Journal of Computer Science

Original Research Paper

Impact of an Extra Layer on the Stacking Algorithm for Classification Problems

Rodolfo Lorbieski and Silvia Modesto Nassar

Department of Informatics and Statistic, UFSC, Florianópolis, Brazil

Article history
Received: 21-03-2017
Revised: 19-10-2017
Accepted: 09-01-2018

Corresponding Author:
Rodolfo Lorbieski
Department of Informatics and Statistic, UFSC, Florianópolis, Brazil
Email: rdlorbieski@gmail.com

Abstract: Classifying and making decisions are tasks performed by any human being in their daily lives. Learning algorithms have been widely studied as tools to aid information management, with an objective to maximize the generalization capacity. Learning algorithms can be used individually or as a committee of machines (ensembles). An ensemble uses the solutions provided by several machines, making different combinations with them to reach a final decision, such as multi-layer algorithm stacking. When combining combination methods, one arrives at a three-layered architecture, which is the focus of this article. The objective of this work was to evaluate the influence of adding one more layer in the stacking meta-learning algorithm in other to obtain accuracy, area under ROC and time in relation to the lower layers, under the influence of the experiment, database and level factors. It was possible to conclude that, statistically, classifiers of the extra layer presented, in a general way, better performance in terms of accuracy and area. However, time grew sharply at each top layer added.

Keywords: Ensemble, Stacking, Multi Layers, Machine Learning

Introduction

Machine learning is a data analysis method intensively developed in the last decades that automates the construction of analytical models. One of its main sub domains is supervised learning, which task is to infer a function to precisely assign unmarked instances (test set) to different predefined classes (Homayouni *et al*., 2010). In the machine learning terminology, classification is considered a supervised learning instance (Alpaydin, 1998), whose objective is to create a mapping between a set of input variables and the output variable through observations of the training data.

An algorithm that implements classification is known as classifier. Classifiers are an invaluable tool for many tasks, such as medical or genomic predictions, spam detection, face recognition and finance (Bost *et al*., 2014). In addition, they can help people explore the knowledge within the data sets and then use it for decision-making (Ballard and Wenjia, 2016).

Numerous classifiers exist in the literature, Delgado *et al*. (2014) described in their work 179 classifiers divided into 17 families. Among the classifier families are those based on probabilities, decision trees, rules-based and others.

The success of learning and acquiring knowledge from the data analyzed depends on several factors, including data quality (Sluban and Lavrac, 2015). However, inconsistencies arise naturally when using real data (da Costa and Abe, 2000). Similarly, Xindong and Zhu (2008) stated that data obtained from real-world problems are never perfect and undergo changes that may hinder the system's performance.

Several studies revealed that there is no general classifier that is suitable for any database, as indicated in the "No Free Lunch" theorem (Wolpert and Macready, 1997). How to build a more reliable ranking system and how to increase ranking accuracy are two major issues that motivate researches in this field (Chen and Wong, 2010). Possible solutions to these problems are ensemble methods, which use meta-classifiers to combine several classifiers together (Polikar, 2006).

Numerous ensemble techniques have been proposed in the literature, such as Boosting (Freund and Schapire, 1996), Bagging (Breiman, 1996) and Stacking (Wolpert, 1992). Additionally, they use a variety of combination methods, including majority voting (Dorigo *et al*., 2006), the weighted majority (Kuncheva, 2004), the fuzzy integral (Cho and Kim, 1995), among others.

While Bagging and Boosting use a linear combination of the same classifiers, Stacking uses a different classifier in a layer called Meta-classifier, whose task is to combine the predictions of different

© 2018 Rodolfo Lorbieski and Silvia Modesto Nassar. This open access article is distributed under a Creative Commons Attribution (CC-BY) 3.0 license.

base classifiers in order to reduce the generalization error (Kadkhodaei and Moghadam, 2016).

The Stacking method offers certain benefits compared to Bagging and Boosting, including the ability to combine different classifiers with simplicity and having a final performance similar to the best classifier of the committee (Menahem *et al.*, 2009). However, in multi-class problems, Stacking may perform worse than other meta-approaches.

Thus, this study proposes an optimization in the use of Stacking by inserting one more level (level-2) in its original structure, so that the final prediction is composed of two ensembles (level-1), instead of isolated base classifiers (level-0). The approach adopted, called Grouping Ensembles Stacking (GES), uses a multilevel strategy to combine groups of classifiers by Stacking.

This study was elaborated to contribute to the researches related to the performance of multilevel Stacking in supervised learning problems. This research has an empirical-analytical content and has as main objective to verify, through Stacking, the potential of ensemble committees in unbalanced datasets when comparing its average performance with lower level classifiers, that generates the knowledge of the level-2 meta-classifier.

This paper is organized as follows: Section 2 presents related works published and Section 3 discusses the design and operation of GES. Section 4 shows details about the experiments and algorithms used. Section 5 and 6 describe the statistical analysis and results. Section 7 and 8 discuss the significance of these findings, conclusion and future directions of this research.

Related Works

The number of publications on ensemble techniques has grown exponentially since its inception (Woźniak *et al.*, 2014). The main idea of combining classifiers is to build a set that will be more effective than any of its individual members operating in isolation. Many ensemble algorithms have been proposed in the literature. Among them, Stacking is one of the most representative methods (Tang *et al.*, 2010).

Recent studies with Stacking involve choosing the algorithm and the characteristics to be used in this meta-classifier (Ledezma *et al.*, 2010). The idea of the Stacking algorithm was first introduced by Wolpert (1992) in the neural networks context and then generalized by Breiman (1996). LeBlanc and Tibshirani (1996) found that Stacking with a non-negative weight restriction may be an efficient way to obtain a better predictive model. Merz (1999) presents the SCANN algorithm, which uses the correspondence analysis to detect correlations between the base classifiers. It selects uncorrelated dimensions as the input variables of the meta-classifier and a closer neighbor method is then used in its learning.

Ting and Witten (1999) used probability distributions for the outputs of each class returned by the base classifiers as input characteristics of the meta-classifier, the authors proposed the use of Multi-response Linear Regression (MLR) technique as a meta-algorithm. Todorovski and Džeroski (2000) propose a Stacking algorithm called Meta-Decision Trees (MDT) that replaces the class value predictions in the leaves by the predictive probabilities returned by the base classifiers. Seewald and Fürnkranz (2001) present an algorithm called Grading that constructs a meta-classifier for each base classifier, where the purpose of each meta-classifier is to determine which base classifier can return a better result. The final prediction is determined by the sum of the correct predictive probabilities returned by the base classifiers.

In empirical tests, Stacking displayed significant performance degradation for data sets of several classes. To solve this problem, Seewald (2002) introduced an alternative algorithm called StackingC. Based on Stacking with MLR, this algorithm reduces the number of probabilities returned by the base classifiers to overcome the weakness of stacking in multi-class problems.

Sill *et al.* (2009) presented a linear technique named Featured-Weighted Linear Stacking (FWLS), where the weights associated with the models are parameterized as linear functions of the meta-characteristics. Abawajy and Kelarev (2012) worked with multi-level ensembles in a systematic investigation, applying the cardiac autonomic progression classification in patients with diabetes.

Stacking efficiency is directly dependent on the number of classes of the problem (Jurek *et al.*, 2014). A new approach called Troika was proposed by Menahem *et al.* (2009) to address multi-class problems. It is based on the four-layer architecture, where the last layer contained only one model: The super classifier, that outputs a vector of probabilities as a final decision of ensemble. Troika performed better than Stacking and StackingC in terms of classification accuracy (Jurek *et al.*, 2014). This paper, differing from Troika's approach, will analyze the effects of an additional layer in the original Stacking algorithm, using only three levels (level-0, level-1 and level-2) and with only 2 meta-classifiers at level-1.

While many Stacking algorithms have been proposed to improve performance in various classification problems, there is no guarantee that this meta-algorithm will outperform all base classifier.

Materials and Methods

Among the most important concepts in a learning algorithm are preprocessing, representation and evaluation (Domingos, 2012). The next subsections discuss, respectively, each of these aspects in the GES.

Preprocessing

The preprocessing step is fundamental. If there is too much irrelevant and redundant information or noisy and unreliable data, the acquisition of knowledge during the training phase becomes harder. Preprocessing filtering steps may greatly optimize the classifiers' training time.

As preprocessing works to improve data quality, naturally it has a positive impact on the generalization performance of a machine learning algorithm (Kotsiantis *et al*., 2007). In this approach, a filter was used to replace null values, another to remove duplicate lines and a third one to generate randomness in the database instances, thus ensuring representativeness in the data.

Representation

Wolpert (1992) proposed the generalization framework by Stacking, which uses a layered architecture. Level-0 classifiers, also known as base classifiers, receive the original dataset as input, each providing a forecast. A meta-classifier at level-1 uses the predictions from the previous layer to produce the final prediction. Stacking's unique architecture focuses on two layers.

Unlike Stacking, this study consists of adding a layer in the part that composes the learning, where a level-2 meta-learner makes a final decision based on the predictions of lower levels meta-learners. For simplicity, the classifiers were grouped in pairs at all levels, whenever heterogeneous classifiers are used.

To increase diversity of ensembles, level-1 meta-learners are trained in different ways, where one of them is composed of base classifiers based on probabilities and the other by base classifiers based on decision trees. These categories were chosen considering that these families are common in the literature and simple to represent. This methodology is expected to result in a general improvement of accuracy and AUC by increasing the diversity among classifiers. Details on the mode of implementation, database and features used are described in the Experiments section.

Evaluation

It is necessary to qualify the result produced by a given classifier in order to estimate its performance when applied to future classifications. In cases where the amount of available data is small, the use of the k-fold cross-validation technique is recommended for this qualification (Prati *et al*., 2008). This technique evaluates the generalization ability of a model using a set of data as input. In the literature, cross validations of 5 and 10 folders are commonly used (McLachlan *et al*., 2005). The present research used 5-fold cross-validation, for reasons of processing and time.

Experiments

For the experiments, several algorithms implemented in the Waikato Environment for Knowledge Analysis (WEKA) were used. This tool includes all the filters and algorithms used to generate the base classifiers, meta-classifiers and ensemble generation algorithms (Witten *et al*., 2016).

Both the base classifiers and the meta-classifiers followed WEKA's standard parameters except for heterogeneous classifiers, where different classifiers must be selected in a distinct way. The databases, classifiers used and other details are described in the following subsections.

Databases

For the GES experiments, five different UCI public repository databases (Blake and Merz, 1998) were used, arranged according to Table 1. The use of bases with variations in the number of attributes, instances and number of output classes (case of the Vehicle database, with four output classes) were prioritized. Such databases have been diversely used in researches in this field.

Base Classifiers

Several base classifiers available from WEKA (Witten *et al*., 2016) were tested and Bayes Net classifiers (Friedman *et al*., 1997) and/or Naive Bayes (John and Langley, 1995) were chosen as probabilistic models. In tree-based models, Decision Stump classifiers (Iba and Langley, 1992) and J48 (Quinlan, 1993) were used.

Each base classifier used was trained with a sample replica of each database. All selected classifiers had a good average performance compared to other classifiers not mentioned.

Table 1: Description of the databases used

Database	Abbreviation	Attributes	Instances	Output classes
Breast Cancer	BC	10	286	2
Diabetes	DB	8	768	2
Horse Colic	HC	368	27	2
Ionosphere	IS	35	351	2
Vehicle	VH	19	846	4

Meta-Classifiers

Meta-learning improves predictive performance by combining different modes of learning, each with different representations and heuristics. By combining different concepts learned, it is expected that meta-classifiers will achieve better accuracy than their individual classifiers (Prodromidis *et al*., 2000).

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

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 (Breiman, 1996).

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 (Freund and Schapire, 1996).

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 (Ting and Witten, 1997).

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 (Witten *et al*., 2016).

StackingC (type II)

An efficient version of Stacking, especially for better performance in multiclass data sets (Seewald, 2002).

Development

The Java language, using the WEKA software API, was used to perform this work. Java language was opted for reasons of portability (Windows/Linux), gratuity, familiarity with the language and easiness in documentation.

Experiment Architecture

The final decision of the ensembles committee is taken by the level-2 meta-classifier, whose learning is based on the predictions of two meta-classifiers level-1, which in turn have their knowledge formed by the predictions of probabilistic base classifiers or based on decision trees.

Meta-classifiers type I (level-1) are formed by BayesNet classifiers, if it is Bayesian and by classifiers J48, if it is based on decision trees. These classifiers were chosen based on their good performance in their respective families. The complete representation of the architecture used can be verified in Fig. 1.

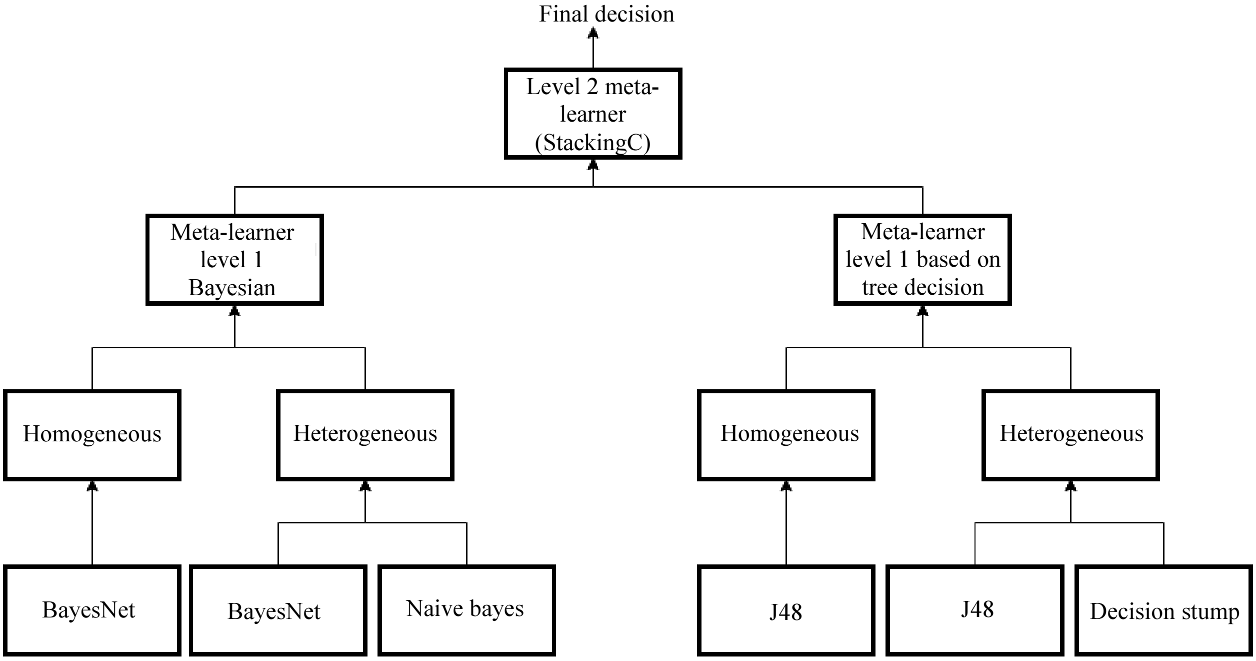

Fig. 1: Architecture representation

Thus, since only one level-2 meta-classifier was analyzed (StackingC) and the base classifiers selected depend exclusively on the meta-classifier of the level above and its type, there is a total of 25 different experiments to be considered in the statistical analysis, one for each possible pair of level-1 meta-classifiers.

Statistical Analysis

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

The data obtained from each experiment were analyzed using IBM SPSS Statistics program (Field, 2013). Simultaneous analysis of the groups was conducted by an analysis of variance (ANOVA three-way) for each dependent variable considering 5 databases ×25 experiments ×3 levels. To locate the differences found, the Tukey test was conducted with a level of significance of 5% ($p<0.05$).

Results

A profile of the factor level was determined with the mean values of the following dependent variables: Accuracy, area under ROC and training time (in seconds) at each level, according to Table 2. It is observed initially that the accuracy and average area of level-2 presented better results than any other level. Furthermore, the base classifiers achieved a better average performance than the level-1 meta-classifiers in all performance measures.

For a deeper analysis, the results will be individually discussed by dependent variable through analysis of variance (ANOVA).

Accuracy factor

The three-way ANOVA results, that is, databases, experiments and levels, did not show significant interaction ($p>0.9999$). Additionally, there was no interaction between level versus experiment ($p = 0.37$) and base versus experiment ($p>0.9999$). On the other hand, according to Fig. 2, there was interaction between database versus level ($p<0.0001$).

Once observed the interaction between database versus level, the respective main effects were ignored (base and level). Therefore, the main effect is highlighted only in the experiment factor ($p = 0.036$) (Fig. 3).

Area Under ROC

The ANOVA (Three-way) results also did not show a significant interaction ($p>0.9999$) for this dependent variable. There was also no interaction between base versus level ($p = 0.525$) and base versus experiment ($p>0.9999$). In contrast, according to Fig. 4, interaction between level versus experiment ($p<0.001$) occurred.

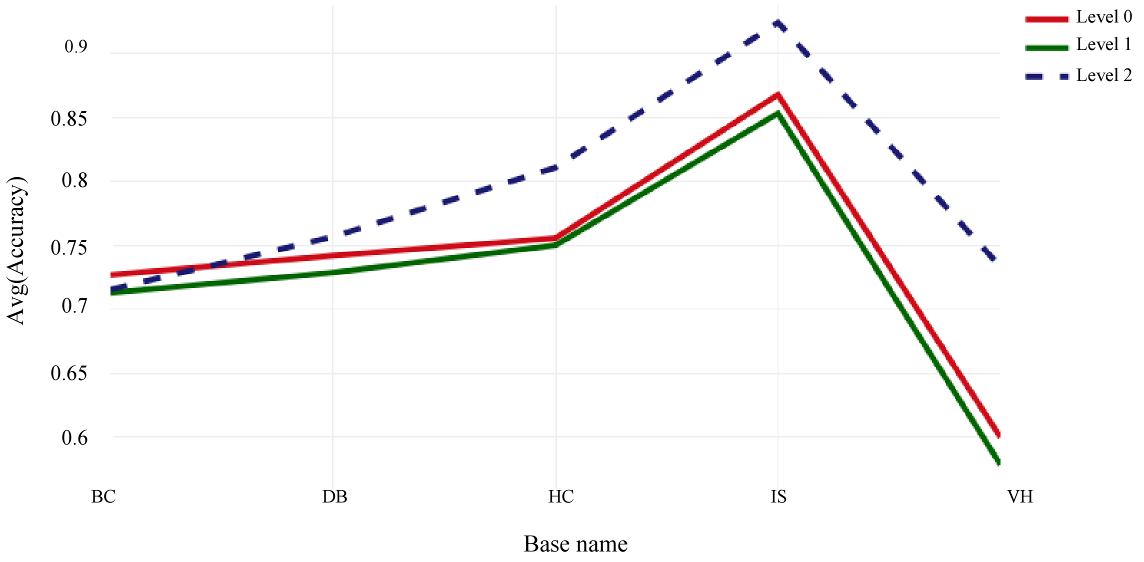

Fig. 2: Interaction between database versus level in accuracy

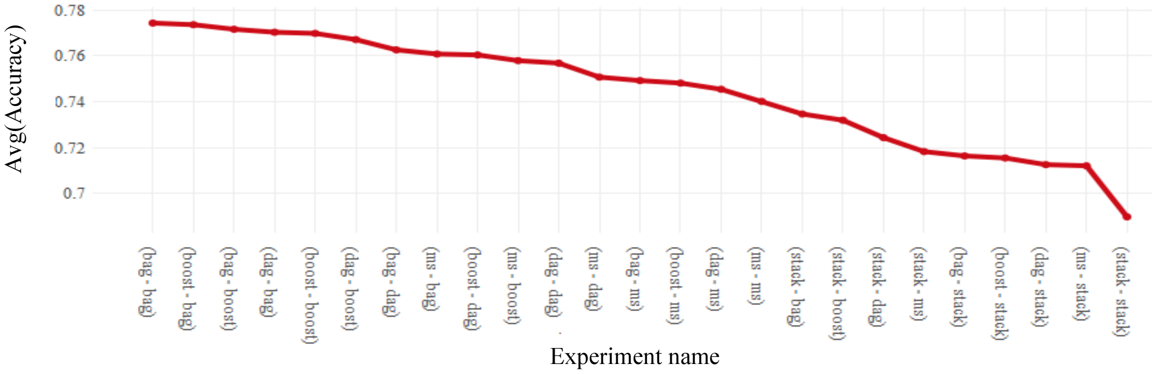

Fig. 3: Experiment factor in the accuracy factor

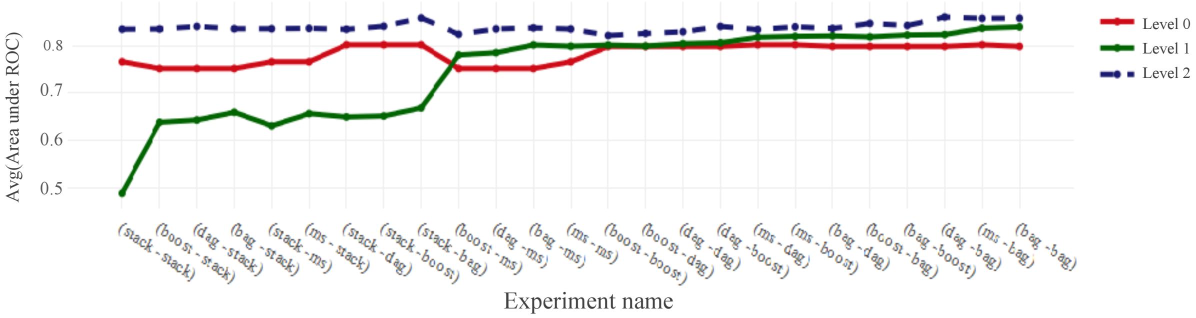

Fig. 4: Interaction between level versus experiment in the area under ROC factor

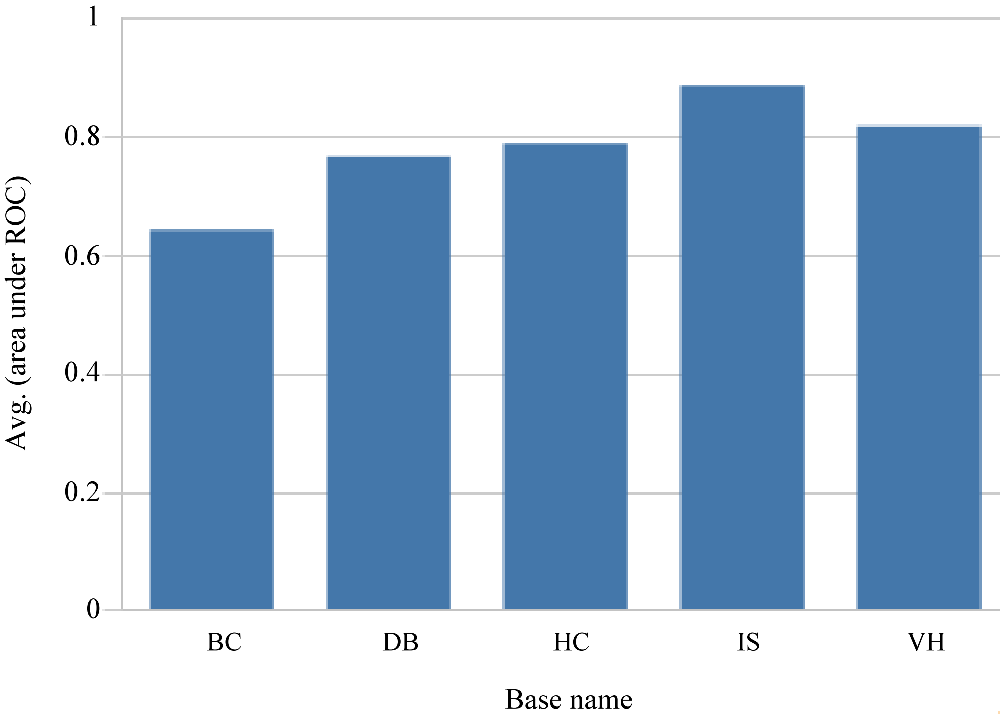

Fig. 5: Database factor in the area under ROC factor

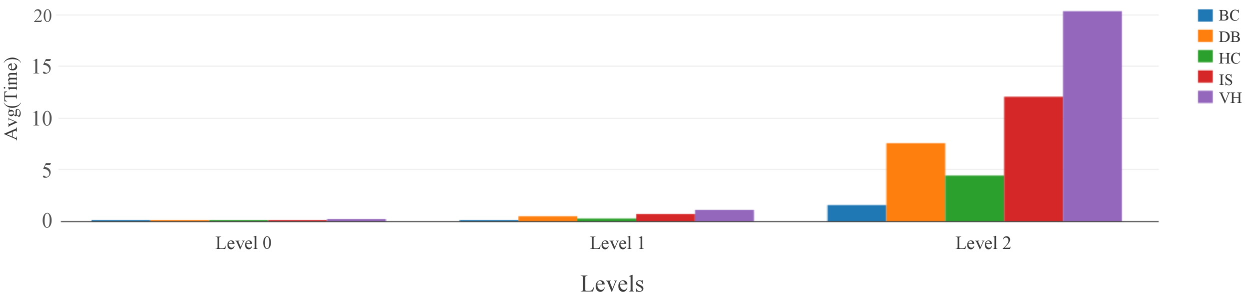

Fig. 6: Interaction of levels on databases in the time factor

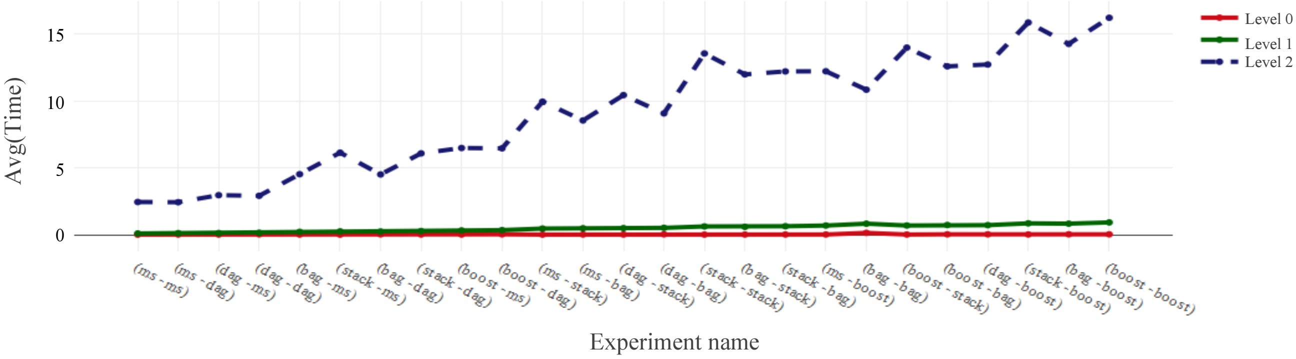

Fig. 7: Interaction of experiments on levels in the time factor

Table 2: Average of performance measures by level

Performance measure	Level-0 (n = 350)	Level-1 (n = 250)	Level-2 (n = 125)
Accuracy	0.727	0.722	0.788
Area under ROC	0.783	0.753	0.839
Time (s)	0.100	0.600	9.200

After this interaction, only the main effect in the base factor (p<0.001) is highlighted. The Fig. 5 illustrates the main database effect.

Time Factor

For the time factor, the results of ANOVA (Three-way) presented significant interaction in all possible combinations between the database, level and experiment variables (all with p<0.001). Figure 6 shows the effect of the levels on the databases with respect to the mean time and Fig. 7 relates to the effect of each level on the experiments over the mean time.

Discussion

In this section, the three dependent variables (accuracy, area under ROC and time) will be discussed individually to analyze the results and understand the impacts of inserting an additional layer into the Stacking combination method.

Accuracy Factor

The present findings did not indicate an interaction between database, experiment and level in accuracy. This result suggests that the level used does not produce significant changes (increase or decrease) in accuracy with the experiments performed. It must be emphasized that this lack of interaction was observed when the five databases were used.

In contrast, the present study points to a significant interaction between bases and level. This shows that both the database and the level used influenced the accuracy. A more detailed analysis shows that level-2 presented a significant increase in accuracy when compared to the databases, especially on multi-class databases. This becomes even more evident when looking at the accuracy of the Vehicle base in Fig. 2.

The results of this study show the main effects on the database, experiment and level factors, in isolation. As an interaction between base and level was verified, the main effects of these two factors will not be discussed separately.

The 25 experiments were compared to verify their accuracy. In general, interestingly, experiments with heterogeneous level-1 meta-classifiers, especially Stacking, performed worse than homogeneous ones (Fig. 3). This may be explained by the limited number of level-1 heterogeneous classifiers (2) when compared to WEKA's standard number for homogeneous classifiers (10). Thus, this result corroborates with the literature (Jurek *et al*., 2014), in which a very small number of classifiers (heterogeneous classifiers grouped in pairs, determined by the research) in an ensemble has a worse average performance than those with several components (homogeneous classifiers that followed WEKA's standard parameters, in which 10 machines are created).

Area under ROC Factor

There was no significant interaction between database, experiment and level. Similar to the accuracy topic, this result suggests that the level used does not produce significant changes (increase or decrease) in the area with the experiments performed in the databases.

This time, the significant interaction occurred only between level and experiment. That is, both the level and the experiment influenced the area average. It can be seen from Fig. 4 that with respect to the area, level-2 showed a better result than the other levels in all experiments, while level-1 performed worse than level-0 in some experiments, especially those with a level-1, type II meta-classifier.

The results of this study evidenced main effects of the database, experiment and level factors, separately, on area under ROC factor. Since an interaction between level and experiment was verified, only the database variable was analyzed. Interestingly, the performance of the Vehicle base in the area factor was not greatly influenced by the fact that the base is multi-class. The percentage difference between the lowest performance (BS) and the best performance (IS) concerning the area was higher than 38%.

Time Factor

There was a significant interaction between database, experiment and level with respect to time, as well as all possible interactions between them. This means that training time is highly correlated with these variables. 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. The result was the same for level-1 in relation to level-0 (Fig. 6 and 7).

The present research is limited to the use of heterogeneous classifiers - not limited to Stacking - in solving prediction problems in supervised learning. A study of which classifiers to use at higher levels than the base classifiers, as well as other preprocessing activities would be ways to optimize the time factor in the training of level-2 meta-classifiers, which proved to be the biggest issue in the present research.

Conclusion

This study contributes to the research related to the performance of multi-level ensemble for prediction in classification problems. The use of a meta-classifier that groups two ensembles in a general way promoted an optimization of the models in terms of accuracy and area under ROC, although the training time was much higher in level-2 in relation to the other levels. These results coincide with much of the work related to ensembles with more than two layers.

In conclusion, for future work other performance measures, such as diversity calculations, should be researched for a more thorough analysis of this topic, which could aid the judgment of the most appropriate meta-classifiers and classifiers for each situation.

Funding Information

Funding for this research was provided by CAPES/CNPQ/UFSC.

Author's Contributions

Rodolfo Lorbieski: Is the main investigator in this research which is part of his M.D. work.

Dr. Silvia Modesto Nassar: Is the author responsible for coordination and supervision.

Ethics

This article is original contribution of the author and contains unpublished material. All of the other authors have read and approved the manuscript. There is no ethical issue involved in this article.

References

Abawajy, J. and A.V. Kelarev, 2012. A multi-tier ensemble construction of classifiers for phishing email detection and filtering. Proceedings of the 4th International Conference on Cyberspace Safety and Security, Dec. 12-13, Springer, Melbourne, Australia, pp: 48-56. DOI: 10.1007/978-3-642-35362-8_5

Alpaydin, E., 1998. Introduction to Machine Learning. 1st Edn., MIT Press, ISBN-13: 978-0-262-01243-0.

Ballard, C. and W. Wang, 2016. Dynamic ensemble selection methods for heterogeneous data mining. Proceedings of the 12th World Congress on Intelligent Control and Automation, Jun. 12-15, IEEE Xplore Press, Guilin, China, pp: 1021-1026. DOI: 10.1109/WCICA.2016.7578244

Blake, C.L. and C.J. Merz, 1998. UCI repository of machine learning databases. University of California. Department of Information and Computer Science, Irvine, CA.

Bost, R., R.A. Popa, S. Tu and S. Goldwasser, 2014. Machine Learning Classification over Encrypted Data. Proceedings of the International Conference on Network and Distributed System Security Symposium, Feb. 8-11, Internet Society, San Diego, CA, USA, DOI: 10.14722/ndss.2015.23241

Breiman, L., 1996. Bagging predictors. Machine Learn., 24: 123-140. DOI: 10.1023/A:1018054314350

Chen, Y. and M.L. Wong, 2010. An ant colony optimization approach for stacking ensemble. Proceedings of the 2nd World Congress on Nature and Biologically Inspired Computing, Dec. 15-17, IEEE Xplore Press, Kitakyushu, Japan, pp: 146-151. DOI: 10.1109/NABIC.2010.5716282

Cho, S.B. and J.H. Kim, 1995. Combining multiple neural networks by fuzzy integral for robust classification. IEEE Trans. Syst. Man Cybernet., 25: 380-384. DOI: 10.1109/21.364825

da Costa, N.C. and J.M. Abe, 2000. Paraconsistência em informática e inteligência artificial. Estudos Avançados, 14: 161-174. DOI: 10.1590/S0103-40142000000200012

Delgado, M.F., E. Cernades, S. Barro and D.A. Amorim, 2014. Do we need hundreds of classifiers to solve real world classification problems. J. Mach. Learn. Res., 15: 3133-3181.

Domingos, P., 2012. A few useful things to know about machine learning. Commun. ACM, 55: 78-87. DOI: 10.1145/2347736.2347755

Dorigo, M., M. Birattari and T. Stutzle, 2006. Ant colony optimization. IEEE Comput. Intell. Magaz., 1: 28-39. DOI: 10.1109/MCI.2006.329691

Field, A., 2013. Discovering Statistics Using IBM SPSS Statistics. 4th Edn., SAGE Publications, London, ISBN: 1446249182, pp: 952.

Freund, Y. and R.E. Schapire, 1996. Experiments with a new boosting algorithm. Proceedings of the 13th International Conference on Machine Learning, Jul. 03-06, Morgan Kaufmann Publishers Inc., San Francisco, pp: 148-156.

Friedman, N., D. Geiger and M. Goldszmidt, 1997. Bayesian network classifiers. Mach. Learn., 29: 131-163. DOI: 10.1023/A:1007465528199

Homayouni, H., S. Hashemi and A. Hamzeh, 2010. Instance-based ensemble learning algorithm with stacking framework. Proceedings of the 2nd International Conference on Software Technology and Engineering, Oct. 3-5, IEEE Xplore Press, San Juan, PR, USA, pp: V2-164-V2-169. DOI: 10.1109/ICSTE.2010.5608830

Iba, W. and P. Langley, 1992. Induction of one-level decision trees. Proceedings of the 9th International Conference on Machine Learning, Jul. 01-03, Morgan Kaufmann Publishers Inc., pp: 233-240.

John, G.H. and P. Langley, 1995. Estimating continuous distributions in Bayesian classifiers. Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence, Aug. 18-20, Morgan Kaufmann Publishers Inc., Montréal, Qué, Canada, pp: 338-345.

Jurek, A., A. Jurek, Y. Bi, S. Wu and C. Nugent, 2014. A survey of commonly used ensemble-based classification techniques. Knowl. Eng. Rev., 29: 551-581. DOI: 10.1017/S0269888913000155

Kadkhodaei, H. and A.M.E. Moghadam, 2016. An entropy based approach to find the best combination of the base classifiers in ensemble classifiers based on stack generalization. Proceedigns of the 4th International Conference on Control, Instrumentation and Automation, Jan. 27-28, IEEE Xplore Press, Qazvin, Iran, pp: 425-429. DOI: 10.1109/ICCIAutom.2016.7483200

Kotsiantis, S.B., I. Zaharakis and P. Pintelas, 2007. Supervised machine learning: A review of classification techniques. Proceedings of the Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies, (RPT' 07), IOS Press, pp: 3-24.

Kuncheva, L.I., 2004. Combining Pattern Classifiers: Methods and Algorithms. 1st Edn., John Wiley and Sons, ISBN-10: 0471210781, pp: 376.

Ledezma, A., R. Aler, A. Sanchis and D. Borrajo, 2010. GA-stacking: Evolutionary stacked generalization. Intell. Data Anal.,□14: 89-119.

LeBlanc, M. and R. Tibshirani, 1996. Combining estimates in regression and classification. J. Am. Stat. Assoc., 91: 1641-1650.

McLachlan, G.J., K.A. Do and C. Ambroise, 2005. Analyzing Microarray Gene Expression Data. 1st Edn., John Wiley and Sons, Hoboken, ISBN-10: 0471726125, pp: 368.

Menahem, E., R. Lior and Y. Elovici, 2009. Troika-An improved stacking schema for classification tasks. Inform. Sci., 179: 4097-4122. DOI: 10.1016/j.ins.2009.08.025

Merz, C.J., 1999. Using correspondence analysis to combine classifiers. Mach. Learn., 36: 33-58. DOI: 10.1023/A:1007559205422

Polikar, R., 2006. Ensemble based systems in decision making. IEEE Circuits Syst. Magaz., 6: 21-45. DOI: 10.1109/MCAS.2006.1688199

Prati, R.C., G.E.A.P.A. Batista and M.C. Monard, 2008. Curvas ROC para avaliação de classificadores. Revista IEEE América Latina, 6: 215-222.

Prodromidis, A., P. Chan and S. Stolfo, 2000. Meta-learning in distributed data mining systems: Issues and approaches. Adv. Distributed Parallel Knowl. Discovery, 3: 81-114.

Quinlan, J.R., 1993. C4. 5: Programs for Empirical Learning. 1st Edn., Morgan Kaufmann. San Francisco, CA.

Seewald, A.K. and J. Fürnkranz, 2001. An evaluation of grading classifiers. Proceedings of the 4th International Conference on Advances in Intelligent Data Analysis, Sept. 13-15, Springer Berlin Heidelberg, pp: 115-124. DOI: 10.1007/3-540-44816-0_12

Seewald, A.K, 2002. How to Make Stacking Better and Faster While Also Taking Care of an Unknown Weakness. Proceedings of the 19th International Conference on Machine Learning, Jul. 8-12, Morgan Kaufmann Publishers, San Francisco, pp: 554-561.

Sill, J., G. Takacs, L. Mackey and D. Lin, 2009. Feature-weighted linear stacking.

Sluban, B. and N. Lavrač, 2015. Relating ensemble diversity and performance: A study in class noise detection. Neurocomputing, 160: 120-131. DOI: 10.1016/j.neucom.2014.10.086

Tang, B., Q. Chen, X. Wang and X. Wang, 2010. Reranking for stacking ensemble learning. Proceedings of the 17th International Conference on Neural Information Processing: Theory and Algorithms, Nov. 22-25, Springer, Sydney, Australia, pp: 575-584. DOI: 10.1007/978-3-642-17537-4_70

Ting, K.M. and I.H. Witten, 1999. Issues in stacked generalization. J. Artif. Intell. Res., 10: 271-289. DOI: 10.1613/jair.594

Todorovski, L. and S. Džeroski, 2000. Combining multiple models with meta decision trees. Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery, Sept. 13-16, Springer-Verlag London, pp: 69-84.

Witten, I.H., E. Frank, M.A. Hall and C.J. Pal, 2016. Data Mining: Practical Machine Learning Tools and Techniques. 4th Edn., Morgan Kaufmann, Cambridge, ISBN-10: 0128042915, pp: 261.

Wolpert, D.H., 1992. Neural Networks 5.2.

Wolpert, D.H. and W.G. Macready, 1997. No free lunch theorems for optimization. IEEE Trans. Evolut. Comput., 1: 67-82. DOI: 10.1109/4235.585893

Woźniak, M., G. Manuel and E. Corchado, 2014. A survey of multiple classifier systems as hybrid systems. Inform. Fus., 16: 3-17. DOI: 10.1016/j.inffus.2013.04.006

Xindong, W. and X. Zhu, 2008. Mining with noise knowledge: Error-aware data mining. IEEE Trans. Syst. Man Cybernet., 38: 917-932. DOI: 10.1109/TSMCA.2008.923034