, it was evident both from the results of the experiments carried out as part of our research, and also from what had been published by others, that most diversity measures proposed for the classification context were clearly not useful by themselves as optimization criteria, even though the static overproduce-and-select paradigm still seemed feasible. In paper I, ten diversity measures and ensemble accuracy were evaluated as selection criteria. The conclusion was that neither diversity nor ensemble accuracy were very useful as a selection criterion. On the other hand, the study showed that combinations of ensemble accuracy and one or several diversity measures (most notably the difficulty measure) was at least as good as selection criteria as any individual measure. Evaluating the use of combinations of measures was, consequently, the focus of paper II and III. The results in paper II were inconclusive regarding whether combinations of performance and diversity measures should be used as selection criteria. In paper III the results actually provided some evidence that a combined measure was useful when selecting an ensemble. The study proposed an algorithm that defined a selection criterion combining several performance and diversity measures where the combination was optimized for each dataset individually. In particular, the optimized combination of measures was significantly better as a selection criterion than using only ensemble accuracy when selecting from the set of ensembles used to optimize the combination of measures. However, using the optimized combination of measures was not very successful when searching among all possible sub-ensembles. Consequently, the optimized combination of measures did not generalize to make it useful as selection criterion outside the set of ensembles already used when defining the selection criterion. In paper IV a static overproduce-and-select algorithm proposed by Zhou et al. [6], GASEN, was compared to implicit algorithms. The sub-ensembles selected by GASEN turned out to perform significantly worse than the ensemble formed by using the entire pool of models that GASEN had used to select from.

The general picture that emerged in the first four papers regarding the static overproduce-and-select paradigm was that it might not be very successful. Consequently, paper V specifically evaluated the basic assumptions motivating the static overproduce-and-select paradigm. It was evident from the results that there is nothing to gain by selecting an ensemble based on any of the optimization criteria evaluated in the study. The evaluated optimization criteria included both performance measures such as ensemble accuracy as well as a number of diversity measures. So, even though the conclusions in the first five papers confirmed previous research regarding to what extent diversity measures were suitable as selection criteria, the explanation was to a large degree complementary to previous research by pointing out the inadequacy of the static overproduce-and-select paradigm, i.e., the strategy of selecting a subset of models from a pool rather than using the entire pool as an ensemble.

It might seem strange that the static overproduce-and-select paradigm has been used in several studies by other authors if it does not work. After all, the previous studies [8–10; 79–81], including the third paper in this thesis, have been performing as well as or better than the techniques used for comparison. However, no one of the studies previously presenting an algorithm using the static overproduce-and-select paradigm had produced results that could show a significant improvement compared to the ensemble achieved by not making any selection at all. Consequently, the natural evaluation that could have indicated the inadequacy of the strategy had not been carried out or, when it was carried out, did not show any significant advantage for the static overproduce-and-select paradigm.

Even though the static overproduce-and-select paradigm is ineffective in the sense that it is unlikely to result in ensembles that perform better than the unreduced ensemble, there could still be valid reasons for wanting to attempt to reduce the size of an ensemble. If the prediction time needs to be kept to a minimum or the production dataset is very large, reducing the size of the ensemble without drastically decreasing the performance of the ensemble could be justified.

It must in this context be noted that the discussion so far applies to the static overproduce-and-select paradigm. The dynamic overproduce-and-select paradigm, where a sub-ensemble is selected for each instance, has not been evaluated and it is possible that it is a more viable solution.

The overproduce-and-select paradigm is not the only way to create ensembles using an explicit learning scheme. The negative correlation learning algorithm is a parallel explicit ensemble learning algorithm which explicitly optimizes the covariance in the bias, variance, and covariance decomposition of the ensemble error (see Equation 2.5). Obviously, the negative correlation learning algorithm uses diversity as a means to achieve accurate ensembles. It was initially defined for regression problems, but it can be used for classification if the models produce measurement results. The only algorithms utilizing negative correlation learning evaluated in any of the papers was the NegBagg algorithm used in paper IV, where the algorithm was used both with and without negative correlation learning. Using the negative correlation learning did not result in any significant improvement of performance. The performance of both the NegBagg algorithms were significantly worse compared to the implicit ensembles used as comparison. This could probably be attributed to the fact that both variants of Negbagg achieved very low average individual accuracy. The NegBagg algorithm also used a constructive approach to decide the architecture of the neural networks and it is possible that an ensemble trained using only negative correlation learning with suitably adjusted architectures (instead of the constructive approach adopted by NegBagg) could perform better. The results indicated that negative correlation learning lead to more diverse ensembles but it was not enough to outweigh the effect of the low average individual accuracy. Chen proposed a regularized negative correlation learning algorithm that was shown to perform better than the negative correlation learning algorithm.

When considering implicit algorithms, i.e., algorithms that might create diversity without actually targeting it, two of the papers evaluated such algorithms. In paper IV, ensembles created with and without bagging were compared. Apart from using bagging, diversity was also introduced by using slightly randomized architectures in the neural networks used in the ensembles. The most obvious result was that both the implicit algorithms outperformed all the explicit algorithms. When considering ensemble accuracy, the two approaches were comparable. However, the explanation for the success was rather different, since the ensemble using bagging achieved significantly lower average individual accuracy while at the same time being significantly more diverse (using the double fault measure). Consequently, it was possible to achieve effective ensembles either by training very accurate but less diverse individual models (without bagging) or by creating less accurate but diverse individual models (using bagging). This was further evaluated in paper VI, were a systematic evaluation of altogether 54 different ways of training neural network ensembles were evaluated and compared. It was evident from the results that there were several different ways to create an effective ensemble. First of all, the same results as seen in paper IV regarding the differences between ensembles trained with or without bagging was strongly confirmed. Again, it was possible to achieve effective ensembles either by training very accurate but less diverse individual models (without bagging) or by creating less accurate but diverse individual models (using bagging). In general it seemed to be slightly better to use bagging. One interesting result that was seen when training the neural networks with or without the random subspace method (randomly disregarding 10% of the input variables when training each network) was that no increase in diversity could be seen. Instead, both diversity (measured using the difficulty and double fault measures) and average individual accuracy decreased. Not surprisingly, the result was that the use of the random subspace method produced the least effective ensembles, regardless of whether bagging was used or not. Three different strategies for deciding the architectures of the networks was used but the difference between the three strategies was remarkably low. The first strategy was to use the same architecture, based on a simple heuristic, in all networks; the second was to randomize the number of hidden nodes in the hidden layer; the third was to use two hidden layers with randomized numbers of nodes in both. The third strategy resulted in slightly lower average individual accuracy, which was compensated for by slightly higher diversity. The most encouraging result was that a new algorithm, random brains, later presented in paper XXII, turned out to be the most effective. The new algorithm mimics how the random subspace method is used internally in random forests by randomly removing internal links between the layers. It turned out that the use of internal random subspace resulted in the most accurate individual networks on average without reducing the degree of diversity when using one hidden layer. The difference between a fixed and a random architecture is small but slightly in favor of a fixed architecture. When instead using two hidden layers, the average individual accuracy is reduced rather a lot compared to when using only one hidden layer. On the other hand, diversity, measured using difficulty, is dramatically increased. The result is that the use of internal random subspace and an architecture with two hidden layers together with bagging results in ensembles with the highest degree of diversity, almost the lowest average individual accuracy, and among the highest ensemble accuracies. Using the same algorithm without bagging is equally effective even though the average individual accuracy is higher and the degree of diversity lower. To summarize the various findings, it is evident that there are several different effective ways to create ensembles of neural networks. An interesting result regards the difference in results when applying the random subspace method in different ways. While the random subspace method (randomly removing variables prior to training) was most clearly ineffective, it turned out to be effective to use random subspace (by randomly removing the input links) for each internal node in the neural networks.

The results achieved in paper IV and VI are in line with the results presented by Kuncheva in a recent paper on diversity. She uses plots to visually show several of the conclusions that could be drawn from the experiments in paper VI.

In recent years, there have been important developments in the understanding of the connection between diversity and accuracy when considering ensembles using majority voting in classification. One of the two most notable developments has been the decomposition of the ensemble error into the average individual error and a diversity part for ensembles using majority voting [46]. The diversity part can in turn be divided into two different diversity parts, representing constructive and destructive forces. By this accomplishment, the same degree of understanding of the ensemble error and its connection to diversity can be said to exist for both regression and classification problems. The second notable development is the contribution by Stapenhurst in his dissertation [60] where he shows how most proposed diversity measures can be expressed as margin measures. Through his accomplishment, it becomes more natural to use the better understood margin theory to analyze ensemble performance. However, the decompositions of ensemble error are obviously still valid, which, in combination with the fact that some of the diversity measures (Q statistics, ρ correlation, and the pairwise κ coefficient measures) could not be expressed as margin measures, means that there could still be some relevance in continued studies of diversity.

6.1.2 Discussion on How to Effectively Utilize Data in Confidence Based Predictions Using Ensembles

Two different papers (papers VII and VIII) addressing different aspects of how to effectively utilize data in confidence based predictions using ensembles are included in this thesis. Both papers use the conformal prediction framework to make confidence based predictions. The two papers complement each other by focusing on two very different aspects of how to effectively utilize data in confidence based predictions. In inductive conformal prediction, part of the data available for training must be withheld during the training of the model to be used as a calibration set when calibrating the confidence bounds. Not being able to use all the data for training is likely to decrease the effectiveness of the model. When using bagging ensembles, it is possible to use the out-of-bag estimate as the calibration set. Since about one-third of the models in the ensemble have not been trained using each training instance, the out-of-bag estimates will be estimated on much smaller ensembles and consequently overestimate the errors (except for very large ensembles). Using the out-of-bag estimates as a calibration set was evaluated for the first time in paper VII, even though it had been proposed previously in the literature. It turned out to be more effective than setting aside part of the data to be used as a calibration set, but also more effective than different forms of aggregated confidence predictors that used cross validation or bootstrapping to be able to utilize all data for training. In a later paper (paper XXV), it was argued that the guarantees provided by the conformal prediction framework would still hold because the out-of-bag estimates are known to overestimate the true error. Since the out-of-bag estimates overestimate the errors (except for very large ensembles), it will at worst result in the conformal predictor's being conservative. One way of dealing with the difference between the ensemble predicting the test instances and the out-of-bag estimates is to use a random proportion of the models in the ensemble when predicting the test instances as well. If the proportion of models is the same as for the out-of-bag estimates, the errors can be expected to be the same on the test instances as on the out-of-bag estimates used as the calibration set. Using the out-of-bag estimates is made possible by training bagging ensembles as the underlying models. Consequently, it is the use of ensembles that provides the opportunity to create more effective confidence based predictions in this case.

Turning to the final paper included in the thesis (paper VIII), another issue within the classification context was addressed: the imbalanced learning problem. When using conformal prediction, the user is guaranteed that the prediction will include the correct class with some user specified confidence. However, no guarantee is given regarding how the errors made will be distributed among the classes. When dealing with imbalanced data, where it is usually the minority class that is of most interest, the distribution of errors becomes vital, since a confidence based predictor that made all its errors on the minority class would be less useful or even useless. The paper investigated how the errors are distributed among classes when using conformal prediction and it was clearly demonstrated that the conformal predictors had a strong tendency to make most of the errors on the minority class, i.e., it was highly biased towards the majority class. As a comparison, a class label conditional conformal predictor was used, which can provide a guarantee also for each class. As expected, the class label conditional conformal predictor was not biased, since the errors were distributed in accordance with the prior class distribution. The implication is that conformal prediction is not effective on even slightly imbalanced datasets since it is probable that the errors will be made primarily on the minority class. If it is important that the proportion of errors be distributed as the prior class distribution, class label conditional conformal prediction must be considered much more effective since it ensures both the error level and unbiased predictions. When using ensembles as the underlying models, if the individual models are biased towards the majority class, the ensemble can be expected to be biased as well. In fact, the ensembles might even aggregate the biases of the individual models, making the ensembles more biased than the average individual model. The results indicate that the random forest ensembles used as underlying models were slightly more biased than the individual decision trees also used as underlying models in the experiments. The motivation for using ensembles as the underlying models in conformal prediction is that they generally result in more efficient conformal predictors. The results imply that even though the ensembles might have been slightly more biased than the decision trees, they still produced more efficient class label conditional conformal predictors.

6.2 Conclusions

First the conclusions regarding the sub-questions will be presented, before presenting the conclusions regarding the research question: *How can ensembles be created effectively in the context of classification?* The first sub-question was: *Which strategy is most effective to use when creating ensembles: the implicit or the explicit learning strategy?*

In this context, effectiveness was defined to be ensemble accuracy on the test set. The more accurate an ensemble is, the more effective it is considered to be.

We will start with the conclusions based on the papers presented in the thesis. Since several of the papers have covered different topics related to the adoption of explicit or implicit learning strategies when creating effective ensembles, the conclusions will also be presented based on these different topics. The first topic to consider is whether it is effective to use a static overproduce-and-select paradigm when creating ensembles. The conclusion is that the static overproduce-and-select paradigm is an ineffective way to create ensembles, at least for smaller datasets. The reason identified is that there is no way to detect, using the available data, which of the smaller ensembles will be better than the unreduced ensemble on the test or production set.

If the static overproduce-and-select paradigm is ineffective it follows that using diversity measures or combinations of diversity and/or performance measures as optimization criterion is also ineffective. The reason is obviously that if it is not effective to search for a smaller ensemble using any optimization criterion, it doesn't matter much what is optimized.

However, there are other ways of creating ensembles using an explicit learning strategy which utilizes diversity in the creation process. Different algorithms using negative correlation learning have been presented as successful in the literature but were not competitive in the evaluation carried out in one of the papers presented in this thesis. One possible reason why they were not competitive could be the constructive approach used to decide the architecture of the neural networks used.

When considering ensembles using an implicit learning strategy and more specifically ensembles of neural networks, which is what were studied, it is possible to get effective ensembles in several different ways. The conclusion was that the only method that was clearly ineffective was to use the random subspace method. However, mimicking the random subspace method internally in the networks, by randomly removing input links to each node in the network, turned out to be the most effective way to create an ensemble. The connection between ensemble accuracy, average individual accuracy, and diversity is evident in the sense that a high ensemble accuracy could be achieved either with very accurate individual models or with very diverse models. An important implication of the presented results is that ensembles of neural networks are generally very robust.

An important fact with direct relevance when creating effective ensembles and analyzing their effectiveness is that almost all diversity measures can be expressed as margin measures. This contribution to the understanding of diversity was recently made in the literature. The implication is that the importance of studying ensemble effectiveness in terms of diversity is probably going to decrease since the margin is a more studied and better understood concept. However, the theoretical results achieved, where the ensemble error has been decomposed into the average individual error and diversity, are obviously still valid.

The second sub-question was: *How should data be utilized effectively in confidence based predictions using ensembles?*

In this context, effectiveness was defined to be efficiency, as the term is used in the conformal prediction framework (see Section 3.1.1).

When predicting with confidence using inductive conformal prediction, some data needs to be set aside during training to enable the calibration of the confidence bounds. When it is possible to use a bagging ensemble, the results show that it is effective to use all the data to train the ensemble and use the out-of-bag estimates as the calibration data. Even if it has not been proven to be valid, theoretical reasoning suggests that it is likely to be valid or even slightly conservative, since the out-of-bag estimates are known to overestimate the actual error.

Inductive conformal prediction is proven to be valid, which means that the prediction will contain the true class with a user defined level of certainty. However, when dealing with imbalanced learning problems it is not only important to know the probability of making an error, it is also important to know which errors are made. It was clearly shown that inductive conformal prediction is strongly biased towards the majority class, i.e., the proportion of errors made on the minority class were much higher than the proportion of errors made on the majority class. On problems with a highly imbalanced class distribution, all or almost all errors were made on the minority class before any errors were made on the majority class. Class label conditional inductive conformal prediction is a specialization of inductive conformal prediction where the calibration data is used in a different way. The validity of the class label conditional inductive conformal prediction is stronger and also applies to each class. As expected, class label conditional inductive conformal prediction was shown to be unbiased, even for strongly imbalanced datasets. The implication is that the general way data is used by inductive conformal prediction is ineffective even for slightly imbalanced datasets and should be exchanged for the way class label conditional inductive conformal prediction uses the data whenever any form of imbalance could be expected in the dataset. The results also imply that it was more effective to use ensembles as underlying models even when they were more biased than the decision trees used as comparison.

The research question was: *How can ensembles be created effectively in the context of classification?*

The papers in this thesis have focused on different aspects of how to create effective ensembles in the context of classification. The two main areas covered relate to implicit versus explicit learning strategies in ensemble creation and the effective use of the data in confidence based prediction. The conclusions strongly indicate that implicit ensembles are generally very robust and effective, making it both generally unnecessary and very hard to find sub-ensembles that are more effective than the unreduced ensembles. Considering prediction with confidence, it was shown that bagging ensembles trained using all the available data made it possible to produce more effective conformal predictors by using the out-of-bag estimates as a calibration set. It was also shown that conformal prediction is very sensitive to class imbalance in the data, resulting in conformal predictors that are strongly biased towards the majority class. When predicting with confidence using imbalanced data, it is therefore strongly recommended to use class label conditional conformal prediction, which was shown to be effective on such problems.

6.3 Future Work

There are a number of questions to look into in future work. First of all, using out-of-bag estimates as a calibration set when using bagging ensembles has been shown to be effective, and so work on proving the validity of this approach will be prioritized.

Another priority is to identify different domains and problems that can be used to showcase the added value of using conformal prediction rather than only using a predictive model. Examples of such domains and problems could be: detecting adverse drug events using patient records; QSAR modeling; anomaly detection in various fields; etc.

Static overproduce-and-select was shown to be ineffective. Several algorithms using dynamic overproduce-and-select, where a sub-ensemble is selected for each instance, exist in the literature. One idea for future work is to use the conformal prediction framework to guide the selection, taking advantage of the possibility of getting confidence estimates for individual instances.

References

[1] Marquis de Condorcet. **Essay on the Application of Analysis to the Probability of Majority Decisions**. *Paris: Imprimerie Royale*, 1785. 1, 17

[2] Lars Kai Hansen and Peter Salamon. **Neural network ensembles**. *IEEE transactions on pattern analysis and machine intelligence*, **12**(10):993–1001, 1990. 2, 4, 11, 23

[3] Anders Krogh and Jesper Vedelsby. **Neural network ensembles, cross validation, and active learning**. *Advances in neural information processing systems*, pages 231–238, 1995. 2, 4, 18

[4] Ludmila I Kuncheva and Christopher J Whitaker. **Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy**. *Machine learning*, **51**(2):181–207, 2003. 2, 5, 20, 22, 23, 34

[5] Gavin Brown, Jeremy Wyatt, Rachel Harris, and Xin Yao. **Diversity creation methods: a survey and categorisation**. *Information Fusion*, **6**(1):5–20, 2005. 2, 3, 29

[6] Zhi-Hua Zhou, Jianxin Wu, and Wei Tang. **Ensembling neural networks: many could be better than all**. *Artificial intelligence*, **137**(1):239–263, 2002. 3, 73

[7] Prem Melville and Raymond J Mooney. **Creating diversity in ensembles using artificial data**. *Information Fusion*, **6**(1):99–111, 2005. 24

[8] Giorgio Giacinto and Fabio Roli. **Design of effective neural network ensembles for image classification purposes**. *Image and Vision Computing*, **19**(9):699–707, 2001. 22, 36, 73

[9] Dragos D Margineantu and Thomas G Dietterich. **Pruning adaptive boosting**. In *ICML*, **97**, pages 211–218. Citeseer, 1997. 36

[10] Giorgio Giacinto and Fabio Roli. **An approach to the automatic design of multiple classifier systems**. *Pattern recognition letters*, **22**(1):25–33, 2001. 36, 73

[11] Gabriele Zenobi and Padraig Cunningham. **Using diversity in preparing ensembles of classifiers based on different feature subsets to minimize generalization error**. In *Machine Learning: ECML 2001*, pages 576–587. Springer, 2001. 24

[12] Guillaume Tremblay, Robert Sabourin, and Patrick Maupin. **Optimizing Nearest Neighbour in Random Subspaces using a Multi-Objective Genetic Algorithm.** In *ICPR (1)*, page 208, 2004.

[13] Eulanda M Dos Santos, Robert Sabourin, and Patrick Maupin. **Pareto analysis for the selection of classifier ensembles**. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 681–688. ACM, 2008.

[14] Eulanda M Dos Santos, Robert Sabourin, and Patrick Maupin. **A dynamic overproduce-and-choose strategy for the selection of classifier ensembles**. *Pattern Recognition*, **41**(10):2993–3009, 2008.

[15] HAYTHAM ELGHAZEL, ALEX AUSSEM, AND FLORENCE PERRAUD. **Trading-off diversity and accuracy for optimal ensemble tree selection in random forests**. In *Ensembles in Machine Learning Applications*, pages 169–179. Springer, 2011. 3

[16] DAVID OPITZ AND RICHARD MACLIN. **Popular Ensemble Methods: An Empirical Study**. *Journal of Artificial Intelligence Research*, **11**:169–198, 1999. 3, 4, 11

[17] LEO BREIMAN. **Random forests**. *Machine learning*, **45**(1):5–32, 2001. 3, 5, 31, 35

[18] AMANDA JC SHARKEY. **Multi-net systems**. In *Combining artificial neural nets*, pages 1–30. Springer, 1999. 3

[19] VLADIMIR VOVK, ALEX GAMMERMAN, AND GLENN SHAFER. *Algorithmic Learning in a Random World*. Springer-Verlag New York, Inc., 2005. 4, 5

[20] LUDMILA I KUNCHEVA. **That elusive diversity in classifier ensembles**. In *Pattern Recognition and Image Analysis*, pages 1126–1138. Springer, 2003. 5, 34

[21] LORENZA SAITTA. **Hypothesis diversity in ensemble classification**. In *Foundations of Intelligent Systems*, pages 662–670. Springer, 2006. 34

[22] E KE TANG, PONNUTHURAI N SUGANTHAN, AND XIN YAO. **An analysis of diversity measures**. *Machine Learning*, **65**(1):247–271, 2006. 5, 27, 34, 35

[23] HAIBO HE AND EDWARDO A GARCIA. **Learning from imbalanced data**. *Knowledge and Data Engineering, IEEE Transactions on*, **21**(9):1263–1284, 2009. 6, 49

[24] THOMAS G DIETTERICH. **Ensemble methods in machine learning**. In *Multiple classifier systems*, pages 1–15. Springer, 2000. 11

[25] TUVE LÖFSTRÖM. *Utilizing Diversity and Performance Measures for Ensemble Creation*. Licentiate thesis, University of Örebro, Sweden, 2009. 11, 55

[26] LEI XU, ADAM KRZYZAK, AND CHING Y SUEN. **Methods of combining multiple classifiers and their applications to handwriting recognition**. *Systems, Man and Cybernetics, IEEE Transactions on*, **22**(3):418–435, 1992. 11

[27] KEVIN WOODS, KEVIN BOWYER, AND W PHILIP KEGELMEYER JR. **Combination of multiple classifiers using local accuracy estimates**. In *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR'96, 1996 IEEE Computer Society Conference on*, pages 391–396. IEEE, 1996.

[28] TIN KAM HO, JONATHAN J. HULL, AND SARGUR N. SRIHARI. **Decision combination in multiple classifier systems**. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **16**(1):66–75, 1994.

[29] LOUISA LAM AND CHING Y SUEN. **Optimal combinations of pattern classifiers**. *Pattern Recognition Letters*, **16**(9):945–954, 1995. 11

[30] VOLKER TRESP. **Committee machines**. *Handbook for neural network signal processing*, pages 1–18, 2001. 11

[31] H SCHWARZE AND J HERTZ. **Generalization in fully connected committee machines**. *EPL (Europhysics Letters)*, **21**(7):785, 1993. 11

[32] MICHAEL I JORDAN AND ROBERT A JACOBS. **Hierarchical mixtures of experts and the EM algorithm**. *Neural computation*, **6**(2):181–214, 1994. 11

[33] ROBERT A JACOBS, MICHAEL I JORDAN, STEVEN J NOWLAN, AND GEOFFREY E HINTON. **Adaptive mixtures of local experts**. *Neural computation*, **3**(1):79–87, 1991. 11

[34] LIN-CHENG WANG, SANDOR Z DER, AND NASSER M NASRABADI. **Composite classifiers for automatic target recognition**. *Optical Engineering*, **37**(3):858–868, 1998. 11

[35] DYMITR RUTA AND BOGDAN GABRYS. **An overview of classifier fusion methods**. *Computing and Information systems*, **7**(1):1–10, 2000. 11

[36] LUDMILA I KUNCHEVA. **A theoretical study on six classifier fusion strategies**. *IEEE Transactions on pattern analysis and machine intelligence*, **24**(2):281–286, 2002. 11

[37] THOMAS G DIETTERICH. **An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization**. *Machine learning*, **40**(2):139–157, 2000. 11, 23

[38] ROBERT HECHT-NIELSEN. **Theory of the backpropagation neural network**. In *Neural Networks, 1989. IJCNN., International Joint Conference on*, pages 593–605. IEEE, 1989. 11

[39] DAVID E RUMELHART, GEOFFREY E HINTON, AND RONALD J WILLIAMS. **Learning internal representations by error propagation**. Technical report, DTIC Document, 1985. 13

[40] PAUL J WERBOS. **Backpropagation through time: what it does and how to do it**. *Proceedings of the IEEE*, **78**(10):1550–1560, 1990. 13

[41] S HAYKIN. *Neural Networks: A Comprehensive Foundation*. Prentice-Hall. Upper Saddle River, NJ, USA, 1998. 14, 31

[42] LEO BREIMAN, JEROME FRIEDMAN, RICHARD OLSHEN, CHARLES STONE, D STEINBERG, AND P COLLA. **CART: Classification and regression trees**. *Wadsworth: Belmont, CA*, **156**, 1983. 14

[43] JOHN H HOLLAND. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence.* U Michigan Press, 1975. 15, 16

[44] RICCARDO POLI, WILLIAM B LANGDON, NICHOLAS F MCPHEE, AND JOHN R KOZA. *A field guide to genetic programming*. Lulu. com, 2008. 16, 17

[45] LUDMILA I KUNCHEVA. *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons, 2004. 18, 32, 33, 36

[46] GAVIN BROWN AND LUDMILA I KUNCHEVA. **"Good" and "bad" diversity in majority vote ensembles**. In *Multiple Classifier Systems*, pages 124–133. Springer, 2010. 20, 25, 76

[47] RICHARD STAPENHURST. **On the Relationship between Ensemble Diversity and Margin Theory**. 2012. 20, 23, 27

[48] G UDNY YULE. **On the association of attributes in statistics: with illustrations from the material of the childhood society, &c**. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, pages 257–319, 1900. 21

[49] TIN KAM HO. **The Random Subspace Method for Constructing Decision Forests**. *IEEE Trans. Pattern Anal. Mach. Intell.*, **20**(8):832–844, 1998. 22

[50] KAMAL M ALI AND MICHAEL J PAZZANI. **Error reduction through learning multiple descriptions**. *Machine Learning*, **24**(3):173–202, 1996. 22

[51] JOSEPH L FLEISS, BRUCE LEVIN, AND MYUNGHEE CHO PAIK. **The measurement of interrater agreement**. *Statistical methods for rates and proportions*, **2**:212–236, 1981. 22, 23

[52] PÁDRAIG CUNNINGHAM. **Overfitting and diversity in classification ensembles based on feature selection**. Technical report, Trinity College Dublin, Department of Computer Science, 2000. 22

[53] RON KOHAVI AND DAVID H WOLPERT. **Bias plus variance decomposition for zero-one loss functions**. In *ICML*, pages 275–283, 1996. 23

[54] DEREK PARTRIDGE AND W KRZANOWSKI. **Software diversity: practical statistics for its measurement and exploitation**. *Information and software technology*, **39**(10):707–717, 1997. 23

[55] ALEXEY TSYMBAL, MYKOLA PECHENIZKIY, AND PDRAIG CUNNINGHAM. **Diversity in Ensemble Feature Selection**, 2003. 23

[56] H. CHEN. *Diversity and regularization in neural network ensembles*. PhD thesis, PhD thesis, University of Birmingham, Birmingham, Great Brittain, 2008. 24

[57] MATTI AKSELA AND JORMA LAAKSONEN. **Using diversity of errors for selecting members of a committee classifier**. *Pattern Recognition*, **39**(4):608–623, 2006. 24, 34, 35

[58] DEREK PARTRIDGE AND WOJTEK KRZANOWSKI. **Distinct failure diversity in multiversion software**. *Res. Rep*, **348**, 1997. 24

[59] LUCA DIDACI, GIORGIO FUMERA, AND FABIO ROLI. **Diversity in classifier ensembles: Fertile concept or dead end?** In *Multiple Classifier Systems*, pages 37–48. Springer, 2013. 26, 27

[60] RICHARD STAPENHURST. *Diversity, margins and non-stationary learning*. PhD thesis, University of Manchester, 2012. 27, 35, 76

[61] NOEL SHARKEY, JOHN NEARY, AND AMANDA SHARKEY. **Searching weight space for backpropagation solution types**. In *Current Trends in Connectionism: Proceedings of the 1995 Swedish Conference on Connectionism*, pages 103–120, 1995. 29

[62] ROBERT PW DUIN AND DAVID MJ TAX. **Experiments with classifier combining rules**. In *Multiple Classifier Systems*, pages 16–29. Springer, 2000. 29, 33

[63] DEREK PARTRIDGE AND WILLIAM B YATES. **Engineering multiversion neural-net systems**. *Neural Computation*, **8**(4):869–893, 1996. 29

[64] YONG LIU. *Negative correlation learning and evolutionary neural network ensembles*. PhD thesis, PhD thesis, University College, The University of New South Wales, Australian Defence Force Academy, Canberra, Australia, 1998. 30

[65] GAVIN BROWN. *Diversity in neural network ensembles*. PhD thesis, PhD thesis, University of Birmingham, 2004. 30

[66] LEO BREIMAN. **Bagging predictors**. *Machine learning*, **24**(2):123–140, 1996. 30, 47

[67] ULF JOHANSSON, TUVE LOFSTROM, AND HENRIK BOSTROM. **Random brains**. In *Neural Networks (IJCNN), The 2013 International Joint Conference on*, pages 1–8. IEEE, 2013. 31

[68] ULF JOHANSSON AND T LOFSTROM. **Producing implicit diversity in ANN ensembles**. In *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pages 1–8. IEEE, 2012. 31

[69] ROBERT E SCHAPIRE. **The strength of weak learnability**. *Machine learning*, **5**(2):197–227, 1990. 31

[70] YOAV FREUND AND ROBERT E SCHAPIRE. **Experiments with a new boosting algorithm**. In *ICML*, **96**, pages 148–156, 1996. 32

[71] DAVID MJ TAX, MARTIJN VAN BREUKELEN, ROBERT PW DUIN, AND JOSEF KITTLER. **Combining multiple classifiers by averaging or by multiplying?** *Pattern recognition*, **33**(9):1475–1485, 2000. 33

[72] DAVID MJ TAX, ROBERT PW DUIN, AND MARTIJN VAN BREUKELEN. **Comparison between product and mean classifier combination rules**. In *Proc. Workshop on Statistical Pattern Recognition, Prague, Czech*, 1997.

[73] JOSEF KITTLER, MOHAMAD HATEF, ROBERT PW DUIN, AND JIRI MATAS. **On combining classifiers**. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **20**(3):226–239, 1998. 33

[74] HENRIK BOSTROM, RONNIE JOHANSSON, AND ALEXANDER KARLSSON. **On evidential combination rules for ensemble classifiers**. In *Information Fusion, 2008 11th International Conference on*, pages 1–8. IEEE, 2008. 33

[75] SHILIANG SUN AND CHANGSHUI ZHANG. **Subspace ensembles for classification**. *Physica A: Statistical Mechanics and its Applications*, **385**(1):199–207, 2007. 34

[76] NIDA MEDDOURI, HÉLA KHOUFI, AND MONDHER SADOK MADDOURI. **Diversity analysis on boosting nominal concepts**. In *Advances in Knowledge Discovery and Data Mining*, pages 306–317. Springer, 2012. 35

[77] CATHERINE A SHIPP AND LUDMILA I KUNCHEVA. **An investigation into how adaboost affects classifier diversity**. In *Proceedings of 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 203–208, 2002. 35

[78] LUDMILA I KUNCHEVA. **A bound on kappa-error diagrams for analysis of classifier ensembles**. *Knowledge and Data Engineering, IEEE Transactions on*, **25**(3):494–501, 2013. 35

[79] FABIO ROLI, GIORGIO GIACINTO, AND GIANNI VERNAZZA. **Methods for designing multiple classifier systems**. In *Multiple Classifier Systems*, pages 78–87. Springer, 2001. 36, 73

[80] ROBERT E BANFIELD, LAWRENCE O HALL, KEVIN W BOWYER, AND W PHILIP KEGELMEYER. **A new ensemble diversity measure applied to thinning ensembles**. In *Multiple Classifier Systems*, pages 306–316. Springer, 2003. 36

[81] ARJUN CHANDRA AND XIN YAO. **DIVACE: Diverse and accurate ensemble learning algorithm**. In *Intelligent Data Engineering and Automated Learning–IDEAL 2004*, pages 619–625. Springer, 2004. 36, 73

[82] JULIEN MEYNET AND JEAN-PHILIPPE THIRAN. **Information theoretic combination of pattern classifiers**. *Pattern Recognition*, **43**(10):3412–3421, 2010. 37

[83] ALEXANDER GAMMERMAN, VOLODYA VOVK, AND VLADIMIR VAPNIK. **Learning by transduction**. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 148–155. Morgan Kaufmann, 1998. 39

[84] CRAIG SAUNDERS, ALEXANDER GAMMERMAN, AND VOLODYA VOVK. **Transduction with confidence and credibility**. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI'99)*, **2**, pages 722–726, 1999. 39

[85] VLADIMIR VOVK, ALEX GAMMERMAN, AND GLENN SHAFER. *Algorithmic learning in a random world*. Springer, 2005. 39

[86] G. SHAFER AND V. VOVK. **A tutorial on conformal prediction**. *The Journal of Machine Learning Research*, **9**:371–421, 2008. 39

[87] TUVE LOFSTROM, ULF JOHANSSON, AND HENRIK BOSTROM. **Effective utilization of data in inductive conformal prediction using ensembles of neural networks**. In *Neural Networks (IJCNN), The 2013 International Joint Conference on*, pages 1–8. IEEE, 2013. 39

[88] TUVE LÖFSTRÖM, HENRIK BOSTRÖM, HENRIK LINUSSON, AND ULF JOHANSSON. **Bias Reduction through Conditional Conformal Prediction**. *Under Review in Intelligent Data Analysis*, 2015.

[89] ULF JOHANSSON, HENRIK BOSTRÖM, TUVE LÖFSTRÖM, AND HENRIK LINUSSON. **Regression conformal prediction with random forests**. *Machine Learning*, pages 1–22, 2014. 39, 48

[90] LESLIE G. VALIANT. **A theory of the learnable**. *Communications of the ACM*, **27**(11):1134–1142, 1984. 39

[91] HARRIS PAPADOPOULOS. **Inductive conformal prediction: Theory and application to neural networks**. *Tools in Artificial Intelligence*, **18**:315–330, 2008. 39, 40

[92] THOMAS MELLUISH, CRAIG SAUNDERS, ILIA NOURETDINOV, AND VOLODYA VOVK. **Comparing the Bayes and typicalness frameworks**. In *Machine Learning: ECML 2001*, pages 360–371. Springer, 2001. 39

[93] ILIA NOURETDINOV, VOLODYA VOVK, MICHAEL VYUGIN, AND ALEX GAMMERMAN. **Pattern recognition and density estimation under the general iid assumption**. In *Computational Learning Theory*, pages 337–353. Springer, 2001. 39

[94] AARON CLAUSET. **A brief primer on probability distributions**. *Santa Fe Institute. http://tuvalu. santafe. edu/˜ aaronc/courses/7000/csci7000-001_2011_L0. pdf*, 2011. 40

[95] VALENTINA FEDOROVA, ALEX GAMMERMAN, ILIA NOURETDINOV, AND VLADIMIR VOVK. **Plug-in martingales for testing exchangeability on-line**. *arXiv preprint arXiv:1204.3251*, 2012. 40

[96] VLADIMIR VOVK. **Conditional validity of inductive conformal predictors**. *Machine Learning*, pages 1–28, 2012. 41, 50

[97] HARRIS PAPADOPOULOS, VLADIMIR VOVK, AND ALEXANDER GAMMERMAN. **Regression conformal prediction with nearest neighbours**. *Journal of Artificial Intelligence Research*, **40**(1):815–840, 2011. 42

[98] VLADIMIR VOVK, VALENTINA FEDOROVA, ILIA NOURETDINOV, AND ALEX GAMMERMAN. **Criteria of efficiency for conformal prediction**. In *AIAI Conference*, 2014. 42, 43

[99] VALENTINA FEDOROVA, ALEX GAMMERMAN, ILIA NOURETDINOV, AND VLADIMIR VOVK. **Conformal Prediction under Hypographical Models**. 2013. 43

[100] ULF JOHANSSON, RIKARD KONIG, TUVE LOFSTROM, AND HENRIK BOSTROM. **Evolved decision trees as conformal predictors**. In *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pages 1794–1801. IEEE, 2013. 43

[101] HENRIK LINUSSON, ULF JOHANSSON, HENRIK BOSTRÖM, AND TUVE LÖFSTRÖM. **Efficiency Comparison of Unstable Transductive and Inductive Conformal Classifiers**. In *Artificial Intelligence Applications and Innovations*, pages 261–270. Springer, 2014. 45

[102] B EFRON. **Bootstrap methods: another look at the jackknife**. *The Annals of Statistics*, **7**(1):1–26, 1979. 47

[103] HENRIK BOSTRÖM. **Concurrent Learning of Large-Scale Random Forests**. In *SCAI*, pages 20–29, 2011. 49

[104] NITESH V CHAWLA, NATHALIE JAPKOWICZ, AND ALEKSANDER KOTCZ. **Editorial: special issue on learning from imbalanced data sets**. *ACM SIGKDD Explorations Newsletter*, **6**(1):1–6, 2004. 49

[105] XU-YING LIU, JIANXIN WU, AND ZHI-HUA ZHOU. **Exploratory undersampling for class-imbalance learning**. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, **39**(2):539–550, 2009. 49

[106] Jie Gu, Yuanbing Zhou, and Xianqiang Zuo. **Making class bias useful: A strategy of learning from imbalanced data**. In *Intelligent Data Engineering and Automated Learning-IDEAL 2007*, pages 287–295. Springer, 2007. 49

[107] Gustavo E. A. P. A. Batista, Ronaldo C. Prati, and Maria Carolina Monard. **A study of the behavior of several methods for balancing machine learning training data**. *SIGKDD Explor. Newsl.*, **6**(1):20–29, June 2004. 49

[108] *Oxford Dictionaries*. Oxford University Press, 2014. "research". Accessed 24th November 2014. 51

[109] R Hevner von Alan, Salvatore T March, Jinsoo Park, and Sudha Ram. **Design science in information systems research**. *MIS quarterly*, **28**(1):75–105, 2004. 51

[110] Ron Kohavi and Foster Provost. **Glossary of terms**. *Machine Learning*, **30**(2-3):271–274, 1998. 51

[111] Egon G Guba. *The paradigm dialog*. Sage Publications, 1990. 51

[112] Thashmee Karunaratne. *Learning predictive models from graph data using pattern mining*. PhD thesis, Stockholm University, 2014. 52

[113] Alan Hevner and Samir Chatterjee. *Design science research in information systems*. Springer, 2010. 52

[114] Robert PW Duin. **A note on comparing classifiers**. *Pattern Recognition Letters*, **17**(5):529–536, 1996. 54

[115] Steven L Salzberg. **On comparing classifiers: Pitfalls to avoid and a recommended approach**. *Data mining and knowledge discovery*, **1**(3):317–328, 1997. 54

[116] A. Asuncion and D. J. Newman. **UCI machine learning repository**, 2007. 54

[117] Tim Menzies, Bora Caglayan, Zhimin He, Ekrem Kocaguneli, Joe Krall, Fayola Peters, and Burak Turhan. **The PROMISE Repository of empirical software engineering data**, June 2012. 54

[118] Craig L Bruce, James L Melville, Stephen D Pickett, and Jonathan D Hirst. **Contemporary QSAR classifiers compared**. *Journal of chemical information and modeling*, **47**(1):219–227, 2007. 55

[119] Jeffrey J Sutherland, Lee A O'Brien, and Donald F Weaver. **A comparison of methods for modeling quantitative structure-activity relationships**. *Journal of Medicinal Chemistry*, **47**(22):5541–5554, 2004. 55

[120] Tom Fawcett. **An introduction to ROC analysis**. *Pattern recognition letters*, **27**(8):861–874, 2006. 56

[121] Thomas G Dietterich. **Approximate statistical tests for comparing supervised classification learning algorithms**. *Neural computation*, **10**(7):1895–1923, 1998. 57

[122] Janez Demšar. **Statistical comparisons of classifiers over multiple data sets**. *The Journal of Machine Learning Research*, **7**:1–30, 2006. 58, 59

[123] R.L. Iman and J.M. Davenport. **Approximations of the critical region of the fbietkan statistics**. *Journal of Communications in Statistics - Theory and Methods*, **9**(6):571–595, 1980. 60

[124] Janez Demšar. **Statistical Comparisons of Classifiers over Multiple Data Sets**. *J. Mach. Learn. Res.*, **7**:1–30, 2006. 61