Use of ensemble methods for indirect test of RF circuits: can it bring benefits?

H. El Badawi [1,2], F. Azais [1], S. Bernard [1], M. Comte [1], V. Kerzérho [1], F. Lefevre [2]
[1] LIRMM, University of Montpellier, CNRS, 161 rue Ada, 34095 Montpellier Cedex, France
[2] NXP Semiconductors, 2 Esplanade Anton Philips, 14000 Caen, France

***Abstract* — Indirect testing of analog and RF integrated circuits is a widely studied approach, which has the benefits of relaxing requirements on test equipment and reducing industrial test cost. It is based on machine-learning algorithms to train a regression model that maps an indirect and low-cost measurement space to the performance parameter space. In this work, we explore the benefit of using ensemble learning. Rather than using one single model to estimate targeted parameters, ensemble learning consists of training multiple individual regression models and combining their outputs in order to improve the predictive power. Different ensemble methods based on bagging, boosting or stacking are investigated and compared to classical individual models. Results are illustrated and discussed on three RF performances of a LNA for which we have production test data.**

Keywords: indirect testing, RF integrated circuits, machine-learning algorithms, ensemble methods, test efficiency

I. INTRODUCTION

Checking whether an IC complies with its specifications after the manufacturing process is an essential task to guarantee the quality of devices shipped to the customer. However, it has a strong impact on the total cost of the product. This is particularly true for analog and RF circuits that necessitates the use of sophisticated and expensive test equipment to measure the device specifications. An interesting approach to reduce the testing costs is to adopt an indirect test strategy. The idea is to measure parameters that require only low-cost test resources and to correlate these measurements, called Indirect Measurements (*IMs*), with the device specifications. This correlation is generally established using machine-learning algorithms.

Indirect testing has been widely studied in the literature for many years [1-10]. Numerous aspects have been researched, such as the choice of the learning algorithm, the definition and optimization of appropriate test stimuli, the processing of complex signatures, the use of embedded sensors to gather pertinent information, the exploitation of multi-Vdd test conditions and procedures for the selection of appropriate indirect measurements.

In this paper, we focus on a new kind of learning algorithms, namely ensemble methods, to see whether they can improve the indirect test efficiency. Recently, ensemble methods have gained in popularity and have shown their superiority over classical learning algorithms in several application domains. However, to the best of our knowledge, no specific studies have addressed the use of these methods in the specific context of analog/RF indirect test.

This paper is organized as follows. Section II summarizes the basics of the indirect test approach. Section III gives an overview of the classical methods commonly used to build a regression model and introduces the principle of three types of ensemble methods. The experimental protocol developed to perform the comparative analysis of classical and ensemble methods is then defined in Section IV and case studies are summarized in Section V. Finally, before conclusion, results are presented and discussed in Section VI.

II. INDIRECT TEST PRINCIPLE

The underlying idea of indirect testing is that process variations that affect the device performance also affect indirect parameters. If the correlation between the indirect parameter space and the specification space can be established, then specifications may be verified using only the low-cost indirect signatures. Unfortunately, the relation between these two sets of parameters is complex and cannot be simply identified with an analytic function. The solution commonly implemented uses machine-learning algorithms.

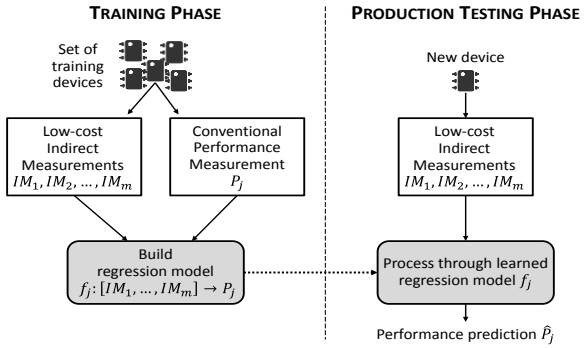

Fig.1. Indirect test synopsis.

The indirect test synopsis is split into two distinct phases, namely training and production testing, as illustrated in Figure 1. The idea is to learn during the training phase the unknown dependency between the low-cost indirect measurements (IM_i) and the conventional performance measurements (P_j). To achieve this, both the specification tests and the low-cost measurements are performed on a set of training devices and a machine-learning algorithm is trained to build regression models that map the indirect parameters space to the performance parameters space. During the production testing phase, only the low-cost indirect measurements are performed, and the specifications of every new device are predicted using the mapping learned in the initial training phase.

III. Regression and Ensemble Methods: Overview

The classical approach to predict the value of a target feature on unseen instances is to build a single regression model. Many different algorithms exist to perform this task. The most popular algorithms used in the context of indirect test are Multiple Linear Regression (MLR), Multi-Adaptive Regression Splines (MARS), and Support Vector Machine (SVM). However, the performances achieved with these algorithms can significantly differ depending on the case study and these is no obvious winner when it comes to choosing a single prediction model.

To cope with the model performance dependency on the size and the structure of the training data, researchers have started to use multiple regression models and aggregate their outcomes to get the final prediction results. The idea is that with an appropriate combination of diverse individual models, it should be possible to exploit the strengths and overcome the weaknesses of the individual models and obtain better overall predictive performance. This approach is called ensemble learning, which refers to the procedures used to train multiple individual regression models (base learners) and combine their outputs in order to improve the stability and the predictive power of the ensemble model. Numerous methods for constructing ensemble models have been proposed in the literature [11], which include parallel and sequential methods, based either on a single type of base learners (homogenous ensemble model) or learners of different types (heterogeneous ensemble model). The general principle of the three most popular methods is described hereafter.

A. Bagging

Bagging stands for bootstrap aggregation. The basic motivation for bagging is to decrease the variance by averaging multiple estimates. The principle consists in using bootstrap resampling (random sampling with replacement) to generate different data subsets from the original training set. Multiple base learners are then trained on these random subsets and the outputs of the base learners are averaged to produce the final estimate. Bagging is a parallel ensemble method that can be applied with any type of prediction model, but the most common application is with decision trees. A very popular algorithm that follows the bagging technique is Random Forest (RF), which uses decision trees as base learners but also randomizes the trees by selecting a random subset of features.

B. Boosting

Boosting is also a method that relies on building multiple base learners on different datasets. However, unlike bagging, boosting is a sequential method. The idea is to incrementally build an ensemble by training at each iteration, a predictor model that will correct its predecessor, by focusing on the under-fitted samples that present a large prediction error. The most popular method of boosting is AdaBoost (Adaptive Boosting). In this technique, the first predictor is learned on the entire dataset, with an equal weight assigned to all training samples. Then at each iteration, the algorithm modifies the weights of the training samples, giving higher weights to under-fitted samples. Finally, results of all predictors are aggregated using a weighting sum to produce the final prediction. Another popular boosting technique is Gradient Boosting. As in the AdaBoost algorithm, a new model is generated at each iteration with the objective to correct the predecessor model; the main difference is that the algorithm tries to fit residual errors made by the previous predictor instead of updating the training samples weights. As for bagging, boosting techniques can be applied with any type of prediction model, but they are usually applied with decision tree methods.

C. Stacking method

Stacking is a heterogeneous ensemble method that exploits a different principle than bagging and boosting techniques, that is based on the concept of a meta learner. The main concept is to use a prediction model to perform the aggregation of multiple base models. Practically, the technique involves two phases. First, multiple base learners are trained on the same dataset, generally using models of different types. The outputs of these base learners are then used to train a higher-level learner, called meta-learner. The two essential differences between stacking and bagging/boosting are: (i) the base models are not obtained by manipulating the training data but by using different model types, and (ii) the aggregation of the different base models is not performed by a simple combiner such as averaging or weighted sum but by a prediction model.

IV. Protocol of Experiments

In order to explore whether ensemble methods can bring benefits over classical methods, the experimental protocol depicted in Figure 2 has been defined. It involves 4 main phases that consist in (i) population partitioning, (ii) feature selection, (iii) model construction and (iv) test efficiency evaluation. Details on these different phases are given hereafter.

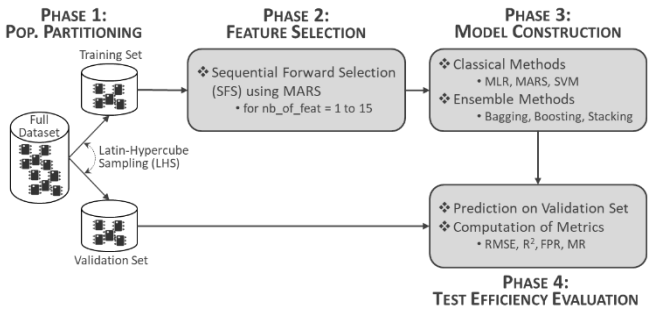

Fig.2. General overview of the experimental protocol.

The first phase involves the partitioning of the population into two different sets. The first one will be used to train the prediction model and the second one with be used to evaluate the constructed model. Note that it is important to evaluate the performance of the model on different instances than the ones used for training, to verify the generalization ability of the model and avoid issues related to overfitting. In this work, we use Latin Hypercube Sampling (LHS) to perform the partitioning. This technique ensures that both the training and validation sets have similar statistical characteristics.

The second phase consists in selecting pertinent *IMs* among the set of available measurements. This problem of selecting a subset of features among a larger set is a recurrent problem in the field of machine-learning, known as feature selection. Various algorithms have been proposed, which can be divided into three categories, namely filters, wrappers and embedded methods [12]. In the context of indirect testing, the solution

commonly employed is a wrapper method based on Sequential Forward Selection (SFS). The procedure starts by building a regression model for each available *IM* and selecting the *IM* that generates the model with the minimum prediction error (lowest *RMSE* score). At the second iteration, a regression model is built for each pair of *IMs* that includes the previously selected *IM*; the pair that gives the best model is then selected. The process then continues with triplets and so on, until a stopping criterion is reached, for instance the number of selected *IMs* reaches a maximum target limit. In this work, we have implemented such a procedure using the MARS algorithm to build the regression models and limiting the search to a maximum of 15 features.

The third phase consists in building a regression model using the features selected in the previous phase. In this work, we have implemented three classical models, namely MLR, MARS and SVM, and five ensemble models belonging to the different categories presented in section II:

- *Bagging*: one ensemble model is built from ten MARS models trained in parallel on ten bootstrap samples of the original training set.
- *Boosting*: one ensemble model is built using the AdaBoost algorithm with a sequential training of ten MARS models, and one ensemble model is built using the Gradient Boosting algorithm with 100 decision trees.
- *Stacking*: one ensemble model is built using the three classical models (MLR, MARS, SVM) as base models, and one ensemble model is built by adding a Random Forest (bagging algorithm applied on 300 decision trees) as 4th base model. In both cases, the aggregation of the base learners is realized by the MARS algorithm.

Finally, the last phase concerns the evaluation of the test efficiency. In this phase, all the models built in the previous phase are used to perform prediction of devices of the validation set. Several metrics are then computed to evaluate the performance of the different models.

The most commonly-used metric to evaluate the accuracy of a model is the *RMSE*, which is a measure of the average prediction error:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2} \quad (1)$$

where y_i is the actual performance value of the ith instance, $\hat{y}_i$ is the predicted performance value of the ith instance, and n is the number of instances in the validation set.

Another very common metric is the coefficient of determination R^2, which is a measure of the goodness of fit of a model. This score is computed as:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (2)$$

where $\bar{y}$ is the mean of the observed data.

The R^2 score is directly related to the *RMSE* score with:

$$R^2 = 1 - \frac{RMSE^2}{{\sigma_y}^2} \quad (3)$$

where ${\sigma_y}^2$ is the variance of the observed data.

The interest of the R^2 score is that it permits comparison across different variables since it is a normalized score that ranges between 0 and 1. In contrast, the *RMSE* score is expressed in units of the variable of interest and its value is dependent of the variable scale; comparison of *RMSE* scores between different variables is therefore invalid. In this paper, we will use the R^2 score to present and comment results.

Another metric has been suggested in [13], which permits to quantify the prediction reliability of a model. This metric, called Failing Prediction Rate (*FPR*), expresses the percentage of circuits with a prediction error that exceeds the conventional measurement uncertainty ε_{meas}:

$$FPR = \frac{1}{n}\sum_{i=1}^{n}(|y_i - \hat{y}_i| > \varepsilon_{meas}) \quad (4)$$

with $(|y_i - \hat{y}_i| > \varepsilon_{meas}) = 1$ if true
$(|y_i - \hat{y}_i| > \varepsilon_{meas}) = 0$ otherwise

Lastly, if the test limits are available, we can compute another metric called the Misclassification Rate (*MR*). This metric simply expresses the ratio of misclassified circuits with respect to the total number of circuits.

V. Case Studies

The test vehicle is a Low-Noise Amplifier (LNA) for which we have production test data on 3,850 devices. More precisely, test data include the conventional measurements of three RF specification performances, namely the gain, the output power at 1dB compression point (P1dB) and the third-order intercept point (IP3). Test data also include 79 low-cost indirect measurements which correspond to DC voltages on internal nodes (the device is equipped with an internal DC bus and internal DC probes) and DC signatures delivered by built-in process monitors. The distribution of the three RF performances is illustrated in Figure 3 and the main characteristics are summarized in Table I.

A first general comment is that the RF performances under investigation do not exhibit a Gaussian distribution. Another important point to highlight is that the three RF performances correspond to three different situations:

- For the gain, we observe a very tight distribution with an excursion of only 0.5dB and a standard deviation that is even smaller than the measurement uncertainty. The test limits are located far away outside the distribution of available samples; as a consequence, there are no bad circuits with respect to the gain performance.
- For P1dB, we observe a slightly larger distribution with an excursion of 1.5dBm and a standard deviation that is around twice the measurement uncertainty. For this performance, the lower test limit is located very close to the left tail of the distribution; three samples have a P1dB performance inferior to this limit, which means that only a negligible portion of the population (less than 0.1%) are bad circuits with respect to the P1dB performance.
- For IP3, we observe a significantly larger distribution with an excursion of more than 3.5dBm but a standard deviation that is only about 1.5 times the measurement uncertainty. For this performance, the lower test limit falls within the distribution of available samples; 807 samples exhibit an IP3 performance inferior to this limit, which means that around 20% of the population are bad circuits with respect to the IP3 performance.

These three RF performances constitute the practical case studies considered in this paper. It will be particularly interesting

to see how the indirect test approach will behave with respect to these three different situations.

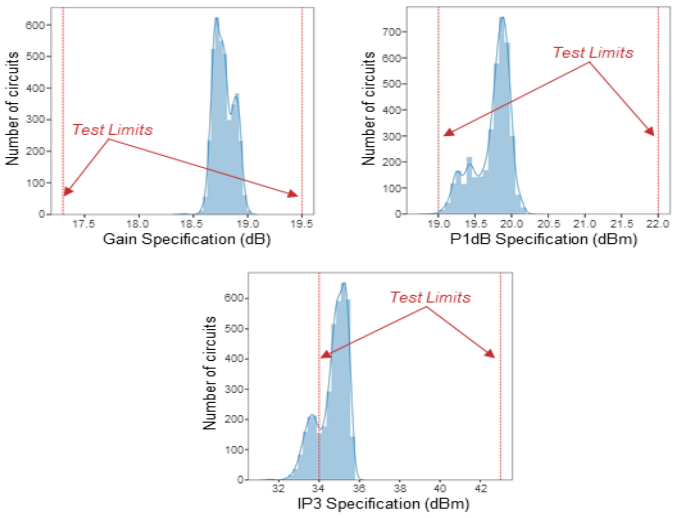

Fig.3. Distribution of the three RF performances under investigation.

TABLE I. SUMMARY OF THE MAIN CHARACTERICTICS FOR THE THREE RF PERFORMANCES UNDER INVESTIGATION

	RF Performance		
	Gain	***P1dB***	***IP3***
Mean value	17.78dB	19.74dBm	34.68dBm
Std deviation	0.09dB	0.24dBm	0.72dBm
Meas. uncertainty	0.1dB	0.1dB	0.5dB
Test limits	[17.3dB;19.5dB]	[19dBm;22dBm]	[34dBm;43dBm]
# good circuits	3850	3847	3043
# bad circuits	0	3	807

VI. RESULTS

The experimental protocol presented in Section IV has been applied on the above 3 case studies. Results are first commented for each RF performance to be predicted; then a global summary and a discussion is provided. Note that all metrics are evaluated on the validation set composed of 1,850 devices.

A. *Prediction of gain (G)*

Figure 4 summarizes the comparison between classical and ensemble methods for the prediction of the gain specification. More precisely, it reports the evolution of R^2 and *FPR* scores (evaluated on the validation set) with respect to the number of features used in the regression model for the different methods.

Several comments arise from the analysis of these graphs. Regarding classical methods, there is a clear advantage to models generated by MARS algorithm compared to MLR and SVM. The best solution is obtained using MARS model built with nine features, with a R^2 score of 0.65 and a *FPR* score of 2.9%. Regarding ensemble methods, models generated using stacking are more performing than models generated using boosting or bagging. The best solution corresponds to an ensemble model built with nine features that combines MLR, MARS, SVM and Random Forest (RF) models. This model permits to reach a R^2 score of 0.72 and a *FPR* score of 1.5%.

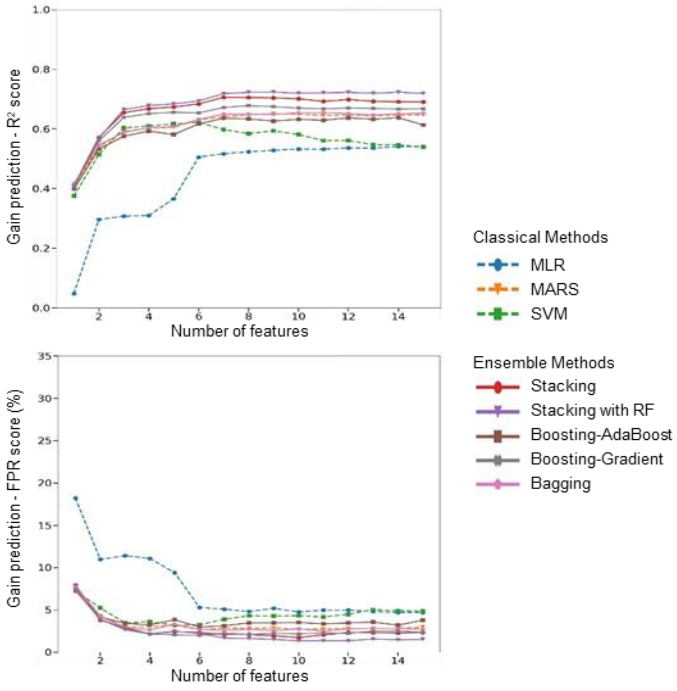

Fig.4. Comparison of R^2 and *FPR* scores achieved for gain prediction using classical and ensemble methods.

More generally for the gain specification, these results show that it is possible to obtain a benefit by using ensemble methods compared to classical methods, especially when stacking is applied. In particular, compared to the best solution achieved using a classical method (MARS model in this case), it is possible to obtain a 10% improvement in the R^2 score and a reduction in the *FPR* score by a factor of almost two.

B. *Prediction of output power at 1dB compression point*

Figure 5 summarizes the comparison between classical and ensemble methods for the prediction of the P1dB specification, in terms of R^2 and *FPR* scores achieved on the validation set by using the different methods. Regarding classical methods, unlike the gain specification, we can observe that SVM models are more powerful than MARS or MLR models, especially when only a limited number of features are used; results are then almost comparable when a higher number of features are used. The best solution is obtained using an SVM model built with eight features, with a R^2 score of 0.85 and a *FPR* score of 12.3%. Regarding ensemble methods, we observe a similar trend than for the gain specification, i.e. models generated using stacking appear more powerful than models generated using boosting or bagging. The best solution corresponds to an ensemble model built with twelve features that combines MLR, MARS, SVM and Random Forest models. This model permits to reach a R^2 score of 0.87 and a *FPR* score of 11.2%.

Globally for the P1dB specification, there is a slight benefit in using ensemble models generated with stacking compared to the best model generated with a classical method (SVM model in this case), with a more limited improvement than for the gain specification. In this case, the R^2 score is improved by only 2.2% and the *FPR* score remains in the same range.

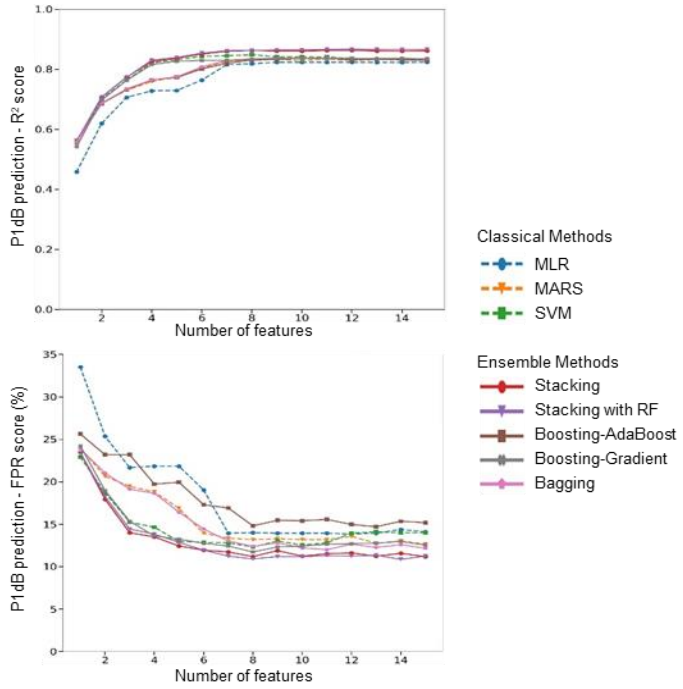

Fig.5. Comparison of R^2 and *FPR* scores achieved for P1dB prediction using classical and ensemble methods.

C. *Prediction of third order intercept point (IP3)*

Figure 6 summarizes the comparison between classical and ensemble methods for the prediction of the IP3 specification, in terms of R^2 and *FPR* scores achieved on the validation set by using the different methods.

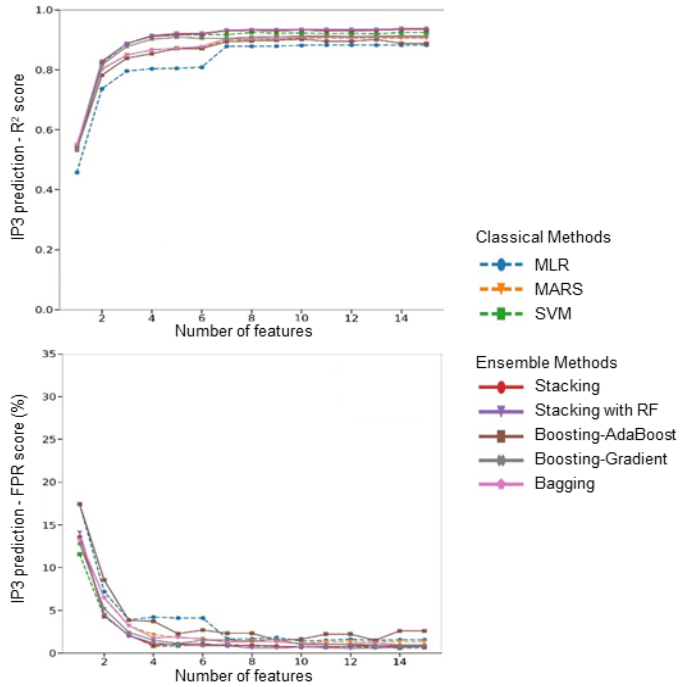

Fig.6. Comparison of R^2 and *FPR* scores achieved for IP3 prediction using classical and ensemble methods.

In case of the IP3 specification, a similar behavior than for the P1dB specification is observed, i.e. the more powerful models obtained with classical methods are SVM models and the more powerful models generated with ensemble methods are models generated with stacking. However, the benefit brought by the use of ensemble methods is not evident in this case. Indeed, the best solution obtained with a classical method is a SVM model built with 14 features that exhibits a R^2 score of 0.93 and a *FPR* score of 0.6%, while the best solution obtained with an ensemble method is a stacked model that exhibits a R^2 score of 0.94 and a *FPR* score of 0.7%. There is therefore a small improvement of the R^2 score but a small degradation of the *FPR* score.

D. *Summary and discussion*

Table II summarizes the best results obtained using either classical or ensemble methods for the three RF specifications. A first general comment is that the use of ensemble methods, and in particular, ensemble methods based on stacking, permits to obtain an improvement in model accuracy for the three specifications. However, the level of improvement is different in each case and seems to depend on the level of accuracy that can be reached by a single model. These results actually tend to indicate that the benefit of using the ensemble model reduces as the accuracy reached by a single model increases.

Still, an important point to underline is that when using classical methods, the type of model that gives the best results differs depending on the specification (MARS or SVM). In contrast, ensemble models built with stacking always lead to the best results. It is an interesting feature to have a solution able to handle a variety of different situations.

Hence globally, the use of ensemble models that are built using stacking appears to be an interesting option. Moreover, it should be mentioned that we didn't explore all the possibilities offered by stacking. Further improvements might be obtained, for instance by including other types of model as base learners which will add more diversity to the model collection, or by changing the type of the aggregating model (MARS model in this study).

TABLE II. COMPARISON BETWEEN CLASSICAL AND ENSEMBLE METHODS: SUMMARY OF BEST RESULTS FOR THE THREE RF PERFORMANCES

	Best solution selected from max(R^2) on validation set					
	RF Perf	***Model***	***R^2 (*)***	***FPR (*)***	***MR (*)***	***# feat.***
Classical method	Gain	MARS	0.65	2.86%	0%	9
	P1dB	SVM	0.85	12.32%	0.1%	8
	IP3	SVM	0.93	0.59%	4.2%	14
Ensemble method	Gain	Stack+RF	0.72	1.51%	0%	9
	P1dB	Stack+RF	0.87	11.24%	0.1%	12
	IP3	Stack+RF	0.94	0.70%	4.2%	14

(*) Score computed on validation set

More generally, this study also opens the question on what a pertinent metric is for indirect test efficiency evaluation. Indeed, results show that performances significantly vary depending on the considered specification and the considered metric.

First, it appears that there is no direct relationship between the accuracy of a model evaluated in terms of R^2 score and its reliability evaluated in terms of *FPR* score. Indeed, for the gain specification, the best model leads to a relatively low accuracy

with a R^2 around 0.7 but a fairly good reliability with less than 3% of the devices that exhibit a prediction error which exceeds the classical measurement uncertainty. In contrast for the P1dB specification, we can obtain a reasonable accuracy with a R^2 around 0.85, but a relatively low reliability with more than 10% of the devices that exhibit a prediction error which exceeds the classical measurement uncertainty. Finally for the IP3 specification, we can have both a good accuracy and a good reliability with a R^2 higher than 0.9 and less than 1% of the devices that exhibit a prediction error which exceeds the classical measurement uncertainty.

Then, it should be highlighted that it is difficult to establish a link between the accuracy or reliability of a model and the misclassification rate. Indeed, the misclassification rate strongly depends on the location of the test limits with respect to the distribution of available samples. For instance, in case of the gain specification, the test limits are located far away from the distribution; despite the relatively low accuracy of the models, all devices are correctly classified as good circuits and a perfect misclassification rate of 0% is achieved. In contrast for the IP3 specification, the low test limit falls within the distribution; so even if we have models with very good accuracy and reliability, around 4% of the circuits are misclassified, which can be considered as a non-negligible number. Yet, this result should be mitigated by the fact that all the misclassified circuits are located relatively close to the test limit, as illustrated in Figure 7, which highlights the location of misclassified circuits on the global IP3 distribution of the validation set. In fact, the computed misclassification rate might not be fully representative of the indirect test efficiency because it does not take into account the uncertainty that can affect the conventional measurement.

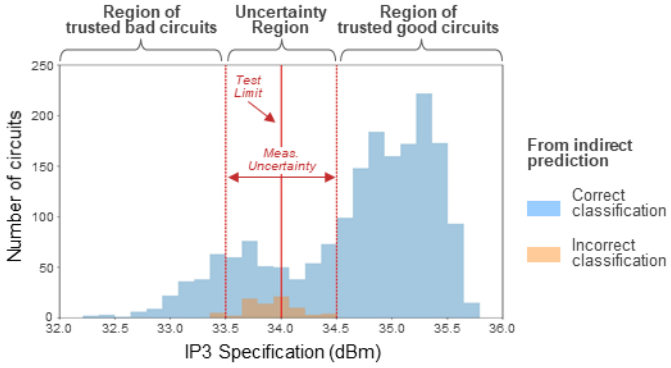

Fig.7. Illustration of misclassified devices by using the "Stack+RF" ensemble model for IP3 specification.

To further explain this point, let us analyze more in details Figure 7. When the measurement uncertainty is considered, it exists a region around the test limit where the circuits might be either good or bad circuits; only circuits outside this region can be trustfully defined as good or bad circuits by the conventional method. For our practical case on the IP3 specification, among the 1,850 circuits of the validation set, 400 are within the uncertainty region, 180 are trusted bad circuits and 1,270 are trusted good circuits. Now looking at the results of the indirect test, it appears that almost all the misclassified circuits are located within the uncertainty region, only five circuits being outside this region. The computed misclassification rate of 4% does not permit to reveal this situation. A more pertinent metric might be to compute the coverage of trusted classifications, i.e. the percentage of circuits that have a correct decision with the indirect prediction among the number of circuits that have a certain decision with the conventional measurement. For our case study, 1,450 circuits have a certain decision with the conventional measurement and 1,445 of them have a correct decision with the indirect prediction, which corresponds to a very good coverage of 99.66%. We believe that this metric can be more representative of the indirect test efficiency than the misclassification rate classically computed.

VII. Conclusion

In this paper, we have explored the use of ensemble methods for indirect test of RF circuits. Different ensemble methods based on bagging, boosting and stacking have been investigated and compared to classical individual models, namely MLR, MARS and SVM models. An experimental protocol has been developed and applied for the prediction of three RF performances on a low-noise amplifier for which we have production test data. Results have demonstrated the superiority of ensemble models built with stacking compared to ensemble models built with bagging or boosting. Results have also shown that such models can outperform the classical individual models, both in terms of accuracy and reliability, and that they offer a superior predictive power over a variety of different situations.

Acknowledgment

This work has been carried out under the framework of PENTA-EUREKA project "HADES: Hierarchy-Aware and secure embedded test infrastructure for Dependability and performance Enhancement of integrated Systems".

References

[1] P. N. Variyam et al., "Prediction of analog performance parameters using fast transient testing," IEEE Trans. On Computer-Aided Design of Integrated Circuits and Systems, Vol. 21, no. 3, pp. 349–361, 2002.

[2] S. Ellouz et al., "Combining internal probing with artficial neural networks for optimal RFIC testing", Proc. IEEE Int'l Test Conference (ITC), p.9, 2006.

[3] L. Abdallah et al., "Sensors for built-in alternate RF test," Proc. IEEE European Test Symposium (ETS), pp. 49-54, 2010.

[4] M.J. Barragan, et al., "Improving the Accuracy of RF Alternate Test Using Multi-VDD Conditions: Application to Envelope-Based Test of LNAs", Proc. IEEE Asian Test Symposium (ATS), pp. 359-364, 2011.

[5] H. Ayari, et al., "Making predictive analog/RF alternate test strategy independent of training set size", Proc. IEEE Int'l Test Conference (ITC), p.9, 2012.

[6] H. Stratigopoulos and S. Mir, "Adaptive Alternate Analog Test", IEEE Design & Test of Computers, Vol. 29, no. 4, pp. 71-79, 2012.

[7] M.J. Barragan, G. Leger, "Efficient selection of signatures for analog/RF alternate test", Proc. European Test Symposium (ETS), p.6, 2013.

[8] S. Larguech, et al., "A Framework for Efficient Implementation of Analog/RF Alternate Test with Model Redundancy", Proc. IEEE Computer Society Annual Symp. on VLSI (ISVLSI), pp. 621-626, 2015.

[9] M. J. Barragan, G. Leger, "A Procedure for Alternate Test Feature Design and Selection", IEEE Design & Test, Vol. 32, no. 1, pp. 18-25, 2015.

[10] H. Stratigopoulos, "Machine learning applications in IC testing", Proc. IEEE European Test Symposium (ETS), p.10, 2018.

[11] Z-H. Zhou, "Ensemble Methods: Foundations and Algorithms", Chapman and Hall/CRC, Machine Learning & Pattern Recognition Series, 1st Edition, 2012.

[12] I. Guyon, A. Elisseeff, "An introduction to variable and feature selection", Journal of Machine Learning Research, Vol. 3, pp. 1157-1182, 2003.