International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:12, No:3, 2018

Performance Assessment of Multi-Level Ensemble for Multi-Class Problems

Rodolfo Lorbieski, Silvia Modesto Nassar

***Abstract*—**Many supervised machine learning tasks require decision making across numerous different classes. Multi-class classification has several applications, such as face recognition, text recognition and medical diagnostics. The objective of this article is to analyze an adapted method of Stacking in multi-class problems, which combines ensembles within the ensemble itself. For this purpose, a training similar to Stacking was used, but with three levels, where the final decision-maker (level 2) performs its training by combining outputs from the tree-based pair of meta-classifiers (level 1) from Bayesian families. These are in turn trained by pairs of base classifiers (level 0) of the same family. This strategy seeks to promote diversity among the ensembles forming the meta-classifier level 2. Three performance measures were used: (1) accuracy, (2) area under the ROC curve, and (3) time for three factors: (a) datasets, (b) experiments and (c) levels. To compare the factors, ANOVA three-way test was executed for each performance measure, considering 5 datasets by 25 experiments by 3 levels. A triple interaction between factors was observed only in time. The accuracy and area under the ROC curve presented similar results, showing a double interaction between level and experiment, as well as for the dataset factor. It was concluded that level 2 had an average performance above the other levels and that the proposed method is especially efficient for multi-class problems when compared to binary problems.

***Keywords*—**Stacking, multi-layers, ensemble, multi-class.

I. Introduction

MACHINE learning explores the construction of algorithms that seek to automatically learn rules that describe a particular behavior or pattern [1]. One of its fields is supervised learning, which uses information from training data to extract knowledge and make predictions in non-labeled instances in test data [2]. Classification is an instance of supervised learning that occurs when the example to be predicted is identified by a categorical value (class) [3]. The classification task is performed by classifying algorithms.

The amount of information increases at an exponential rate, making classification tasks more complex in modern applications [4]. There are two basic types of problems in classification, multi-class and binary classification. The most practical applications require multi-class classification [5], which is generally more difficult than the classification of binary problems [6].

Classification plays an essential role in the most diverse areas, since accurate predictions can lead to better decisions in information management [7]. Classifiers with similar characteristics are divided into groups of the same category, which is called family [8]. The present research focuses only on Bayesian families and decision trees.

Data quality is a determining factor for the success of classifier predictions [9]. Alternatives to the optimization of classifiers have been constantly developed [10], such as ensembles, that take advantage of the knowledge of several classifiers to promote a better generalization [11].

The machine committee study involves several issues, such as the selection of instances and attributes to be considered, use of appropriate data partitioning methods and the combination of classifiers [10]. For the combination of classifiers linear statistical methods, voting and the combination of meta-learning are used [12], the latter being the focus of this article.

Through the experience gained with the application of one or more classifiers, the meta-learning training occurs, which arises as an alternative to traditional strategies for not involving specialist knowledge or costly trial and error processes [13].

The most common ensemble methods are the Bagging and Boosting techniques [14] for homogeneous classifiers and Stacking for heterogeneous classifiers. The Stacking method, whose training is by meta-learning, can combine different classifiers with simplicity and with a final performance similar to the best classifier of the committee [15]. However, in multi-class problems, Stacking may perform worse than other meta-approaches.

Lorbieski research emerges as a differentiated Stacking method, where the effect of combining two different groups of similar (i.e., family-like) pairs of classifiers using tree-based and Bayesian families is analyzed [16]. Following the same training principle, the combination was divided into three layers, one more than the original Stacking: one layer with two classifiers base for each family (level 0), another with two classifiers representing the different families (level 1) and the last layer with the final decision maker (level 2). Performance was measured in terms of accuracy, area under the ROC curve (AUC) and training time. The current paper is based on the work done in Lorbieski [16] and has the objective to expand the quoted research to multi-class problems by analyzing what effects levels, datasets, and experiments have on performance measures.

This paper is organized as follows: in Section II the related works are presented. Section III explains the operation of the proposed method. In Section IV the experiments performed are detailed, as well as the datasets, tools and algorithms used. The statistical analysis, the results obtained and the discussion about the research are presented in Section V. Finally, section 6 presents the conclusions and future work.

R. Lorbieski is with Department of Informatics and Statistics in Federal University of Santa Catarina, Florianópolis, Brazil (e-mail: r.lorbieski@posgrad.ufsc.br).

S. M. Nassar is with Department of Informatics and Statistics in Federal University of Santa Catarina, Florianópolis, Brazil.

II. Related Work

Stacking studies involve the choice of algorithm and the characteristics to be used in this meta-classifier [17]. The Stacking algorithm was originally developed by Wolpert [18]. The main idea is to teach a meta-classifier with meta-instances created from the outputs of the base classifiers, generally estimated via cross-validation [19]. Wolpert had applied his research based on neural networks. Breiman later had the task of generalizing the Stacking algorithm [20].

Ting and Witten (1999) have shown that Stacking has an optimized performance when meta-instances are formed by probability distributions for each class rather than simple class labels [21]. Seewald (2003) has improved the efficiency of Ting and Witten's Stacking with the creation of StackingC [14], which removes non-relevant attributes and reduces the dimension of the dataset before using it as input for the meta-classifier. After this treatment, the probability of occurrence of a specific class is used for each linear model, disregarding the probabilities of the other classes.

Tsirogianis et al. (2004) used four medical diagnostic problems to group combination methods such as bagging and boosting [22]. In their research a multi-level approach similar to this paper was used, however they chose different combination strategies of Stacking training.

The work of Li et al. (2006), in addition to presenting a helpful review on multi-class classification, also depicts an experimental investigation using discriminant analysis [23]. Tanwani et al. (2009) present guidelines for classification problems with multiple classes [24]. Several research for multi-class classification using ensemble techniques are present in the literature [25]-[27].

Stacking efficiency is directly dependent on the number of classes of the problem [28]. An innovative approach called Troika was proposed by Menaheme et al. (2009) to address multi-class problems [29]. It is based on the four-layer architecture, where the last layer contains only one model: the superclassifier, that outputs a vector of probabilities of the set. Troika performed better than Stacking and StackingC in terms of classification accuracy [28]. In contrast to the Troika, the present article seeks to analyze the effects of only one additional layer in the StackingC algorithm, that is, using three levels and with only two meta-classifiers at level 1.

In this study, to evaluate the discriminative performances of features selected by ensemble algorithms, we will make use of five meta classifiers. The name of the algorithms and their abbreviations are given in Table I. For the sake of convenience, we will use abbreviation of classifiers where needed in this study.

III. Proposed Model

The purpose of this paper is to evaluate the performance of the three-layer ensembles grouping for multi-class problems formed by two groups: one consisting of base classifiers based on decision trees and another Bayesian group using only multi-class datasets. These categories were chosen for being common families in the literature and simple to represent.

The proposed research framework is shown in Fig. 1, which is consisted of several steps. In the first step, the data acquired from the data set is pre-processed, removing duplicates, misclassified and missing data. Then, still in the pre-processing, the data are randomized and then, in the second step, replicated for the training of each base classifier. In the next step, the outputs of the Bayesian and probabilistic base classifiers will be, respectively, the entries for the Bayesian and probabilistic meta-classifiers in the level above. Finally, in the last step, the meta-learner StackingC (level 2) makes a final decision based on the outputs of the two meta-classifiers in the previous level. The evaluation of this research was conducted using a 10-fold cross-validation. Details on the implementation, dataset, and features used are described in the next section.

IV. Experimental Design

For the experiments the Weka API tool was used [30]. This tool was the basis for the pre-processing, training, classification and validation activities of the classifiers used. The implementation was developed on the Java SE platform, that was opted for reasons of portability (Windows/Linux), gratuity, familiarity with the language, good documentation and easy communication with the Weka tool. For simplicity, all classifiers used Weka's default values. The datasets, experiments and levels were considered as influence factors in performance measures. The next subsections clarify details about the algorithms used and the dataset and experiments factors.

A. Combination Methods Used

Meta-learning improves predictive performance by combining different modes of learning, each with distinct representations and heuristics. By merging different concepts learned, it is expected that meta-classifiers will achieve better accuracy than their individual classifiers [31].

Meta-classifiers type I are composed of homogeneous classifiers, while the ones composed of heterogeneous classifiers are type II. Since a level 2 meta-learner necessarily combines two different classifiers, it is type II and the use of StackingC was chosen. In present research, five different level 1 meta-classifiers were used and are described as follows.

1) Bagging (type I): Voting scheme in which n models of the same type are built. The class chosen is the one with majority voting between the models' predictions [20].

2) AdaBoosting (type I): An implementation of boosting. It works similarly to Bagging, but the boosting is interactive and each classifier has individual weights for its predictions. Base classifiers focus on difficult-to-classify examples [32].

3) Dagging (type I): A meta-classifier similar to Bagging, which provides disjoint subsets of training data for the chosen base classifier to make a final decision [21].

4) MultiScheme (type II): Selects a classifier among others using cross-validation in training data or performance in training data. Performance is measured based on the correct percentage [7].

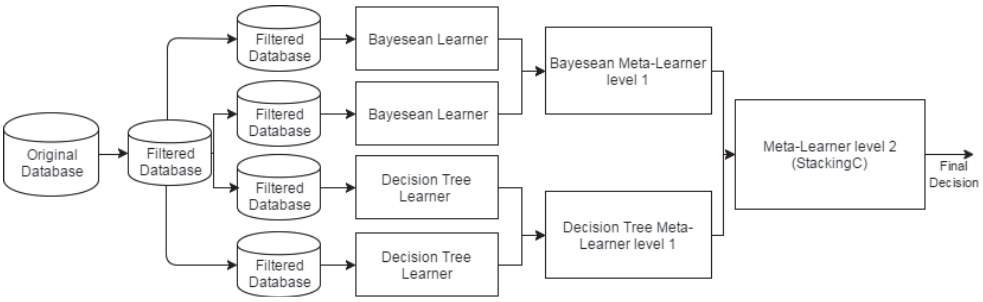

Fig. 1 Research framework

5) StackingC (type II): An efficient version of Stacking, especially for better performance in multi-class data sets [14].

For the sake of convenience, we will use abbreviation of meta-classifiers in charts of present research. The name of the classifiers and their abbreviations are given in Table I.

TABLE I
COMBINATION METHODS USED

Meta-Classifiers	Abbreviation
Bagging	bag
Adaboosting	boost
Dagging	dag
MultiScheme	ms
StackingC	stack

B. Base Classifiers

Several base classifiers available from WEKA [30] were tested. When using a meta-classifier at level 1, it is necessary to determine which base classifiers will form their knowledge base according to their type. For homogeneous classifiers, BayesNet classifiers [33] and J48 [34] were selected for, respectively, the Bayesian and tree-based families. For heterogeneous Bayesian base classifiers the BayesNet and NaïveBayes [35] classifiers were selected. In the heterogeneous tree-based models the DecisionStump based classifiers [36] and J48 were used. The cited algorithms were selected for being widely published in the literature and easy to learn. All selected classifiers had a good average performance in the bases evaluated in comparison to other classifiers not mentioned.

C. Dataset Factor

For the experiments, five different UCI public repository datasets were used [37]. Only multi-class datasets were used, with variations in the number of attributes and instances (Table I). Such datasets have been used vastly in works of the Artificial Intelligence area.

TABLE II
DATASET SETTINGS USED

Datasets	Abbreviation	Attributes number	Instances	Classes
Wine	WI	13	178	3
Vehicle	VE	18	946	4
Glass Identification	GI	13	214	6
Segmentation	SE	19	2310	7
Ecoli	EC	8	366	8

D. Experiment Factor

The experiment factor considers which pair of combination meta-classifiers at level 1 was used for each experiment. Each meta-classifier level 1 has five combination options to be applied in the base classifiers. Since there are two meta-classifiers at level 1, a total of 25 different experiments are analyzed.

V. RESULTS AND DISCUSSION

In present research it was established that the performance measures (dependent variables) that are most important for determining the quality of a classifier are: (1) training time, (2) accuracy and (3) area under the ROC curve (AUC). Among the control factors, the ones that most affect the dependent variables are: (1) the dataset used, (2) the level of classifier and (3) the experiment.

To evaluate performance of multi-class dataset in terms of AUC was used the weighted average of AUC, where each target class is weighted according to its prevalence; accuracy is the average of correct predictions divided by the total number of predictions for all class [38].

The data obtained from each experiment were analyzed using IBM SPSS Statistics program and the implementation was performed on an Intel Xeon computer with a 3.3GHz processor, 16GB of RAM and the Windows 8 operating system. Simultaneous analysis of the groups was conducted by an analysis of variance (ANOVA three-way) for each dependent variable considering 5 multi-class datasets x 25 experiments x 3 levels. To locate the differences found, the Tukey test was conducted with a significance level of 5% ($p<0.05$).

For comparison purposes, the results obtained for multi-class datasets are summarized in Table II, where the average performance of the different levels in each dependent variable is shown. The time shown in the table is calculated. It is estimated that the performance in the accuracy and in the area under the ROC curve of level 1 has shown worse results compared to the other levels because of the diversity, since the meta-classifiers of this level learn with classifiers of the same family.

The triple interaction on independent variables occurred only in the time variable ($p<0.001$) as shown in Figs. 2 and 3. In Fig. 2 it is illustrated the average time, in seconds, spent in each dataset per level. The Segmentation dataset was the one that most required training time for the classifiers.

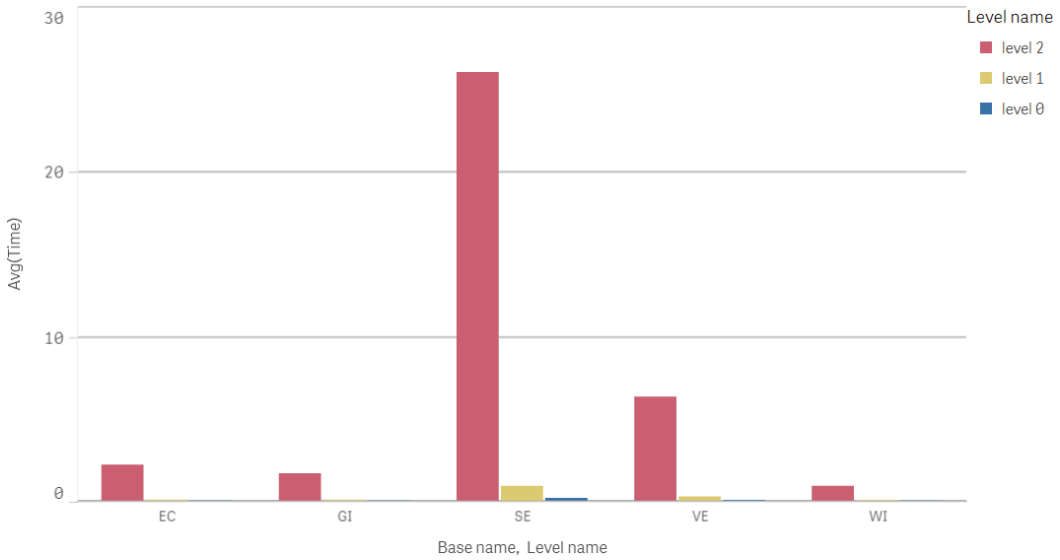

Fig. 2 Average time spent per level in each dataset

TABLE III
PERFORMANCES COMPARISON IN MULTI-CLASS PROBLEMS

Performance Measures	Multi-class databases		
	Level 0 (n=350)	Level 1 (n=250)	Level 2 (n=125)
1 - Accuracy	0.804	0.759	0.907
2 - Area under ROC	0.911	0.859	0.974
3 - Time (s)	0.05	0.3	7.4

The graph of Fig. 3 shows the sum of the time spent per experiment at each level. Experiments with meta-classifiers level 1, type II (StackingC and MultiScheme) showed a significantly higher training time than the type I meta classifiers (Bagging, Boosting, Dagging). The StackingC - StackingC experiment stands out, in which the total time spent on level 2 exceeds 100 seconds.

For the performance measures area under the ROC curve and accuracy, the interactions were similar (Figs. 4 and 5). Both for accuracy and for area there was double interaction only for the level versus experiment factors ($p < 0.001$).

The unsatisfactory performance of the StackingC level 1 meta-classifier also remains in the performance measures area under the ROC curve and accuracy. The loss of accuracy and area in Stacking levels presented in Figs. 4 and 5 is a consequence of using stack approaches with similar classifiers (same family) and in small amounts, such as in the case of Level 1 classifiers. Performance was markedly higher at level 2 when diversity was gained by using a larger number of classifiers from different families.

Once observed the interaction between level versus experiment, the respective main effects were ignored (if the two-way interaction is statistically significant, there will be significant differences in the simple interaction effects in each case [39]). Therefore, the main effect is highlighted only in the base factor ($p<0.001$ for both performance measures) (Fig. 6).

It is noted a better performance in level 2 than level 0 (which showed the second best average performance) among the multi-class datasets. With respect to accuracy, level 2 achieved an average performance 12.81% higher than level 0 - a performance 3.4 times higher than in binary problems obtained in Lorbieski [16]. In contrast, for the area under ROC curve, level 2 obtained an average performance of 6.87% higher than level 0, a result almost equal to Lorbieski's research.

Regarding the mean of training time, the factors dataset, level and experiment were highly correlated. As expected, level 2 exceeded the time of the other levels in all analyzes, since it requires that all classifiers that form its knowledge have already been trained.

There was an increase in the average training time of 7.4 seconds in relation to the level 0. Therefore, the present method is feasible to be used for small and medium bases. This increase was slightly higher (approximately 1 second) for binary problems obtained by Lorbieski [16].

These results suggest that the proposed method leads to significant performance gains in multi-class problems. Such findings are similar to those obtained by Lorbieski in binary problems [16]. Finally, the good performance of the method in multi-class problems reported in this research also coincides with results of related work in multi-class classification problems in ensemble, such as the Troika [29] and other authors [26], [27].

VI. CONCLUSION

This research presented the results of an investigation in multi-class problems using Stacking algorithms that perform groupings of classifiers of different families, contributing to the studies of classification problems using multi-level ensemble.

These investigations differ from related work by analyzing a particular case of Stacking for multi-class data and measuring the efficiency not only in terms of accuracy and time, but also of the area under the ROC curve. In multi-class problems an

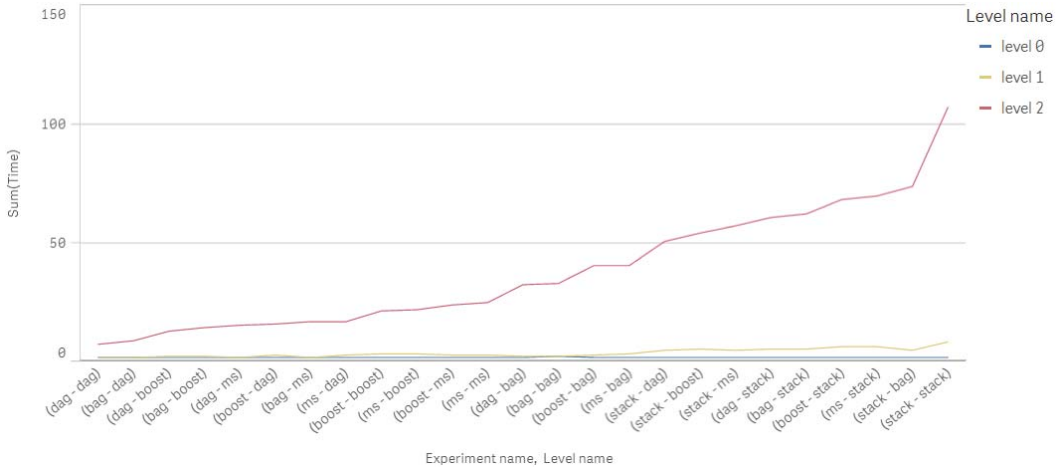

Fig. 3 Sum of the time spent by experiment at each level

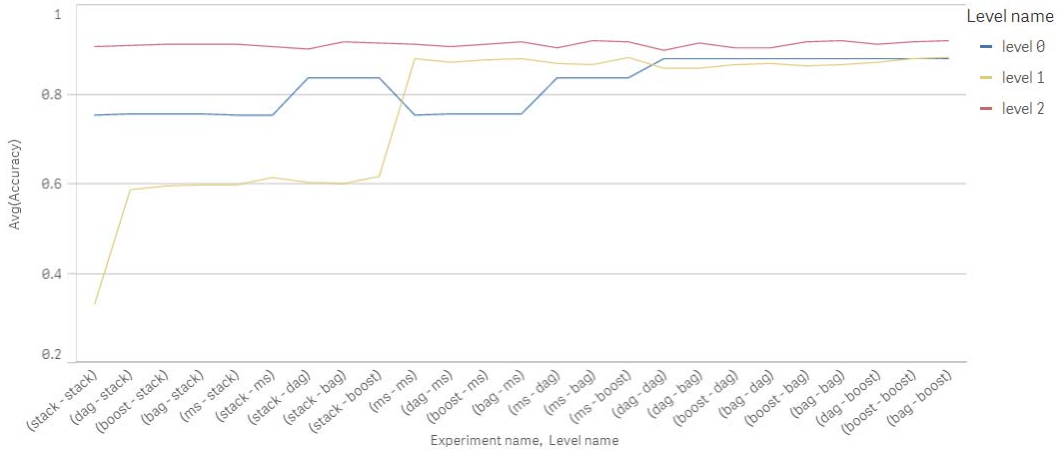

Fig. 4 Interaction between experiment versus level in accuracy

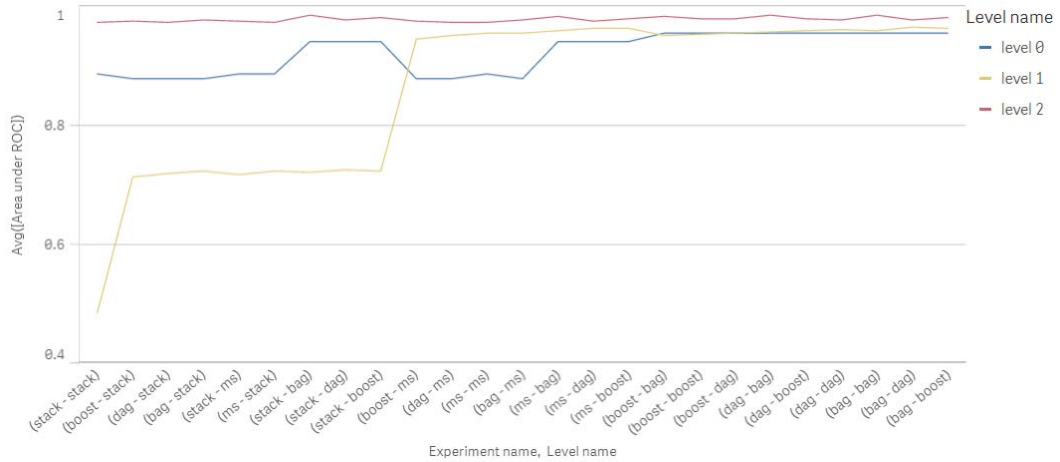

Fig. 5 Interaction between experiment versus level in area under ROC

increase of accuracy and area in relation to binary problems was promoted. In relation to time, there was no significant increase in the use of level 2 in the multi-class datasets compared to the binary ones.

The StackingC level 1 meta-classifier decreased its performance on time, accuracy and area under the ROC curve in all the experiments in which it participated. An optimization of this meta-classifier would certainly provide even better

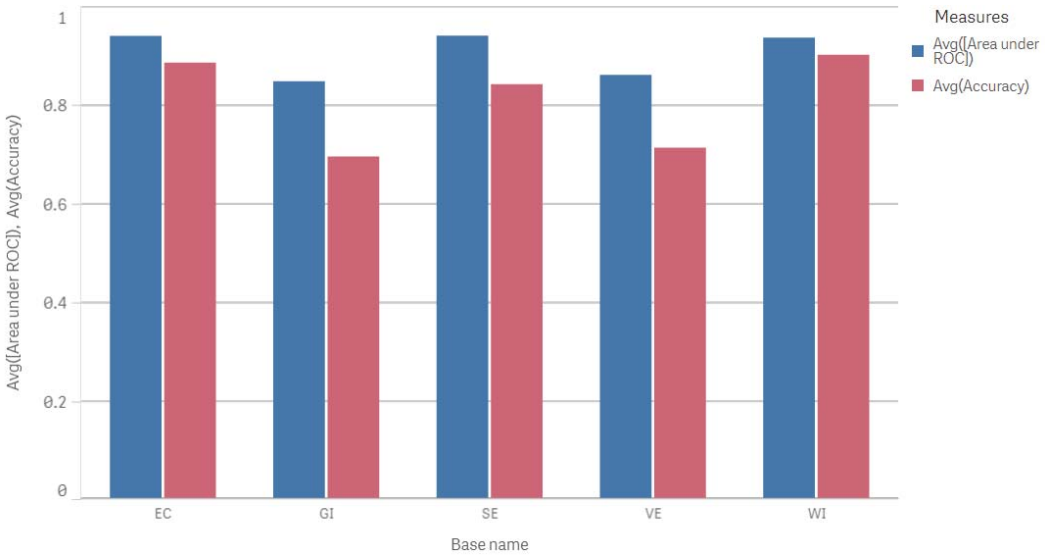

Fig. 6 Effect of the base factor for accuracy and area under the ROC curve

results for the method used. Therefore, for future work it is suggested an improvement or adaptation of StackingC to the proposed method and the use of calculations of diversity measures, which could aid the judgment of the most appropriate meta-classifiers and classifiers for each situation.

References

[1] M. F. F. Oliveira, "Análise de mercado: uma ferramenta de mapeamento de oportunidades de negócio em técnicas de geomarketing e aprendizado de máquina," 2016.
[2] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," 2007.
[3] S. Chebrolu, A. Abraham, and J. P. Thomas, "Feature deduction and ensemble design of intrusion detection systems," *Computers & security*, vol. 24, no. 4, pp. 295–307, 2005.
[4] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. U. Khan, "The rise of "big data" on cloud computing: Review and open research issues," *Information Systems*, vol. 47, pp. 98–115, 2015.
[5] G. Malik and M. Tarique, "On machine learning techniques for multi class classification," *International Journal of Advancements in Research & Technology*, vol. 3, no. 2, 2014.
[6] J. Liu, S. Ranka, and T. Kahveci, "Classification and feature selection algorithms for multi-class cgh data," *Bioinformatics*, vol. 24, no. 13, pp. i86–i95, 2008.
[7] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.
[8] M. Fernandez-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems," *J. Mach. Learn. Res*, vol. 15, no. 1, pp. 3133–3181, 2014.
[9] J. A. Saez, J. Luengo, and F. Herrera, "Evaluating the classifier behavior with noisy data considering performance and robustness: the equalized loss of accuracy measure," *Neurocomputing*, vol. 176, pp. 26–35, 2016.
[10] Y. Ren, L. Zhang, and P. N. Suganthan, "Ensemble classification and regression-recent developments, applications and future directions [review article]," *IEEE Computational Intelligence Magazine*, vol. 11, no. 1, pp. 41–53, 2016.
[11] J.-C. Levesque, C. Gagne, and R. Sabourin, "Bayesian hyperparameter optimization for ensemble learning," *arXiv preprint arXiv:1605.06394*, 2016.
[12] L. M. Vriesmann, *Seleção Dinâmica de Subconjunto de Classificadores*. PhD thesis, Pontifícia Universidade Católica do Paraná, 2012.
[13] R. Vilalta and Y. Drissi, "A perspective view and survey of meta-learning," *Artificial Intelligence Review*, vol. 18, no. 2, pp. 77–95, 2002.
[14] A. K. Seewald, *Towards understanding stacking: studies of a general ensemble learning scheme*. na, 2003.
[15] S. Džeroski and B. Ženko, "Is combining classifiers with stacking better than selecting the best one?," *Machine learning*, vol. 54, no. 3, pp. 255–273, 2004.
[16] R. Lorbieski and S. Nassar, "Performance evaluation in multi-level ensemble.," 2017. Manuscript submitted for publication.
[17] A. Ledezma, R. Aler, A. Sanchis, and D. Borrajo, "Ga-stacking: Evolutionary stacked generalization," *Intelligent Data Analysis*, vol. 14, no. 1, pp. 89–119, 2010.
[18] D. H. Wolpert, "Stacked generalization," *Neural networks*, vol. 5, no. 2, pp. 241–259, 1992.
[19] G. Sigletos, G. Paliouras, C. D. Spyropoulos, and M. Hatzopoulos, "Combining information extraction systems using voting and stacked generalization," *Journal of Machine Learning Research*, vol. 6, no. Nov, pp. 1751–1782, 2005.
[20] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.
[21] K. M. Ting and I. H. Witten, "Issues in stacked generalization," *J. Artif. Intell. Res.(JAIR)*, vol. 10, pp. 271–289, 1999.
[22] G. Tsirogiannis, D. Frossyniotis, J. Stoitsis, S. Golemati, A. Stafylopatis, and K. Nikita, "Classification of medical data with a robust multi-level combination scheme," in *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, vol. 3, pp. 2483–2487, IEEE, 2004.
[23] T. Li, S. Zhu, and M. Ogihara, "Using discriminant analysis for multi-class classification: an experimental investigation," *Knowledge and information systems*, vol. 10, no. 4, pp. 453–472, 2006.
[24] A. K. Tanwani, J. Afridi, M. Z. Shafiq, and M. Farooq, "Guidelines to select machine learning scheme for classification of biomedical datasets," in *European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*, pp. 128–139, Springer, 2009.
[25] T. Windeatt and R. Ghaderi, "Coding and decoding strategies for multi-class learning problems," *Information Fusion*, vol. 4, no. 1, pp. 11–21, 2003.
[26] G. Tsoumakas and I. Vlahavas, "Random k-labelsets: An ensemble method for multilabel classification," in *European Conference on Machine Learning*, pp. 406–417, Springer, 2007.
[27] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes," *Pattern Recognition*, vol. 44, no. 8, pp. 1761–1776, 2011.
[28] A. Jurek, Y. Bi, S. Wu, and C. Nugent, "A survey of commonly used ensemble-based classification techniques," *The Knowledge Engineering Review*, vol. 29, no. 05, pp. 551–581, 2014.
[29] E. Menahem, L. Rokach, and Y. Elovici, "Troika–an improved stacking schema for classification tasks," *Information Sciences*, vol. 179, no. 24, pp. 4097–4122, 2009.

[30] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.
[31] A. Prodromidis, P. Chan, and S. Stolfo, "Meta-learning in distributed data mining systems: Issues and approaches," *Advances in distributed and parallel knowledge discovery*, vol. 3, pp. 81–114, 2000.
[32] Y. Freund, R. E. Schapire, *et al.*, "Experiments with a new boosting algorithm," in *icml*, vol. 96, pp. 148–156, 1996.
[33] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Machine learning*, vol. 29, no. 2-3, pp. 131–163, 1997.
[34] J. Quinlan, "C4. 5: Programs for empirical learning morgan kaufmann," *San Francisco, CA*, 1993.
[35] G. H. John and P. Langley, "Estimating continuous distributions in bayesian classifiers," in *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pp. 338–345, Morgan Kaufmann Publishers Inc., 1995.
[36] W. Iba and P. Langley, "Induction of one-level decision trees," in *Proceedings of the ninth international conference on machine learning*, pp. 233–240, 1992.
[37] K. Bache and M. Lichman, "Uci machine learning repository," 2013.
[38] D. J. Hand and R. J. Till, "A simple generalisation of the area under the roc curve for multiple class classification problems," *Machine learning*, vol. 45, no. 2, pp. 171–186, 2001.
[39] B. Cohen, *Explaining Psychological Statistics*. Wiley, 2013.