Pós-Graduação em Ciência da Computação

Thiago José Marques Moura

MINE - A framework for dynamic regressor selection.

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
http://cin.ufpe.br/~posgraduacao

Recife
2019

Thiago José Marques Moura

MINE - A framework for dynamic regressor selection.

Tese de Doutorado apresentada ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de Doutor em Ciência da Computação.

Área de Concentração: Inteligência Computacional
Orientador: Dr. George Darmiton da Cunha Cavalcanti
Coorientador: Dr. Luiz Eduardo Soares de Oliveira

Recife
2019

Catalogação na fonte
Bibliotecário Vimário Carvalho CRB4-1204

M929m Moura, Thiago José Marques.
Mine - a framework for dynamic regressor selection / Thiago José Marques Moura. – 2019.
87 f.: il., fig., tab.

Orientador: Dr. George Darmiton da Cunha Cavalcanti.
Tese (Doutorado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2019.
Inclui referências e apêndice.

1. Inteligência computacional. 2. Regressores. 3. Medidas. 4. Dinâmica de Regressores. I. Cavalcanti, George Darmiton da Cunha (orientador). II. Oliveira, Luiz Eduardo Soares de (coorientador). III. Título.

006.3 CDD (23. ed.) UFPE-MEI 2019-147

Thiago José Marques Moura

“MINE - A framework for dynamic regressor selection”

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Doutor em Ciência da Computação.

Aprovado em: 12/08/2019.

_____

Orientador: Prof. Dr. George Darmiton da Cunha Cavalcanti

BANCA EXAMINADORA

_____

Prof. Dr. Tsang Ing Ren
Centro de Informática/UFPE

_____

Prof. Dr. Luciano de Andrade Barbosa
Centro de Informática/UFPE

_____

Profa. Dra. Renata Maria Cardoso Rodrigues de Souza
Centro de Informática/UFPE

_____

Profa. Dra. Isis Didier Lins
Departamento de Engenharia de Produção/UFPE

_____

Prof. Dr. Alceu de Souza Britto Junior
Programa de Pós-Graduação em Informática Aplicada/ PUC-PR

I dedicate this work to my wife Alana and my children Júlia and Matheus.

ACKNOWLEDGEMENTS

First of all, I would like to thank my family: my wife Alana, my children Júlia and Matheus. They were always with me in the most difficult and painful hours for the conclusion of this work. Many moments I could not be with them to dedicate myself to the research.

I also express my gratitude to my supervisors, Prof. George Cavalcanti, and Prof. Luiz Eduardo Oliveira, to go along with me in this journey. Special thanks to Prof. George for always being ready, having the patience to read and reread what I wrote and get my questions in the most complicated moments.

I thank the members of my thesis committee: Prof. Tsang, Prof. Luciano Barbosa, Profa. Renata Souza, Profa. Isis Lins, and Prof. Alceu Junior for evaluating this thesis and providing constructive comments.

I would like to thank cousins Bruno, Juliana, and César Augusto for supporting during the days that I had to sleep in Recife to attend classes of the subjects.

Thank all the IFPB colleagues for granting me the qualification license. In particular, I would like to thank Profa. Damires Yluska for all the support I needed to start my PhD. I would also like to thank Prof. Leandro Almeida for helping me in the execution of experiments on IFPB servers.

Finally, I want to thank some people that encouraged or inspired me in different moments over the last 5 years: Wagner Jorge, Luiz Felipe, Edileuza Leão, Alysson Bispo, Roberto Hugo Pinheiro, and Dayvid Victor. A special thanks to Wagner Jorge Firmino da Silva which helped me through the most difficult moments in the research. It was a pleasure meeting you all.

ABSTRACT

Dynamic Regressor Selection (DRS) systems work by selecting the most competent regressors from an ensemble to estimate the target value of a given test pattern. Hence, the central issue in dynamic selection techniques is how to define the competence of the regressors to select the most competent ones. This competence is usually quantified using a single measure, such as the performance of the regressors in local regions of the feature space around the test pattern, called the region of competence. However, to decide what is the best measure to correctly calculate the level of competence is a hard task, because no one is the best for any task. Works using ensemble of classifiers present a wide variety of measures that are used to calculate the competence. Using ensemble of regressors, many of these measures can not be used or adapted. Thus, in this work, we present a framework for DRS, called Meta INtEgration (MINE), that aims at selecting and combining the most competent regressors from a homogeneous ensemble during the evaluation of a given test pattern. The proposed framework uses the combination of different measures extracted from the region of competence, as a criterion for the selection and combination of the regressors. Also, we have done a survey in the literature on some measures used with regression problems to test the performance of the dynamic regression selection algorithms found in the literature. The measures are extracted from region of competence and they are aimed at capturing different behaviors of the regressors. Thus, for each test pattern, only the most competent regressors are selected and combined. Using the MINE framework, comprehensive experiments on 20 regression datasets show that MINE improves the final estimate performance when compared to state-of-the-art techniques. Also, experiments are performed on 15 real regression problems datasets using the state-of-the-art dynamic regressor selection techniques by changing only the measure that computes the competence. The results show that the measures have different performance throughout the datasets and none of them are better in all situations.

Keywords: Regressors, Ensemble of Regressors, Measures, Combination, Dynamic Regressor Selection.

RESUMO

Sistemas de seleção dinâmica de regressores (Dynamic Regressor Selection - DRS) funcionam selecionando os regressores mais competentes de um *ensemble* com o objetivo de estimar o valor de um dado padrão de teste. Assim, a questão central nas técnicas de seleção dinâmica é como definir a competência dos regressores para selecionar os mais competentes. Essa competência é geralmente quantificada usando uma única medida, como o desempenho dos regressores em regiões locais do espaço de características em torno do padrão de teste, chamado de região de competência. No entanto, decidir qual é a melhor medida para calcular corretamente o nível de competência é uma tarefa difícil, porque nenhuma delas é a melhor para qualquer tarefa. Trabalhos usando *ensemble* de classificadores apresentam uma grande variedade de medidas que são usadas para calcular a competência. Usando *ensemble* de regressores, muitas dessas medidas não podem ser usadas ou adaptadas. Assim, neste trabalho, apresentamos um *framework* para DRS, chamado *Meta INtEgration* (MINE), que visa selecionar e combinar os regressores mais competentes de um *ensemble* homogêneo durante a avaliação de um dado padrão de teste. O *framework* proposto utiliza a combinação de diferentes medidas extraídas da região de competência como critério para a seleção e combinação dos regressores. Além disso, fizemos um levantamento na literatura sobre algumas medidas utilizadas com problemas de regressão para testar o desempenho dos algoritmos de seleção dinâmica de regressores encontrados na literatura. As medidas são extraídas da região de competência e visam capturar diferentes comportamentos dos regressores. Assim, para cada padrão de teste, apenas os regressores mais competentes são selecionados e combinados. Usando o *framework* MINE, experimentos foram realizados em 20 bases de dados de regressão mostrando que o MINE melhora o desempenho da estimativa final quando comparado com as técnicas da literatura. Também, experimentos foram realizados com 15 bases de dados de problemas reais de regressão, usando técnicas de seleção dinâmica da literatura, alterando apenas a medida que calcula a competência. Os resultados mostram que as medidas têm desempenho diferente ao longo das bases de dados e nenhuma delas é melhor em todas as situações.

Palavras-chaves: Regressores, Ensemble de Regresssores, Medidas, Combinação, Seleção Dinâmica de Regressores.

LIST OF FIGURES

LIST OF TABLES

CONTENTS

1 INTRODUCTION

In machine learning, techniques that use ensembles are those that generate different models with some degree of diversity and combine the models to make a prediction. Ensembles are used either in classification or regression problems. The advantage of ensembles concerning single models has been reported in terms of increased robustness and accuracy for both classification (HO, 1998; DOMENICONI; YAN, 2004; SINGH; SINGH, 2005), and regression problems (DRUCKER, 1997; SHRESTHA; SOLOMATINE, 2006; ZHANG; ZHANG; WANG, 2008).

Ensemble systems have three general phases (CRUZ; SABOURIN; CAVALCANTI, 2018): (1) Generation, when a training set is used to generate an ensemble; (2) Selection, when a subset from the ensemble is selected to perform the prediction; and (3) Combination (Fusion), when the final prediction is the result of the combination of the models previous selected. Figure 1 shows the general phases of ensemble systems: generation, selection and combination.

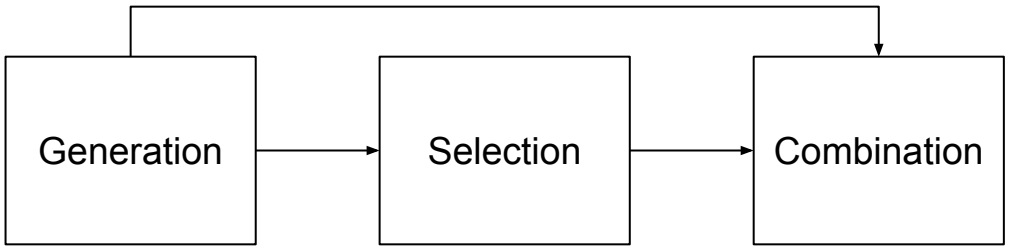


Figure 1 – Ensemble systems general phases: generation, selection, and combination.

In the generation phase, a training set is used to create the ensemble. Ensemble is said to be homogeneous when a single learning algorithm is used to train all models and when more than one learning algorithm is used, the ensemble is said heterogeneous.

In the second phase, only a single model or a subset of the ensemble is selected. Selection is optional and can be both static and dynamic. In the static approach, the selection is performed before the evaluation of the test pattern, using information from the training set (ORTIZ-BOYER; HERVáS-MARTíNEZ; GARCíA-PEDRAJAS, 2005) or the validation set (PARTALAS et al., 2008). Then, the selected models are used to estimate the target value of all test patterns. In the dynamic approach, a different subset is selected for each new test pattern. In the dynamic selection techniques, each model is expected to be specialized in a specific region of the feature space, which is known as region of competence. Thus, for each test pattern, the most competent models are selected in the region of competence where the test pattern is located. Recent works show that dynamic selection techniques perform better than static selection (KO; SABOURIN; BRITTO, 2008; BRITTO; SABOURIN; OLIVEIRA, 2014; CRUZ; SABOURIN; CAVALCANTI, 2018; MENDES-MOREIRA et

al., 2009). In this work, dynamic selection systems that generate models for regression problems is called Dynamic Regression Selection (DRS) systems.

When the subset of the selected models from the ensemble contains more than one model, they must be combined (fused). The combination can be done using a simple technique such as mean or weighted mean. The weighted mean has better accuracy than the mean (PERRONE; COOPER, 1993), and the weights can be defined statically or dynamically. Statically (constant weights), the weighted mean of the models uses the same vector of weights for any test pattern, while in the dynamic form (nonconstant weights) the weights are defined according to the performance of the models in the region of competence.

1.1 STATEMENT OF THE PROBLEM

The rationale behind dynamic selection systems is that different models are competent (or experts) in different local regions of the feature space, which means that no model is competent to estimate all the test patterns, so, the idea is to select the most competent models for each new test pattern.

The crucial issue in dynamic selection systems is to define which criterion is used to measure the competence of the models. It is expected that the better the competence of the selected models, the higher the accuracy of the whole system. A common alternative used to measure the competence is to calculate the cumulative error of the model in the neighborhood of the test pattern (ROONEY et al., 2004; MENDES-MOREIRA et al., 2009; WOODS; KEGELMEYER; BOWYER, 1997; GIACINTO; ROLI, 1999). However, the Dynamic Classifier Selection (DCS) (SANTANA et al., 2006; SANTOS; SABOURIN; MAUPIN, 2008) literature shows that using only the cumulative error in the region of competence is not sufficient to correctly measure the competence of the classifiers. Recent classification works (CRUZ et al., 2015; CRUZ; SABOURIN; CAVALCANTI, 2016; CRUZ; SABOURIN; CAVALCANTI, 2017) use the composition of many measures to determine the competence of the classifiers, selecting and combining them to predict a given test pattern. The literature for classification problems, which discuss measures to evaluate the competence of classifiers, is richer than regression problems. The problem is that the measures used for classification are not directly transferable to regression problems and usually, only the error measure is used in dynamic regressor selection.

In both Dynamic Classifier Selection (DCS) as in Dynamic Regressor Selection (DRS) systems, the region of competence is used to find the most similar patterns according to the test pattern x_{query}. Figure 2 shows an example of the region of competence calculated using regression data. Some similarity measure, as Euclidian Distance, can be used to find the most similar patterns from the training set or the validation set. In that figure, k is the size of the region of competence, and x_{query} is the test pattern.

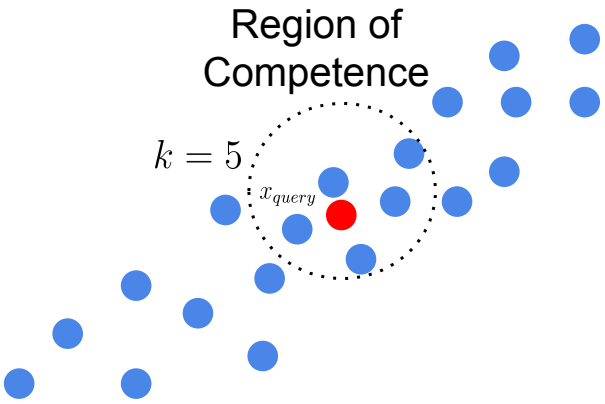


Figure 2 – Definition of the Region of Competence. Blue dots (●) represent patterns from the training set or validation set. The red dot (●) represents the test pattern x_{query}. In this example, K is the size of the region of competence.

In works with DRS (ROONEY et al., 2004; MENDES-MOREIRA et al., 2009), only one error measure is used to estimate the competence of the regressors in the region of competence. For example, both Root Sum Squared Error (RSSE) and Sum Squared Error (SSE) can be used as measures of competence. For the selection of a single model, RSSE and SSE have the same result, selecting the same model to predict the x_{query} pattern. When more than one model is selected, these measures have different performance in the combination, because they are measures with different magnitudes.

Table 1 presents the results using RSSE and SSE as measures of competence. The final result of the combination of the regressors $\hat{f}_1$, $\hat{f}_2$, and $\hat{f}_3$ was calculated through the weighted mean using the RSSE and SSE as weights for the regressors ($\hat{f}_n$) prediction. It is easy to see that there is a difference (0.064 vs 0.0103) between the weighted means.

Thus, some questions arise:

- Are there more measures of competence that can be used with regression problems? Which ones?
- Is there any measure that is better than the others in all situations?
- Are the measures problem-dependent?
- If there are more measures, how to combine them to achieve better performance in a dynamic regressor selection system?

Finally, it is necessary to carry out in the literature of regression problems a survey of probable measures of competence that can be used in DRS. Also, evaluate the measures individually, checking if any of them performs better than others in all situations and, otherwise, find a way to combine them by increasing the whole performance of DRS techniques.

Table 1 – Weighted Mean of the regressors using Root Sum Squared Error and Sum Squared Error. For each regressor $\hat{f}_n$, the RSSE and SSE are calculated in the region of competence around the test pattern x_{query}. $\hat{f}_n(x_{query})$ is the estimated value for the test pattern.

Regressor	$\hat{f}_n(x_{query})$	**RSSE**	**SSE**	$\hat{f}_n(x_{query})\times$ RSSE	$\hat{f}_n(x_{query})\times$ SSE
$\hat{f}_1$	0.1	0.1	0.01	0.01	0.001
$\hat{f}_2$	0.12	0.2	0.04	0.024	0.0048
$\hat{f}_3$	0.2	0.15	0.0225	0.03	0.0045
			Weighted Mean	**0.064**	**0.0103**

1.2 OBJECTIVES

The objectives of this thesis are: (i) to develop a new technique that combines the measures of competence enhancing dynamic selection and thereby increasing the whole performance of the system; (ii) to survey in the literature some measures used with regression problems that can be used to measure the competence of the regressors in DRS algorithms.

To acomplish those objectives, in this thesis we propose the Meta INtEgration (MINE) framework for DRS. It uses a combination of measures extracted from the region of competence as a criterion to select and combine the regressors. MINE was designed to work in the following scenarios: (i) select a single regressor, given the test pattern (MINE-Selection (MINE-S)); (ii) all the ensemble regressors are combined through the weighted mean (MINE-Weighting (MINE-W)); and (iii) a subset of the ensemble is dynamically selected for each test pattern (MINE-Weighting with Selection (MINE-WS)). Our hypothesis is that the DRS can benefit from the combination of several measures instead of relying on a single one. Also, in order to expand the research with the use of homogeneous ensembles, this thesis brings a robust study using homogeneous ensembles for DRS.

Also, we define eight measures to be extracted from the region of competence for each test pattern and compare them individually with DRS algorithms find in the literature (ROONEY et al., 2004; MENDES-MOREIRA et al., 2009). The hypothesis raised is whether the DRS algorithms have different performance when the measure used to calculate the competence of the regressors is modified. Another point is to verify if some measure performs better than others in all situations.

1.3 CONTRIBUTIONS

This thesis has two contributions for dynamic regressor selection systems, they are:

- a framework for dynamic regressor selection, called MINE, that uses the combination of the measures of competence as a criterion to select and combine the regressors from an ensemble;

- a study of a survey of measures of competence evaluated with state-of-the-art dynamic regression selection techniques;

These contributions generated the following submitted articles (Chapter 4 and 3):

- MOURA, T. J. M.; CAVALCANTI, G. D.; OLIVEIRA, L. S. MINE: A Framework for Dynamic Regressor Selection. **Submitted to Pattern Recognition**, 2019.
- MOURA, T. J. M.; CAVALCANTI, G. D.; OLIVEIRA, L. S. Evaluating Competence Measures for Dynamic Regressor Selection, **In proceedings of International Joint Conference on Neural Networks (IJCNN)**, 2019.

1.4 ORGANIZATION

This thesis is organized into five chapters. Figure 3 presents this thesis overview. Boxes are chapters and arrows indicate the flow of the thesis. The thesis starts with Chapter 1 (current chapter) presenting the introduction of the thesis. Chapter 2 presents a background of Dynamic Regressor Selection (DRS) pointing the main works in the three steps of the construction of DRS systems: (i) ensemble generation; (ii) selection; and (iii) combination, for a better understanding of the following chapters.

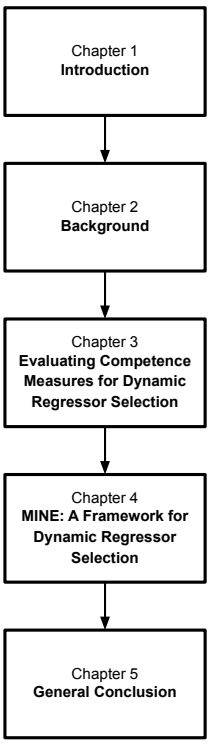


Figure 3 – Thesis overview. The boxes are chapters and the arrows are the flow of the thesis.

Chapter 3 shows a study with the evaluation of eight measures used to calculate the competence of the regressors from an ensemble. This chapter presents the definition of the

eight measures to be extracted from the region of competence for each test pattern and a comparative tests of DRS algorithms. The algorithms are evaluated using 15 regression problems from different data repositories and they are compared against an individual regressor and against the Mean and Median. Experiments were performed to validate the hypothesis that DRS algorithms have a different performance when the measure used to calculate the competence of the regressors is modified. The content of this chapter is going to published in the proceedings of the International Joint Conference on Neural Networks (IJCNN), this year.

In Chapter 4, MINE framework is introduced. The chapter begins with a brief introduction, pointing to the central issue of the dynamic regressors selection that is the choice of the measure of competence, followed by a detailed description of the framework, with its steps and modules. MINE framework can operate in three different scenarios: (i) the selection of a single regressor given a test pattern (MINE-Selection (MINE-S)); (ii) all the regressors in the ensemble are weighted and combined (MINE-Weighting (MINE-W)); and, (iii) a subset the ensemble is dynamically selected per test pattern (MINE-Weighting with Selection (MINE-WS)). Finally, a series of experiments that demonstrate that MINE outperforms state-of-the-art DRS techniques and static techniques as Mean and Median. The content of this chapter has been submitted to the Pattern Recognition journal.

In Chapter 5, the conclusion and future works are presented.

2 BACKGROUND

Dynamic Regressor Selection (DRS) consists in selecting regressors from an ensemble for each new test pattern. If only a single regressor is selected, its output is the prediction of the test pattern, otherwise, the selected regressors are combined in order to predict the test pattern. Thus, for each test pattern x_{query} and an ensemble of regressors $\mathcal{F}$ of size N, a subset $\mathcal{F}'$ of size $M \leq N$ is selected containing the most competent regressors.

The idea involved in the DRS techniques is that each regressor from the ensemble has different performance in distinct regions of the feature space around the test pattern, called the region of competence (MERZ, 1996; KUNCHEVA; RODRíGUEZ, 2007). No regressor is competent to correctly predict all test patterns, so it is interesting to select the most competent one or a subset with the most competent to predict each test pattern.

The crucial issue in DRS is how to measure the competence of the regressors from the ensemble for each test pattern. In general, DRS systems use the error generated in the region of competence, selecting the regressor(s) with the lowest error rate(s). These errors generated in the region of competence also can be used in the weighted combination of the regressors. The weights are inversely proportional to the error rates, that is, the smaller the error of the regressor, the greater is its weight.

In general, DRS systems have three main steps (Figure 4):

1. Ensemble Generation, when the ensemble $\mathcal{F}$ is generated.

2. Selection, when the most competent regressors from the ensemble are selected.

3. Combination (Fusion), when the selected regressors are combined using some criterion.

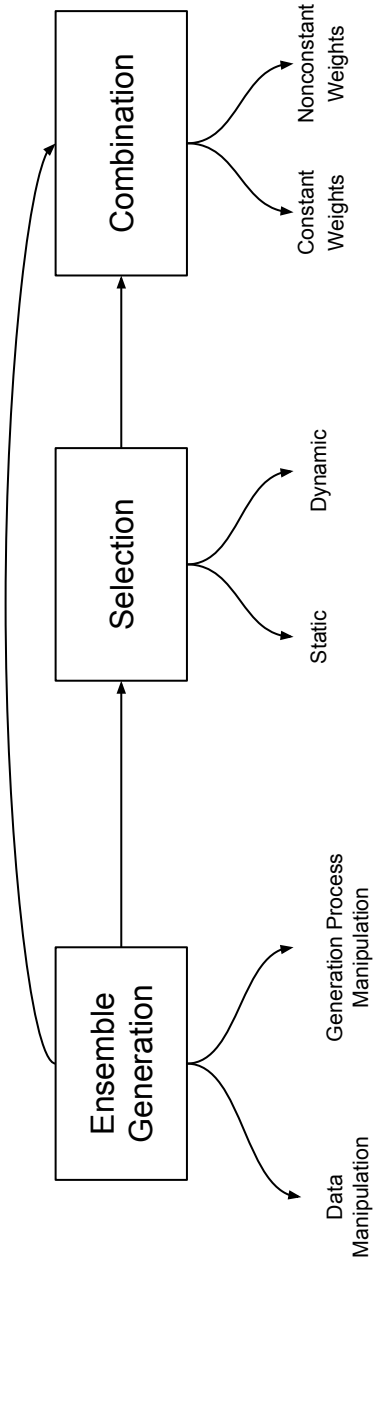


Figure 4 – Steps of DRS systems.

The next sections describe the steps shown in Figure 4, as well as some works from the literature.

2.1 ENSEMBLE GENERATION

The first step of DRS systems is the generation of the ensemble. The goal of this step is to generate the models $\hat{f}_n, \forall n \in \{1, 2, ..., N\}$ to compose the ensemble $\mathcal{F} = \{\hat{f}_1, \hat{f}_2, ..., \hat{f}_N\}$.

If the generation of the regressors is performed using the same learning algorithm, the ensemble is said homogeneous, otherwise, is heterogeneous. Homogeneous ensembles are more discussed in the literature (DIETTERICH, 1997; BROWN et al., 2005; ROONEY et al., 2004), because it is more difficult to control the interaction between the different learning processes. Figure 5 shows the process to generate a single regressor $\hat{f}_n$.

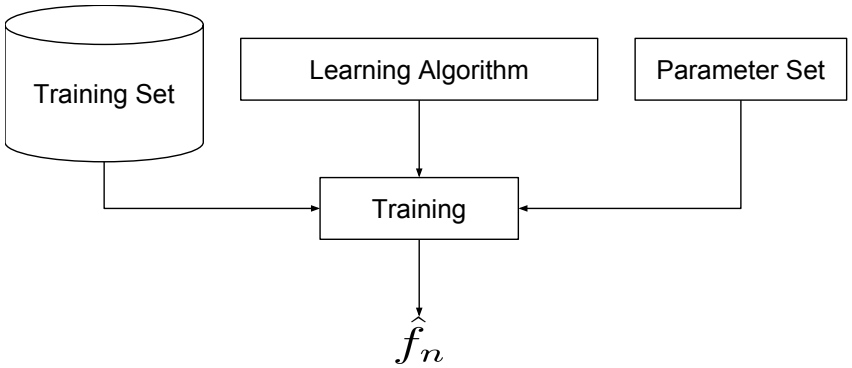


Figure 5 – Generation process of a regressor $\hat{f}_n$ (figure inspired by (MENDES-MOREIRA et al., 2012)).

The training set is used to train the regressor $\hat{f}_n$ and the parameters of the learning algorithm can be manipulated to generate more diverse and accurate regressors.

This section presents the main methods for the ensemble generation step. With the goal to generate diverse and accurate models, ensemble generation methods are classified into two groups: (i) Data Manipulation; and (ii) Generation Process Manipulation. These two groups are detailed in the next sections.

2.1.1 Data Manipulation

In this section, it is discussed methods of data manipulation in two different ways: subsampling from the training set and manipulating the features.

2.1.1.1 Subsampling from the Training Set

This approach generates different subsamples of the training set and each subsample is used to train a model. This approach assumes that learning algorithms are unstable, that is, small changes in training data result in large changes in the results. Decision trees and artificial neural networks are examples of unstable learning algorithms (ZHOU,

2012), (BREIMAN, 1996b), (DIETTERICH, 1997). However, some sampling methods such as Bagging and Boosting have been used successfully in algorithms considered stable, such as Support Vector Machines (SVM) (KIM et al., 2002).

Bagging

The first method to be pointed out is Bagging (Bootstrap AGGegatING) (BREIMAN, 1996a). Bagging generates distinct datasets, using sampling with replacement, which means that some instances are repeated in each dataset, and on average only 63% of instances are unique. The outputs of the Bagging are N training sets $\{\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_N\}$, responsible for training each model $\hat{f}_n \in \mathcal{F}$. All sets $\mathcal{T}_n$ have the same size as the original training set $\mathcal{T}$. For more details about Bagging, see (BREIMAN, 1996a) and (DOMINGOS, 1997).

Boosting

Other important method using subsampling is Boosting. Freund and Schapire present the algorithm AdaBoost (FREUND; SCHAPIRE, 1996), which is the most popular Boosting algorithm. The main idea is the possibility of converting a weak model into a strong model, that is, a model that can reach high accuracy. A weak model is one that achieves performs slightly better than random prediction. In the Bagging, the training patterns are selected randomly with replacement, but in AdaBoost the patterns have different probabilities to be selected. Initially, the probabilities are the same, but during the iterations, the patterns that are more difficult to classify are more likely to be selected.

AdaBoost was designed for classification problems and its first adaptation for regression problems was AdaBoost.R (FREUND; SCHAPIRE, 1997). This method assumes the target values y are just two possible labels $y = [0, 1]$, that is, it transforms the regression dataset into a classification problem (binary classification) as follows: (i) the range of the target values y is split into S sub-ranges having the same size and lower limits $l_1 = 1/(S+1), l_2 = 2/(S+1), ..., l_s = S/(S+1)$; (ii) each pattern i from training set is replaced by S of its copies, where the target value of each copy is replaced with two new variables. The first variable has the value l_j and the second is a new target variable of the binary classification problem, defined as 0 if $y_i < l_j$ or 1, otherwise. Therefore, each new instance j associated with the original pattern i represents the question "is $y_i < l_j$?". The new dataset has $N \times S$ patterns (MENDES-MOREIRA et al., 2012).

In (AVNIMELECH; INTRATOR, 1999b) is proposed an adaptation of AdaBoost for regression problems. In each iteration, the regression errors calculated by the models in the previous iteration are considered Correct or Reject. Correct if the error is less than 0.5 and Reject, otherwise. So, the weights of the patterns are updated using the regression error of the model. An improvement to this work, called AdaBoost.RT, is presented in (SHRESTHA; SOLOMATINE, 2006). At each iteration, the error is calculated using a sub-

set of the patterns. This subset contains the patterns with an error higher than a given threshold.

Another variation of the AdaBoost algorithm is the AdaBoost.R2 (DRUCKER, 1997). AdaBoost.R2 normalizes the error values, guaranteeing that the average error for all of the patterns in the training set is in the interval $[0, 1]$. The interactions finish when the average error is lower than 0.5. This method was tested by (BORRA; CIACCIO, 2002) with different learning algorithms (DART (FRIEDMAN, 1996), PPR (FRIEDMAN; STUETZLE, 1981), and MARS (FRIEDMAN, 1991)).

Other works (ZEMEL; PITASSI, 2001; RÄTSCH; DEMIRIZ; BENNETT, 2002) based on AdaBoost for regression problems present variations in the function that calculates the weight of the patterns in each iteration.

In (CHANDRAHASAN et al., 2011) is presented a comparative study between Bagging and Boosting for classification problems. The conclusions are: (i) no single algorithm performed well for all problems used in the experiments. The algorithms depends more on dataset than any other factors; and (ii) to be competitive and feasible, it is important to consider the processing time. In their experiments, AdaBoost runs in a reasonable time in all the three medical datasets used.

2.1.1.2 Manipulating the Features

In this approach, different training sets $\{\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_N\}$ are generated by changing the representation of the patterns from the original training set $\mathcal{T}$. Each training set $\mathcal{T}_n, \forall n \in \{1, 2, ..., N\}$ is generated by replacing the original representation $\{(\mathbf{x}_i, y)\}$, with a new one $\{(\mathbf{x}'_i, y)\}$, where $\mathbf{x}_i$ and $\mathbf{x}'_i$ are vectors of features, and y is the target value. In this approach, the new training sets are generated in two ways: (i) feature selection, that is, $\mathbf{x}'_i \subset \mathbf{x}_i$; (ii) the new features are obtained by applying some transformation into the original ones.

One of the most popular methods of feature selection is the Random Subspace (HO, 1998). In this method, the training sets are generated selecting randomly the features. Originally, decision trees were used as learning algorithms and the ensemble was called decision forest. For each test pattern, the final prediction is the combination of all the trees from the ensemble.

Also, iterative search methods can be used to select the features. In (OPITZ, 1999) is used a genetic algorithm to generate new subsets of features, starting with a randomly selected subset. The proposal was evaluated on classification problems using neural networks. The criterion used to select the features is the minimization of the individual model error and the maximization of ambiguity (KROGH; VEDELSBY, 1994). The ambiguity is defined as: $\sum_{i=1}^{N}[\alpha_i \times (\hat{f}_i(x_{query}) - \hat{f}_{ens}(x_{query}))^2]$, where N is the size of the ensemble, $\hat{f}_i$ is some regressor from the ensemble, $\hat{f}_{ens}$ is the final estimation of the ensemble, and α_i is a weight.

Feature selection can be used to generate ensembles using k-nearest-neighbors (WOODS; KEGELMEYER; BOWYER, 1997) as learning algorithm. kNN is stable regarding the training set, but unstable regarding the set of features. Therefore, small changes in the training set of the kNN, cause irrelevant changes in the final result, but to use different subsets of the features to find the similar data, cause bigger changes in the final result. In (DOMENICONI; YAN, 2004), feature selection is combined with adaptive sampling (THOMPSON; SEBER, 1996) to reduce the risk of discarding some useful information. When the work of Domeniconi et al. is compared to random feature selection, it reduces diversity among the models and increases the accuracy of the ensemble.

In (RODRÍGUEZ; KUNCHEVA; ALONSO, 2006) is presented a method that combines selection and transformation, called rotation forests. The original set of features is divided into d disjoint subsets and Principal Component Analysis (PCA) (JOLLIFFE, 2002) is applied to each subset. All principal components are kept in order to preserve the variability information in the data. Thus, d axis rotations occur to form the new features for training a decision tree. The idea of the rotation approach is to establish simultaneously individual accuracy and diversity among the trees from the ensemble. In (ZHANG; ZHANG; WANG, 2008) is applied rotation forest to regression problems.

2.1.2 Generation Process Manipulation

This section presents methods that manipulate the generation process of a model. This can be performed in two ways: (i) manipulating the parameter sets; and (ii) manipulating the learning algorithm.

2.1.2.1 Manipulating the Parameter Sets

There are many learning algorithms and each one is sensitive to changes in the input parameters. These input parameters can be changed before training a new model and thus generate more diverse and accurate ensembles. For example, neural networks can use different initial weights to obtain different models. The models trained with different sets of initial weights, generate different predictions for the same test pattern (KOLEN; POLLACK, 1990).

In (ROSEN, 1996) is generated randomly the initial weights to train different models, but the architecture of the neural network remains the same. In (PERRONE; COOPER, 1993) the initial weights are generated randomly but also the architecture of the neural network is manipulated changing the number of hidden layers and neurons . In (YANKOV; DECOSTE; KEOGH, 2006) is proposed a method which the ensemble is generated using k-nearest-neighbors. The ensemble proposed by Yankov et al. have only two models: one of them has a small number of nearest neighbors and the other has large nearest neighbors. Using a small value for k, the model becomes unstable. With a bigger k the estimation is much smoother.

The main goal of these techniques is to generate more diverse and accurate models, but there is no guarantee that this will happen. Varying the parameters of a learning algorithm can be useful when the training dataset is small.

2.1.2.2 Manipulating the Learning Algorithm

Another technique to generate the models is to change the internal characteristics of the learning algorithms. The same dataset is used for training, but the trained models have different results. The ensemble can be generated in two ways: (i) sequential; and (ii) parallel. In the first way, the training of a model is influenced only by the previously trained model. In the second way, the overall quality of the ensemble is taken into account and information about the models is exchanged between the generation processes.

Sequential

The most common sequential approach is to check the ensemble error (e.g., (GRANITTO; VERDES; CECCATTO, 2005) and (ISLAM; MURASE, 2003)). Granitto et al. presented SECA (Stepwise Ensemble Construction Algorithm), an algorithm that trains the models via Bagging. Each bootstrap sample is used to train a new neural network. SECA controls the training time of the models by checking the overall ensemble error, thus the training can stop early or later. Another method, called Cooperative Neural Network Ensembles (CNNE)(ISLAM; MURASE, 2003), starts the ensemble with two neural networks and new models are added trying to reduce the ensemble error. In addition, before adding new neural networks, the technique manipulates the number of neurons, adding new neurons to the network.

Parallel

In parallel techniques, the models are trained simultaneously, but the learning processes are not independent. They interact to guarantee that the training of each model is trying to accomplish objectives of the overall ensemble. The main difference between the parallel approaches and the sequential ones is that the ensemble generation is performed simultaneously taking into account (in each model of the ensemble) the behavior of the other models in previous iterations.

Some parallel methods use evolutionary algorithms. ADDEMUP (Accurate anD Diverse Ensemble-Maker giving United Predictions) (OPITZ; SHAVLIK, 1996). ADDEMUP generates new models from previous ones, using a fitness function that tries to weights the accuracy of the model, and the diversity of it within the other ones in the ensemble, according to the bias/variance decomposition (KROGH; VEDELSBY, 1994). As in AdaBoost (Section 2.1.1.1), the training of new models is focused on misclassified examples. The

process generates models in parallel and selects the best ones in each iteration of the genetic algorithm, stopping until a criterion is reached.

The method Ensemble Learning via Negative Correlation (ELNC) (LIU; YAO, 1999) also trains neural networks in parallel. However, it used the negative correlation term (UEDA; NAKANO, 1996) as error function, instead of the bias/variance decomposition (KROGH; VEDELSBY, 1994) used in ADDEMUP.

Random Forest (BREIMAN, 2001) is a method where trees are generated using Bagging (Section 2.1.1.1), but during the training of each model, the nodes are created taking into account a randomly selected feature subset. The subset used in one node is independent of the subset used in the previous one. This method, based on the manipulation of the learning algorithm, is a combination of bootstrap sampling and random feature selection.

2.2 SELECTION

Many methods presented in the ensemble generation step try to generate diverse and accurate ensembles, however, this result is not guaranteed. Some of those methods use random process (e.g. Bagging and AdaBoost) and this action cannot guarantee the diversity among the generated models.

Ensemble Selection consists of selecting a subset of models $\mathcal{F}'$ from an ensemble $\mathcal{F}$ previously generated, where $\mathcal{F}' \subseteq \mathcal{F}$. There are two selection approaches: (i) static; and (ii) dynamic. In (KUNCHEVA; RODRíGUEZ, 2007), the definition of static selection for classification problems is described as follows: the regions of competence of each classifier are specified during the training phase. During the evaluation of a test pattern x_{query}, the region around x_{query} is first found, and the classifier responsible for this region is called upon to label x_{query} (AVNIMELECH; INTRATOR, 1999a; VERIKAS et al., 1999). In recent works on dynamic classifier selection (OLIVEIRA; CAVALCANTI; SABOURIN, 2017; CRUZ et al., 2015; CRUZ; SABOURIN; CAVALCANTI, 2016), the static selection definition is presented as follows: given the initially generated ensemble $\mathcal{F}$, a subset is selected to satisfy all the patterns in the test set. This last definition is confused with the definition of ensemble pruning, and is the same one presented in the works of (ROONEY et al., 2004) and (MENDES-MOREIRA et al., 2009). This last definition is used from here to onwards.

In dynamic selection, for each test pattern x_{query}, the selection of the models is performed during the evaluation phase and the subset of the models $\mathcal{F}'$ is combined to predict the test pattern.

The next sections describe in details the two selection approaches and their main works from the literature.

2.2.1 Static Selection or Ensemble Pruning

The main goal of the static selection/pruning approach is to improve the predictive ability or reduce computational costs. Even in methods that are designed to use all the generated models, e.g Bagging (BREIMAN, 1996a), works as (ZHOU; WU; TANG, 2002; HERNANDEZ-LOBATO; MARTINEZ-MUNOZ; SUAREZ, 2006) show that the addition of a selection phase can reduce computational costs and improve the prediction accuracy.

Static selection methods can be classified as: (i) partitioning-based; and (ii) search-based. Partitioning-based methods divide the initial ensemble $\mathcal{F}$ into subensembles using some partitioning criterion. Then, for each subensemble, one or more models are selected using some selection criterion. Search-based methods search for a subset from the original ensemble $\mathcal{F}$ by iteratively adding or removing models from the subset according to a given evaluation measure and search algorithm. The next sections show both static selection methods.

2.2.1.1 Partitioning-Based Approaches

In this approach, it is believed that the ensemble $\mathcal{F}$ contains many similar and redundant models. The main idea of partitioning-based approaches is to divide the models into subensembles using a partitioning criterion and to select representative models (one or more) from each subensemble. In the partitioning-based approaches, the subensembles are generated using clustering algorithms. The goal is to build a subensemble $\mathcal{F}'$ with the best models, which typically means that the best models are accurate and diverse. With this, some diversity is guaranteed in $\mathcal{F}'$.

In (LAZAREVIC; OBRADOVIC, 2001), k-means clustering (LLOYD, 1982) algorithm is used to obtain clusters of similar models. The number of clusters is an input parameter of this approach and is used the prediction vectors made by the models as partitioning criterion. In (COELHO; ZUBEN, 2006) is presented the ARIA - Adaptive Radius Immune Algorithm for clustering. This algorithm does not require the number of clusters to be defined as an input parameter.

2.2.1.2 Search-Based Approaches

Simpler and more common than the previous one, this approach works like this: given an initial ensemble $\mathcal{F}$, search-based approaches search for the best subensemble $\mathcal{F}'$, using some search algorithm for iteratively adding or removing models in $\mathcal{F}'$. A simple, but costly way, is to choose the best subensemble $\mathcal{F}'$ using some criterion with a greedy search in space $2^N - 1$, where N is the size of the ensemble $\mathcal{F}$. This search is a NP complete problem (TAMON; XIANG, 2000) and intractable for ensembles with $N > 30$ (MARTíNEZ-MUñOZ; SUáREZ, 2006). This method may be useful for small ensembles.

Another search-based approach is randomized algorithms. Randomized Algorithms perform a heuristic search in the input space using stochastic methods, such as evolutionary algorithms. In (RUTA; GABRYS, 2001) is used three randomized algorithms to search for the best subensemble: genetic algorithms (KUNCHEVA; JAIN, 2000), tabu search (GLOVER; LAGUNA, 1998), and population-based incremental learning (BALUJA, 1994). The main result of the experiments is that the three algorithms outperform the greedy search. These results are performed using an ensemble $\mathcal{F}$ with small size.

Some works use sequential search algorithms. Sequential algorithms generate an initial solution (subensemble) and iteratively change by adding or removing models. Three types of sequential search algorithms are used:

- Forward: In this approach, the ensemble is initialized with an empty set. Models are appended to the ensemble in each iteration. This is referred to as Forward Selection (PARTALAS et al., 2008).
- Backward: The search begins with the entire initial ensemble and models are eliminated in each iteration. This is referred to as Backward Elimination (PARTALAS et al., 2008).
- Combined: Apply consecutive forward and backward steps (MOLINA; BELANCHE; NEBOT, 2002).

The work (MARTíNEZ-MUñOZ; SUáREZ, 2007) shows a sequential forward search method based on AdaBoost. In each iteration, this method selects one model from the ensemble $\mathcal{F}$ that minimizes the ensemble error. Although this method was originally proposed for classification, it can be directly applied to regression using AdaBoost.R or AdaBoost.R2.

Combined methods are more difficult to implement because of mix both forward and backward steps. Some examples of combined search methods for static selection are (MENDES-MOREIRA. et al., 2006) and (MARGINEANTU; DIETTERICH, 1997).

In (MENDES-MOREIRA. et al., 2006) is presented an algorithm that starts by randomly selecting a predefined number of M models. In each iteration, one forward step and one backward step are applied. The forward step selects a model from the initial ensemble which improves the accuracy of the subensemble. In the second step, one model is removed from the subensemble, leaving only the M models with higher ensemble accuracy. The process stops when the same model is selected in both steps.

In (MARGINEANTU; DIETTERICH, 1997) is shown another combined search algorithm called Reduce-Error Pruning with Backfitting. Firstly, a subensemble of three models $\mathcal{F}' = \{\hat{f}_1, \hat{f}_2, \hat{f}_3\}$ is created. In next iterations, model $\hat{f}_1$ is removed and other $\hat{f}_4$ is tested together $\hat{f}_2$ and $\hat{f}_3$, with the goal to reduce the subensemble error. The process is repeated for the models $\hat{f}_2$ and $\hat{f}_3$ and it continues until a predefined number of iterations is reached.

2.2.2 Dynamic Selection

In Dynamic Selection, the selection of the models is done during the evaluation of the test pattern x_{query}. For each pattern x_{query}, the region of competence is defined with the most similar patterns and a subensemble $\mathcal{F}'$ is selected to be combined in the final prediction $\hat{f}_{ens}(x_{query})$.

Some measure is used in the region of competence to select the models. The individual performance of each of the models is evaluated in the region of competence and the M models with the best performance are selected. In (MERZ, 1996) is described as the dynamic selection approach works for classification problems. Merz uses a performance matrix to evaluate the models in the region of competence. In classification problems, many measures were tested in the region of the competence (CRUZ; SABOURIN; CAVALCANTI, 2017), but many of them cannot be used or adapted for regression problems. In regression problems, the error measure can be, for instance, the squared error, the absolute error, or another error measure. If at the end of the dynamic selection phase only one model is selected, the prediction of x_{query} is the prediction of the selected model. If more than one model is selected, the models are combined to predict the pattern x_{query}.

Dynamic Selection consists of the following steps:

1. Given a test pattern x_{query}, find the region of competence with the K most similar patterns from the training set $\mathcal{T}$ or from the validation set $\mathcal{V}$.

2. Select a subensemble $\mathcal{F}'$ from the ensemble $\mathcal{F}$, where $\mathcal{F}' \subseteq \mathcal{F}$, according to each individual performance of the models in the region of competence.

3. Obtain the prediction $\hat{f}_n(x_{query})$ for each selected model $\hat{f}_n \in \mathcal{F}'$.

4. Obtain the ensemble prediction $\hat{f}_{ens}(x_{query})$. If more than one model is selected, the ensemble prediction is the result of the combination of the models in the subensemble $\mathcal{F}'$. Combination functions are explained in Section 2.3.

The standard method for obtaining similar data and find the region of competence in dynamic systems is the well-known k-nearest-neighbors ($k$nn) with the Euclidean distance (WOODS; KEGELMEYER; BOWYER, 1997). The choice of neighborhood size K can be decisive for the performance of the system. In the work of (MENDES-MOREIRA et al., 2009), the size of the neighborhood is tested with values in the set $\{2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 25, 30\}$ and the conclusion is that the best value is problem-dependent.

In (ROONEY et al., 2004) is adapted the Dynamic Classifier Selection (DCS) techniques proposed by (TSYMBAL; PUURONEN, 2000; TSYMBAL; PECHENIZKIY; CUNNINGHAM, 2006) for regression problems by defining three dynamic regressor selection algorithms: Dynamic Selection (DS), Dynamic Weighting (DW), and Dynamic Weighting with Selection (DWS). The algorithms dynamically select the most competent regressors in the

region of competence per test pattern. Such algorithms use the performance of the regressors in the region of competence as a selection criterion; it means that the regressors with the smallest cumulative error in the neighborhood are chosen to estimate the test pattern. In (MENDES-MOREIRA et al., 2009) is also used the error similarly to (ROONEY et al., 2004), however, in their work, the estimated errors are weighted by the distance between the patterns in the region of competence and the test pattern.

All of the three algorithms use the concept of region of competence. Given a test pattern x_{query}, its region of competence is a set Ψ composed of the K nearest neighbours of x_{query} in the validation or training set given by $\{t_1, t_2, ..., t_K\}$. These three algorithms were described by (MENDES-MOREIRA et al., 2009; MENDES-MOREIRA et al., 2015) as follows:

- Dynamic Selection (DS) - it selects the regressor with the lowest accumulated error in the region of competence. The errors are weighted by the distance between the neighborhood pattern and the test pattern. Only a single regressor is selected and no combination is required. The estimation of the test pattern is the value returned by the selected regressor.

- Dynamic Weighting (DW) - it combines all the regressors of the ensemble using the weighted mean. For each test pattern x_{query}, its region of competence Ψ is calculated; Ψ is composed of K patterns. For each pattern in Ψ, a weight is calculated using Equation 2.1.

$$d_k = \frac{\frac{1}{dist_k}}{\sum_{j=1}^{K}(\frac{1}{dist_j})} \tag{2.1}$$

  where $dist_k$ is a distance measure between a pattern $t_k \in \Psi$ and the test pattern x_{query}.

  The vector $\{d_1, d_2, ..., d_k\}, k \in \{1, 2, ..., K\}$, is used to calculate the weight α_i of the regressor $\hat{f}_i$ using Equation 2.2:

$$\alpha_i = \frac{\frac{1}{\sqrt{\sum_{k=1}^{K}(d_k \times sqe_{k,i})}}}{\sum_{n=1}^{N}\left(\frac{1}{\sqrt{\sum_{k=1}^{K}(d_k \times sqe_{k,n})}}\right)} \tag{2.2}$$

  where N is the ensemble size, k represents the index of the neighbor, and $sqe_{k,i}$ is the squared error of the regressor i calculated using the pattern $t_k \in \Psi$.

  This is a combination technique with the weights dynamically calculated and not constants. Some combination tecniques will be explained in details in Section 2.3.

- Dynamic Weighting with Selection (DWS) - it combines a subset of the regressors. The regressors with the accumulated error in the upper half of the error interval

$E_i > (E_{max} - E_{min})/2$ are discarded, where E_{max} is the largest accumulated error of any regressor and E_{min} is the lowest accumulated error of any regressor. The measure to calculate the performance of the regressors from the ensemble is the same than the DW algorithm and the remaining regressors are combined using the same strategy of the DW.

After selecting the most competent models, the combination is executed. In the next section, the main works for the combination of the regressors are presented.

2.3 COMBINATION

After the selection of the models, both in static or dynamic approaches, the next step is the combination of the models in $\mathcal{F}'$ to predict the test pattern x_{query}. In regression problems, the combination is performed using a linear function as shown in Equation 2.3.

$$\hat{f}_{ens}(x_{query}) = \sum_{i=1}^{M} \alpha_i \times \hat{f}_i(x_{query}) \tag{2.3}$$

where α_i is the weight of the model $\hat{f}_i$ and M is the size of $\mathcal{F}'$.

The combination functions can calculate the vector of weights $\{\alpha_1, \alpha_2, ..., \alpha_M\}$ using two approaches: (i) constant weights; (ii) nonconstant weights (MERZ, 1998). In the first approach, the vector of weights is constant and remain the same for all test patterns. In the second approach, the vector of weights varies according to the test pattern x_{query}.

In (PERRONE; COOPER, 1993) is defined two ways to combine the models from an ensemble: Basic Ensemble Method (BEM), and Generalized Ensemble Method (GEM). In the BEM, the combination of the models is performed using the mean among the regressors, where all the regressors have the same importance, according the Equation (2.4).

$$\hat{f}_{BEM} = \frac{1}{M} \sum_{i=1}^{M} \hat{f}_i(x_{query}) \tag{2.4}$$

In the GEM, the models are combined using the weighted mean where the weights are inversely proportional to the errors generated in the training set or the validation set, and the sum of the weights must be equals to 1. These weights are constants; it means that the weights do not change during the evaluation of test patterns.

Bragging (BUHLMANN, 2012) is a method to combine statically the models generated via Bagging using a median, instead of the mean. All the weights are equals and constant.

Breiman presents the stacked regression (BREIMAN, 1996c) method based on the stacked generalization framework (WOLPERT, 1992; ALDAVE; DUSSAULT, 2014) that was first presented in the context of classification. Given a training set with T examples, the

goal is to obtain the vector of weights α that minimize the error in the training set, using Equation (2.5).

$$\sum_{j=1}^{T}\left[f(t_j)-\sum_{i=1}^{M}\alpha_i\times\hat{f}_i(t_j)\right]^2 \tag{2.5}$$

where M is the ensemble size, α_i is the weight of the model $\hat{f}_i$ and $f(t_j)$ is the observed value of the training pattern t_j.

Dynamic Weighting (DW) (ROONEY et al., 2004; MENDES-MOREIRA et al., 2009), as mentioned in Section 2.2.2, calculates the vector of weights dynamically according to the test pattern x_{query}. For each model, the weight is inversely proportional to the accumulated error in the region of competence. Thus, for each test pattern, a different vector of weights is used.

In (CARUANA et al., 2004) is presented a forward selection approach with a static combination of the models. In this approach, models can be selected multiple times. The combination is a weighted mean, but the models added to the ensemble multiple times receive more weight.

The main disadvantage of using constant weights is that equal weights for the whole test set, can, at least theoretically, be less adequate for some test patterns. This is the main argument for using nonconstant weights (VERIKAS et al., 1999).

2.3.1 Other combination methods

One approach that can be explored and seems to be promising is to combine different ensemble combination methods. The method wMetaComb (ROONEY; PATTERSON, 2007) is a technique that fuses two combination techniques: Stacking (WOLPERT, 1992) and the DWS algorithm (ROONEY et al., 2004; MENDES-MOREIRA et al., 2009). In the wMetaComb, the estimated value of the test pattern is the weighted mean of the predictions of two combination techniques. The weights to combine the techniques are calculated based on the errors during the training phase.

Another method is the cocktail ensemble for regression (YU; ZHOU; TING, 2007). This method combines different ensemble approaches using forward selection to choose the one that most reduces the general error during combination. The method starts with the ensemble that produces the lowest estimated error and continues adding new ensembles, in order to reduce the error. The same ensemble can be selected more than once. In this method, both combination methods with constant or non-constant weights can be used to combine the models from an ensemble.

2.4 FINAL REMARKS

Many studies present better results with the use of ensembles rather than individual models (GARCÍA-PEDRAJAS; HERVÁS-MARTÍNEZ; ORTIZ-BOYER, 2005; MOURA; CAVALCANTI; OLIVEIRA, 2019; CAVALCANTI et al., 2016). The disadvantage of using ensembles lies in finding out which best techniques will be used to generate, select and combine the models. The "No Free Lunch" theorem (WOLPERT, 1996; WOLPERT; MACREADY, 1997) stipulates that a universal algorithm does not exist. So, we can say that there is no algorithm, into these three steps, that is better than all others to all problems.

The techniques of ensembles generation have different characteristics, each of them has a different way to construct the models. But, according to some studies (PRAMANIK et al., 2010; CHANDRAHASAN et al., 2011), none of them is better than the others for all the problems. The performance of the techniques depends much more on the dataset than any other factor.

As pointed out, dynamic selection has better results than static selection. In classification, much work of dynamic selection was produced (CRUZ; SABOURIN; CAVALCANTI, 2016; CRUZ et al., 2015; CRUZ; SABOURIN; CAVALCANTI, 2017; OLIVEIRA; CAVALCANTI; SABOURIN, 2017), but in regression, the work of (MENDES-MOREIRA et al., 2009) is the main reference.

Finally, as commented previously in Section 2.3, in general, the weighted combination of the models is better than using a simple mean.

3 EVALUATING COMPETENCE MEASURES FOR DYNAMIC REGRESSOR SELECTION

Evaluating Competence Measures for Dynamic Regressor Selection

Thiago J. M. Moura[1,2], George D. C. Cavalcanti[1], and Luiz S. Oliveira[3]
[1]Centro de Informática (CIn) - Universidade Federal de Pernambuco - Brazil
[2]Instituto Federal da Paraíba (IFPB) - Brazil
[3]Departamento de Informática (DInf) - Universidade Federal do Paraná - Brazil

ABSTRACT

Dynamic regressor selection (DRS) systems work by selecting the most competent regressors from an ensemble to estimate the target value of a given test pattern. This competence is usually quantified using the performance of the regressors in local regions of the feature space around the test pattern. However, choosing the best measure to calculate the level of competence correctly is not straightforward. The literature of dynamic classifier selection presents a wide variety of competence measures, which cannot be used or adapted for DRS. In this paper, we review eight measures used with regression problems, and adapt them to test the performance of the DRS algorithms found in the literature. Such measures are extracted from a local region of the feature space around the test pattern, called region of competence, therefore competence measures. To better compare the competence measures, we perform a set of comprehensive experiments on 15 regression datasets. Three DRS systems were compared against individual regressor and static systems that use the Mean and the Median to combine the outputs of the regressors from the ensemble. The DRS systems were assessed varying the competence measures. Our results show that DRS systems outperform individual regressors and static systems but the choice of the competence measure is problem-dependent.

3.1 INTRODUCTION

Model selection systems consist of two main phases (CRUZ; SABOURIN; CAVALCANTI, 2018): Generation and Selection. In the first phase, a training set is used to generate the ensemble. The ensemble is said homogeneous when a single learning algorithm is used to train the models. Otherwise, it is called heterogeneous. In the second phase, a model or a subset of models are selected to evaluate the test pattern. Such a selection can be done according to two distinct approaches: static and dynamic. In the static approach, selection occurs using the performance of the models in the training set (ORTIZ-BOYER; HERVáS-MARTíNEZ; GARCíA-PEDRAJAS, 2005) or using a separated validation set after the training stage (PARTALAS et al., 2008). In the static selection, the models are used to evaluate all test patterns. In the dynamic approach, a different model or subset of

models from the ensemble are selected for each new test pattern. In dynamic selection techniques, each model is an expert in a specific local region of the feature space, which is known as region of competence. So, for each test pattern, the most competent models are selected for the region of competence where the test pattern is located. The region of competence contains the patterns from the training set or the validation set which are neighbors of the test pattern, also known as the neighborhood of the test pattern. The standard method for defining the region of competence is the k-nearest neighbors (kNN) algorithm with Euclidean distance (WOODS; KEGELMEYER; BOWYER, 1997). Dynamic selection is a growing research area in machine learning, and recent works have shown that dynamic selection techniques are more efficient than static selection for both classification and regression problems (BRITTO; SABOURIN; OLIVEIRA, 2014; MENDES-MOREIRA et al., 2012).

The central issue in dynamic selection is to define the criterion to measure the competence of each model in the ensemble, i.e., the competence measure. In general, the accuracy of the models in the region of competence is used as a criterion for determining the competence. Some works of dynamic classifier selection (DCS) (SANTANA et al., 2006), (SANTOS; SABOURIN; MAUPIN, 2008) use other measures, beyond accuracy, to calculate the competence. Recent works on DCS (CRUZ et al., 2015), (CRUZ; SABOURIN; CAVALCANTI, 2016), (CRUZ; SABOURIN; CAVALCANTI, 2017) use the composition of many measures to determine the competence of the classifiers, selecting and combining them to predict the class of the test pattern.

Many of the measures used in DCS cannot be directly used for regression. So, dynamic regressor selection (DRS) literature methods commonly use the error of the predictions in the region of competence as a criterion to dynamically select the best regressors, e.g., Rooney et al. (ROONEY et al., 2004) and Moreira et al. (MENDES-MOREIRA et al., 2009). Rooney et al. adapted the DCS technique proposed by Tsymbal et al. (TSYMBAL; PUURONEN, 2000), (TSYMBAL; PECHENIZKIY; CUNNINGHAM, 2006) for regression problems by defining three DRS algorithms: Dynamic Selection (DS), Dynamic Weighting (DW), and Dynamic Weighting with Selection (DWS). The algorithms dynamically select the most competent regressors in the region of competence per test pattern. Such algorithms use the performance of the regressors in the region of competence as a selection criterion; it means that the regressors with the smallest cumulative error in the neighborhood are chosen to estimate the test pattern. Moreira et al. (MENDES-MOREIRA et al., 2009) also use the error similarly to Rooney et al., however, in their work, the estimated errors are weighted by the distance between the patterns in the region of competence and the test pattern.

Having in mind that literature uses the prediction error in the region of competence as a criterion to select the best regressors, we assume that DRS algorithms can benefit from different criteria to select the best regressors per query pattern. So, we perform an

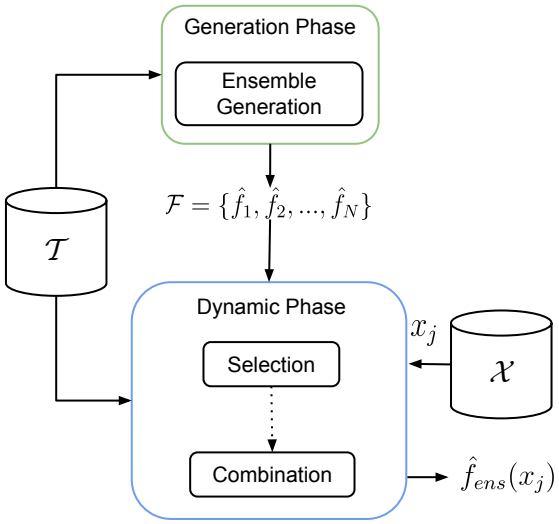


Figure 6 – Overview of the Dynamic Regressor Selection architecture. $\mathcal{T}$ and $\mathcal{X}$ are the training and testing sets respectively. $\mathcal{F}$ is the ensemble of regressors generated in the Generation Phase, x_j is a test pattern and $\hat{f}_{ens}(x_j)$ is the result of the test pattern estimate.

empirical evaluation of eight measures that can employed as a criterion to measure the competence (competence measures) of the regressors for DRS.

To the best of our knowledge, seven of these measures are adapted for the first time to this task, and they capture different information, such as weighted error, variance, and similarity. These eight competence measures are evaluated using 15 regression problems from different data repositories and three literature algorithms: DS, DW, and DWS.

The contributions of this work are: i) Evaluation of eight competence measures that are used as a criterion to select the best regressors per query pattern. Seven of these eight measures are evaluated for the first time in this task; ii) Comparative study using three dynamic selection algorithms (DS, DW, and DWS) and all the competence measures; iii) Comparison between dynamic systems and individual regressor; iv) Comparison between dynamic and static systems that use the Mean and the Median to combine the outputs of the regressors from the ensemble.

This paper is organized as follows. Section 3.2 presents the dynamic regressors selection algorithms. Section 3.3 describes the eight measures. The experimental results are shown in Section 3.4 and the final remarks are presented in Section 3.5.

3.2 DRS ALGORITHMS

A general overview of the DRS architecture is depicted in Figure 6. The architecture is divided into two phases: Generation and Dynamic. They are described in the following subsections.

3.2.1 Generation Phase

This phase generates an ensemble $\mathcal{F} = \{\hat{f}_1, \hat{f}_2, ..., \hat{f}_N\}$ using the training set $\mathcal{T}$, where N is the number of regressors. The ensemble can be homogeneous or heterogeneous. The former uses different sets to train each regressor $\hat{f}_n, \forall n \in \{1, 2, ..., N\}$. These different sets are commonly generated using Bagging (BREIMAN, 1996a), Boosting (SHRESTHA; SOLOMATINE, 2006), or Random Subspace (HO, 1998). Heterogeneous ensembles, on the other hand, are generated using different learning algorithms for training the regressors.

3.2.2 Dynamic Phase

The Dynamic phase selects a subset of the regressors per test pattern $x_j \in \mathcal{X}$ and it can work in three different ways: (I) select only one regressor from the ensemble $\mathcal{F}$; (II) weighted combination of all the regressors and; (III) select a subset of the regressors and combine them. The result of this phase is the ensemble prediction $\hat{f}_{ens}(x_j)$.

Any of the three algorithms proposed in (ROONEY et al., 2004) for the Dynamic phase can be used in the architecture depicted in Figure 6. All of them use the concept of region of competence. Given a test pattern x_j, its region of competence is a set Ψ composed of the K nearest neighbours of x_j in the validation or training set given by $\{t_1, t_2, ..., t_K\}$. These three algorithms were described by Moreira et al. (MENDES-MOREIRA et al., 2009; MENDES-MOREIRA et al., 2015) as follows:

- Dynamic Selection (DS) - it selects the regressor with the lowest accumulated error in the region of competence. The errors are weighted by the distance between the neighborhood pattern and the test pattern. Only a single regressor is selected and no combination is required. The estimation of the test pattern is the value returned by the selected regressor.

- Dynamic Weighting (DW) - it combines all the regressors of the ensemble using the weighted mean. For each test pattern x_j, its region of competence Ψ is calculated; Ψ is composed of K patterns. For each pattern in Ψ, a weight is calculated using Equation 3.1:

$$d_k = \frac{\frac{1}{dist_k}}{\sum_{j=1}^{K}(\frac{1}{dist_j})} \tag{3.1}$$

  where $dist_k$ is a distance measure between a pattern $t_k \in \Psi$ and the test pattern x_j.

The vector $\{d_1, d_2, ..., d_k\}, k \in \{1, 2, ..., K\}$, is used to calculate the weight α_i of the regressor $\hat{f}_i$ using Equation 3.2:

$$\alpha_i = \frac{\frac{1}{\sqrt{\sum_{k=1}^{K}(d_k \times sqe_{k,i})}}}{\sum_{n=1}^{N}\left(\frac{1}{\sqrt{\sum_{k=1}^{K}(d_k \times sqe_{k,n})}}\right)} \tag{3.2}$$

where N is the ensemble size, k represents the index of the neighbor, and $sqe_{k,i}$ is the squared error of the regressor i calculated using the pattern $t_k \in \Psi$.

- Dynamic Weighting with Selection (DWS) - it combines a subset of the regressors. The regressors with the accumulated error in the upper half of the error interval $E_i > (E_{max} - E_{min})/2$ are discarded, where E_{max} is the largest accumulated error of any regressor and E_{min} is the lowest accumulated error of any regressor. The measure to calculate the performance of the regressors from the ensemble is the same than the DW algorithm and the remaining regressors are combined using the same strategy of the DW.

In (MENDES-MOREIRA et al., 2009), the authors use the Root Sum Squared Error as a competence measure to select a regressor from the ensemble for the three aforementioned algorithms. In Section 3.3, this competence measure is defined as m_7. In spite of the good results reported in (MENDES-MOREIRA et al., 2009), we show that other competence measures perform better and should not be neglected. To the best of our knowledge, this is the first work that analyzes other competence measures for DRS algorithms.

3.3 COMPETENCE MEASURES

Table 2 shows a summary of all competence measures used in this work. These measures correspond to different criteria to measure the behavior of each regressor from the ensemble $\mathcal{F}$. Each measure expresses one of the following information: (i) the error calculated in the region of competence; (ii) the variance of the estimated values in the neighborhood; or (iii) the similarity between the observed and the estimated values of the test pattern x_j.

Some competence measures are calculated using the distances between the test pattern and the nearest neighbors. However, instead of using the distance, we use the inverse of the normalized distance (d_k) in the interval [0,1]. So, smaller the distance greater the value of d_k, according to Equation 3.1.

The measures described below are extracted using the region of competence $\{t_1, t_2, ..., t_K\}$ of the test pattern x_j. In the next equations, $f(t_k)$ stands for the observed value of the neighborhood pattern and $\hat{f}_n(t_k)$ is the estimated value of the pattern t_k given by the regressor $\hat{f}_n$.

Table 2 – Summary of the Competence Measures.

Measure	**Acronym**	**Equation**
Variance	m_1	$Var(\hat{f}_n(t_1), \hat{f}_n(t_2), ..., \hat{f}_n(t_K))$
Sum Absolute Error	m_2	$\sum_{k=1}^{K} \left\| f(t_k) - \hat{f}_n(t_k) \right\| \times d_k$
Sum Squared Error	m_3	$\sum_{k=1}^{K} (f(t_k) - \hat{f}_n(t_k))^2 \times d_k$
Minimum Squared Error	m_4	$\min_{1 \leq k \leq K} \{(f(t_k) - \hat{f}_n(t_k))^2 \times d_k\}$
Maximum Squared Error	m_5	$\max_{1 \leq k \leq K} \{(f(t_k) - \hat{f}_n(t_k))^2 \times d_k\}$
Neighbor's Similarity	m_6	$\sum_{k=1}^{K} (f(t_k) - \hat{f}_n(x_j))^2 \times d_k$
Root Sum Squared Error	m_7	$\sum_{k=1}^{K} \sqrt{(f(t_k) - \hat{f}_n(t_k))^2 \times d_k}$
Closest Squared Error	m_8	$(f(t_1) - \hat{f}_n(t_1))^2$

- m_1 - *Variance*: the variance of the neighbors estimated values. The variance is calculated for each regressor using the estimated values of the patterns in the region of competence, according to Equation 3.3:

$$m_1 = Var(\hat{f}_n(t_1), \hat{f}_n(t_2), ..., \hat{f}_n(t_K)) \tag{3.3}$$

  This competence measure is inspired in the work of Tresp et al. (TRESP; TANIGUCHI, 1995), whose variance of the estimated values is used as weight in the static combination of artificial neural networks.

- m_2 - *Sum Absolute Error*: the sum of the absolute errors is calculated in the region of competence, weighted by d_k, according to Equation 3.4:

$$m_2 = \sum_{k=1}^{K} \left| f(t_k) - \hat{f}_n(t_k) \right| \times d_k \tag{3.4}$$

- m_3 - *Sum Squared Error*: the sum of the squared errors is calculated using the inverse of the distances d_k as weights, according to Equation 3.5:

$$m_3 = \sum_{k=1}^{K} (f(t_k) - \hat{f}_n(t_k))^2 \times d_k \tag{3.5}$$

- m_4 - *Minimum Squared Error*: the minimum value of squared errors is calculated using the inverse of the distances d_k as weights. The measure m_4 is computed using Equation 3.6:

$$m_4 = \min_{1 \leq k \leq K} \{(f(t_k) - \hat{f}_n(t_k))^2 \times d_k\} \tag{3.6}$$

- m_5 - *Maximum Squared Error*: the maximum value of squared errors is calculated using the inverse of the distances d_k as weights. The measure m_5 is computed using Equation 3.7:

$$m_5 = \max_{1 \leq k \leq K} \{(f(t_k) - \hat{f}_n(t_k))^2 \times d_k\} \tag{3.7}$$

  Considering that m_4 and m_5 define an interval, these measures present mean and variance, it means that, the interval contains information about implicit measures of dispersion (error variance) and centrality (error mean) of the squared error in the region of competence.

- m_6 - *Neighbor's Similarity*: the sum of the differences between the estimated values of the test pattern from test set $\mathcal{X}$ and the observed value of each neighborhood pattern, weighted by the inverse of the distance. The measure m_6 is computed using Equation 3.8:

$$m_6 = \sum_{k=1}^{K} (f(t_k) - \hat{f}_n(x_j))^2 \times d_k \tag{3.8}$$

  where $\hat{f}_n(x_j)$ is the estimated value by the regressor $\hat{f}_n$ for x_j.

  The goal of the competence measure m_6 is to find the degree of similarity between the estimation of the pattern x_j and the observed values of the nearest neighbors. This is the only measure that uses in its calculation the estimated value for the test pattern ($\hat{f}_n(x_j)$). So far as we know, this measure is unprecedented and is defined by the authors of this work.

- m_7 - *Root Sum Squared Error*: the root sum squared errors in region of competence, weighted by d_k. The measure m_7 is computed using Equation 3.9:

$$m_7 = \sum_{k=1}^{K} \sqrt{(f(t_k) - \hat{f}_n(t_k))^2 \times d_k} \tag{3.9}$$

  The root squared error is more stable and less sensitivity to the difference between the maximum and the minimum errors, while the squared error is very sensitive to extreme error values. The measures m_3 and m_7 present different points of view from the error calculated in the region of competence. These two measures produce the same result when a single regressor is chosen to estimate a test pattern, but different results in the combination of the regressors.

- m_8 - *Closest Squared Error*: the error obtained by the regressor only on the nearest neighbor. The measure m_8 is computed using Equation 3.10:

$$m_8 = (f(t_1) - \hat{f}_n(t_1))^2 \tag{3.10}$$

Table 3 – Datasets used in the experiments.

Dataset	Instances	Features	Source
Airfoil Self Noise	1503	5	UCI
Bank32NH	8192	32	Delve
Bank8FM	8192	8	Delve
Breast Cancer	194	32	Torgo
CCPP	9568	4	UCI
Concrete	1030	8	UCI
Delta Ailerons	7129	6	Torgo
Delta Elevators	9517	6	Torgo
Housing	506	13	UCI
Kinematics	8192	8	Delve
Machine	209	6	Torgo
Puma32H	8192	32	Delve
Puma8NH	8192	8	Delve
Stock	950	9	Torgo
Triazines	186	60	Torgo

3.4 EXPERIMENTS

3.4.1 Datasets

To show the performance of the DRS algorithms, a total of 15 regression datasets were used in the comparative study. The main features of each dataset are shown in Table 3. These are public datasets, which are available in the following repositories: personal page of Prof. Luís Torgo[1], UCI machine learning repository[2], and Delve repository[3].

3.4.2 Experimental Protocol

For each dataset, all data attributes are normalized into the interval [0,1], and the experiments were conducted using 30 replications. For each replication, the data is randomly split into ten parts of approximately the same size. Then, a 10-fold cross-validation is carried out using 90% of the folds as the training set ($\mathcal{T}$) and 10% as the testing set ($\mathcal{X}$).

In the experiments, homogeneous ensembles of size $N = 100$ were generated using Bagging (BREIMAN, 1996a). Bagging generates different datasets, using sampling with replacement. Each generated dataset has the same size of the training set. Using replacement, some instances will be repeated in each dataset, and on average only 63% of instances will be unique. The learning algorithm CART (BREIMAN et al., 1984) was used with default settings found in MATLAB without any specific adjustment.

[1] http://www.dcc.fc.up.pt/∼ltorgo/Regression/DataSets.html
[2] http://http://archive.ics.uci.edu/ml/
[3] http://www.cs.toronto.edu/∼delve/

In (MENDES-MOREIRA et al., 2009), experiments were performed varying the size of the region of competence K in the set $\{2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 25, 30\}$. They concluded that the appropriate size for the neighborhood is problem-dependent, so they fixed the size of the region of competence with $K = 10$. Analyzing works of classification (CRUZ et al., 2015; CRUZ; SABOURIN; CAVALCANTI, 2016), time-series forecasting (SERGIO; LIMA; LUDERMIR, 2016), and regression (ROONEY et al., 2004), it can be verified that the size of the region of competence is fixed for better validation and comparison of the results. So, all the experiments in these sections use $K = 10$ as the size of the region of competence.

For each test set, the Mean Squared Error - MSE is computed. The result shows the arithmetic mean of the MSE calculated for the 10 test sets used in the cross-validation. A single individual regressor was trained with the entire training set without the use of Bagging. The performance of this regressor is compared with the dynamic selection algorithm (DS), as will be presented in Section 3.4.3.

3.4.3 DS Results

This section presents the results of the experiments performed using the DS algorithm. The experiments aim to compare the results obtained by the DS using each competence measure described in Section 3.3. Table 4 shows the arithmetic mean of the results calculated over 30 replications for each dataset. Individual Regressor represents a single model trained using the whole training set $\mathcal{T}$, as explained in Section 3.4.2.

According to Table 4, the DS algorithm was better in 11 out of 15 and the individual regressor was better in only 4 out of 15 datasets. So, DRS is a good way to predict new test patterns, instead of a single regressor. Second, measure m_6 achieved the best performance. As pointed out earlier, only this measure uses the estimated value of the test pattern in its calculation. With the use of an ensemble with many regressors, this competence measure is interesting when a single regressor is selected from the ensemble.

Figure 7(a) shows the difference of the errors calculated in Table 4 between measures m_7 and the best measure ($m*$) for each dataset. The difference of the errors in the datasets Bank8FM, Concrete, Housing, and Puma32H are zero or close to zero. We conclude that m_7 measure is not better in any dataset when DS algorithm is used.

3.4.4 DW and DWS Results

Tables 5 and 6 show the results to DW and DWS algorithms, respectively. Both algorithms combine the regressors from the ensemble. DW combine all the regressors using weighted mean and DWS selects and combine a subset of them.

Analyzing the results in Table 5, DW algorithm achieved better performance when compared with Mean and Median. Mean reached better performance in 3 out of 15 datasets, and Median has only one tie at dataset CCPP. Among the eight competence

Table 4 – Mean and standard deviation of the results of the MSE over 30 replications obtained for the DS algorithm and Individual Regressor. The best results are in bold. Line "Win/Tie/Loss" shows the total of the results. Error values are in the scale 10^{-4}.

		DS							
Dataset	**Individual Regressor**	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8
Airfoil Self Noise	6.17(0.25)	10.47(0.36)	**4.58(0.21)**	4.66(0.18)	7.34(0.40)	4.70(0.22)	6.65(0.10)	4.66(0.18)	5.99(0.34)
Bank32NH	20.35(0.36)	21.16(0.37)	21.21(0.42)	21.14(0.34)	21.13(0.39)	21.14(0.38)	**19.51(0.06)**	21.14(0.34)	21.14(0.32)
Bank8FM	**2.69(0.04)**	3.05(0.06)	3.02(0.06)	3.01(0.05)	3.02(0.05)	3.01(0.05)	7.30(0.04)	3.01(0.05)	3.02(0.05)
Breast Cancer	122.17(7.29)	107.29(9.19)	128.44(9.23)	128.53(10.81)	129.84(9.34)	128.00(9.71)	**68.36(1.13)**	128.53(10.81)	126.62(10.12)
CCPP	3.04(0.06)	3.32(0.06)	3.30(0.06)	3.28(0.08)	3.75(0.08)	3.27(0.08)	**2.30(0.01)**	3.28(0.08)	3.69(0.07)
Concrete	6.26(0.27)	10.30(0.67)	6.23(0.40)	6.23(0.40)	8.60(0.58)	**6.21(0.48)**	9.74(0.23)	6.23(0.40)	7.65(0.47)
Delta Ailerons	2.27(0.05)	2.15(0.04)	2.53(0.05)	2.53(0.05)	2.53(0.06)	2.53(0.06)	**1.57(0.01)**	2.53(0.05)	2.52(0.05)
Delta Elevators	4.50(0.05)	4.25(0.05)	5.14(0.08)	5.13(0.07)	5.01(0.08)	5.10(0.07)	**3.04(0.01)**	5.13(0.07)	5.06(0.08)
Housing	9.73(0.95)	13.61(1.67)	9.95(1.22)	**9.71(1.48)**	11.10(1.34)	9.99(1.43)	10.49(0.55)	**9.71(1.48)**	10.84(1.36)
Kinematics	20.52(0.40)	19.94(0.37)	21.04(0.37)	21.01(0.36)	23.84(0.35)	20.96(0.36)	**6.41(0.03)**	21.01(0.36)	23.88(0.36)
Machine	**3.65(1.05)**	8.31(0.84)	4.15(1.53)	4.04(1.25)	5.11(1.89)	3.87(0.94)	4.35(0.68)	4.04(1.25)	4.23(1.40)
Puma32H	**3.54(0.03)**	3.91(0.05)	3.86(0.06)	3.86(0.05)	3.91(0.06)	3.87(0.05)	8.98(0.05)	3.86(0.05)	3.89(0.05)
Puma8NH	30.01(0.35)	31.13(0.46)	32.39(0.40)	32.40(0.39)	32.38(0.42)	32.36(0.52)	**22.94(0.09)**	32.40(0.39)	32.51(0.53)
Stock	1.48(0.14)	1.76(0.17)	1.41(0.16)	1.38(0.17)	1.89(0.21)	1.38(0.15)	**0.73(0.01)**	1.38(0.17)	1.76(0.18)
Triazines	**32.85(3.39)**	37.61(3.15)	36.94(4.84)	37.71(4.71)	39.56(4.30)	36.94(4.35)	32.92(0.95)	37.71(4.71)	38.60(5.21)
Win/Tie/Loss	4/0/11	0/0/15	1/0/14	0/1/14	0/0/14	1/0/14	8/0/7	1/0/14	0/0/15

Table 5 – Mean and standard deviation of the results of the MSE over 30 replications obtained for the DW algorithm, Mean and Median. The best results are in bold. Line "Win/Tie/Loss" shows the total of the results. Error values are in the scale 10^{-4}.

			DW							
Dataset	**Mean**	**Median**	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8
Airfoil Self Noise	2.84(0.06)	3.12(0.08)	3.96(0.17)	2.46(0.07)	2.31(0.08)	2.92(0.09)	**2.27(0.09)**	3.12(0.06)	2.50(0.07)	4.15(0.25)
Bank32NH	**10.81(0.06)**	11.26(0.08)	10.94(0.07)	10.87(0.06)	11.39(0.09)	11.15(0.07)	11.54(0.10)	13.34(0.05)	10.88(0.06)	15.35(0.18)
Bank8FM	**1.52(0.01)**	1.59(0.01)	**1.52(0.01)**	**1.52(0.01)**	1.55(0.01)	1.57(0.01)	1.56(0.01)	1.62(0.01)	**1.52(0.01)**	2.18(0.03)
Breast Cancer	72.58(1.82)	79.52(2.97)	71.99(1.88)	72.82(2.06)	74.26(2.59)	73.89(2.48)	74.73(2.71)	**70.82(1.69)**	72.84(2.01)	91.01(5.65)
CCPP	1.92(0.01)	**1.85(0.01)**	1.96(0.01)	**1.85(0.02)**	**1.85(0.02)**	1.98(0.02)	**1.85(0.02)**	1.87(0.01)	**1.85(0.02)**	2.62(0.04)
Concrete	3.91(0.13)	3.96(0.15)	4.19(0.13)	3.48(0.15)	**3.44(0.17)**	3.82(0.13)	**3.44(0.17)**	3.97(0.14)	3.48(0.14)	5.11(0.31)
Delta Ailerons	1.43(0.01)	1.48(0.01)	**1.42(0.01)**	1.45(0.01)	1.49(0.02)	1.45(0.01)	1.49(0.02)	1.43(0.01)	1.45(0.01)	1.58(0.02)
Delta Elevators	2.92(0.01)	3.02(0.01)	**2.89(0.01)**	2.96(0.01)	3.02(0.02)	2.93(0.01)	3.03(0.01)	**2.89(0.01)**	2.95(0.01)	3.18(0.02)
Housing	5.48(0.28)	5.48(0.55)	5.81(0.33)	5.21(0.31)	**5.17(0.36)**	5.51(0.26)	5.20(0.38)	6.01(0.30)	5.20(0.31)	6.98(0.71)
Kinematics	9.89(0.05)	9.46(0.06)	9.65(0.05)	9.42(0.05)	9.21(0.07)	10.33(0.06)	9.27(0.07)	**6.79(0.02)**	9.41(0.05)	15.32(0.19)
Machine	2.73(0.70)	2.86(0.98)	5.04(0.88)	**2.53(0.80)**	2.69(0.83)	2.88(0.64)	2.78(0.84)	3.26(0.81)	2.54(0.81)	3.42(0.98)
Puma32H	**1.94(0.01)**	1.98(0.01)	**1.94(0.01)**	1.95(0.01)	1.98(0.01)	2.04(0.01)	2.01(0.01)	2.05(0.01)	1.95(0.01)	3.04(0.04)
Puma8NH	**17.96(0.06)**	18.78(0.08)	17.97(0.06)	18.04(0.07)	18.36(0.10)	18.30(0.08)	18.48(0.11)	18.28(0.06)	18.05(0.07)	23.76(0.29)
Stock	0.87(0.04)	0.87(0.06)	0.95(0.08)	0.78(0.05)	0.75(0.05)	0.86(0.04)	0.74(0.05)	**0.62(0.03)**	0.77(0.05)	1.05(0.08)
Triazines	**23.59(1.35)**	25.40(2.08)	26.97(1.51)	23.91(1.53)	24.69(1.77)	24.12(1.45)	25.04(1.79)	25.97(1.26)	23.96(1.53)	29.89(3.04)
Win/Tie/Loss	3/2/10	0/1/14	1/3/11	1/2/12	1/2/12	0/0/15	1/2/12	3/1/11	0/2/13	0/0/15

Table 6 – Mean and standard deviation of the results of the MSE over 30 replications obtained for the DWS algorithm, Mean and Median. The best results are in bold. Line "Win/Tie/Loss" shows the total of the results. Error values are in the scale 10^{-4}.

			DWS							
Dataset	**Mean**	**Median**	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8
Airfoil Self Noise	2.84(0.06)	3.12(0.08)	4.98(0.16)	**2.25(0.08)**	2.29(0.09)	2.92(0.09)	**2.25(0.09)**	4.01(0.08)	2.29(0.08)	4.15(0.25)
Bank32NH	**10.81(0.06)**	11.26(0.08)	11.06(0.08)	10.93(0.06)	11.41(0.09)	11.15(0.07)	11.55(0.10)	13.98(0.05)	10.95(0.06)	15.35(0.18)
Bank8FM	**1.52(0.01)**	1.59(0.01)	1.64(0.02)	1.54(0.01)	1.55(0.01)	1.57(0.01)	1.56(0.01)	3.03(0.03)	1.54(0.01)	2.18(0.03)
Breast Cancer	72.58(1.82)	79.52(2.97)	72.91(2.33)	74.37(2.67)	74.58(2.67)	73.89(2.48)	75.29(2.78)	**70.26(1.51)**	75.26(2.52)	91.01(5.65)
CCPP	1.92(0.01)	1.85(0.01)	2.08(0.01)	**1.83(0.02)**	1.84(0.02)	1.98(0.02)	1.85(0.02)	1.99(0.01)	**1.83(0.02)**	2.62(0.04)
Concrete	3.91(0.13)	3.96(0.15)	5.23(0.21)	**3.43(0.15)**	**3.43(0.17)**	3.82(0.13)	3.44(0.17)	5.33(0.15)	**3.43(0.14)**	5.11(0.31)
Delta Ailerons	**1.43(0.01)**	1.48(0.01)	**1.43(0.01)**	1.50(0.02)	1.50(0.02)	1.45(0.01)	1.50(0.02)	1.46(0.01)	1.51(0.02)	1.58(0.02)
Delta Elevators	2.92(0.01)	3.02(0.01)	**2.89(0.01)**	3.06(0.02)	3.04(0.02)	2.93(0.01)	3.04(0.02)	2.91(0.01)	3.09(0.02)	3.18(0.02)
Housing	5.48(0.28)	5.48(0.55)	7.42(0.43)	5.20(0.31)	**5.17(0.37)**	5.51(0.26)	5.20(0.38)	7.06(0.55)	5.21(0.32)	6.98(0.71)
Kinematics	9.89(0.05)	9.46(0.06)	9.75(0.06)	9.05(0.06)	9.16(0.07)	10.33(0.06)	9.23(0.07)	**6.50(0.02)**	9.01(0.07)	15.32(0.19)
Machine	2.73(0.70)	2.86(0.98)	6.15(0.78)	2.89(0.93)	**2.69(0.85)**	2.88(0.64)	2.78(0.85)	3.69(0.84)	2.97(0.96)	3.42(0.98)
Puma32H	**1.94(0.01)**	1.98(0.01)	2.04(0.01)	1.97(0.01)	1.99(0.01)	2.04(0.01)	2.01(0.01)	3.94(0.04)	1.98(0.01)	3.04(0.04)
Puma8NH	**17.96(0.06)**	18.78(0.08)	19.27(0.15)	18.37(0.09)	18.42(0.10)	18.30(0.08)	18.53(0.11)	19.70(0.07)	18.45(0.11)	23.76(0.29)
Stock	0.87(0.04)	0.87(0.06)	0.97(0.08)	0.75(0.05)	0.74(0.05)	0.86(0.04)	0.74(0.05)	**0.63(0.02)**	0.74(0.05)	1.05(0.08)
Triazines	**23.59(1.35)**	25.40(2.08)	27.50(1.58)	25.26(1.96)	24.99(1.84)	24.12(1.45)	25.30(1.83)	28.45(1.13)	25.53(1.96)	29.89(3.04)
Win/Tie/Loss	5/1/19	0/0/15	1/1/13	1/2/12	2/1/12	0/0/15	0/1/14	3/0/12	0/2/13	0/0/15

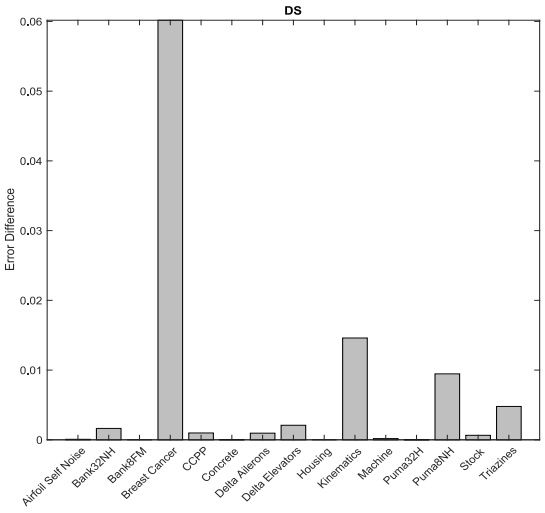
measures tested, six of them emerge with better results in DW, that is, none of them has superior performance for all datasets.

Figure 7(b) shows the difference of the errors calculated in Table 5 between m_7 and the best measure ($m*$) for each dataset. Using the measures as weights for the combination of the regressors from the ensemble, as is done in DW algorithm, we conclude that m_7 is not better in any database when used with the DW algorithm. For the datasets Bank8FM, CCPP, and Puma32H, the difference of the errors are zero or close to zero.

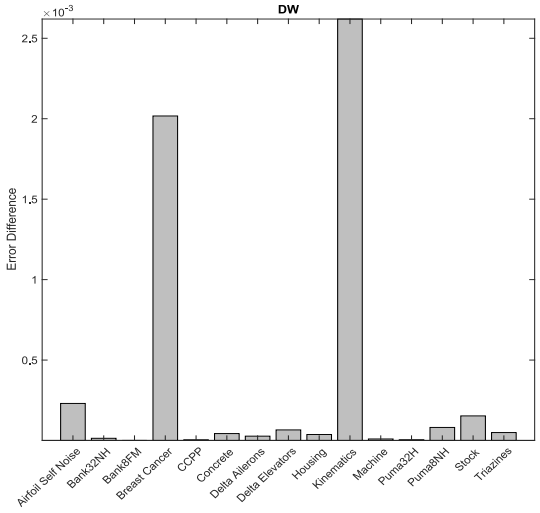
In Table 6, the DWS algorithm has better performance when compared with Mean and Median. Mean has better performance in 5 out of 15 datasets, and Median does not perform better in any dataset. Among the eight competence measures tested, six of them emerge with better results in DWS, that is, none of them has superior performance for all datasets.

Figure 7(c) shows the difference of the errors calculated in Table 6 between m_7 and the best measure. ($m*$) for each dataset. The same behavior observed with DW algorithm can be noticed here. m_7 is not better in any dataset and for the datasets Bank8FM, CCPP, Concrete, and Puma32H, the difference of the errors are zero or close to zero.

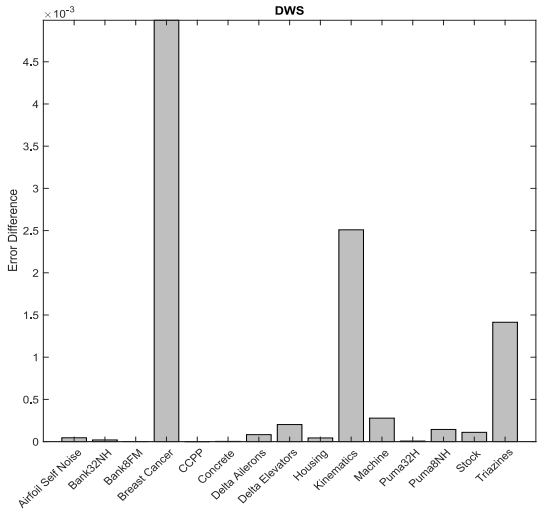
The experimental results show that m_7 proposed as competence measure in (MENDES-MOREIRA et al., 2009) performs better in some datasets, but for others, there are competence measures that bring better overall performance for DRS systems.



(a) DS algorithm



(b) DW algorithm



(c) DWS algorithm

Figure 7 – Comparison between measure m_7 and the best measure for each dataset. The bars present the difference of the errors between m_7 and $m*$, where $m*$ is the lowest error rate among the other measures.

3.5 CONCLUSION

Differently from the literature on DCS, where several competence measures have been proposed and assessed during the last decade (BRITTO; SABOURIN; OLIVEIRA, 2014; CRUZ; SABOURIN; CAVALCANTI, 2018), the number of works dealing with DRS is quite limited. The central issue in DRS, i.e., defining competence measures to help selecting the best regressor or ensemble of regressors, has been neglected in most of the works. To fill this gap, in this work we review eight competence measures, which were assessed using three different DRS systems and 15 datasets.

As presented in Section 3.4, DRS techniques perform better when compared to a single individual regressor or to classic statistical techniques such as Mean and Median. Another situation is that the reduction in the variance achieved by weighted average can explain why DW and DWS are better than DS (TSYMBAL; PECHENIZKIY; CUNNINGHAM, 2006).

It is possible to conclude that the competence measure used to select the regressors is problem-dependent. As previously mentioned, the literature techniques presented by (MENDES-MOREIRA et al., 2009), Section 3.2, use m_7 to select and combine the regressors, but our experiments pointed out that it does not have the best performance in all situations.

For future works, we can test a solution to select, for each regression problem, the best measure to be used in DRS techniques. Another solution is to combine the measures to select the most competent regressor or to use this combination as weighting to fuse the regressors from the ensemble.

4 MINE: A FRAMEWORK FOR DYNAMIC REGRESSOR SELECTION

MINE: A Framework for Dynamic Regressor Selection

Thiago J. M. Moura[1,2], George D. C. Cavalcanti[1], and Luiz S. Oliveira[3]
[1]Centro de Informática (CIn) - Universidade Federal de Pernambuco - Brazil
[2]Instituto Federal da Paraíba (IFPB) - Brazil
[3]Departamento de Informática (DInf) - Universidade Federal do Paraná - Brazil

ABSTRACT

Dynamic Regressor Selection (DRS) techniques aim to select the most competent regressors from an ensemble and combine them to estimate the target value of a given test pattern. For each test pattern, only the most competent regressors are selected and combined. Hence, the central issue in dynamic selection techniques is how to define the competence of the regressors to select the most competent ones. This competence usually is defined using a single measure, such as the performance of the regressor in the local region of the feature space around the test pattern, called the region of competence. However, no single measure is the best for any task. In this work, we present a framework for DRS, called Meta INtEgration (MINE), that aims at selecting and combining the most competent regressors from a homogeneous ensemble during the evaluation of a given test pattern. The proposed framework uses the combination of different measures extracted from the region of competence, as a criterion for the selection and combination of the regressors. Comprehensive experiments on 20 regression datasets show that MINE improves the final estimate performance when compared to state-of-the-art techniques.

4.1 INTRODUCTION

Ensemble learning refers to techniques that generate different models, with some degree of diversity, which are combined to make a prediction, either in classification or regression problems. The advantage of ensembles concerning single models has been reported in terms of increased robustness and accuracy for both classification (HO, 1998; DOMENICONI; YAN, 2004; SINGH; SINGH, 2005), and regression problems (DRUCKER, 1997; SHRESTHA; SOLOMATINE, 2006; ZHANG; ZHANG; WANG, 2008).

Ensemble-based systems contain three main modules (CRUZ; SABOURIN; CAVALCANTI, 2018): (1) Generation, (2) Selection, and (3) Combination. In the generation module, a training set is used to create the ensemble. The ensemble is said homogeneous when a single learning algorithm is used to train all the models; otherwise, it is called heterogeneous. In the second module, only one model or a subset of the ensemble is selected. Finally, when a subset of the ensemble is selected, the models are combined to estimate the target value of a given test pattern. Over the last two decades, researchers have been

dedicating efforts to improve the quality of the ensemble (SHRESTHA; SOLOMATINE, 2006; RODRÍGUEZ; KUNCHEVA; ALONSO, 2006), and also searching for alternatives to better select and combine the models (GIACINTO; ROLI, 1999; GIACINTO; ROLI, 2001; PERRONE; COOPER, 1993).

Regarding the selection module, it can be either static or dynamic. In the static approach, the selection is performed before the evaluation of the test pattern using the information extracted from the training (ORTIZ-BOYER; HERVáS-MARTíNEZ; GARCíA-PEDRAJAS, 2005) or validation set (PARTALAS et al., 2008). So, the selected models are used to estimate the target value of all test patterns. In the dynamic approach, a different subset of the ensemble is selected for each new test pattern. In dynamic selection techniques, each model is expected to be an expert in a specific local region of the feature space that is known as region of competence. So, for each test pattern, the most competent models are selected in the region of competence where the test pattern is located. Recent works have shown that dynamic selection techniques outperform static selection (KO; SABOURIN; BRITTO, 2008; BRITTO; SABOURIN; OLIVEIRA, 2014; CRUZ; SABOURIN; CAVALCANTI, 2018).

When the selected subset of the ensemble contains more than one model, they should be combined. The combination can be performed using a simple rule such as the mean or the weighted mean. In general, the weighted mean presents better precision than the mean (PERRONE; COOPER, 1993), and its weight can be defined statically or dynamically. The former uses the same weight vector for any test pattern, while in the latter, the weights are defined according to the performance of the models in the region of the feature space where the test pattern is located (MENDES-MOREIRA et al., 2012).

The crucial issue in dynamic selection systems is to define the criterion to measure the competence of the models. It is expected that the better the competence of the dynamically selected models, the higher the precision of the whole system. An usual manner to measure the competence consists in calculating the accumulated error of a given model in the neighborhood of the test pattern (ROONEY et al., 2004; MENDES-MOREIRA et al., 2009). However, the literature on dynamic classifier selection (DCS) shows that using only the accumulated error in the region of competence is not enough to correctly calculate the competence of the classifiers (CRUZ et al., 2015; CRUZ; SABOURIN; CAVALCANTI, 2016; CRUZ; SABOURIN; CAVALCANTI, 2017).

Besides, a preliminary study in the DRS literature compares eight measures of competence and concludes that none of them has superior performance for different tasks (MOURA; CAVALCANTI; OLIVEIRA, 2019). In other words, selecting one different measure per task or combining all the measures may increase in the precision of DRS systems. With that in mind, in this work we hypothesize that the DRS can benefit from the combination of several measures instead of relying on a single one.

To validate such a hypothesis, in this work, we introduce the Meta INtEgration

(MINE) framework for DRS. It uses a combination of measures extracted from the region of competence as a criterion to select and combine the regressors. Since the competence measures used in the context of classification are not suitable for regression, we survey different measures found in the literature of regression, time-series, and forecasting problems.

The contribution of this work is two-fold. Firstly, we proposed a DRS framework that can operate in three different scenarios: (i) the selection of a single regressor given a test pattern (MINE-Selection (MINE-S)); (ii) all the regressors in the ensemble are weighted and combined (MINE-Weighting (MINE-W)); and, (iii) a subset the ensemble is dynamically selected per test pattern (MINE-Weighting with Selection (MINE-WS)). Secondly, we present a robust study that constructs homogeneous ensembles where the base learning algorithm is selected per regression problem.

To evaluate the performance of the MINE framework and show the relevance of the measures adopted with homogeneous ensembles, we carried out a set of extensive experiments on 20 regression problems. We compare the MINE framework against state-of-the-art DRS techniques, and individual regressor trained with the whole training set. Our experimental results show that the adopted measures are useful for the DRS with homogeneous ensembles and validate our hypothesis that better results are achieved when using multiple measures.

This paper is organized as follows: Section 4.2 presents the related works. Section 4.3 describes the proposed framework for DRS. Section 4.4 shows the methodology and experiments used to evaluate the proposed framework. Section 4.5 presents the conclusions about the research.

4.2 RELATED WORKS

This section reviews the literature about selection and combination of regressors. Table 7 presents the related works focusing on three aspects: i) selection strategy that indicates whether the technique is static or dynamic; ii) ensemble type that indicates whether the ensemble is homogeneous or heterogeneous; and iii) selection criterion that indicates what is the measure used as the criterion to define the competence of the regressors from the ensemble. The value "error" in the column "Selection Criterion" indicates that an error measure is used as a criterion to select the regressors.

Table 7 – Related Works

Method	**Static/Dynamic**	**Ensemble Type**	**Selection Criterion**
Perrone et al. (PERRONE; COOPER, 1993)	Static	homogeneous	error
Partalas et al. (PARTALAS et al., 2008)	Static	homogeneous	error
Rooney et al. (ROONEY et al., 2004)	Dynamic	homogeneous	error
Moreira et al. (MENDES-MOREIRA et al., 2009)	Dynamic	heterogeneous	error
Rooney et al. (ROONEY; PATTERSON, 2007)	Dynamic	homogeneous	error

Perrone et al. (PERRONE; COOPER, 1993) defined two ways to combine the models from an ensemble: Basic Ensemble Method (BEM), and Generalized Ensemble Method (GEM). In the BEM, the combination of the models is performed using the mean among the regressors, where all the regressors have the same importance. In the GEM, the models are combined using the weighted mean where the weights are inversely proportional to the errors generated in the training set or the validation set. These weights are constants; it means that the weights do not change during the evaluation of query patterns.

Partalas et al. (PARTALAS et al., 2008) presented an algorithm to select the best subset of regressors from an ensemble. Their algorithm uses greedy search (forward selection, and backward elimination) to select the best subset of the regressors based on the performance in a validation set. The selection is static, and once the subset of the ensemble is defined, it will be the same for all test patterns.

Rooney et al. (ROONEY et al., 2004) proposed three DRS algorithms that use as selection criterion the accumulated error in the region of competence. Two different learning algorithms were used: linear regression and 5-NN (5 nearest neighbors). For each learning algorithm, they generated homogeneous ensembles using Random Subspace (HO, 1998). Later, Moreira et al. (MENDES-MOREIRA et al., 2009) used these three DRS algorithms with the difference that the errors are weighted by the distance between the test pattern and its neighbors. This work is the latest on dynamic selection and combination of regressors and it uses the DRS algorithms with homogeneous ensembles. The three algorithms are described by Moreira et al. (MENDES-MOREIRA et al., 2009; MENDES-MOREIRA et al., 2015) as follows:

- Dynamic Selection (DS) - it selects the regressor with the lowest accumulated error in the region of competence. The errors are weighted by the distance between the neighborhood patterns and the test pattern. Only a single regressor is selected and no combination is needed. The estimation of the test pattern is the value returned by the selected regressor.

- Dynamic Weighting (DW) - it combines all the regressors of the ensemble using the weighted mean. For each test pattern x_j, its region of competence Ψ is calculated; Ψ is composed of K patterns. For each pattern in Ψ, a weight is calculated using Equation 4.1:

$$d_k = \frac{\frac{1}{dist_k}}{\sum_{j=1}^{K}(\frac{1}{dist_j})} \tag{4.1}$$

  where $dist_k$ is the distance between a pattern $t_k \in \Psi$ and the test pattern x_j.

The vector $\{d_1, d_2, ..., d_K\}$ is used to calculate the weight α_i of the regressor $\hat{f}_i$, using Equation 4.2:

$$\alpha_i = \frac{\frac{1}{\sqrt{\sum_{k=1}^{K}(d_k \times sqe_{k,i})}}}{\sum_{n=1}^{N}\left(\frac{1}{\sqrt{\sum_{k=1}^{K}(d_k \times sqe_{k,n})}}\right)} \tag{4.2}$$

where N is the ensemble size, k represents the index of the neighbor, and $sqe_{k,i}$ is the squared error of the regressor i calculated using the pattern $t_k \in \Psi$.

- Dynamic Weighting with Selection (DWS) - it combines a subset of the regressors. The regressors with the accumulated error in the upper half of the error interval $E_i > (E_{max} - E_{min})/2$ are discarded, where E_{max} is the largest accumulated error of any regressor and E_{min} is the lowest accumulated error of any regressor. The measure to calculate the performance of the regressors from the ensemble is the same than the DW algorithm and the remaining regressors are combined using the same strategy of the DW.

Finally, wMetaComb (ROONEY; PATTERSON, 2007) is a technique for regression problems that fuses two combination techniques: Stacking (WOLPERT, 1992) and the DWS algorithm. In the wMetaComb, the estimated value of the test pattern is the weighted mean of the predictions of two combination techniques. The weights to combine the techniques are calculated based on the errors during the training phase.

It is important to notice that the technical literature uses only the error either as a selection criterion or as the measure to calculate the weights for the combination of the regressors. In the static selection, the error is calculated using the training set or the validation set. In the dynamic selection, the error calculated in the region of competence is used as a selection criterion. The proposed framework presents an approach that uses not only the error but the composition of other measures as a criterion for the DRS. In addition, the new framework is not limited to the use of a specific learning algorithm for the generation of homogeneous ensembles, but it chooses a suitable one for each regression problem.

4.3 MINE FRAMEWORK

The Meta INtEgration (MINE) framework architecture (Figure 8) is divided into four phases: Learning Algorithm Selection, Generation, Optimization, and Generalization. In the first phase, the best learning algorithm is selected for the task under analysis and a homogeneous ensemble using this learning algorithm is generated in the second phase. After, some competence measures are extracted and the Optimization phase calculates a

weight for each measure; the more important the measure, the higher the weight. The last phase selects a subset of the ensemble to predict the value of the query pattern. These four phases are detailed in the following sections.

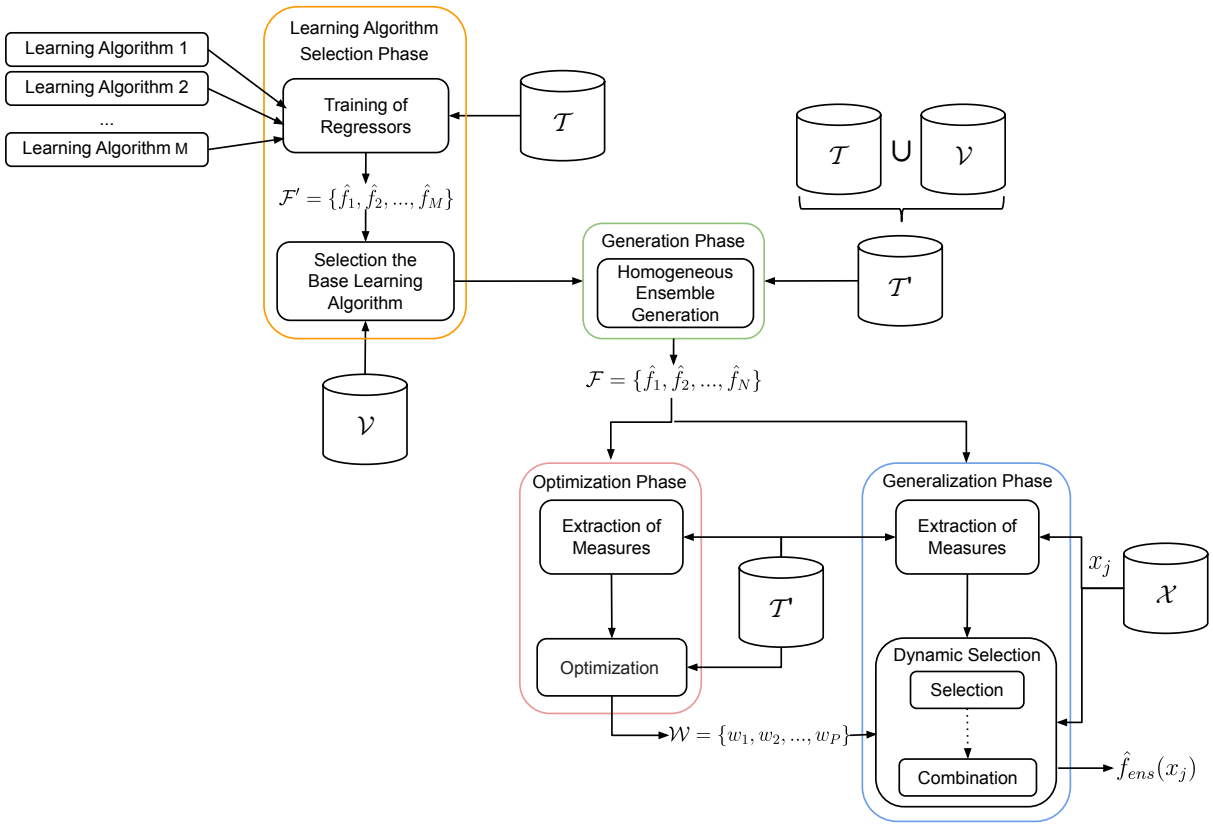


Figure 8 – Architecture of MINE framework. $\mathcal{T}$, $\mathcal{V}$, and $\mathcal{X}$ are the sets of Training, Validation, and Test respectively. $\mathcal{T}'$ is the training set used to train the homogeneous ensemble. $\mathcal{F}' = \{\hat{f}_1, \hat{f}_2, ..., \hat{f}_M\}$ and $\mathcal{F} = \{\hat{f}_1, \hat{f}_2, ..., \hat{f}_N\}$ are the regressors generated in the Learning Algorithm Selection Phase and Generation Phase, respectively. $\mathcal{W} = \{w_1, w_2, ..., w_P\}$ is the vector of weight resulting from the Optimization Phase. x_j is a pattern from test set $\mathcal{X}$ and $\hat{f}_{ens}(x_j)$ is the ensemble estimative for the pattern x_j.

4.3.1 Learning Algorithm Selection

This phase aims at selecting the learning algorithm (among M) given the training set $\mathcal{T}$, and the validation set $\mathcal{V}$. So, M regressors $\mathcal{F}' = \{\hat{f}_1, \hat{f}_2, ..., \hat{f}_M\}$ are trained, each one using a distinct learning algorithm (Training of Regressors module). After, the performance of the M regressors is evaluated using the validation set $\mathcal{V}$, and the base learning algorithm that minimizes the MSE on $\mathcal{V}$ is selected (Selection the Base Learning Algorithm module). This learning algorithm is used to generate the homogeneous ensemble for the task under analysis in the next phase.

4.3.2 Generation

This phase generates a homogeneous ensemble $\mathcal{F}$ containing N regressors. The learning algorithm selected in previous phase is employed to train all the regressors using distinct sets generated with the Bagging (Bootstrap AGGegatING) algorithm (BREIMAN, 1996a).

4.3.3 Optimization

Dynamic regressor selection systems use the error of the predictions in the region of competence as a criterion to dynamically select the best regressors. Moura et al. (MOURA; CAVALCANTI; OLIVEIRA, 2019) evaluated eight different measures and showed that none of them is the ideal choice when used isolated and also that the best measure depends on the task. As stated before, we advocate that the combination of different measures is a better alternative than using only one. So, this phase aims at generating a vector of weights $\mathcal{W} = \{w_1, w_2, ..., w_P\}$ that gives different importance for each measure $m_i, i = \{1, 2, \ldots, P\}$, and it is composed of two modules: Extraction of Measures, and Optimization. In the next section, eight measures are defined, so, $P = 8$.

4.3.3.1 Extraction of Measures

A total of eight measures $\{m_1, m_2, \ldots, m_8\}$ is extracted from the region of competence and they correspond to different criteria to analyze the behavior of each regressor. Measure m_1 captures the diversity among the regressors $\hat{f}_n \in \mathcal{F}$ using the variance of their estimations. On the other hand, m_2, m_3, and m_7 capture different points of view of the prediction error. The dispersion and centrality of the error in the region of competence are calculated using m_4 and m_5 respectively. The similarity between the estimation of a pattern and the observed values of its nearest neighbors is calculated using measure m_6. And finally, m_8 measures the error of the nearest neighbor.

In the next equations, $f(t_k)$ refers to the observed value of the neighborhood pattern t_k, $\hat{f}_n(t_k)$ is the estimated value of the pattern t_k given by the regressor $\hat{f}_n$, and d_k is the inverse of the normalized distance in the interval $[0; 1]$. So, the smaller the distance the greater the value of d_k, according to

$$d_k = \frac{\frac{1}{dist_k}}{\sum_{j=1}^{K}(\frac{1}{dist_j})} \tag{4.3}$$

where $\{dist_1, dist_2, ..., dist_K\}$ is a vector where each element is a distance measure between the neighbor pattern from the training set $\mathcal{T}'$ and the training pattern t_i, and K is the neighborhood size. The measures are extracted from the region of competence $\Psi = \{t_1, t_2, ..., t_K\}$ for each pattern t_i, where t_k is a pattern from the same training set $\mathcal{T}'$, $\forall k \in \{1, 2, ..., K\}$. The eight measures calculated for each regressor $\hat{f}_n$ are described below.

- m_1 - *Variance*: the variance of the neighbors estimated values. The variance is calculated for each regressor using the estimated values for the patterns in the region of competence, according to Equation 4.4:

$$m_{1,n} = Var(\hat{f}_n(t_1), \hat{f}_n(t_2), ..., \hat{f}_n(t_K)) \tag{4.4}$$

  This measure is inspired in the work of Tresp et al. (TRESP; TANIGUCHI, 1995), whose variance of the estimated values is used as weight in the static combination of artificial neural networks.

- m_2 - *Sum Absolute Error*: the sum of the absolute errors is calculated in the region of competence, weighted by d_k, according to Equation 4.5:

$$m_{2,n} = \sum_{k=1}^{K} \left| f(t_k) - \hat{f}_n(t_k) \right| \times d_k \tag{4.5}$$

- m_3 - *Sum Squared Error*: the sum of the squared errors is calculated using the inverse of the distances d_k as weights, according to Equation 4.6:

$$m_{3,n} = \sum_{k=1}^{K} (f(t_k) - \hat{f}_n(t_k))^2 \times d_k \tag{4.6}$$

- m_4 - *Minimum Squared Error*: the minimum value of squared errors is calculated using the inverse of the distances d_k as weights. The measure m_4 is computed using Equation 4.7:

$$m_{4,n} = \min_{1 \leq k \leq K} \{(f(t_k) - \hat{f}_n(t_k))^2 \times d_k\} \tag{4.7}$$

- m_5 - *Maximum Squared Error*: the maximum value of squared errors is calculated using the inverse of the distances d_k as weights. The measure m_5 is computed using Equation 4.8:

$$m_{5,n} = \max_{1 \leq k \leq K} \{(f(t_k) - \hat{f}_n(t_k))^2 \times d_k\} \tag{4.8}$$

  Considering that m_4 and m_5 define an interval, these measures present mean and variance, it means that, the interval contains information about implicit measures of dispersion (error variance) and centrality (error mean) of the squared error in the region of competence.

- m_6 - *Neighbor's Similarity*: the sum of the differences between the estimated value of the validation pattern from validation set $\mathcal{T}'$ and the observed values of each neighborhood pattern, weighted by the inverse of the distance. The measure m_6 is computed using Equation 4.9:

$$m_{6,n} = \sum_{k=1}^{K} (f(t_k) - \hat{f}_n(t_i))^2 \times d_k \tag{4.9}$$

where $\hat{f}_n(t_i)$ is the estimated value of the regressor $\hat{f}_n$ for t_i. t_i is the pattern being tested in the leave-one-out process.

The goal of the measure m_6 is to find the degree of similarity between the estimation of the pattern $t_i \in \mathcal{T}'$ and the observed values of the nearest neighbors $\{t_1, t_2, ..., t_K\}$. This is the only measure that uses the estimated value for the test pattern ($\hat{f}_n(t_i)$). So far as we know, this measure is unprecedented and is defined by the authors of this work.

- m_7 - *Root Sum Squared Error*: the root of sum squared errors in the region of competence, with the errors weighted by d_k. The measure m_7 is computed using Equation 4.10:

$$m_{7,n} = \sqrt{\sum_{k=1}^{K} (f(t_k) - \hat{f}_n(t_k))^2 \times d_k} \tag{4.10}$$

  Root squared error is more stable and less sensitivity to the difference between the maximum and the minimum errors, while squared error is very sensitive to extreme error values. The measures m_3 and m_7 present different points of view from the error calculated in the region of competence. These two measures have a high correlation, but using them together allows a better balance in the weights of the combination (ADHIKARI, 2015). Also, m_3 and m_7 produce the same result when a single regressor is chosen to estimate a test pattern, but different results in the combination of the regressors (MOURA; CAVALCANTI; OLIVEIRA, 2019).

- m_8 - *Closest Squared Error*: the error obtained by the regressor only on the nearest neighbor. The measure m_8 is computed using Equation 4.11:

$$m_{8,n} = (f(t_1) - \hat{f}_n(t_1))^2 \tag{4.11}$$

For each pair (pattern $t_i \in \mathcal{T}'$, regressor $\hat{f}_n$), the eight measures are extracted from the region of competence of the pattern t_i and produces a vector $M_{i,n} = \{m_{1,n}, m_{2,n}, ..., m_{8,n}\}$ where each element is the value of one measure.

4.3.3.2 Optimization

This module uses a Genetic Algorithm (GA) (EIBEN; SMITH, 2003) to obtain one weight per measure using the vectors $M_{i,n}$ described in the last section. Algorithm 1 shows the pseudo-code of the optimization process whose output is the vector $\mathcal{W} = \{w_1, w_2, ..., w_P\}$ that minimizes the Mean Squared Error (MSE) of the training set $\mathcal{T}'$. The mutation, crossover, and elitism parameters are discussed in Section 4.4.2.

Algorithm 1 Optimization Process

Input: Ensemble $\mathcal{F}$; Training set $\mathcal{T}'$; Neighborhood size K; Population size L
Output: $\mathcal{W}_{best}$: Best Individual

1: $Pop = InitialPopulation(L)$;
2: **repeat**
3: $MSE_{Pop} = \emptyset$; {set with the MSE of all individuals}
4: **for** each individual $\{w_1, w_2, ..., w_P\}$ in Pop **do**
5: $SE = 0$
6: **for** each pattern t_i in $\mathcal{T}'$ **do**
7: $\mathcal{T}' = \mathcal{T}'$ - t_i {Leave-one-out}
8: Find the region of competence Ψ of t_i using $\mathcal{T}'$
9: **for** each $\hat{f}_n$ in $\mathcal{F}$ **do**
10: Calculate the measures $\{m_{1,n}, m_{2,n}, ..., m_{P,n}\}$ using Ψ
11: $\alpha_n = \sum_{p=1}^{P} w_p \times m_{p,n}$
12: **end for**
13: $\hat{f}_{ens}(t_i) = DynamicSelection(\mathcal{F}, \mathcal{A}, t_i)$
14: $SE = SE + (f(t_i) - \hat{f}_{ens}(t_i))^2$
15: **end for**
16: $MSE = SE/|\mathcal{T}'|$
17: $MSE_{Pop} = MSE_{Pop} \cup MSE$
18: **end for**
19: $BestInds = SaveBestIndsElitism(MSE_{Pop}, Pop)$
20: $Pop = GenerateOffspring() \cup BestInds$
21: **until** $MSE = 0$ or reach maximum iteration
22: $\mathcal{W}_{best} = BestInd(MSE_{Pop}, Pop)$
23: **return** $\mathcal{W}_{best}$

In line 1, the initial population is generated. The population is composed of L individuals, and each is a vector of weights $\mathcal{W}$ whose size is given by the number of measures. In this way, each individual is represented by a set of P genes and each gene is a real value $w_p \in IR$, $\forall p \in \{1, 2, ..., P\}$.

For each pattern $t_i \in \mathcal{T}'$, the region of competence Ψ is defined (line 8) and the measures are extracted for each regressor $\hat{f}_n \in \mathcal{F}$ (lines 9 and 10). Line 11 shows the weighted combination of the measures to compute a new vector $\mathcal{A} = \{\alpha_1, \alpha_2, ..., \alpha_N\}$, where N is the number of regressors from the ensemble.

Dynamic Selection uses the vector $\mathcal{A}$ to estimate the value $\hat{f}_{ens}(t_i)$ for the pattern t_i (line 13) and the squared error is computed in line 14. The estimated value $\hat{f}_{ens}(t_i)$ can be the result of one the following DRS techniques: (i) MINE-S - dynamic selection of a single regressor from the ensemble; (ii) MINE-W - combination of all the regressors from the ensemble; or (iii) MINE-WS - dynamic selection and combination of a subset of regressors from the ensemble. These DRS techniques are explained in Section 4.3.4. The framework works for each DRS technique separately. In other words, the optimization process is technique-dependent.

The MSE is computed in line 16, and this is the fitness function (Eq. 4.12) of the

optimization procedure.

$$fit(ind) = \frac{1}{|\mathcal{T}'|} \sum_{i=1}^{|\mathcal{T}'|} (f(t_i) - \hat{f}_{ens}(t_i))^2 \tag{4.12}$$

where ind is an individual that belongs to Pop, t_i is a pattern from the training set $\mathcal{T}'$, $f(t_i)$ is the observed value of the pattern t_i, and $|\mathcal{T}'|$ is the number of instances in the training set $\mathcal{T}'$.

The MSE of all individuals are stored into the set MSE_{Pop}, and after finishing all the individuals, the best ones (lower MSE), are selected (line 19) to compose the new offspring (line 20). At the end of the algorithm, the best individual (lowest MSE) is selected and stored into $\mathcal{W}_{best}$ (line 22).

4.3.4 Generalization

In this phase, the estimated value $\hat{f}_{ens}(x_j)$ is calculated for each test pattern x_j. This phase consists of two modules: Extraction of Measures and Dynamic Selection. The Extraction of Measures module receives as input the ensemble $\mathcal{F}$, the test set $\mathcal{X}$ and the training set $\mathcal{T}'$. This module works similarly as described in Section 4.3.3.1, where for each test pattern $x_j \in \mathcal{X}$, the region of competence is defined using $\mathcal{T}'$ and the measures are extracted for each regressor $\hat{f}_n \in \mathcal{F}$. The Dynamic Selection module receives as input the measures extracted in the previous module, the ensemble $\mathcal{F}$, the test set $\mathcal{X}$, and the weights $\mathcal{W}$ calculated in the Optimization Phase. This module is responsible for calculating the competence of the regressors using as criterion the combination of the measures. After the combination of the measures, $\hat{f}_{ens}(x_j)$ is computed as the final estimation for test pattern x_j.

The Dynamic Selection module contains two submodules: Selection and Combination. The first one is responsible to select one or more regressors from the ensemble per test pattern. If more than one regressor is selected, the Combination submodule is performed. The Combination submodule can also combine all the regressors directly, without executing the Selection submodule.

In this work, three ways of using the MINE framework are proposed: (i) MINE-S - dynamic selection of a single regressor from the ensemble; (ii) MINE-W - combination of all the regressors from the ensemble; and (iii) MINE-WS - dynamic selection and combination of a subset of regressors from the ensemble.

4.3.4.1 Dynamic Selection

This module aims at selecting the best regressor(s) per test pattern x_j from the ensemble of regressors $\mathcal{F}$ given the vector of weights $\mathcal{W}$ calculated in the Optimization Phase. If more than one regressor are selected, they should be combined to produce the estimated value of the test pattern.

The selection process is based on the competence of each regressor $\hat{f}_n$ in the estimation of the value of x_j. The competence of $\hat{f}_n$ is calculated using α_n (Eq.4.13) that multiplies each measure $m_{p,n}$ by its respective weight $w_p \in \mathcal{W}$. It is important to remember that each vector of measures $M_{j,n} = \{m_{1,n}, m_{2,n}, ..., m_{P,n}\}$ is calculated using the regressor $\hat{f}_n$ and the region of competence of the test pattern x_j; so, this vector is regressor-dependent.

$$\alpha_n = \sum_{p=1}^{P} w_p \times m_{p,n} \tag{4.13}$$

where $m_{p,n}$ is the measure p calculated for the regressor $\hat{f}_n$ in the region of competence, w_p is the weight of the measure p in the vector $\mathcal{W}$, and α_n is the result of the measures combination for each regressor $\hat{f}_n$ from the ensemble $\mathcal{F}$.

After the evaluation of the competence of each regressor, we have a vector $\mathcal{A}$, which is used to select and combine the regressors. Depending on how the regressors are selected using $\mathcal{A}$, we propose three techniques of DRS using the MINE framework: MINE-S, MINE-W, and MINE-WS that are described below. These variations are similar to the ones in (MENDES-MOREIRA et al., 2009), but they use a different measure to calculate the weights to combine the regressors.

Using the MINE framework, one of the three proposed techniques can be used during the execution of the Optimization and Generalization phases. In addition to the proposed techniques, MINE framework can be modified to meet another strategy not foreseen in this work.

MINE-S

Some measures ($m_{p,n}$) capture different points of view of the error calculated in the region of competence per regressor $\hat{f}_n$. So, it is correct to say that the lower the weighted sum of these measures given by α_n, the more competent is the regressor $\hat{f}_n$. Thus, MINE-Selection selects the regressor that obtains the lowest value of $\alpha_n \in \mathcal{A}$, for each test pattern $x_j \in \mathcal{X}$. The regressor index is selected using Equation 4.14:

$$index = \underset{1 \leq n \leq N}{\operatorname{argmin}}(\{\alpha_1, \alpha_2, ..., \alpha_N\}) \tag{4.14}$$

and the estimated value for the test pattern is calculated using Equation 4.15:

$$\hat{f}_{ens}(x_j) = \hat{f}_{index}(x_j) \tag{4.15}$$

where $\hat{f}_{index}(x_j)$ is the estimated value for the test pattern x_j. Algorithm 2 presents the pseudo-code of the MINE-S.

MINE-W

MINE-Weighting combines all the regressors from the ensemble $\mathcal{F}$ using the vector $\mathcal{A} = \{\alpha_1, \alpha_2, ..., \alpha_N\}$. For each test pattern, the estimated value is the weighted mean of

the regressors estimates. The values $\alpha_n \in \mathcal{A}$ are normalized using Equation 4.16:

$$\tilde{\alpha}_n = \frac{\frac{1}{\alpha_n}}{\sum_{n=1}^{N} \frac{1}{\alpha_n}}. \tag{4.16}$$

So, the estimated value for the test pattern x_j is computed using Equation 4.17:

$$\hat{f}_{ens}(x_j) = \sum_{n=1}^{N} \tilde{\alpha}_n \times \hat{f}_n(x_j). \tag{4.17}$$

Algorithm 3 presents the pseudo-code of the MINE-W.

Algorithm 2 Selecting using MINE-S

Input: Ensemble $\mathcal{F}$; Training set $\mathcal{T}'$; Test set $\mathcal{X}$; Vector of Weights $\mathcal{W}$; Neighborhood size K

Output: MSE: Mean Squared Error

$SE = 0$
for each test pattern x_j in $\mathcal{X}$ **do**
  Find the region of competence Ψ of x_j using $\mathcal{T}'$
  for each $\hat{f}_n$ in $\mathcal{F}$ **do**
    Calculate the measures $\{m_{1,n}, m_{2,n}, ..., m_{P,n}\}$ using Ψ
    $\alpha_n = \sum_{p=1}^{P} w_p \times m_{p,n}$
  end for
  $index = \underset{1 \leq n \leq N}{\operatorname{argmin}}(\{\alpha_1, \alpha_2, ..., \alpha_N\})$
  $\hat{f}_{ens}(x_j) = \hat{f}_{index}(x_j)$
  $SE = SE + (f(x_j) - \hat{f}_{ens}(x_j))^2$
end for
$MSE = SE/|\mathcal{X}|$
return MSE

Algorithm 3 Combining all the regressors using MINE-W

Input: Ensemble $\mathcal{F}$; Training set $\mathcal{T}'$; Test set $\mathcal{X}$; Vector of Weights $\mathcal{W}$; Neighborhood size K

Output: MSE: Mean Squared Error

$SE = 0$
$\mathcal{A} = \emptyset$
for each test pattern x_j in $\mathcal{X}$ **do**
  Find the region of competence Ψ of x_j using $\mathcal{T}'$
  for each $\hat{f}_n$ in $\mathcal{F}$ **do**
    Calculate the measures $\{m_{1,n}, m_{2,n}, ..., m_{P,n}\}$ using Ψ
    $\alpha_n = \sum_{p=1}^{P} w_p \times m_{p,n}$
    $\mathcal{A} = \mathcal{A} \cup \alpha_n$
  end for
  for each α_n in $\mathcal{A}$ **do**
    $\tilde{\alpha}_n = (1/\alpha_n)/(\sum_{n=1}^{N}(1/\alpha_n))$
  end for
  $\hat{f}_{ens}(x_j) = \sum_{n=1}^{N} \tilde{\alpha}_n \times \hat{f}_n(x_j)$
  $SE = SE + (f(x_j) - \hat{f}_{ens}(x_j))^2$
end for
$MSE = SE/|\mathcal{X}|$
return MSE

MINE-WS

In MINE-Weighting with Selection, for each test pattern, the regressors with $\alpha_n > (\alpha_{max} - \alpha_{min})/2$ are removed from the ensemble, i.e., the values of $\mathcal{A}$ in the upper half of the difference between the largest and lowest values are discarded. For the remaining regressors, they are combined using Equations 4.16 and 4.17. Algorithm 4 presents the pseudo-code of the MINE-WS.

Algorithm 4 Selecting and Combining the regressors using MINE-WS

Input: Ensemble $\mathcal{F}$; Training set $\mathcal{T}'$; Test set $\mathcal{X}$; Vector of Weights $\mathcal{W}$; Neighborhood size K

Output: MSE: Mean Squared Error

$SE = 0$
$\mathcal{A} = \emptyset$
for each test pattern x_j in $\mathcal{X}$ **do**
  Find the region of competence Ψ of x_j using $\mathcal{T}'$
  for each $\hat{f}_n$ in $\mathcal{F}$ **do**
    Calculate the measures $\{m_{1,n}, m_{2,n}, ..., m_{P,n}\}$ using Ψ
    $\alpha_n = \sum_{p=1}^{P} w_p \times m_{p,n}$
    $\mathcal{A} = \mathcal{A} \cup \alpha_n$
  end for
  $\tilde{\mathcal{F}} = \mathcal{F}$
  $\tilde{\mathcal{A}} = \mathcal{A}$
  for each $\hat{f}_n$ in $\mathcal{F}$ **do**
    if $\alpha_n > (\alpha_{max} - \alpha_{min})/2$ **then** {Selecting}
      $\tilde{\mathcal{F}} = \tilde{\mathcal{F}} - \hat{f}_n$
      $\tilde{\mathcal{A}} = \tilde{\mathcal{A}} - \alpha_n$
    end if
  end for
  $N = size(\tilde{\mathcal{F}})$
  for each α_n in $\tilde{\mathcal{A}}$ **do** {Combining}
    $\tilde{\alpha}_n = (1/\alpha_n)/(\sum_{n=1}^{N}(1/\alpha_n))$
  end for
  $\hat{f}_{ens}(x_j) = \sum_{n=1}^{N} \tilde{\alpha}_n \times \hat{f}_n(x_j)$ $\{\hat{f}_n \in \tilde{\mathcal{F}}\}$
  $SE = SE + (f(x_j) - \hat{f}_{ens}(x_j))^2$
end for
$MSE = SE/|\mathcal{X}|$
return MSE

4.4 EXPERIMENTS

The experiments were performed using a total of 20 regression datasets. Table 8 shows the main features of the datasets including the sources that are: the personal page of Prof. Luís Torgo[1], UCI Repository[2], and Delve Repository[3]. To facilitate the implementation of the framework, we used datasets with only numeric (integer or real) attributes, except for the Abalone dataset, in which the categorical attribute sex was converted to binary using two bits.

In Section 4.4.1 the entire experimental protocol is described. Section 4.4.2 presents the parameters of the genetic algorithm used in the optimization module (Section 4.3.3). In Section 4.4.3, the experiments present the results of the Learning Algorithm Selection

[1] http://www.dcc.fc.up.pt/∼ltorgo/Regression/DataSets.html
[2] http://http://archive.ics.uci.edu/ml/
[3] http://www.cs.toronto.edu/∼delve/

Table 8 – Datasets characteristics.

Dataset	Instances	Features	Source
Abalone	4177	8	UCI
Airfoil Self Noise	1503	5	UCI
Bank32NH	8192	32	Delve
Bank8FM	8192	8	Delve
Breast Cancer	194	32	Torgo
CCPP (TüFEKCI, 2014)	9568	4	UCI
Comp Act	8192	22	Delve
Comp Act Small	8192	8	Delve
Concrete (YEH, 1998)	1030	9	UCI
Delta Ailerons	7129	6	Torgo
Delta Elevators	9517	6	Torgo
Housing	506	13	UCI
Kinematics	8192	8	Delve
Machine	209	6	Torgo
Puma32H	8192	32	Delve
Puma8NH	8192	8	Delve
Stock	950	9	Torgo
Triazines (HIRST; KING; STERNBERG, 1995),(1994)	186	60	Torgo
Wine Q. Red (CORTEZ et al., 2009)	1599	12	UCI
Wine Q. White (CORTEZ et al., 2009)	4898	12	UCI

Phase, where the regressors are tested using a validation set. In this phase, for each dataset, the best learning algorithm is chosen. Also, the experiments present the results of MINE-S compared against the DS algorithm (Section 4.4.4). In Section 4.4.5, the results of MINE-W and MINE-WS are compared against DW and DWS algorithms respectively. In Section 4.4.6, the results of MINE techniques are compared against Individual Regressor, Mean, and Median. Section 4.4.7 analyzes the importance of each measure extracted from the region of competence per dataset.

4.4.1 Experimental Protocol

For each dataset, all data attributes were normalized into the interval [0,1]. The experiments were conducted using 20 replications, and for each replication, the configurations used are described in the next subsections.

4.4.1.1 Ensemble Generation

In the first phase (Learning Algorithm Selection Phase), a set of regressors with size $M = 10$ is generated. All the regressores are generated using the whole training set $\mathcal{T}$. Ten learning algorithms were used in this phase: CART (BREIMAN et al., 1984), LINEAR, feedforward neural network with one hidden layer (FANN-1) with 10 neurons, a feedforward neural network with two hidden layers (FANN-2), with 5 and 10 neurons in each of

the layers, Support Vector Regression (SVR) with RBF kernel, SVR with Linear kernel, SVR with polynomial order 3 kernel, RBF network with 10 neurons, 3-nearest neighbor (3-NN) and 5-nearest neighbor (5-NN). The learning algorithms were used with default settings found in MATLAB[4] without any specific adjustment.

In the second phase (Generation Phase), homogeneous ensembles with different sizes $N = \{5, 10, 15, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$ are generated using Bagging, a sampling with replacement, as in Bootstrap AGGregatING (BREIMAN, 1996a). Bagging generates distinct datasets, using sampling with replacement. The outputs of the Bagging are N training sets $\{\mathcal{T}_1', \mathcal{T}_2', ..., \mathcal{T}_N'\}$, and each is used to train one regressor $\hat{f}_i \in \mathcal{F}$. All sets $\mathcal{T}_i'$ have the same size as the original training set $\mathcal{T}'$.

4.4.1.2 Framework Validation

For each replication in the Learning Algorithm Selection Phase, a 10-fold cross-validation is carried out using 70% of the folds for the training set ($\mathcal{T}$) and 20% for the validation set ($\mathcal{V}$). From the Generation Phase onwards, a 10-fold cross-validation is carried out, and each replication uses 90% of the folds for training ($\mathcal{T}'$) and 10% for testing set ($\mathcal{X}$). The result of each replication is the arithmetic mean of the MSE calculated for the 10 testing sets used in the cross-validation.

4.4.1.3 Region of Competence

In (MENDES-MOREIRA et al., 2009), experiments were performed varying the size of the region of competence K in the set $\{2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 25, 30\}$. They concluded that the appropriate size for the neighborhood is problem-dependent, so they fixed the size of the region of competence with $K = 10$. Analyzing works of classification (CRUZ et al., 2015; CRUZ; SABOURIN; CAVALCANTI, 2016), time-series forecasting (SERGIO; LIMA; LUDERMIR, 2016), and regression (ROONEY et al., 2004), it can be verified that the size of the region of competence is fixed for better validation and comparison of the results. The main objective is to compare and validate state-of-the-art techniques against the proposed techniques regardless the size of the region of competence. Thus, according to (MENDES-MOREIRA et al., 2009), we fixed the size of the region of competence to $K = 10$ for all the experiments using DRS techniques.

4.4.1.4 State-of-the-art techniques

The algorithms DS, DW, and DWS use only one error measure as a criterion to select the most competent regressor (MENDES-MOREIRA et al., 2009). For these techniques, we used the same experimental protocol of the MINE *framework*: the same data sets, learning

[4] https://www.mathworks.com/products/matlab.html

algorithms to generate the ensemble and the size of the region of competence was fixed to K = 10.

For each regressor used in comparison, a 10-fold cross-validation is carried out using 90% of the folds for the training set ($\mathcal{T}'$) and 10% for testing set ($\mathcal{X}$). For the state-of-the-art techniques, the result for each replication is the arithmetic mean of the MSE calculated for the 10 testing sets used in the cross-validation.

4.4.1.5 Hypothesis Tests

Non-parametric hypothesis tests were performed for pairwise comparison between the results obtained using the proposed techniques against the results obtained using state-of-the-art DRS techniques, and against the results obtained using classical combination techniques. Wilcoxon signed rank tests were used to compare two paired samples from the same population, each pair being independent, randomly selected, as suggested in (DEMŠAR, 2006). The null hypothesis H_0 indicates whether the two methods have the same performance and the alternative hypothesis H_1 verifies whether the proposed techniques performs better (lowest error). A significance level of 5% was adopted for left-tailed. Values marked with • indicate that the null hypothesis must be rejected and there is evidence that the alternative hypothesis is correct ($pValue \leq 0.05$). In other words, the proposed technique achieves superior performance compared to the other techniques.

4.4.2 Genetic Algorithm Configurations

This section presents the parameters of the genetic algorithm used in the optimization module and all of them were defined empirically. For all replications, the genetic algorithm was configured as follows:

- Population Size: 80.
- Fitness Limit: 0.
- Crossover fraction: 0.8.
- Mutation Function: Gaussian with 0 mean and standard deviation 1.0.
- Maximum Generations: 100 × 8 genes = 800.
- Stall Generations Limit: 40.
- Elitism: Best 8 individuals move on to the next generation.
- Initial Population: 71 individuals randomly generated with the values of the genes in the interval [0,1], and nine individuals initialized according to Matrix 4.18. The first line of the matrix shows the first chromosome initialized with 1 for all genes. The other chromosomes of the matrix have 1 in only one gene.

$$firstPop = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.18}$$

The following sections present the results of the experiments.

4.4.3 Learning Algorithm Selection Phase results

This section presents the Learning Algorithm Selection Phase results. For each dataset, a set of regressors was generated with the size $M = 10$. The used learning algorithms are described in the previous section. Table 9 shows the performance of the regressors in the validation set $\mathcal{V}$. The results were calculated using 20 replications. The best results are in bold.

Table 9 – Mean and standard deviation of the results calculated in 20 replications, obtained for each regressor used to compare. For each dataset, the best result is in bold. Error values are in the scale 10^{-4}.

Dataset	CART	LINEAR	FANN-1	FANN-2	SVR (RBF)	SVR (Linear)	SVR (POLY 3)	RBF	3-NN	5-NN
Abalone	97.8 (1.3)	62.9 (0.3)	58.5 (1.5)	**57.2 (0.6)**	64.3 (0.1)	64.9 (0.2)	60.0 (0.5)	62.0 (0.5)	72.6 (0.9)	66.6 (0.6)
Airfoil Self Noise	70.1 (2.5)	164.6 (0.5)	59.2 (7.6)	**37.3 (5.5)**	112.0 (0.9)	169.4 (0.5)	105.3 (1.4)	123.4 (3.0)	70.2 (2.8)	98.9 (2.6)
Bank32NH	202.1 (3.0)	103.9 (0.1)	**100.8 (1.1)**	102.0 (1.3)	135.7 (0.4)	115.3 (0.1)	272.8 (3.2)	171.0 (1.3)	233.8 (1.4)	215.6 (1.2)
Bank8FM	27.4 (0.4)	23.4 (0.0)	13.3 (0.1)	**13.0 (0.1)**	17.6 (0.1)	23.8 (0.0)	16.6 (0.2)	34.8 (1.3)	159.0 (1.6)	159.5 (1.2)
Breast Cancer	1202.0 (75.0)	808.2 (44.2)	1023.6 (79.8)	953.2 (76.1)	761.1 (20.0)	754.1 (33.7)	3127.3 (444.4)	**751.5 (22.7)**	872.6 (34.9)	765.4 (30.4)
CCPP	32.7 (0.6)	36.5 (0.0)	29.5 (0.2)	29.3 (0.3)	30.6 (0.0)	36.7 (0.0)	30.8 (0.0)	31.6 (0.1)	26.6 (0.3)	**25.8 (0.2)**
Comp Act	12.3 (0.1)	95.9 (1.0)	**6.4 (0.4)**	6.5 (0.4)	11.0 (0.1)	156.4 (3.3)	7.2 (0.4)	42.6 (2.8)	9.0 (0.5)	9.4 (0.5)
Comp Act Small	16.2 (0.3)	99.9 (0.7)	**9.6 (0.2)**	10.0 (0.4)	12.7 (0.1)	153.3 (3.5)	10.1 (0.5)	25.7 (1.0)	11.1 (0.2)	10.5 (0.2)
Concrete	77.5 (4.5)	17.08 (0.12)	**5.91 (0.29)**	6.43 (0.29)	9.62 (0.10)	18.23 (0.26)	7.27 (0.20)	10.05 (0.37)	13.68 (0.38)	13.81 (0.32)
Delta Ailerons	22.8 (0.4)	16.0 (0.0)	15.3 (1.0)	15.3 (0.2)	15.8 (0.0)	16.4 (0.0)	**15.1 (0.0)**	16.2 (0.1)	18.4 (0.2)	16.8 (0.1)
Delta Elevators	45.6 (0.5)	28.8 (0.0)	**28.0 (0.1)**	28.0 (0.2)	28.2 (0.2)	28.9 (0.0)	28.3 (0.0)	29.3 (0.2)	36.4 (0.2)	32.9 (0.2)
Housing	106.8 (11.0)	119.0 (3.2)	87.7 (10.4)	92.8 (12.7)	95.1 (2.3)	127.8 (2.0)	**65.1 (8.4)**	104.6 (6.1)	114.4 (9.3)	129.2 (7.1)
Kinematics	216.1 (3.2)	203.0 (0.1)	44.9 (1.0)	**40.4 (1.2)**	48.9 (0.2)	207.8 (0.2)	103.4 (0.6)	161.2 (2.2)	83.9 (0.8)	74.0 (0.6)
Machine	49.3 (16.0)	40.7 (6.6)	73.8 (30.0)	75.9 (34.3)	76.8 (3.6)	47.1 (2.1)	54.3 (19.0)	**30.8 (10.1)**	53.0 (9.1)	61.5 (6.5)
Puma32H	36.1 (0.4)	230.9 (0.3)	**13.2 (0.7)**	14.8 (4.2)	212.2 (0.6)	235.3 (0.3)	190.4 (2.7)	275.8 (0.7)	285.3 (2.0)	257.6 (1.5)
Puma8NH	302.1 (3.2)	337.8 (0.2)	**170.7 (0.6)**	171.3 (0.6)	181.7 (0.6)	351.2 (0.5)	181.0 (0.5)	318.4 (1.5)	283.3 (2.5)	255.2 (2.2)
Stock	17.3 (1.3)	70.4 (0.4)	11.5 (0.8)	12.4 (0.9)	13.5 (0.2)	73.2 (0.9)	12.4 (0.2)	35.6 (3.0)	**6.6 (0.3)**	7.8 (0.3)
Triazines	219.1 (20.4)	298.2 (29.8)	346.8 (44.9)	316.4 (34.1)	**215.8 (7.5)**	243.7 (14.7)	440.3 (56.8)	250.4 (6.3)	225.2 (12.7)	228.1 (12.4)
Wine Q. Red	240.0 (8.6)	170.4 (0.7)	174.9 (3.0)	175.6 (5.0)	**166.1 (1.0)**	173.0 (0.8)	173.0 (3.0)	168.3 (1.2)	197.5 (3.9)	189.1 (2.5)
Wine Q. White	190.3 (3.4)	158.2 (0.4)	**144.6 (2.1)**	146.5 (2.4)	146.8 (0.3)	159.1 (0.2)	147.3 (2.6)	155.8 (00.7)	148.3 (2.1)	144.7 (1.4)

According to the literature, we observe in Table 9 that no learning algorithm is better than the others for all situations. The best learning algorithm is problem dependent. For each dataset in the next experiments, the best-performing learning algorithm is used to generate the homogeneous regressor ensemble.

The worst-performing regressors were those trained with the following learning algorithms: CART, LINEAR, and SVR with Linear kernel. These regressors did not perform as the best one in any dataset, so these algorithms are not selected for any dataset in the next phases.

4.4.4 MINE-S results

This section presents the results of the experiments performed using the MINE-S technique that selects the most suitable regressor per test pattern. Table 10 compares MINE-S with DS for different ensemble sizes $N = \{5, 10, 15, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$. The results are the MSE arithmetic mean and the standard deviation of the 20 replications for each dataset.

According to Table 10, MINE-S has better results for any ensemble size when compared to the DS technique. For MINE-S and DS, increasing the size of the ensemble does not guarantee better results. In some datasets, the error increases when the size of the ensemble increases. A possible explanation to this fact is that selecting a suitable regressor among too many is a difficult task.

4.4.5 MINE-W and MINE-WS results

This section presents the results of the experiments performed using MINE-W and MINE-WS techniques, Tables 11 and 12, respectively. The results show the arithmetic mean and the standard deviation of the MSE computed for the 20 replications using different sizes of the ensemble $N = \{5, 10, 15, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$.

According to Table 11, MINE-W performs better than DW for any ensemble size, and an increase in the ensemble size leaded to a decrease in the error rates MINE-W obtained superior performance on average in 13 out of 20 datasets, and reached smaller error rates when compared with MINE-S.

According to Table 12, MINE-WS performs better than DWS for any ensemble size. Similarly to MINE-W, in MINE-WS, increasing the ensemble size leaded to a decrease of the error rates. MINE-WS obtained superior performance on average in 11 out of 20 datasets. Also, MINE-WS reached smaller error rates when compared with MINE-S, but worse results when compared to MINE-W. The reduction in the variance achieved by weighted average of all regressors can explain why using all of them is better than the selection of a subset of the regressors or just one of them.

Table 10 – Mean and standard deviation of the results calculated in 20 replications. For each dataset, the best result is in bold. Line "Win/Tie/Loss" shows the total of the results compared with MINE-S. The values marked with a • indicate that the null hypothesis must be rejected ($pValue \leq 0.05$), in other words, the result of MINE-S achieves superior performance. The values are in the scale 10^{-4}.

	5		10		15		20		30		40	
Dataset	DS	MINE-S	DS	MINE-S	DS	MINE-S	DS	MINE-S	DS	MINE-S	DS	MINE-S
Abalone	58.83(1.26)•	**57.16(0.84)**	60.20(1.50)•	**58.06(1.03)**	60.68(1.75)•	**59.01(2.88)**	61.05(1.62)•	**58.42(0.91)**	62.11(1.62)•	**59.00(1.11)**	62.55(1.98)•	**58.95(0.88)**
Airfoil Self Noise	18.24(3.71)•	**17.23(3.81)**	15.61(3.82)•	**14.60(3.70)**	14.56(3.79)•	**13.13(1.07)**	13.43(0.92)•	**12.69(0.77)**	12.59(0.89)•	**12.05(0.79)**	12.21(0.81)•	**11.68(0.70)**
Bank32NH	109.73(3.80)•	**106.42(1.54)**	110.97(3.00)•	**108.16(1.81)**	113.20(2.97)•	**109.00(1.96)**	113.82(2.45)•	**109.31(1.72)**	115.23(3.36)•	**109.73(2.54)**	116.58(3.41)•	**111.35(3.27)**
Bank8FM	**13.05(0.15)**	13.07(0.14)	**13.12(0.18)**	13.13(0.17)	13.29(0.32)	**13.17(0.15)**	13.30(0.16)	**13.26(0.14)**	13.34(0.12)	**13.29(0.15)**	13.40(0.16)•	**13.32(0.18)**
Breast Cancer	716.15(13.38)	**714.54(14.81)**	**715.74(14.92)**	718.17(13.82)	**713.96(12.80)**	716.78(14.10)	**712.50(12.42)**	715.17(15.67)	**713.02(14.21)**	713.24(25.73)	712.96(14.47)	**712.25(20.84)**
CCPP	26.73(0.29)	**26.68(0.53)**	**26.84(0.28)**	27.27(0.67)	**26.94(0.28)**	27.23(0.54)	**27.16(0.24)**	27.31(0.61)	27.36(0.31)•	**26.97(0.75)**	27.48(0.30)•	**26.96(0.59)**
Comp Act	5.84(0.19)•	**5.65(0.08)**	5.89(0.73)•	**5.60(0.12)**	5.90(0.69)•	**5.65(0.16)**	5.97(0.71)•	**5.70(0.29)**	5.85(0.34)•	**5.65(0.14)**	5.88(0.34)•	**5.69(0.17)**
Comp Act Small	8.55(0.54)•	**8.31(0.11)**	8.57(1.02)•	**8.14(0.14)**	8.93(1.61)•	**8.04(0.14)**	8.62(1.18)•	**8.00(0.12)**	8.64(1.20)•	**7.96(0.12)**	8.64(1.19)•	**7.93(0.08)**
Concrete	47.93(36.21)	**39.40(7.20)**	43.89(15.05)•	**36.31(3.46)**	41.24(13.66)	**37.96(9.44)**	**41.30(13.29)**	42.48(21.48)	37.20(4.19)•	**35.47(3.07)**	35.46(3.39)	**35.33(4.38)**
Delta Ailerons	**14.79(0.04)**	14.80(0.06)	**14.71(0.04)**	14.75(0.05)	**14.68(0.04)**	14.72(0.05)	**14.66(0.04)**	14.69(0.05)	**14.64(0.04)**	14.67(0.04)	**14.63(0.04)**	14.65(0.06)
Delta Elevators	28.37(0.76)•	**28.02(0.12)**	28.61(0.69)•	**28.11(0.13)**	29.24(2.09)•	**28.22(0.14)**	29.38(1.85)•	**28.22(0.21)**	29.67(1.96)•	**28.20(0.13)**	29.83(1.94)•	**28.24(0.19)**
Housing	58.76(8.35)•	**52.56(5.44)**	55.03(6.24)•	**51.71(5.27)**	56.20(6.15)•	**50.07(6.02)**	56.31(7.32)•	**52.22(6.21)**	55.77(6.38)•	**49.36(6.03)**	54.83(5.30)•	**51.93(5.43)**
Kinematics	31.60(0.36)•	**31.42(0.39)**	30.23(0.25)•	**30.02(0.27)**	29.64(0.29)•	**29.52(0.26)**	29.32(0.29)•	**29.21(0.25)**	29.06(0.24)•	**28.93(0.25)**	28.89(0.27)•	**28.78(0.29)**
Machine	**57.06(11.97)**	58.66(11.42)	**50.48(10.60)**	51.68(10.26)	**49.18(10.87)**	50.47(10.37)	**48.33(10.88)**	49.29(10.54)	**48.00(11.06)**	49.67(10.80)	**47.44(11.00)**	48.18(10.98)
Puma32H	13.05(0.24)•	**12.92(0.27)**	13.01(0.16)•	**12.86(0.25)**	12.98(0.14)•	**12.83(0.25)**	13.05(0.14)•	**12.79(0.21)**	13.10(0.16)•	**12.85(0.24)**	13.17(0.12)•	**12.84(0.15)**
Puma8NH	173.52(0.72)•	**173.01(0.75)**	174.63(0.92)•	**173.51(0.90)**	175.31(0.80)•	**173.30(0.84)**	175.85(0.88)•	**173.51(0.77)**	176.81(0.90)•	**173.22(1.00)**	177.48(0.97)•	**173.19(0.98)**
Stock	5.95(0.32)•	**5.82(0.27)**	5.95(0.35)	**5.93(0.28)**	**5.90(0.30)**	**5.90(0.26)**	5.91(0.25)	**5.85(0.24)**	5.93(0.26)	**5.85(0.29)**	5.96(0.27)	**5.94(0.28)**
Triazines	**207.96(8.81)**	208.78(8.31)	**206.84(7.69)**	208.86(11.21)	**206.37(7.79)**	208.95(11.87)	**205.76(6.36)**	208.38(10.66)	**204.88(7.06)**	206.75(9.82)	**202.31(7.34)**	206.47(11.12)
Wine Q. Red	164.18(1.49)	**164.00(1.53)**	**163.47(1.16)**	163.74(1.36)	163.62(1.20)	**163.61(1.11)**	163.76(1.13)	**163.66(1.13)**	163.99(1.40)	**163.92(1.27)**	164.23(1.07)	**164.18(1.21)**
Wine Q. White	142.28(4.21)•	**136.39(1.44)**	141.83(3.83)•	**136.74(3.27)**	142.11(3.61)•	**136.15(1.24)**	142.61(3.78)•	**136.21(1.50)**	150.11(21.26)•	**137.29(2.18)**	153.72(22.37)•	**136.43(1.29)**
Win/Tie/Loss	4/0/16	**16/0/4**	7/0/13	**13/0/7**	5/1/14	**14/1/5**	6/0/14	**14/0/6**	4/0/16	**16/0/4**	3/0/17	**17/0/3**

	50		60		70		80		90		100	
Dataset	DS	MINE-S	DS	MINE-S	DS	MINE-S	DS	MINE-S	DS	MINE-S	DS	MINE-S
Abalone	62.95(2.35)•	**58.85(1.15)**	64.00(3.43)•	**59.05(1.09)**	64.37(3.30)•	**59.20(1.00)**	64.87(3.31)•	**59.32(1.06)**	65.49(3.46)•	**59.53(1.04)**	65.64(3.33)•	**59.06(1.05)**
Airfoil Self Noise	11.91(0.76)•	**11.35(0.68)**	11.64(0.83)•	**11.21(0.66)**	11.57(0.84)•	**11.05(0.70)**	11.67(1.71)•	**11.18(1.75)**	11.55(1.66)•	**10.75(0.66)**	11.42(1.71)•	**10.84(0.88)**
Bank32NH	117.44(3.60)•	**110.97(2.65)**	118.70(4.81)•	**111.49(2.29)**	118.97(4.83)•	**111.60(2.60)**	119.55(4.95)•	**111.08(2.30)**	120.10(5.14)•	**110.80(3.31)**	120.60(4.99)•	**110.87(2.45)**
Bank8FM	13.51(0.22)•	**13.32(0.20)**	13.56(0.22)•	**13.40(0.30)**	13.67(0.29)•	**13.41(0.26)**	13.75(0.32)•	**13.46(0.23)**	13.81(0.34)•	**13.45(0.25)**	13.87(0.41)•	**13.53(0.43)**
Breast Cancer	712.68(13.54)	**707.98(17.93)**	713.75(12.22)•	**705.98(14.15)**	712.96(12.53)	**710.66(14.10)**	712.30(12.55)	**707.22(18.53)**	713.54(12.46)	**711.42(17.19)**	713.60(10.68)	**713.56(19.43)**
CCPP	27.58(0.43)•	**26.62(0.64)**	27.68(0.43)•	**26.64(0.54)**	27.78(0.46)•	**26.61(0.61)**	27.91(0.44)•	**26.41(0.66)**	28.01(0.50)•	**26.35(0.99)**	28.09(0.46)•	**26.22(0.63)**
Comp Act	5.90(0.38)•	**5.69(0.15)**	5.96(0.53)•	**5.72(0.16)**	5.96(0.52)	**5.75(0.21)**	5.96(0.43)	**5.70(0.19)**	5.91(0.40)	**5.80(0.30)**	5.88(0.33)	**5.75(0.28)**
Comp Act Small	8.63(1.23)•	**7.95(0.10)**	8.63(1.21)•	**7.91(0.10)**	8.65(1.18)•	**7.90(0.10)**	8.67(1.24)•	**7.92(0.15)**	8.74(1.26)•	**7.89(0.11)**	8.64(1.20)•	**7.88(0.10)**
Concrete	36.05(4.50)	**35.33(4.01)**	35.59(4.38)	**35.52(4.04)**	35.23(4.25)	**35.18(4.72)**	35.30(3.83)	**34.85(4.39)**	36.01(4.85)	**34.89(3.71)**	36.07(4.60)	**34.42(3.33)**
Delta Ailerons	**14.62(0.04)**	14.64(0.04)	**14.61(0.05)**	14.63(0.06)	**14.60(0.05)**	14.63(0.06)	**14.60(0.04)**	14.62(0.07)	**14.59(0.04)**	14.62(0.06)	**14.59(0.04)**	14.62(0.06)
Delta Elevators	29.94(1.92)•	**28.30(0.16)**	30.11(1.89)•	**28.23(0.21)**	30.10(1.88)•	**28.27(0.14)**	30.21(1.86)•	**28.26(0.19)**	30.61(2.43)•	**28.27(0.17)**	30.69(2.45)•	**28.25(0.18)**
Housing	54.38(4.46)•	**51.92(5.41)**	55.24(6.27)•	**50.84(4.90)**	55.15(6.01)•	**50.45(5.41)**	55.32(5.91)•	**51.89(5.38)**	55.15(6.52)•	**51.86(8.07)**	54.68(6.47)	**53.22(6.95)**
Kinematics	28.71(0.23)•	**28.62(0.26)**	28.62(0.29)	**28.56(0.26)**	28.52(0.28)	**28.47(0.27)**	28.50(0.25)•	**28.42(0.26)**	28.45(0.22)•	**28.35(0.25)**	28.43(0.24)	**28.36(0.26)**
Machine	**47.55(10.97)**	48.73(11.47)	**47.09(10.87)**	49.00(10.54)	**46.99(10.88)**	48.17(10.80)	**46.93(10.91)**	48.27(11.12)	**46.85(10.82)**	48.81(10.34)	**47.01(10.90)**	48.59(10.50)
Puma32H	13.19(0.12)•	**12.88(0.21)**	13.20(0.15)•	**12.90(0.21)**	13.23(0.16)•	**12.87(0.18)**	13.20(0.17)•	**12.92(0.21)**	13.22(0.17)•	**12.89(0.23)**	13.27(0.18)•	**12.98(0.20)**
Puma8NH	177.99(1.05)•	**173.28(0.62)**	178.27(1.06)•	**173.33(0.90)**	178.79(1.00)•	**173.25(0.72)**	179.17(1.24)•	**173.39(0.80)**	179.45(1.22)•	**173.22(0.85)**	179.73(1.22)•	**173.43(0.79)**
Stock	5.89(0.27)	**5.87(0.31)**	**5.87(0.26)**	5.90(0.29)	5.85(0.28)	**5.81(0.27)**	**5.84(0.34)**	5.85(0.31)	**5.85(0.34)**	5.88(0.24)	5.89(0.37)	**5.88(0.32)**
Triazines	**201.48(7.83)**	207.24(10.12)	**201.58(7.45)**	205.07(10.62)	**202.65(7.83)**	206.92(13.75)	**201.60(7.96)**	207.14(11.10)	**201.72(7.83)**	210.19(12.61)	**201.23(7.85)**	210.55(11.21)
Wine Q. Red	**164.26(1.06)**	164.58(1.08)	**164.00(1.33)**	164.06(1.07)	**164.02(1.28)**	164.36(2.02)	**164.10(1.14)**	164.24(1.44)	**164.14(1.07)**	164.70(1.39)	**164.20(1.03)**	164.38(1.16)
Wine Q. White	152.05(21.02)•	**137.08(2.14)**	152.02(21.08)•	**136.90(1.31)**	146.27(6.54)•	**137.56(2.29)**	147.22(6.37)•	**137.70(2.10)**	149.48(8.00)•	**138.09(2.41)**	148.92(7.32)•	**138.89(5.78)**
Win/Tie/Loss	4/0/16	**16/0/4**	5/0/15	**15/0/5**	4/0/16	**16/0/4**	5/0/15	**15/0/5**	5/0/15	**15/0/5**	4/0/16	**16/0/4**

Table 11 – Mean and standard deviation of the results calculated in 20 replications. For each dataset, the best result is in bold. Line "Win/Tie/Loss" shows the total of the results compared with MINE-W. The values marked with a • indicate that the null hypothesis must be rejected ($pValue \leq 0.05$), in other words, the result of MINE-W achieves superior performance. The values are in the scale 10^{-4}.

	5		10		15		20		30		40	
Dataset	DW	MINE-W	DW	MINE-W	DW	MINE-W	DW	MINE-W	DW	MINE-W	DW	MINE-W
Abalone	55.32(0.39)•	**55.25(0.40)**	54.90(0.28)	**54.87(0.24)**	54.74(0.21)	**54.72(0.20)**	**54.68(0.18)**	**54.68(0.19)**	**54.62(0.18)**	54.63(0.22)	54.59(0.19)	**54.58(0.20)**
Airfoil Self Noise	20.45(0.70)•	**16.81(1.14)**	18.65(0.61)•	**14.33(1.02)**	18.15(0.54)•	**13.46(0.72)**	17.92(0.50)•	**13.07(0.67)**	17.69(0.50)•	**12.56(0.54)**	17.54(0.42)•	**12.29(0.46)**
Bank32NH	92.69(0.68)•	**92.52(0.64)**	**90.42(0.43)**	90.44(0.43)	**89.76(0.46)**	89.84(0.45)	**89.45(0.34)**	89.55(0.35)	**89.09(0.29)**	89.23(0.30)	**88.93(0.28)**	89.06(0.31)
Bank8FM	**12.45(0.07)**	**12.45(0.07)**	12.35(0.05)•	**12.34(0.05)**	12.31(0.04)•	**12.30(0.04)**	12.30(0.04)•	**12.29(0.04)**	12.27(0.03)•	**12.26(0.03)**	12.26(0.03)•	**12.25(0.03)**
Breast Cancer	**722.52(11.52)**	724.68(12.42)	**718.31(9.00)**	721.26(10.58)	**716.69(7.31)**	719.39(7.62)	**716.53(7.12)**	720.59(7.49)	**716.33(7.17)**	719.81(7.88)	**715.85(6.95)**	719.12(7.46)
CCPP	24.16(0.17)•	**24.03(0.16)**	23.61(0.13)•	**23.48(0.13)**	23.42(0.12)•	**23.29(0.12)**	23.32(0.12)•	**23.18(0.11)**	23.23(0.12)•	**23.09(0.11)**	23.18(0.12)•	**23.04(0.11)**
Comp Act	5.51(0.06)•	**5.43(0.05)**	5.41(0.05)•	**5.34(0.05)**	5.37(0.03)•	**5.31(0.04)**	5.36(0.03)•	**5.30(0.03)**	5.34(0.02)•	**5.29(0.03)**	5.33(0.02)•	**5.28(0.03)**
Comp Act Small	8.52(0.08)•	**8.37(0.05)**	8.41(0.09)•	**8.24(0.06)**	8.36(0.06)•	**8.18(0.04)**	8.33(0.04)•	**8.15(0.03)**	8.31(0.04)•	**8.12(0.03)**	8.30(0.03)•	**8.10(0.03)**
Concrete	39.35(4.44)•	**35.71(1.35)**	36.47(1.71)•	**33.25(1.30)**	35.73(1.15)•	**32.52(1.09)**	35.32(1.04)•	**32.17(1.19)**	34.90(1.20)•	**32.32(4.04)**	34.55(1.06)•	**31.63(2.63)**
Delta Ailerons	15.04(0.04)•	**15.03(0.04)**	15.03(0.03)•	**15.02(0.03)**	**15.02(0.02)**	**15.02(0.03)**	**15.02(0.03)**	**15.02(0.03)**	15.02(0.02)•	**15.01(0.03)**	15.02(0.02)•	**15.01(0.02)**
Delta Elevators	27.69(0.77)•	**27.50(0.08)**	27.45(0.19)	**27.41(0.05)**	27.40(0.10)	**27.39(0.04)**	**27.37(0.06)**	27.38(0.04)	**27.34(0.03)**	27.35(0.03)	**27.33(0.03)**	27.34(0.03)
Housing	54.98(5.06)•	**50.74(3.16)**	52.69(2.93)•	**49.25(2.64)**	51.41(3.04)•	**48.20(2.74)**	50.79(3.21)•	**47.78(2.95)**	50.36(2.92)•	**47.38(2.67)**	50.12(2.97)•	**47.37(2.96)**
Kinematics	33.28(0.36)•	**32.14(0.35)**	32.38(0.33)•	**31.00(0.29)**	32.13(0.28)•	**30.63(0.25)**	31.98(0.23)•	**30.42(0.21)**	31.83(0.19)•	**30.20(0.17)**	31.78(0.17)•	**30.10(0.16)**
Machine	**71.19(9.13)**	74.52(7.05)	**68.83(8.60)**	71.32(8.24)	**68.49(8.06)**	71.17(7.74)	**68.10(7.96)**	71.55(7.80)	**67.71(7.91)**	72.15(7.92)	**67.67(7.89)**	70.45(7.71)
Puma32H	11.26(0.19)•	**11.25(0.18)**	10.94(0.15)•	**10.93(0.14)**	10.82(0.10)•	**10.81(0.09)**	10.79(0.08)•	**10.77(0.07)**	10.73(0.05)•	**10.72(0.05)**	10.72(0.04)•	**10.70(0.04)**
Puma8NH	**168.08(0.31)**	168.13(0.31)	**167.51(0.28)**	167.55(0.27)	**167.30(0.24)**	167.34(0.22)	**167.19(0.19)**	167.23(0.19)	**167.11(0.16)**	167.15(0.15)	**167.05(0.17)**	167.09(0.16)
Stock	5.40(0.21)•	**5.23(0.22)**	5.18(0.19)•	**5.02(0.20)**	5.11(0.17)•	**4.97(0.17)**	5.07(0.16)•	**4.94(0.16)**	5.03(0.16)•	**4.91(0.17)**	5.02(0.16)•	**4.90(0.17)**
Triazines	207.73(6.57)	**207.58(6.65)**	**206.08(5.43)**	206.77(5.79)	**206.28(5.49)**	206.51(6.25)	**206.46(5.28)**	207.71(6.56)	**205.93(5.24)**	207.53(7.09)	**205.53(4.95)**	206.28(6.08)
Wine Q. Red	164.74(1.23)•	**164.65(1.34)**	164.31(0.85)	**164.25(0.94)**	164.25(0.66)•	**164.16(0.68)**	164.15(0.63)•	**164.09(0.66)**	164.09(0.65)•	**164.03(0.69)**	164.04(0.64)•	**163.99(0.69)**
Wine Q. White	135.45(1.48)•	**133.82(1.36)**	133.75(1.06)•	**132.30(0.97)**	133.17(0.71)•	**131.69(0.73)**	132.91(0.66)•	**131.50(0.79)**	132.67(0.65)•	**131.14(0.52)**	132.48(0.59)•	**131.18(0.76)**
Win/Tie/Loss	3/1/16	**16/1/3**	5/0/15	**15/0/5**	5/1/14	**14/1/5**	6/2/12	**12/2/6**	7/0/13	**13/0/7**	6/0/14	**14/0/6**

	50		60		70		80		90		100	
Dataset	DW	MINE-W	DW	MINE-W	DW	MINE-W	DW	MINE-W	DW	MINE-W	DW	MINE-W
Abalone	**54.55(0.20)**	**54.55(0.21)**	54.55(0.23)	**54.54(0.23)**	**54.53(0.20)**	**54.53(0.21)**	54.51(0.19)	**54.50(0.19)**	**54.50(0.18)**	**54.50(0.18)**	**54.50(0.18)**	**54.50(0.17)**
Airfoil Self Noise	17.48(0.41)•	**12.08(0.40)**	17.41(0.39)•	**11.93(0.38)**	17.40(0.38)•	**11.85(0.35)**	17.35(0.38)•	**11.75(0.33)**	17.34(0.36)•	**11.69(0.31)**	17.33(0.37)•	**11.65(0.33)**
Bank32NH	**88.83(0.28)**	88.95(0.34)	**88.77(0.31)**	88.91(0.37)	**88.75(0.32)**	88.86(0.37)	**88.70(0.29)**	88.83(0.34)	**88.69(0.29)**	88.78(0.33)	**88.68(0.27)**	88.77(0.29)
Bank8FM	12.26(0.03)•	**12.24(0.03)**	12.25(0.03)•	**12.24(0.03)**	12.25(0.02)•	**12.23(0.03)**	12.24(0.03)•	**12.23(0.03)**	12.24(0.02)•	**12.23(0.03)**	12.24(0.02)•	**12.23(0.03)**
Breast Cancer	**715.39(6.69)**	720.70(7.30)	**715.11(6.83)**	720.00(7.64)	**714.94(6.57)**	718.83(7.45)	**714.75(6.33)**	718.36(7.94)	**714.82(6.55)**	718.09(8.42)	**714.91(6.53)**	719.04(8.18)
CCPP	23.14(0.13)•	**23.00(0.12)**	23.12(0.12)•	**22.98(0.12)**	23.10(0.12)•	**22.96(0.12)**	23.10(0.13)•	**22.96(0.12)**	23.09(0.13)•	**22.95(0.12)**	23.09(0.12)•	**22.95(0.12)**
Comp Act	5.32(0.02)•	**5.28(0.03)**	5.32(0.02)•	**5.27(0.02)**	5.32(0.02)•	**5.27(0.02)**	5.32(0.02)•	**5.27(0.02)**	5.31(0.02)•	**5.26(0.02)**	5.31(0.02)•	**5.26(0.02)**
Comp Act Small	8.29(0.02)•	**8.10(0.03)**	8.29(0.03)•	**8.10(0.05)**	8.28(0.03)•	**8.09(0.04)**	8.28(0.02)•	**8.08(0.03)**	8.28(0.03)•	**8.08(0.03)**	8.28(0.03)•	**8.08(0.03)**
Concrete	34.37(1.02)•	**31.23(2.02)**	34.30(1.03)•	**31.11(1.91)**	34.21(0.94)•	**30.90(1.55)**	34.16(0.93)•	**30.76(1.36)**	34.13(0.91)•	**30.69(1.21)**	34.17(0.88)•	**30.74(1.18)**
Delta Ailerons	15.02(0.02)•	**15.01(0.02)**	15.02(0.02)•	**15.01(0.02)**	15.02(0.02)•	**15.01(0.02)**	15.02(0.02)•	**15.01(0.02)**	15.02(0.02)•	**15.00(0.02)**	15.02(0.02)•	**15.00(0.02)**
Delta Elevators	**27.32(0.03)**	27.34(0.04)	**27.32(0.03)**	27.33(0.04)	**27.31(0.03)**	27.33(0.04)	**27.31(0.03)**	27.33(0.04)	**27.31(0.03)**	27.32(0.03)	**27.31(0.03)**	27.32(0.03)
Housing	50.08(2.89)•	**47.23(3.05)**	49.91(2.79)•	**47.14(2.83)**	49.89(2.80)•	**47.17(2.80)**	49.86(2.84)•	**47.12(2.73)**	49.79(2.85)•	**47.08(2.80)**	49.77(2.90)•	**47.06(2.92)**
Kinematics	31.73(0.16)•	**30.02(0.15)**	31.70(0.11)•	**29.97(0.13)**	31.66(0.11)•	**29.90(0.11)**	31.65(0.10)•	**29.88(0.12)**	31.64(0.09)•	**29.86(0.10)**	31.63(0.08)•	**29.83(0.09)**
Machine	**67.48(7.73)**	69.61(7.74)	**67.04(7.66)**	69.95(7.56)	**66.84(7.66)**	68.91(7.09)	**66.62(7.64)**	69.77(6.77)	**66.70(7.63)**	70.13(6.39)	**66.79(7.74)**	70.10(6.62)
Puma32H	10.70(0.04)•	**10.69(0.04)**	10.68(0.03)•	**10.67(0.03)**	10.68(0.03)•	**10.66(0.03)**	10.67(0.03)•	**10.65(0.03)**	10.67(0.03)•	**10.65(0.03)**	10.66(0.03)•	**10.65(0.03)**
Puma8NH	**167.00(0.18)**	167.05(0.17)	**166.98(0.17)**	167.04(0.16)	**166.96(0.17)**	167.01(0.16)	**166.95(0.16)**	167.00(0.16)	**166.94(0.16)**	167.00(0.16)	**166.94(0.15)**	166.99(0.15)
Stock	4.99(0.16)•	**4.88(0.17)**	4.98(0.15)•	**4.86(0.16)**	4.97(0.15)•	**4.86(0.16)**	4.97(0.15)•	**4.86(0.16)**	4.97(0.15)•	**4.85(0.16)**	4.97(0.15)•	**4.86(0.16)**
Triazines	**205.39(4.89)**	206.31(6.01)	**205.31(4.95)**	206.63(6.21)	**205.33(4.93)**	206.16(5.66)	**205.25(5.08)**	206.17(5.62)	**205.23(5.05)**	206.04(5.00)	**205.12(5.14)**	206.05(5.44)
Wine Q. Red	164.00(0.60)	**163.95(0.67)**	163.98(0.65)•	**163.92(0.70)**	163.96(0.66)•	**163.88(0.70)**	163.97(0.63)•	**163.90(0.66)**	163.97(0.61)•	**163.91(0.65)**	163.96(0.62)•	**163.91(0.67)**
Wine Q. White	132.43(0.52)•	**131.01(0.75)**	132.37(0.55)•	**130.89(0.63)**	132.32(0.50)•	**130.76(0.45)**	132.30(0.51)•	**130.69(0.45)**	132.27(0.46)•	**130.68(0.42)**	132.23(0.43)•	**130.61(0.44)**
Win/Tie/Loss	6/1/13	**13/1/6**	6/0/14	**14/0/6**	6/1/13	**13/1/6**	6/0/14	**14/0/6**	6/1/13	**13/1/6**	6/1/13	**13/1/6**

Table 12 – Mean and standard deviation of the results calculated in 20 replications. For each dataset, the best result is in bold. Line "Win/Tie/Loss" shows the total of the results compared with MINE-WS. The values marked with a • indicate that the null hypothesis must be rejected ($pValue \leq 0.05$), in other words, the result of MINE-WS achieves superior performance. The values are in the scale 10^{-4}.

	5		10		15		20		30		40	
Dataset	DWS	MINE-WS	DWS	MINE-WS	DWS	MINE-WS	DWS	MINE-WS	DWS	MINE-WS	DWS	MINE-WS
Abalone	55.34(0.40)	**55.29(0.39)**	54.94(0.30)	**54.87(0.29)**	54.81(0.23)	**54.78(0.24)**	**54.76(0.20)**	54.78(0.21)	**54.69(0.18)**	54.75(0.35)	**54.77(0.28)**	54.80(0.30)
Airfoil Self Noise	19.45(0.83)•	**15.73(1.06)**	16.52(0.88)•	**13.34(0.79)**	15.19(0.79)•	**12.78(0.67)**	14.59(0.70)•	**12.54(0.55)**	13.81(0.59)•	**12.55(0.43)**	13.56(0.53)•	**12.35(0.51)**
Bank32NH	92.87(0.80)•	**92.65(0.67)**	90.93(0.68)•	**90.72(0.47)**	90.66(0.68)•	**90.22(0.42)**	90.56(0.63)•	**90.15(0.39)**	90.84(0.81)•	**90.28(0.88)**	91.17(1.02)•	**90.36(0.52)**
Bank8FM	**12.45(0.07)**	12.47(0.08)	**12.35(0.05)**	12.37(0.06)	**12.31(0.04)**	12.35(0.05)	**12.30(0.05)**	12.35(0.07)	**12.28(0.05)**	12.34(0.08)	**12.27(0.05)**	12.32(0.05)
Breast Cancer	**722.52(11.52)**	724.90(14.29)	**718.31(9.00)**	722.44(13.81)	**716.69(7.31)**	720.95(11.54)	**716.53(7.12)**	720.81(11.65)	**716.33(7.17)**	722.07(11.50)	**715.85(6.95)**	718.61(10.98)
CCPP	24.21(0.22)•	**24.06(0.20)**	23.74(0.21)•	**23.54(0.19)**	23.65(0.21)•	**23.32(0.15)**	23.62(0.19)•	**23.26(0.17)**	23.65(0.21)•	**23.19(0.19)**	23.67(0.20)•	**23.13(0.18)**
Comp Act	5.54(0.12)•	**5.47(0.06)**	5.45(0.08)•	**5.37(0.06)**	5.42(0.05)•	**5.35(0.09)**	5.41(0.05)•	**5.31(0.05)**	5.41(0.12)•	**5.26(0.04)**	5.38(0.03)•	**5.25(0.06)**
Comp Act Small	8.53(0.08)•	**8.37(0.05)**	8.51(0.40)•	**8.20(0.08)**	8.42(0.12)•	**8.08(0.06)**	8.38(0.12)•	**8.02(0.04)**	8.34(0.11)•	**7.96(0.06)**	8.31(0.08)•	**7.93(0.05)**
Concrete	43.92(27.89)•	**35.87(1.61)**	35.89(3.11)•	**33.97(3.01)**	34.31(2.35)•	**33.55(5.73)**	33.74(2.09)•	**32.02(1.27)**	32.95(2.79)•	**31.33(1.59)**	31.94(1.61)•	**31.17(1.66)**
Delta Ailerons	**15.04(0.04)**	**15.04(0.04)**	**15.03(0.03)**	**15.03(0.03)**	**15.02(0.02)**	**15.02(0.03)**	**15.02(0.03)**	**15.02(0.04)**	**15.02(0.02)**	15.03(0.03)	**15.02(0.02)**	**15.02(0.03)**
Delta Elevators	27.70(0.77)	**27.52(0.09)**	27.46(0.19)	**27.43(0.06)**	**27.41(0.11)**	27.45(0.11)	**27.39(0.07)**	27.40(0.08)	**27.36(0.05)**	27.37(0.05)	**27.39(0.16)**	27.40(0.10)
Housing	55.95(5.11)•	**51.39(3.60)**	53.56(3.07)•	**49.74(3.06)**	52.54(3.33)•	**48.70(3.85)**	51.89(3.50)•	**47.80(5.33)**	50.90(3.18)•	**47.01(4.12)**	50.63(3.39)•	**46.59(3.42)**
Kinematics	33.25(0.36)•	**31.76(0.32)**	32.27(0.33)•	**30.10(0.29)**	31.93(0.30)•	**29.26(0.27)**	31.72(0.24)•	**28.85(0.24)**	31.42(0.21)•	**28.38(0.17)**	31.25(0.18)•	**28.12(0.21)**
Machine	**66.01(11.05)**	67.05(10.32)	58.78(10.42)	**58.27(10.23)**	**55.10(10.51)**	56.43(10.16)	**54.43(10.30)**	57.41(10.22)	**53.90(10.30)**	58.23(10.40)	**53.50(9.99)**	55.57(10.86)
Puma32H	**11.26(0.19)**	11.27(0.19)	**10.95(0.15)**	**10.95(0.15)**	**10.83(0.10)**	10.84(0.10)	**10.80(0.08)•**	**10.80(0.08)**	10.76(0.06)•	**10.75(0.05)**	10.76(0.05)•	**10.73(0.05)**
Puma8NH	**168.08(0.31)**	168.14(0.31)	**167.51(0.28)**	167.56(0.27)	**167.30(0.24)**	167.35(0.24)	**167.19(0.19)**	167.24(0.17)	**167.11(0.16)**	167.28(0.51)	**167.05(0.16)**	167.14(0.18)
Stock	5.44(0.24)•	**5.39(0.21)**	5.31(0.22)•	**5.23(0.22)**	5.25(0.20)•	**5.15(0.22)**	5.17(0.19)•	**5.07(0.20)**	5.07(0.17)•	**4.91(0.13)**	5.02(0.16)•	**4.82(0.15)**
Triazines	**208.31(6.72)**	208.44(7.68)	**207.33(5.65)**	208.45(7.43)	**208.08(5.42)**	208.44(7.17)	**208.16(5.45)**	210.21(8.38)	**207.65(5.66)**	208.58(8.04)	**207.00(5.50)**	207.87(6.16)
Wine Q. Red	164.68(1.25)•	**164.50(1.27)**	164.16(0.95)	**164.12(1.08)**	**164.01(0.69)**	164.06(0.83)	**163.86(0.67)**	163.87(0.81)	163.75(0.72)	**163.70(0.79)**	**163.66(0.76)**	163.68(0.77)
Wine Q. White	135.71(1.66)•	**133.80(1.08)**	134.05(1.44)•	**132.56(1.12)**	133.71(1.10)•	**131.86(0.81)**	133.68(1.22)•	**131.71(0.68)**	133.51(1.41)•	**131.04(0.54)**	133.70(2.06)•	**131.01(0.92)**
Win/Tie/Loss	6/1/13	**13/1/6**	4/2/14	**14/2/4**	8/1/11	**11/1/8**	8/2/10	**10/2/8**	8/0/12	**12/0/8**	8/1/11	**11/1/8**

	50		60		70		80		90		100	
Dataset	DWS	MINE-WS	DWS	MINE-WS	DWS	MINE-WS	DWS	MINE-WS	DWS	MINE-WS	DWS	MINE-WS
Abalone	54.74(0.24)	**54.71(0.35)**	**54.77(0.28)**	54.78(0.30)	**54.76(0.25)**	54.85(0.33)	**54.76(0.43)**	54.81(0.32)	**54.78(0.47)**	55.13(0.61)	**54.80(0.40)**	54.93(0.34)
Airfoil Self Noise	13.25(0.41)•	**12.25(0.39)**	13.17(0.39)•	**12.17(0.39)**	13.07(0.44)•	**12.11(0.51)**	13.11(0.46)•	**12.14(0.42)**	13.17(0.45)•	**11.92(0.48)**	13.23(0.48)•	**12.03(0.45)**
Bank32NH	91.60(1.01)•	**90.62(0.60)**	91.90(0.98)•	**90.59(0.54)**	92.41(1.51)•	**90.43(0.92)**	92.68(1.73)•	**90.08(0.57)**	93.06(1.73)•	**89.94(0.57)**	93.25(1.38)•	**89.99(0.73)**
Bank8FM	**12.27(0.05)**	12.32(0.07)	**12.26(0.05)**	12.31(0.06)	**12.26(0.05)**	12.32(0.06)	**12.26(0.04)**	12.34(0.13)	**12.27(0.07)**	12.36(0.16)	**12.27(0.07)**	12.33(0.12)
Breast Cancer	**715.39(6.69)**	721.03(8.87)	**715.11(6.83)**	721.21(9.86)	**714.94(6.57)**	720.55(10.53)	**714.75(6.33)**	720.80(10.90)	**714.82(6.55)**	718.71(10.24)	**714.91(6.53)**	721.39(12.96)
CCPP	23.70(0.26)•	**23.07(0.18)**	23.70(0.28)•	**23.06(0.19)**	23.74(0.28)•	**23.01(0.15)**	23.80(0.26)•	**23.03(0.16)**	23.84(0.29)•	**23.02(0.18)**	23.90(0.29)•	**23.00(0.15)**
Comp Act	5.36(0.03)•	**5.23(0.06)**	5.35(0.03)•	**5.22(0.02)**	5.34(0.03)•	**5.21(0.03)**	5.35(0.04)•	**5.21(0.05)**	5.35(0.04)•	**5.20(0.03)**	5.35(0.03)•	**5.19(0.02)**
Comp Act Small	8.29(0.07)•	**7.92(0.05)**	8.28(0.08)•	**7.90(0.05)**	8.27(0.08)•	**7.89(0.07)**	8.25(0.07)•	**7.87(0.06)**	8.23(0.05)•	**7.87(0.09)**	8.22(0.04)•	**7.85(0.05)**
Concrete	31.89(1.54)•	**30.87(1.35)**	31.41(1.56)	**31.06(1.43)**	**31.02(1.42)**	**31.02(1.29)**	**31.03(1.39)**	31.18(1.56)	31.11(1.38)	**30.88(1.13)**	30.95(1.25)	**30.70(1.04)**
Delta Ailerons	**15.02(0.02)**	**15.02(0.03)**	**15.02(0.02)**	**15.02(0.03)**	15.02(0.02)	**15.01(0.03)**	**15.01(0.02)**	**15.01(0.03)**	**15.02(0.02)**	**15.02(0.03)**	**15.02(0.02)**	**15.02(0.03)**
Delta Elevators	27.39(0.17)	**27.37(0.06)**	**27.40(0.17)**	27.79(1.85)	27.39(0.17)	**27.36(0.06)**	27.51(0.53)	**27.38(0.06)**	27.51(0.53)	**27.41(0.17)**	27.52(0.53)	**27.40(0.11)**
Housing	50.64(3.49)•	**45.98(3.55)**	50.38(3.46)•	**45.64(3.67)**	50.27(3.22)•	**45.16(3.70)**	50.10(3.16)•	**45.04(3.15)**	49.98(3.21)•	**45.28(3.46)**	49.74(3.29)•	**44.39(2.60)**
Kinematics	31.08(0.17)•	**27.97(0.20)**	30.97(0.17)•	**27.84(0.17)**	30.83(0.16)•	**27.74(0.16)**	30.73(0.13)•	**27.71(0.16)**	30.67(0.12)•	**27.68(0.15)**	30.61(0.12)•	**27.62(0.14)**
Machine	**53.16(10.15)**	55.96(10.05)	**52.67(9.82)**	55.31(10.06)	**52.50(9.81)**	56.60(10.42)	**52.46(9.88)**	55.84(10.37)	**52.34(9.91)**	56.24(10.73)	**52.34(9.97)**	56.60(10.35)
Puma32H	10.76(0.05)•	**10.72(0.04)**	10.76(0.04)•	**10.71(0.04)**	10.77(0.05)•	**10.70(0.04)**	10.77(0.04)•	**10.70(0.04)**	10.78(0.04)•	**10.69(0.03)**	10.78(0.04)•	**10.69(0.04)**
Puma8NH	**167.01(0.17)**	167.08(0.20)	**166.99(0.16)**	167.10(0.19)	**166.96(0.17)**	167.08(0.19)	**166.96(0.16)**	167.04(0.18)	**166.95(0.17)**	167.02(0.15)	**166.95(0.16)**	167.01(0.15)
Stock	4.98(0.17)•	**4.77(0.16)**	4.92(0.19)•	**4.75(0.15)**	4.87(0.18)•	**4.74(0.17)**	4.85(0.18)•	**4.75(0.16)**	4.84(0.16)•	**4.74(0.17)**	4.84(0.16)•	**4.74(0.15)**
Triazines	**206.74(5.54)**	206.86(6.49)	**206.49(5.87)**	208.91(7.75)	**206.39(5.74)**	209.36(6.57)	**206.78(5.74)**	209.59(7.32)	**206.68(5.82)**	210.11(7.24)	**206.69(5.74)**	208.41(6.78)
Wine Q. Red	**163.63(0.73)**	163.68(0.78)	**163.59(0.76)**	163.65(0.79)	**163.53(0.75)**	163.69(0.67)	**163.54(0.72)**	163.75(0.69)	**163.58(0.74)**	163.73(0.69)	**163.55(0.75)**	163.77(0.71)
Wine Q. White	133.65(1.63)•	**130.69(0.70)**	133.50(1.44)•	**130.79(1.58)**	133.43(1.38)•	**130.70(1.44)**	133.30(1.26)•	**130.64(1.58)**	133.19(1.13)•	**130.11(0.76)**	133.11(1.11)•	**130.54(1.28)**
Win/Tie/Loss	7/1/13	**13/1/7**	8/1/11	**11/1/8**	7/1/12	**12/1/7**	8/1/11	**11/1/8**	7/1/12	**12/1/7**	7/1/12	**12/1/7**

4.4.6 Comparing MINE with static techniques

This section compares the three MINE techniques against Individual Regressor, and the combination of the regressors using Mean and Median as the combination rule (Table 13). The "Individual Regressor" column shows the results per dataset when only one regressor is applied. Each dataset is trained using the best performing regressor as listed in Table 9. For the sake of simplicity, the size of the ensemble was defined as 90 given that this value reached competitive precision as reported in Tables 11 and 12.

Table 13 – Mean and standard deviation of the results calculated in 20 replications. For each dataset, the best result is in bold. The values are in the scale 10^{-4}. Ensemble Size = 90.

Dataset	Individual Regressor	Mean	Median	MINE-S	MINE-W	MINE-WS
Abalone	56.58(0.68)	54.51(0.19)	**54.46(0.17)**	59.53(1.04)	54.50(0.18)	55.13(0.61)
Airfoil Self Noise	32.00(4.54)	21.48(0.43)	20.89(0.42)	10.75(0.66)	**11.69(0.31)**	11.92(0.48)
Bank32NH	98.04(0.96)	88.79(0.29)	89.15(0.25)	110.80(3.31)	**88.78(0.33)**	89.94(0.57)
Bank8FM	12.79(0.08)	12.27(0.02)	12.32(0.02)	13.45(0.25)	**12.23(0.03)**	12.36(0.16)
Breast Cancer	730.74(9.40)	715.72(6.52)	716.95(7.10)	**711.42(17.19)**	718.09(8.42)	718.71(10.24)
CCPP	24.46(0.15)	23.37(0.12)	23.42(0.12)	26.35(0.99)	**22.95(0.12)**	23.02(0.18)
Comp Act	6.03(0.16)	5.37(0.02)	5.38(0.02)	5.80(0.30)	5.26(0.02)	**5.20(0.03)**
Comp Act Small	9.28(0.20)	8.42(0.02)	8.45(0.02)	7.89(0.11)	8.08(0.03)	**7.87(0.09)**
Concrete	52.83(2.70)	39.36(0.89)	38.08(0.76)	34.89(3.71)	**30.69(1.21)**	30.88(1.13)
Delta Ailerons	15.05(0.02)	15.03(0.02)	15.03(0.02)	**14.62(0.06)**	15.00(0.02)	15.02(0.03)
Delta Elevators	27.76(0.10)	**27.32(0.03)**	27.35(0.03)	28.27(0.17)	**27.32(0.03)**	27.41(0.17)
Housing	55.79(4.32)	51.75(2.62)	51.01(2.25)	51.86(8.07)	47.08(2.80)	**45.28(3.46)**
Kinematics	39.61(1.03)	33.01(0.10)	33.04(0.12)	28.35(0.25)	29.86(0.10)	**27.68(0.15)**
Machine	82.03(5.37)	78.79(6.21)	81.56(5.44)	**48.81(10.34)**	70.13(6.39)	56.24(10.73)
Puma32H	12.29(0.43)	10.71(0.03)	**10.64(0.02)**	12.89(0.23)	10.65(0.03)	10.69(0.03)
Puma8NH	169.48(0.64)	**166.93(0.16)**	166.95(0.17)	173.22(0.85)	167.00(0.16)	167.02(0.15)
Stock	5.52(0.22)	5.26(0.14)	5.23(0.16)	5.88(0.24)	4.85(0.16)	**4.74(0.17)**
Triazines	211.96(4.99)	206.27(4.89)	209.16(5.19)	210.19(12.61)	**206.04(5.00)**	210.11(7.24)
Wine Q. Red	164.81(0.55)	164.30(0.59)	164.67(0.62)	164.70(1.39)	163.91(0.65)	**163.73(0.69)**
Wine Q. White	143.12(3.19)	133.35(0.39)	133.56(0.34)	138.09(2.41)	130.68(0.42)	**130.11(0.76)**

The "Individual Regressor" did not obtain the best performance in any of the used datasets. In contrast, the MINE techniques achieved the best results in 17 out of 20 datasets; a special highlight to MINE-WS that obtained similar performance when compared with MINE-W, however, it uses only a subset of the ensemble while MINE-W uses the whole ensemble.

In Appendix 6 the MINE techniques are compared among them. Also, the MINE-S is individually compared to the Individual Regressor and the MINE-W is compared to the Mean and Median. The same hypothesis tests configurations presented in Section 4.4.1 were performed in the comparisons and added into the tables of Appendix 6.

4.4.7 Evaluating the Measures

As explained in the previous sections, all the eight measures presented in Section 4.3.3.1 were combined using a vector of weights $\mathcal{W}$ calculated in the Optimization Phase of the

MINE framework. For each test pattern, the combination of the measures generates a new vector of weights $\mathcal{A}$ that was used to select the most competent regressor in MINE-S, to select and combine the regressors in MINE-WS, and to combine all the regressors in MINE-W.

Figures 9, 10, and 11 show the arithmetic mean of the weights over 20 replications per datasets for the MINE-S, MINE-W, and MINE-WS, respectively. These tables also show the mean of the weights per measure (these values are at the bottom of each figure).

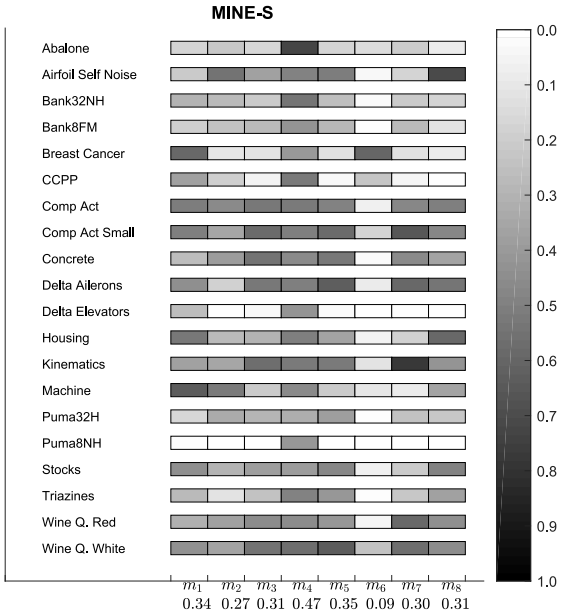


Figure 9 – Mean of the weights of the measures calculated for MINE-S.

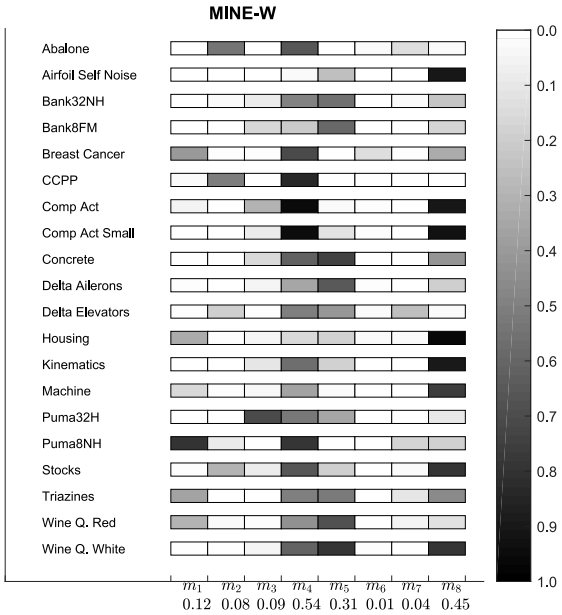


Figure 10 – Mean of the weights of the measures calculated for MINE-W.

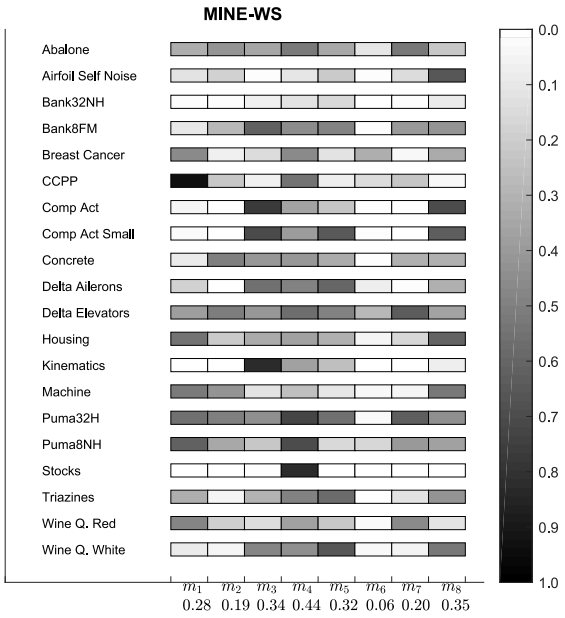


Figure 11 – Mean of the weights of the measures calculated for MINE-WS.

We can observe that the weights of the measures vary depending on the technique under analysis. In MINE-S, the range of the weights is wider than in MINE-W and MINE-WS. For instance, in MINE-W, some weights are zero or close to it. This means that this measure has little or no influence in the decision process, observe m_6, for instance. MINE-WS uses more measures, on average, when compared with MINE-W, but the values of the weights are not as high as in MINE-S.

This analysis shows that the importance of the measures varies from dataset to dataset and also from technique to technique indicating that their combination is more advantageous than using only one.

Table 14 – Mean and standard deviation of the results calculated in 20 replications for the individual measures applied to DS compared against MINE-S. For each dataset, the best result is in bold. Line "Win/Tie/Loss" shows the total of the results. The values are in the scale 10^{-4}.

	Ensemble Size = 90								
Dataset	DS (m_1)	DS (m_2)	DS (m_3)	DS (m_4)	DS (m_5)	DS (m_6)	DS (m_7)	DS (m_8)	MINE-S
Abalone	61.09(2.18)	65.12(2.51)	65.49(3.46)	**58.76(1.13)**	67.43(7.83)	59.40(0.31)	65.49(3.46)	68.24(5.20)	59.53(1.04)
Airfoil Self Noise	99.75(3.94)	10.93(0.89)	11.55(1.66)	34.17(2.19)	11.68(1.73)	58.43(1.16)	11.55(1.66)	17.36(1.11)	**10.75(0.66)**
Bank32NH	115.17(2.90)	119.37(5.11)	120.10(5.14)	**110.32(2.93)**	118.93(3.50)	161.83(1.16)	120.10(5.14)	115.47(2.08)	110.80(3.31)
Bank8FM	15.08(0.45)	13.91(0.34)	13.81(0.34)	**13.31(0.16)**	13.64(0.22)	20.08(0.40)	13.81(0.34)	14.41(0.33)	13.45(0.25)
Breast Cancer	765.38(12.95)	728.61(12.79)	713.54(12.46)	741.22(16.63)	748.31(12.13)	**684.88(9.57)**	713.54(12.46)	774.26(23.38)	711.42(17.19)
CCPP	28.20(0.27)	27.59(0.33)	28.01(0.50)	30.60(0.52)	28.52(0.47)	**22.99(0.11)**	28.01(0.50)	30.21(0.42)	26.35(0.99)
Comp Act	7.04(0.43)	5.86(0.32)	5.91(0.40)	6.69(0.58)	6.01(0.32)	6.95(0.28)	5.91(0.40)	6.29(0.25)	**5.80(0.30)**
Comp Act Small	10.40(1.41)	8.95(1.36)	8.74(1.26)	9.92(0.85)	8.80(0.94)	9.05(0.13)	8.74(1.26)	9.64(1.64)	**7.89(0.11)**
Concrete	78.17(3.74)	35.60(3.48)	36.01(4.85)	60.77(7.63)	38.48(13.35)	83.61(1.74)	36.01(4.85)	46.49(12.05)	**34.89(3.71)**
Delta Ailerons	15.48(0.07)	14.76(0.06)	**14.59(0.04)**	15.20(0.07)	14.69(0.07)	14.94(0.03)	**14.59(0.04)**	15.10(0.07)	14.62(0.06)
Delta Elevators	28.78(2.13)	31.60(5.30)	30.61(2.43)	28.53(0.55)	30.44(1.84)	28.71(0.07)	30.61(2.43)	31.30(2.22)	**28.27(0.17)**
Housing	118.23(8.19)	55.58(4.98)	55.15(6.52)	76.81(9.12)	55.48(4.88)	87.51(3.34)	55.15(6.52)	53.76(6.07)	**51.86(8.07)**
Kinematics	47.29(0.64)	28.92(0.22)	28.45(0.22)	38.66(0.39)	29.44(0.37)	49.32(0.35)	28.45(0.22)	35.96(0.39)	**28.35(0.25)**
Machine	139.30(6.66)	46.94(10.58)	**46.85(10.82)**	87.37(12.22)	48.43(10.37)	53.03(9.34)	**46.85(10.82)**	54.60(10.11)	48.81(10.34)
Puma32H	15.26(0.18)	13.24(0.17)	13.22(0.17)	13.51(0.13)	13.23(0.14)	27.24(0.23)	13.22(0.17)	13.86(0.16)	**12.89(0.23)**
Puma8NH	177.19(1.04)	179.43(1.06)	179.45(1.22)	173.67(0.80)	182.07(1.06)	191.49(0.81)	179.45(1.22)	184.57(1.24)	**173.22(0.85)**
Stock	12.13(0.67)	5.85(0.29)	5.85(0.34)	7.93(0.60)	**5.78(0.28)**	7.22(0.12)	5.85(0.34)	6.18(0.29)	5.88(0.24)
Triazines	224.83(7.45)	210.51(8.11)	**201.72(7.83)**	216.85(8.56)	210.57(10.13)	210.25(5.23)	**201.72(7.83)**	208.02(9.27)	210.19(12.61)
Wine Q. Red	170.56(1.52)	165.72(1.03)	**164.14(1.07)**	171.32(1.86)	168.27(1.54)	170.22(0.82)	**164.14(1.07)**	170.03(1.23)	164.70(1.39)
Wine Q. White	150.38(1.96)	148.97(8.65)	149.48(8.00)	148.13(3.08)	147.72(5.95)	141.29(0.60)	149.48(8.00)	151.43(5.73)	**138.09(2.41)**
Win/Tie/Loss	0/0/20	0/0/20	0/4/16	3/0/17	1/0/19	2/0/18	0/4/16	0/0/20	**10/0/10**

Table 15 – Mean and standard deviation of the results calculated in 20 replications for the individual measures applied to DW compared against MINE-W. For each dataset, the best result is in bold. Line "Win/Tie/Loss" shows the total of the results. The values are in the scale 10^{-4}.

	Ensemble Size = 90								
Dataset	DW (m_1)	DW (m_2)	DW (m_3)	DW (m_4)	DW (m_5)	DW (m_6)	DW (m_7)	DW (m_8)	MINE-W
Abalone	54.68(0.24)	54.50(0.19)	54.54(0.19)	56.79(0.94)	54.62(0.17)	**54.31(0.14)**	54.50(0.18)	58.71(2.31)	54.50(0.18)
Airfoil Self Noise	42.71(3.10)	17.05(0.37)	14.54(0.35)	26.97(1.16)	13.52(0.34)	22.17(0.42)	17.34(0.36)	15.00(0.64)	**11.69(0.31)**
Bank32NH	89.18(0.27)	88.70(0.29)	**88.66(0.28)**	99.37(1.71)	88.74(0.29)	92.12(0.31)	88.69(0.29)	98.88(1.26)	88.78(0.33)
Bank8FM	12.27(0.02)	12.24(0.02)	12.22(0.02)	12.79(0.09)	**12.21(0.03)**	12.32(0.03)	12.24(0.02)	12.80(0.09)	12.23(0.03)
Breast Cancer	728.31(9.38)	714.75(6.50)	714.05(6.59)	730.41(10.28)	716.52(6.58)	**713.02(6.43)**	714.82(6.55)	720.89(10.33)	718.09(8.42)
CCPP	23.41(0.10)	**22.95(0.12)**	22.98(0.15)	26.46(0.35)	23.14(0.19)	23.04(0.11)	23.09(0.13)	27.18(0.43)	**22.95(0.12)**
Comp Act	5.37(0.02)	5.31(0.02)	5.28(0.02)	6.09(0.49)	5.30(0.02)	5.33(0.02)	5.31(0.02)	5.79(0.13)	**5.26(0.02)**
Comp Act Small	8.49(0.02)	8.27(0.03)	8.17(0.04)	9.01(0.37)	8.18(0.04)	8.29(0.02)	8.28(0.03)	9.07(1.42)	**8.08(0.03)**
Concrete	40.70(0.81)	33.55(0.94)	31.50(0.95)	52.01(10.76)	30.84(1.07)	37.33(0.71)	34.13(0.91)	38.00(3.28)	**30.69(1.21)**
Delta Ailerons	15.04(0.02)	15.02(0.02)	**15.00(0.02)**	15.10(0.06)	**15.00(0.02)**	15.01(0.02)	15.02(0.02)	15.09(0.03)	**15.00(0.02)**
Delta Elevators	27.35(0.02)	27.31(0.03)	27.31(0.03)	27.93(0.33)	27.33(0.03)	**27.30(0.03)**	27.31(0.03)	28.34(0.72)	27.32(0.03)
Housing	57.71(2.91)	49.83(2.74)	49.38(3.19)	63.64(5.13)	49.17(3.42)	49.81(2.22)	49.79(2.85)	50.53(3.57)	**47.08(2.80)**
Kinematics	33.30(0.11)	31.59(0.09)	30.50(0.09)	35.13(0.33)	30.54(0.09)	32.56(0.10)	31.64(0.09)	32.84(0.24)	**29.86(0.10)**
Machine	93.93(6.24)	68.09(7.37)	58.23(9.04)	85.20(8.43)	**55.41(10.02)**	72.22(6.36)	66.70(7.63)	56.76(9.95)	70.13(6.39)
Puma32H	10.72(0.03)	10.67(0.02)	**10.65(0.02)**	12.13(0.08)	10.66(0.03)	10.80(0.03)	10.67(0.03)	12.02(0.11)	**10.65(0.03)**
Puma8NH	166.99(0.16)	166.95(0.17)	166.98(0.17)	170.58(0.63)	167.08(0.17)	167.01(0.16)	**166.94(0.16)**	170.63(0.70)	167.00(0.16)
Stock	6.10(0.19)	4.95(0.15)	4.80(0.16)	5.17(0.16)	**4.76(0.15)**	5.43(0.13)	4.97(0.15)	5.95(0.28)	4.85(0.16)
Triazines	210.81(3.95)	205.93(5.09)	204.50(5.24)	210.12(6.41)	**203.45(5.28)**	206.36(4.85)	205.23(5.05)	206.82(6.92)	206.04(5.00)
Wine Q. Red	164.79(0.61)	163.96(0.62)	163.67(0.65)	167.97(1.24)	**163.48(0.65)**	164.07(0.64)	163.97(0.61)	166.12(0.64)	163.91(0.65)
Wine Q. White	135.57(0.45)	132.19(0.47)	131.39(0.53)	140.40(1.51)	131.23(0.55)	131.19(0.28)	132.27(0.46)	140.03(4.23)	**130.68(0.42)**
Win/Tie/Loss	0/0/20	0/1/19	1/2/17	0/0/20	5/1/14	3/0/17	1/0/19	0/0/20	**7/3/10**

Tables 14, 15, and 16 present the results of the DS, DW, and DWS techniques when performed varying each of the measures of competence studied in this work. The tables compare the results with the MINE-S, MINE-W and MINE-WS, respectively. In the tables is possible to notice that the techniques MINE-S, MINE-W and MINE-WS have better performance when compared to the techniques of the literature.

In addition, we can observe that MINE techniques present better results for most datasets when compared individually with each of the measures of competence.

Table 16 – Mean and standard deviation of the results calculated in 20 replications for the individual measures applied to DWS compared against MINE-WS. For each dataset, the best result is in bold. Line "Win/Tie/Loss" shows the total of the results. The values are in the scale 10^{-4}.

	Ensemble Size = 90								
Dataset	DWS (m_1)	DWS (m_2)	DWS (m_3)	DWS (m_4)	DWS (m_5)	DWS (m_6)	DWS (m_7)	DWS (m_8)	MINE-WS
Abalone	54.92(0.27)	54.87(0.30)	56.33(0.98)	56.80(0.94)	56.28(0.46)	54.80(0.17)	**54.78(0.47)**	59.81(2.28)	55.13(0.61)
Airfoil Self Noise	55.05(3.30)	13.13(0.47)	13.97(0.34)	26.97(1.16)	13.04(0.34)	27.55(0.77)	13.17(0.45)	14.90(0.64)	**11.92(0.48)**
Bank32NH	91.18(0.35)	92.30(1.32)	91.22(0.74)	99.37(1.71)	**89.78(0.63)**	96.61(0.44)	93.06(1.73)	99.33(1.28)	89.94(0.57)
Bank8FM	12.35(0.07)	**12.25(0.03)**	12.43(0.16)	12.79(0.09)	12.37(0.05)	13.11(0.14)	12.27(0.07)	12.96(0.10)	12.36(0.16)
Breast Cancer	730.65(8.95)	**714.75(6.50)**	714.84(8.21)	731.28(10.38)	720.38(9.91)	714.94(7.95)	714.82(6.55)	741.95(16.70)	718.71(10.24)
CCPP	23.83(0.10)	23.67(0.24)	23.82(0.23)	26.46(0.35)	24.06(0.22)	**22.92(0.10)**	23.84(0.29)	27.53(0.41)	23.02(0.18)
Comp Act	5.57(0.08)	5.37(0.07)	5.24(0.06)	6.10(0.49)	5.25(0.03)	5.60(0.16)	5.35(0.04)	5.75(0.13)	**5.20(0.03)**
Comp Act Small	8.71(0.04)	8.21(0.11)	7.89(0.05)	9.01(0.37)	7.92(0.06)	8.28(0.06)	8.23(0.05)	8.93(1.42)	**7.87(0.09)**
Concrete	47.90(1.43)	31.17(1.26)	**30.45(1.02)**	52.01(10.76)	30.58(1.09)	44.69(1.04)	31.11(1.38)	37.89(3.28)	30.88(1.13)
Delta Ailerons	15.05(0.04)	15.02(0.02)	**14.99(0.02)**	15.10(0.05)	15.00(0.03)	15.04(0.04)	15.02(0.02)	15.12(0.04)	15.02(0.03)
Delta Elevators	27.41(0.04)	27.37(0.05)	27.72(0.50)	27.95(0.32)	27.72(0.11)	**27.36(0.04)**	27.51(0.53)	28.76(0.73)	27.41(0.17)
Housing	68.25(4.29)	50.60(3.50)	46.36(3.56)	63.67(5.15)	46.50(3.69)	56.61(3.06)	49.98(3.21)	48.06(3.92)	**45.28(3.46)**
Kinematics	36.48(0.30)	30.36(0.14)	**27.63(0.14)**	35.13(0.33)	28.44(0.13)	34.47(0.21)	30.67(0.12)	32.47(0.24)	27.68(0.15)
Machine	123.87(6.46)	**52.10(10.17)**	54.95(9.36)	85.13(8.43)	53.18(9.87)	62.17(9.12)	52.34(9.91)	56.09(10.16)	56.24(10.73)
Puma32H	10.73(0.03)	10.83(0.05)	11.01(0.04)	12.13(0.08)	10.73(0.03)	11.09(0.10)	10.78(0.04)	12.04(0.11)	**10.69(0.03)**
Puma8NH	168.84(0.42)	166.96(0.17)	167.99(0.36)	170.59(0.63)	169.68(0.74)	168.29(0.31)	**166.95(0.17)**	173.02(0.91)	167.02(0.15)
Stock	7.62(0.28)	4.82(0.18)	4.75(0.16)	5.17(0.16)	**4.72(0.16)**	5.76(0.13)	4.84(0.16)	5.95(0.28)	4.74(0.17)
Triazines	212.01(3.86)	207.88(5.72)	**203.60(7.52)**	210.13(6.43)	205.04(7.53)	206.76(5.05)	206.68(5.82)	205.79(7.27)	210.11(7.24)
Wine Q. Red	166.45(0.73)	163.62(0.71)	164.22(0.82)	168.27(1.20)	164.73(0.96)	164.77(0.82)	**163.58(0.74)**	166.93(1.13)	163.73(0.69)
Wine Q. White	136.16(0.47)	132.76(0.88)	131.02(0.76)	140.45(1.51)	130.88(0.83)	130.82(0.45)	133.19(1.13)	139.66(4.17)	**130.11(0.76)**
Win/Tie/Loss	0/0/20	3/0/17	4/0/16	0/0/20	2/0/18	2/0/18	3/0/17	0/0/20	**6/0/14**

4.5 CONCLUSION

This paper proposed the MINE framework for dynamic regressor selection that aims to select and combine the best regressors per query pattern from a homogeneous ensemble. MINE uses information extracted from the region of competence as a criterion to select the competent regressors. Instead of using only one measure from the region of competence, knowing that no single measure is the best for any task, the proposal combines a set of measures to better select the competent regressors.

Three algorithms were presented, and their difference resides in how many regressors are selected from the ensemble. MINE-S selects only the most competent regressor while MINE-W combines all the regressors. MINE-WS, in turn, selects a subset of the regressors. Experiments showed that the MINE techniques presented in this work perform better compared to state-of-the-art DRS techniques, and classical combination techniques, such as Mean and Median. Among the MINE family, a highlight to MINE-W because it performed similarly to MINE-WS but required fewer regressors in the combination phase.

The results showed that the combination of multiple measures extracted from the region of competence generates more accurate results than using only a single measure. We also observed that some measures received zero-weight for some datasets. In other words, the set of measures is problem-dependent and can be selected instead of using all of them. The proposed framework is modular and can be evaluated using more significant set measures. Also, as presented in (MENDES-MOREIRA et al., 2009; MENDES-MOREIRA et al., 2015), the size of the region of competence is problem-dependent and for better error rates a study must be done to find the ideal neighborhood size for each dataset.

For future work, we intend to evaluate different optimization algorithms in the Optimization Phase, such as PSO (Particle Swarm Optimization) (EIBEN; SMITH, 2003), and Differential Evolution (EIBEN; SMITH, 2003). We also intend to analyze some parameters of the framework, such as the size of the region of competence.

5 CONCLUSION

This thesis brought contributions to Dynamic Regressor Selection (DRS). First we have identified and adapted some measures that can be used to measure the competence of the regressors in the region of competence and the experiments show that the performance of these measures is problem-dependent. Then, we proposed the MINE (Meta INtEgration), a framework that combines the measures previously identified for the selection and fusion of the regressors from an initially generated homogeneous ensemble. The framework can operate in three different scenarios and the proposed framework improves the estimation performance when compared against DRS techniques and well-known static techniques.

We did a survey of eight measures found in works of regression problems and adapted them to the measurement of the competence of the regressors. To the best of our knowledge, seven of these measures are adapted for the first time to this task, and they capture different information, such as weighted error, variance, and similarity among the regressors. These eight competence measures are evaluated using 15 regression problems from different data repositories and three literature algorithms: DS, DW, and DWS. It is possible to conclude that the competence measure used to select the regressors is problem-dependent. DRS techniques perform better when compared to a single individual regressor or to classic statistical techniques such as Mean and Median. Another situation is that the reduction in the variance achieved by weighted mean can explain why DW and DWS are better than DS (TSYMBAL; PECHENIZKIY; CUNNINGHAM, 2006).

We proposed the MINE framework for dynamic regressor selection (DRS). MINE aims to select and combine the best regressors per query pattern from a homogeneous ensemble. The framework uses the combination of a set of measures extracted from the region of competence as a criterion to select the competent regressors. Three algorithms were presented, and their difference resides in how many regressors are selected from the ensemble. The algorithms are: (i) MINE-Selection (MINE-S): selects a single regressor given a test pattern; (ii) MINE-Weighting (MINE-W): all ensemble regressors are combined by the weighted mean; and, (iii) MINE-Weighting with Selection (MINE-WS): a subset of the ensemble is dynamically selected per test pattern. Also, this chapter presents a robust study of homogeneous ensembles used with these three algorithms. Experiments shown that the MINE techniques perform better compared to state-of-the-art DRS techniques, and classical combination techniques, such as Mean and Median. Among the MINE family, a highlight to MINE-W because it performed similarly to MINE-WS but required fewer regressors in the combination phase. The results have shown that the combination of multiple measures extracted from the region of competence generates more accurate results than using only a single measure. We also observed that the set of measures is problem-dependent and can be selected instead of using all of them.

Overall, this thesis presented a study with eight measures to compute the behavior of the regressors in the region of competence, comparing them with DRS techniques found in the literature. Also, it was presented a new framework to DRS that use a combination of some measures, instead of, the state-of-the-art DRS techniques that use only a single measure. Besides that, the framework can be used in three different scenarios, selecting just one regressor, combining all of them, or selecting and combining a subset of regressors from the original ensemble.

5.1 FUTURE WORKS

The findings of this thesis suggest the following points for future works:

- A new solution to select, for each regression problem, the best measure to be used. In MINE framework, the combination of the measures has shown significant improvement in the performance when compared to DRS techniques. Despite this, it can be seen in the experiments at the end of Chapter 4 that the combination of the measures is not the best solution for the majority of the datasets. To select dynamically the measures, for each regression problem, can present better results.

- Evaluate different optimization algorithms in the Optimization Phase of the MINE. There are many optimization algorithms in machine learning literature and only Genetic Algorithm (GA) were used in this thesis. In addition, little tuning work was done at GA. It is understood that in the MINE optimization phase, other optimization algorithms can be used as PSO (Particle Swarm Optimization) (EIBEN; SMITH, 2003), and Differential Evolution (EIBEN; SMITH, 2003), with their adjusted parameters, in an attempt to improve the performance of the MINE.

- Analyze the region of competence. As presented in (MENDES-MOREIRA et al., 2009; MENDES-MOREIRA et al., 2015), the size of the region of competence is problem-dependent and for better error rates a study must be done to find the ideal neighborhood size for each problem.

- Work with heterogeneous ensembles. This thesis presented a robust study with homogeneous ensembles. A study with heterogeneous ensembles is necessary. Heterogeneous ensembles are composed by trained regressors with different learning algorithms and because of this, they can present greater diversity than homogeneous ensembles.

- Adapt the framework to address time series problems. Time series can be classified as a regression problems. The measures used in the MINE can be used to verify the behavior of the regressors trained with data representing time series.

REFERENCES

ADHIKARI, R. A neural network based linear ensemble framework for time series forecasting. *Neurocomputing*, v. 157, p. 231–242, 2015.

ALDAVE, R.; DUSSAULT, J.-P. Systematic Ensemble Learning for Regression. *Journal of Machine Learning*, mar 2014. Disponível em: <http://arxiv.org/abs/1403.7267>.

AVNIMELECH, R.; INTRATOR, N. Boosted mixture of experts: An ensemble learning scheme. *Neural Computation*, v. 11, n. 2, p. 483–497, Feb 1999.

AVNIMELECH, R.; INTRATOR, N. Boosting regression estimators. *Neural computation*, v. 11, p. 499–520, 03 1999.

BALUJA, S. *Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning*. Pittsburgh, PA, 1994.

BORRA, S.; CIACCIO, A. D. Improving nonparametric regression methods by bagging and boosting. *Computational Statistics and Data Analysis*, v. 38, n. 4, p. 407–420, 2002. ISSN 0167-9473. Nonlinear Methods and Data Mining.

BREIMAN, L. Bagging predictors. *Machine Learning*, Kluwer Academic Publishers, v. 24, n. 2, p. 123–140, 1996.

BREIMAN, L. Heuristics of instability and stabilization in model selection. *Annals of Statistics*, The Institute of Mathematical Statistics, v. 24, n. 6, p. 2350–2383, 12 1996.

BREIMAN, L. Stacked regressions. *Machine Learning*, v. 24, n. 1, p. 49–64, 1996.

BREIMAN, L. Random forests. *Machine Learning*, v. 45, n. 1, p. 5–32, 2001.

BREIMAN, L.; FRIEDMAN, J. H.; OLSHEN, R. A.; STONE, C. J. *Classification and Regression Trees*. [S.l.: s.n.], 1984. v. 19. 368 p. ISBN 0412048418.

BRITTO, A. S.; SABOURIN, R.; OLIVEIRA, L. S. O. Dynamic selection of classifiers - a comprehensive review. *Pattern Recognition*, v. 47, n. 11, p. 3665–3680, 2014.

BROWN, G.; WYATT, J.; HARRIS, R.; YAO, X. Diversity creation methods: a survey and categorisation. *Information Fusion*, v. 6, n. 1, p. 5 – 20, 2005. ISSN 1566-2535. Diversity in Multiple Classifier Systems.

BUHLMANN, P. *Bagging, Boosting and Ensemble Methods*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. 985–1022 p.

CARUANA, R.; NICULESCU-MIZIL, A.; CREW, G.; KSIKES, A. Ensemble selection from libraries of models. In: *Proceedings of the Twenty-first International Conference on Machine Learning - ICML*. [S.l.]: ACM, 2004. p. 18–.

CAVALCANTI, G. D.; OLIVEIRA, L. S.; MOURA, T. J.; CARVALHO, G. V. Combining diversity measures for ensemble pruning. *Pattern Recognition Letters*, v. 74, p. 38 – 45, 2016. ISSN 0167-8655.

CHANDRAHASAN, R. K.; Y, A. C.; SRIDHAR, U. R.; L, A. An empirical comparison of boosting and bagging algorithms. In: *International Journal of Computer Science and Information Security - IJCSIS*. [S.l.]: SCRIBD, 2011. p. 147–152.

COELHO, G. P.; ZUBEN, F. J. V. The influence of the pool of candidates on the performance of selection and combination techniques in ensembles. In: *The 2006 IEEE International Joint Conference on Neural Network Proceedings*. [S.l.: s.n.], 2006. p. 5132–5139. ISSN 2161-4393.

CORTEZ, P.; CERDEIRA, A.; ALMEIDA, F.; MATOS, T.; REIS, J. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, v. 47, n. 4, p. 547–553, 2009.

CRUZ, R. M.; SABOURIN, R.; CAVALCANTI, G. D. Meta-des.h: A dynamic ensemble selection technique using meta-learning and a dynamic weighting approach. In: *Proceedings of the International Joint Conference on Neural Networks*. [S.l.: s.n.], 2016. p. 216–221.

CRUZ, R. M.; SABOURIN, R.; CAVALCANTI, G. D. Meta-des.oracle: Meta-learning and feature selection for dynamic ensemble selection. *Information Fusion*, v. 38, p. 84–103, 2017.

CRUZ, R. M.; SABOURIN, R.; CAVALCANTI, G. D. Dynamic classifier selection: Recent advances and perspectives. *Information Fusion*, Elsevier, v. 41, p. 195–216, 2018.

CRUZ, R. M. C.; SABOURIN, R.; CAVALCANTI, G. D.; REN, T. I. Meta-des: A dynamic ensemble selection framework using meta-learning. *Pattern Recognition*, v. 48, n. 5, p. 1925–1935, 2015.

DEMŠAR, J. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, v. 7, p. 1–30, 2006.

DIETTERICH, T. G. Machine-learning research – four current directions. *AI MAGAZINE*, v. 18, p. 97–136, 1997.

DOMENICONI, C.; YAN, B. Nearest neighbor ensemble. In: *International Conference, Pattern Recognition*. [S.l.: s.n.], 2004. v. 1, p. 228–231.

DOMINGOS, P. Why does bagging work? a bayesian account and its implications. In: *International Conference on Knowledge Discovery and Data Mining*. [S.l.]: AAAI Press, 1997. p. 155–158.

DRUCKER, H. Improving regressors using boosting techniques. In: *Proceedings of the 14th International Conference on Machine Learning (ICML'97)*. [S.l.]: Morgan Kaufmann Publishers, 1997. p. 107–115.

EIBEN, A. E.; SMITH, J. E. *Introduction to Evolutionary Computing*. [S.l.]: Springer, 2003.

FREUND, Y.; SCHAPIRE, R. E. Experiments with a new boosting algorithm. In: *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1996. (ICML'96), p. 148–156. ISBN 1-55860-419-7.

FREUND, Y.; SCHAPIRE, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, v. 55, n. 1, p. 119–139, 1997.

FRIEDMAN, J. Local learning based on recursive covering. In: . [S.l.: s.n.], 1996.

FRIEDMAN, J. H. Multivariate adaptive regression splines. *Annals of Statistics*, The Institute of Mathematical Statistics, v. 19, n. 1, p. 1–67, 03 1991.

FRIEDMAN, J. H.; STUETZLE, W. Projection pursuit regression. *Journal of the American Statistical Association*, Taylor Francis, v. 76, n. 376, p. 817–823, 1981.

GARCÍA-PEDRAJAS, N.; HERVÁS-MARTÍNEZ, C.; ORTIZ-BOYER, D. Cooperative coevolution of artificial neural network ensembles for pattern classification. *IEEE Transactions on Evolutionary Computation*, v. 9, n. 3, p. 271–302, 2005.

GIACINTO, G.; ROLI, F. Methods for dynamic classifier selection. In: *Proceedings of International Conference on Image Analysis and Processing*. [S.l.: s.n.], 1999. p. 659–664.

GIACINTO, G.; ROLI, F. Dynamic classifier selection based on multiple classifier behaviour. *Pattern Recognition*, v. 34, n. 9, p. 1879–1881, 2001.

GLOVER, F.; LAGUNA, M. *Tabu Search*. oston, MA: Springer US, 1998. 2093–2229 p. ISBN 978-1-4613-0303-9.

GRANITTO, P.; VERDES, P.; CECCATTO, H. Neural network ensembles: evaluation of aggregation algorithms. *Artificial Intelligence*, v. 163, n. 2, p. 139–162, 2005. ISSN 0004-3702.

HERNANDEZ-LOBATO, D.; MARTINEZ-MUNOZ, G.; SUAREZ, A. Pruning in ordered regression bagging ensembles. In: *The 2006 IEEE International Joint Conference on Neural Network Proceedings*. [S.l.: s.n.], 2006. p. 1266–1273.

HIRST, J. D.; KING, R. D.; STERNBERG, M. J. E. Quantitative structure-activity relationships by neural networks and inductive logic programming. ii. the inhibition of dihydrofolate reductase by triazines. *Journal of Computer-Aided Molecular Design*, v. 8, n. 4, p. 421–432, 1994.

HIRST, J. D.; KING, R. D.; STERNBERG, M. J. E. Comparison of artificial intelligence methods for modeling pharmaceutical qsars. *Applied Artificial Intelligence*, Taylor & Francis Group, v. 9, n. 2, p. 213–233, 1995.

HO, T. K. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 20, n. 8, p. 832–844, 1998.

ISLAM, M.; MURASE, K. A constructive algorithm for training cooperative neural network ensembles. *IEEE Transactions on Neural Networks*, v. 14, n. 4, p. 820–834, 2003. ISSN 1045-9227.

JOLLIFFE, I. *Principal Component Analysis*. 2. ed. [S.l.]: Springer, 2002. ISBN 9780387954424.

KIM, H.-C.; PANG, S.; JE, H.-M.; KIM, D.; BANG, S.-Y. Support vector machine ensemble with bagging. In: LEE, S.-W.; VERRI, A. (Ed.). *Pattern Recognition with Support Vector Machines*. [S.l.]: Springer Berlin Heidelberg, 2002. p. 397–408.

KO, A. H.; SABOURIN, R.; BRITTO, A. S. From dynamic classifier selection to dynamic ensemble selection. *Pattern Recognition*, v. 41, n. 5, p. 1718–1731, 2008.

KOLEN, J. F.; POLLACK, J. B. Back propagation is sensitive to initial conditions. In: *Proceedings of the 1990 Conference on Advances in Neural Information Processing Systems 3*. [S.l.]: Morgan Kaufmann Publishers Inc., 1990. p. 860–867.

KROGH, A.; VEDELSBY, J. Neural network ensembles, cross validation and active learning. In: *Proceedings of the 7th International Conference on Neural Information Processing Systems*. [S.l.]: MIT Press, 1994. (NIPS'94), p. 231–238.

KUNCHEVA, L.; RODRíGUEZ, J. Classifier ensembles with a random linear oracle. *Knowledge and Data Engineering, IEEE Transactions on*, v. 19, p. 500–508, 05 2007.

KUNCHEVA, L. I.; JAIN, L. C. Designing classifier fusion systems by genetic algorithms. *IEEE Transactions on Evolutionary Computation*, v. 4, n. 4, p. 327–336, Nov 2000. ISSN 1089-778X.

LAZAREVIC, A.; OBRADOVIC, Z. Effective pruning of neural network classifier ensembles. In: *IJCNN'01. International Joint Conference on Neural Networks. Proceedings*. [S.l.: s.n.], 2001. v. 2, p. 796–801. ISSN 1098-7576.

LIU, Y.; YAO, X. Ensemble learning via negative correlation. *Neural Networks*, v. 12, n. 10, p. 1399–1404, 1999. ISSN 0893-6080.

LLOYD, S. P. Least squares quantization in pcm. *IEEE Trans. Inf. Theor.*, v. 28, n. 2, p. 129–137, set. 1982.

MARGINEANTU, D. D.; DIETTERICH, T. G. Pruning adaptive boosting. In: *Proceedings of the Fourteenth International Conference on Machine Learning*. [S.l.]: Morgan Kaufmann Publishers Inc., 1997. (ICML '97), p. 211–218. ISBN 1-55860-486-3.

MARTíNEZ-MUñOZ, G.; SUáREZ, A. Pruning in ordered bagging ensembles. In: *Proceedings of the 23rd International Conference on Machine Learning*. [S.l.]: ACM, 2006. (ICML '06), p. 609–616.

MARTíNEZ-MUñOZ, G.; SUáREZ, A. Using boosting to prune bagging ensembles. *Pattern Recognition Letters*, v. 28, n. 1, p. 156 – 165, 2007.

MENDES-MOREIRA, J.; JORGE, A. M.; SOARES, C.; SOUSA, J. F. D. Ensemble learning: A study on different variants of the dynamic selection approach. In: *Machine Learning and Data Mining in Pattern Recognition*. [S.l.: s.n.], 2009. v. 5632, p. 191–205.

MENDES-MOREIRA, J.; JORGE, A. M.; SOUSA, J. F. de; SOARES, C. Improving the accuracy of long-term travel time prediction using heterogeneous ensembles. *Neurocomputing*, v. 150, p. 428 – 439, 2015.

MENDES-MOREIRA, J.; SOARES, C.; JORGE, A. M.; SOUSA, J. F. D. Ensemble approaches for regression: A survey. *ACM Computing Surveys*, ACM, v. 45, n. 1, p. 10:1–10:40, 2012.

MENDES-MOREIRA., J.; SOUSA, J. F.; JORGE, A. M.; SOARES, C. An ensemble regression approach for bus trip time prediction. In: *Proceedings of the EWGT2006 joint conferences*. [S.l.: s.n.], 2006. p. 317–321.

MERZ, C. J. Dynamical selection of learning algorithms. In: *International Workshop on Artificial Intelligence and Statistics*. [S.l.]: Springer, 1996.

MERZ, C. J. *Classification and regression by combining models*. Tese (Doutorado) — University of California Irvine, 1998.

MOLINA, L. C.; BELANCHE, L.; NEBOT, A. Feature selection algorithms: a survey and experimental evaluation. In: *2002 IEEE International Conference on Data Mining, 2002. Proceedings.* [S.l.: s.n.], 2002. p. 306–313.

MOURA, T. J. M.; CAVALCANTI, G. D. C.; OLIVEIRA, L. S. *Evaluating Competence Measures for Dynamic Regressor Selection*. 2019.

OLIVEIRA, D. V.; CAVALCANTI, G. D.; SABOURIN, R. Online pruning of base classifiers for dynamic ensemble selection. *Pattern Recognition*, v. 72, p. 44 – 58, 2017. ISSN 0031-3203.

OPITZ, D. W. Feature selection for ensembles. In: *Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference*. [S.l.]: American Association for Artificial Intelligence, 1999. p. 379–384.

OPITZ, D. W.; SHAVLIK, J. W. Generating accurate and diverse members of a neural-network ensemble. In: *Advances in Neural Information Processing Systems*. [S.l.]: MIT Press, 1996. p. 535–541.

ORTIZ-BOYER, D.; HERVáS-MARTíNEZ, C.; GARCíA-PEDRAJAS, N. Cixl2: A crossover operator for evolutionary algorithms based on population features. *Journal of Artificial Intelligence Research*, AI Access Foundation, v. 24, n. 1, p. 1–48, 2005.

PARTALAS, I.; TSOUMAKAS, G.; HATZIKOS, E. V.; VLAHAVAS, I. Greedy regression ensemble selection: Theory and an application to water quality prediction. *Information Sciences*, v. 178, n. 20, p. 3867–3879, 2008.

PERRONE, M. P.; COOPER, L. N. When networks disagree: Ensemble methods for hybrid neural networks. In: *Neural Networks for Speech and Image Processing*. [S.l.: s.n.], 1993. p. 123–140.

PRAMANIK, S.; CHOWDHURY, U.; PRAM, B. K.; HUDA, N. A comparative study of bagging, boosting and c4.5: The recent improvements in decision tree learning algorithm. *Asian Journal of Information Technology*, v. 9, p. 300–306, 06 2010.

RÄTSCH, G.; DEMIRIZ, A.; BENNETT, K. P. Sparse regression ensembles in infinite and finite hypothesis spaces. *Machine Learning*, v. 48, n. 1, p. 189–218, 2002.

RODRÍGUEZ, J. J.; KUNCHEVA, L. I.; ALONSO, C. J. Rotation forest: A new classifier ensemble method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 28, n. 10, p. 1619–1630, 2006.

ROONEY, N.; PATTERSON, D. A weighted combination of stacking and dynamic integration. *Pattern Recognition*, v. 40, n. 4, p. 1385–1388, 2007.

ROONEY, N.; PATTERSON, D.; ANAND, S.; TSYMBAL, A. Dynamic integration of regression models. *Proceedings of the International Workshop on Multiple Classifier Systems*, v. 3077, p. 164–173, 2004.

ROSEN, B. E. Ensemble learning using decorrelated neural networks. *Connection Science*, Taylor Francis, v. 8, n. 3-4, p. 373–384, 1996.

RUTA, D.; GABRYS, B. Application of the evolutionary algorithms for classifier selection in multiple classifier systems with majority voting. In: *Proceedings of the Second International Workshop on Multiple Classifier Systems*. [S.l.]: Springer-Verlag, 2001. (MCS '01), p. 399–408. ISBN 3-540-42284-6.

SANTANA, A.; SOARES, R. G. F.; CANUTO, A. M. P.; SOUTO, M. C. P. d. A dynamic classifier selection method to build ensembles using accuracy and diversity. In: *Proceedings of the Ninth Brazilian Symposium on Neural Networks*. [S.l.: s.n.], 2006. p. 36–41.

SANTOS, E. M. D.; SABOURIN, R.; MAUPIN, P. A dynamic overproduce-and-choose strategy for the selection of classifier ensembles. *Pattern Recognition*, v. 41, n. 10, p. 2993–3009, 2008.

SERGIO, A. T.; LIMA, T. P. F. de; LUDERMIR, T. B. Dynamic selection of forecast combiners. *Neurocomputing*, v. 218, p. 37–50, 2016.

SHRESTHA, D. L.; SOLOMATINE, D. P. Experiments with adaboost.rt, an improved boosting scheme for regression. *Neural computation*, v. 18, n. 7, p. 1678–1710, 2006.

SINGH, S.; SINGH, M. A dynamic classifier selection and combination approach to image region labelling. *Signal Processing: Image Communication*, v. 20, n. 3, p. 219–231, 2005.

TAMON, C.; XIANG, J. On the boosting pruning problem. In: MÁNTARAS, R. López de; PLAZA, E. (Ed.). *Machine Learning: ECML 2000*. Heidelberg Berlin: Springer Berlin Heidelberg, 2000. p. 404–412.

THOMPSON, S. K.; SEBER, G. A. F. *Adaptive Sampling*. [S.l.]: Wiley-Interscience, 1996.

TRESP, V.; TANIGUCHI, M. Combining estimators using non-constant weighting functions. In: *Proceedings of the 7th International Conference on Neural Information Processing Systems*. [S.l.: s.n.], 1995. p. 419–426.

TSYMBAL, A.; PECHENIZKIY, M.; CUNNINGHAM, P. Dynamic integration with random forests. Berlin, Heidelberg, p. 801–808, 2006.

TSYMBAL, A.; PUURONEN, S. Bagging and boosting with dynamic integration of classifiers. *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, p. 116–125, 2000.

TüFEKCI, P. Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods. *International Journal of Electrical Power and Energy Systems*, v. 60, p. 126–140, 2014.

UEDA, N.; NAKANO, R. Generalization error of ensemble estimators. In: *Proceedings of International Conference on Neural Networks (ICNN'96)*. [S.l.: s.n.], 1996. v. 1, p. 90–95.

VERIKAS, A.; LIPNICKAS, A.; MALMQVIST, K.; BACAUSKIENE, M.; GELZINIS, A. Soft combination of neural classifiers: A comparative study. *Pattern Recogn. Lett.*, Elsevier Science Inc., v. 20, n. 4, p. 429–444, 1999.

WOLPERT, D. H. Stacked generalization. *Neural Networks*, v. 5, n. 2, p. 241–259, 1992.

WOLPERT, D. H. The lack of a priori distinctions between learning algorithms. *Neural Computation*, v. 8, n. 7, p. 1341–1390, 1996.

WOLPERT, D. H.; MACREADY, W. G. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, v. 1, n. 1, p. 67–82, 1997.

WOODS, K.; KEGELMEYER, W. P.; BOWYER, K. Combination of multiple classifiers using local accuracy estimates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 19, n. 4, p. 405–410, 1997.

YANKOV, D.; DECOSTE, D.; KEOGH, E. Ensembles of nearest neighbor forecasts. In: *Machine Learning: ECML 2006*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. p. 545–556.

YEH, I.-C. Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete Research*, v. 28, n. 12, p. 1797–1808, 1998.

YU, Y.; ZHOU, Z.; TING, K. M. Cocktail ensemble for regression. In: *Seventh IEEE International Conference on Data Mining (ICDM 2007)*. [S.l.: s.n.], 2007. p. 721–726.

ZEMEL, R. S.; PITASSI, T. *Advances in NeuralInformation Processing Systems*. [S.l.]: MIT Press, 2001. v. 13. 696–702 p.

ZHANG, C.-X.; ZHANG, J.-S.; WANG, G.-W. An empirical study of using rotation forest to improve regressors. *Applied Mathematics and Computation*, v. 195, n. 2, p. 618–629, 2008.

ZHOU, Z.-H. *Ensemble Methods: Foundations and Algorithms*. 1st. ed. [S.l.]: Chapman & Hall/CRC, 2012. ISBN 1439830037, 9781439830031.

ZHOU, Z.-H.; WU, J.; TANG, W. Ensembling neural networks: Many could be better than all. *Artificial Intelligence*, v. 137, n. 1, p. 239–263, 2002. ISSN 0004-3702.

6 APPENDIX

6.1 COMPARING MINE TECHNIQUES

According to (MENDES-MOREIRA et al., 2009), combine more than one model is better than the selection of just one. This can be explained by the reduction in the variance achieved by averaging of the predictions of the regressors for the test pattern.

So, looking the results, we find the same conclusion as (MENDES-MOREIRA et al., 2009), by combining more than one regressor from the ensemble (MINE-W or MINE-WS) using weighted mean, we achieve better error rates than the MINE-S. We can see this by looking at the results of Tables 10, 11, and 12.

Table 17 presents the error rates of the proposed techniques when using the ensemble size $N = 90$. It is possible to see that MINE-W performs better in 9 out of 20 datasets and MINE-WS performs better in 7 out of 20 datasets.

Table 17 – Mean and standard deviation of the results calculated in 20 replications. For each dataset, the best result is in bold. Line "Win/Tie/Loss" shows the total of the results. The values marked with a • indicate that the null hypothesis must be rejected ($pValue \leq 0.05$), in other words, the result of MINE-W achieves superior performance. The values are in the scale 10^{-4}.

	Ensemble Size = 90		
Dataset	MINE-S	MINE-W	MINE-WS
Abalone	59.53(1.04)•	**54.50(0.18)**	55.13(0.61)•
Airfoil Self Noise	**10.75(0.66)**	11.69(0.31)	11.92(0.48)•
Bank32NH	110.80(3.31)•	**88.78(0.33)**	89.94(0.57)•
Bank8FM	13.45(0.25)•	**12.23(0.03)**	12.36(0.16)•
Breast Cancer	**711.42(17.19)**	718.09(8.42)	718.71(10.24)
CCPP	26.35(0.99)•	**22.95(0.12)**	23.02(0.18)
Comp Act	5.80(0.30)•	5.26(0.02)	**5.20(0.03)**
Comp Act Small	7.89(0.11)	8.08(0.03)	**7.87(0.09)**
Concrete	34.89(3.71)•	**30.69(1.21)**	30.88(1.13)
Delta Ailerons	**14.62(0.06)**	15.00(0.02)	15.02(0.03)•
Delta Elevators	28.27(0.17)•	**27.32(0.03)**	27.41(0.17)•
Housing	51.86(8.07)•	47.08(2.80)	**45.28(3.46)**
Kinematics	28.35(0.25)	29.86(0.10)	**27.68(0.15)**
Machine	**48.81(10.34)**	70.13(6.39)	56.24(10.73)
Puma32H	12.89(0.23)•	**10.65(0.03)**	10.69(0.03)•
Puma8NH	173.22(0.85)•	**167.00(0.16)**	167.02(0.15)
Stock	5.88(0.24)•	4.85(0.16)	**4.74(0.17)**
Triazines	210.19(12.61)	**206.04(5.00)**	210.11(7.24)•
Wine Q. Red	164.70(1.39)•	163.91(0.65)	**163.73(0.69)**
Wine Q. White	138.09(2.41)•	130.68(0.42)	**130.11(0.76)**
Win/Tie/Loss	4/0/16	**9/0/11**	7/0/13

According to Table 17, when comparing MINE-W against MINE-S, we can observe that MINE-W performs better in 14 out of 20 datasets and has a significant difference in 13 out of 20 datasets. When compared against MINE-WS, MINE-W has superior performance in 12 out of 20 datasets and significant difference in 8 out of 20 datasets.

6.2 COMPARING WITH STATIC TECHNIQUES

In this section the MINE techniques are compared individually with static techniques. For each dataset, a single regressor was trained using the whole training set, called "Individual Regressor".

Table 18 presents the comparison of the results between the Individual Regressor and the MINE-S technique. It can be verified that MINE-S has a better error rate in 12 out of 20 datasets and the hypothesis tests present a significant difference in 10 out of 20 datasets.

Table 18 – Mean and standard deviation of the results calculated in 20 replications. For each dataset, the best result is in bold. Line "Win/Tie/Loss" shows the total of the results compared between Individual Regressor and MINE-S. The values marked with a • indicate that the null hypothesis must be rejected ($pValue \leq 0.05$), in other words, the result of MINE-S achieves superior performance. The values are in the scale 10^{-4}.

		Ensemble Size = 90
Dataset	Individual Regressor	MINE-S
Abalone	**56.58(0.68)**	59.53(1.04)
Airfoil Self Noise	32.00(4.54)•	**10.75(0.66)**
Bank32NH	**98.04(0.96)**	110.80(3.31)
Bank8FM	**12.79(0.08)**	13.45(0.25)
Breast Cancer	730.74(9.40)•	**711.42(17.19)**
CCPP	**24.46(0.15)**	26.35(0.99)
Comp Act	6.03(0.16)•	**5.80(0.30)**
Comp Act Small	9.28(0.20)•	**7.89(0.11)**
Concrete	52.83(2.70)•	**34.89(3.71)**
Delta Ailerons	15.05(0.02)•	**14.62(0.06)**
Delta Elevators	**27.76(0.10)**	28.27(0.17)
Housing	55.79(4.32)•	**51.86(8.07)**
Kinematics	39.61(1.03)•	**28.35(0.25)**
Machine	82.03(5.37)•	**48.81(10.34)**
Puma32H	**12.29(0.43)**	12.89(0.23)
Puma8NH	**169.48(0.64)**	173.22(0.85)
Stock	**5.52(0.22)**	5.88(0.24)
Triazines	211.96(4.99)	**210.19(12.61)**
Wine Q. Red	164.81(0.55)	**164.70(1.39)**
Wine Q. White	143.12(3.19)•	**138.09(2.41)**
Win/Tie/Loss	8/0/12	**12/0/8**

Table 19 presents the results compared among Mean, Median, and MINE-W. MINE-W has better result in 15 out of 20 datasets. Comparing only with the Mean, MINE-W has better results in 16 out of 20 datasets and MINE-W has a significant difference (•) in 14 out of 20 datasets. When compared to the Median, the results are better in 16 out of 20 datasets and significant difference in 16 out of 20 datasets.

The results of this section show that the proposed techniques are better than static ones. Using DRS in homogeneous ensembles presents satisfactory results and better error rates.

Table 19 – Mean and standard deviation of the results calculated in 20 replications. For each dataset, the best result is in bold. Line "Win/Tie/Loss" shows the total of the results. The values marked with a • indicate that the null hypothesis must be rejected ($pValue \leq 0.05$), in other words, the result of MINE-W achieves superior performance. The values are in the scale 10^{-4}.

	Ensemble Size = 90		
Dataset	Mean	Median	MINE-W
Abalone	54.51(0.19)	**54.46(0.17)**	54.50(0.18)
Airfoil Self Noise	21.48(0.43)•	20.89(0.42)•	**11.69(0.31)**
Bank32NH	88.79(0.29)	89.15(0.25)•	**88.78(0.33)**
Bank8FM	12.27(0.02)•	12.32(0.02)•	**12.23(0.03)**
Breast Cancer	**715.72(6.52)**	716.95(7.10)	718.09(8.42)
CCPP	23.37(0.12)•	23.42(0.12)•	**22.95(0.12)**
Comp Act	5.37(0.02)•	5.38(0.02)•	**5.26(0.02)**
Comp Act Small	8.42(0.02)•	8.45(0.02)•	**8.08(0.03)**
Concrete	39.36(0.89)•	38.08(0.76)•	**30.69(1.21)**
Delta Ailerons	15.03(0.02)•	15.03(0.02)•	**15.00(0.02)**
Delta Elevators	**27.32(0.03)**	27.35(0.03)•	**27.32(0.03)**
Housing	51.75(2.62)•	51.01(2.25)•	**47.08(2.80)**
Kinematics	33.01(0.10)•	33.04(0.12)•	**29.86(0.10)**
Machine	78.79(6.21)•	81.56(5.44)•	**70.13(6.39)**
Puma32H	10.71(0.03)•	**10.64(0.02)**	10.65(0.03)
Puma8NH	**166.93(0.16)**	166.95(0.17)	167.00(0.16)
Stock	5.26(0.14)•	5.23(0.16)•	**4.85(0.16)**
Triazines	206.27(4.89)	209.16(5.19)•	**206.04(5.00)**
Wine Q. Red	164.30(0.59)•	164.67(0.62)•	**163.91(0.65)**
Wine Q. White	133.35(0.39)•	133.56(0.34)•	**130.68(0.42)**
Win/Tie/Loss	2/1/17	2/0/18	**15/1/4**