

# Safety-Relevant AI-Based System Robustification with Neural Network Ensembles

Adrià Aldomà<sup>†,‡</sup>, Axel Brando<sup>†</sup>, Francisco J. Cazorla<sup>†</sup>, Jaume Abella<sup>†</sup>

<sup>†</sup>Barcelona Supercomputing Center (BSC), Spain

<sup>‡</sup>Universitat Politècnica de Catalunya (UPC), Spain

**Abstract**—Functional safety requirements of AI-based safety-critical applications challenge AI models, whose accuracy can be limited. In this paper, we show how using several cooperative deep learning (DL) models helps to raise global accuracy and reject making a prediction when confidence is below a pre-established threshold.

## I. INTRODUCTION

Deep Learning (DL) algorithms are increasingly needed in safety-critical systems as they become more autonomous [1], [2]. However, the limited accuracy of those DL models challenges their use in safety-critical systems [3].

In the context of DL, *ensemble learning* has been considered to enhance performance, increase robustness, and improve generalization [4]–[6]. Ensembles build on diverse models to capture different aspects of the data and provide complementary insights, leading to more robust and accurate predictions [7], [8]. However, how to architect ensembles in the context of safety-relevant AI-based systems leveraging diversity has not been studied so far [3]. In particular, it is unclear how to specialize multiple models, and how to organize them, so that erroneous predictions are reduced, and diversity can be tailored to mitigate weaknesses of some models.

This paper proposes a scheme to architect ensembles of DL models fitting the needs of safety-critical AI-based systems. We exploit some key ML concepts like *boosting* algorithms [9], [10] and *learning to defer* [11] to properly specialize DL models and organize them hierarchically. Our solution relies on the use of meta-classifiers that decide whether a specific model performs well for a given input data in a specific context and, if it is not the case, they forward the prediction task to a subsequent model or group of models. Moreover, our solution allows the system abstaining from making a prediction if confidence is low, which is particularly amenable for safety-critical systems to trigger safety measures rather than taking unreasonable risks.

## II. RELATED WORK

There are some popular techniques for creating ensembles: *bagging*, which stands for “bootstrap aggregation” [12], [13], and *boosting* [9], [10]. In the case of boosting, of particular relevance for our work, models are trained sequentially with errors given in a round (iteration) having more emphasis in posterior rounds, i.e. DL models are specialized on data mispredicted by previous models [14]. Predictions of the

<sup>1</sup>The research leading to these results has received funding from the European Union’s Horizon Europe Programme under the SAFEXPLAIN Project (www.safexplain.eu), grant agreement num. 101069595. This work has also been supported by the Project PLEC2023-010240 funded by MICIU/AEI/10.13039/501100011033. Authors thank the support given to the Research Group SSAS (Code: 2021 SGR 00637) by the Research and University Department of the Generalitat de Catalunya.

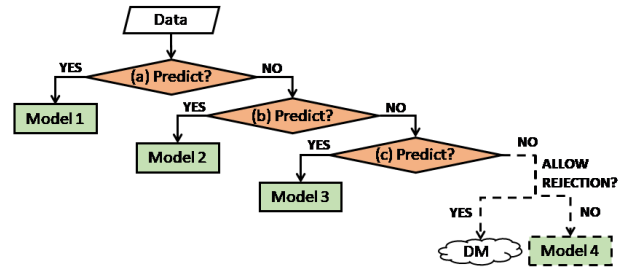


Fig. 1. Scheme of the proposed ensemble framework. A rejection option may be allowed or not at the last stage.

different models can be averaged, or the most repeated output be chosen [15].

In the case of *cascade* ensembles [16], every new model receives as input the instance and all the previous models’. We get inspiration from this approach to build later our proposed ensemble approach.

## III. ENSEMBLES TO INCREASE DIVERSITY

**Proposed Ensemble Framework.** A schematic of our proposed approach to build ensembles is shown in Figure 1. First, data is delivered to meta-classifier (a), which is indeed an DL model trained to determine whether it is appropriate using *Model 1* (previously trained) or not by providing an estimate of the probability of it being correct. If the meta-classifier’s prediction exceeds a given threshold<sup>2</sup>, then *Model 1* is regarded as appropriate and it will deliver the prediction. Else, data is forwarded to meta-classifier (b), which is also trained with all the training data, but targeting the specialization of *Model 2*. The process repeats to decide whether *Model 2* is appropriate to perform the prediction. This is repeated until either a model is enforced to predict (i.e. rejection option is not allowed), or the prediction is rejected and control is transferred to an external decision maker, denoted as DM in Figure 1 (i.e. typically triggering a safety measure).

**Rationale behind the Proposed Ensemble Framework.** Using multiple cooperative DL models helps mitigating lack of confidence by specializing different models for different input data subsets. One technique that facilitates this process is *rejection learning* [17], which allows models not to raise a prediction if the expected loss<sup>3</sup> is not low enough [11]. However, rejection learning fails to incorporate the auxiliary expertise from other agents (models) to manage rejected predictions. Alternatively, *learning to defer* [11] takes into account the confidence of an auxiliary expert (e.g., *Model 2* in Figure 1)

<sup>2</sup>Using this approach, we convert the continuous  $[0, 1]$  output of the meta-classifier into a binary one. In the rest of this work, we set thresholds above 0.9 for bias reasons, so the corresponding model is allowed to predict if the meta-classifier’s prediction is above 0.9, and the prediction deferred otherwise.

<sup>3</sup>The loss function of an ML model evaluates the similarity between the predicted output and the true value.

regarding the same prediction prior to rejecting a prediction, which occurs solely when the previous models (the main and the auxiliary ones) have low confidence (e.g., using *Model 3* in Figure 1 when *Models 1 and 2* have low confidence). We adapt *learning to defer* by training an auxiliary model specifically on those input data rejected by previous models using *meta-classifiers*.

**Design and Training of the Ensemble.** We train the first model,  $h_1$  (e.g., *Model 1* in Figure 1), with all input data and the first meta-classifier,  $\gamma_1$  (meta-classifier (a) in Figure 1), also with all input data, with the latter learning the behavior of the former. Inference is performed with all input data, and the next subset of data to be forwarded to  $h_2$  is obtained with  $\gamma_1$ , retaining only rejected data. The following model,  $h_k$  (where  $k > 1$ ), is trained with the rejected data by all previous meta-classifiers, namely  $\gamma_1, \gamma_2, \dots, \gamma_{k-1}$ . Note that we let the cascade process happen automatically so the meta-classifiers are the ones filtering the data used to feed the subsequent models. The meta-classifier used to train model  $h_k$ , namely  $\gamma_{k-1}$ , is retrained to satisfy a *learning to defer* approach between  $h_{k-1}$  and  $h_k$ , so that it weights not only whether  $h_{k-1}$  is good or bad predicting some data, but also how  $h_k$  behaves predicting such data. We train the following meta-classifier,  $\gamma_k$ , with all input data again so it can learn the behavior of  $h_k$ , not only on the remaining data at that stage, but on all of them and work as a rejector.

#### IV. EVALUATION

**Setup.** The architecture of the neural networks used for the predictive models is a (32, 16, 8) dense network, and a  $5 \times 5$  convolution layer with a  $2 \times 2$  max pooling layer before a (256, 128, 64, 32, 16) dense network for the meta-classifiers. Since we tackle classification problems, the models are trained with a multi-class cross entropy and the meta-classifiers, with a binary cross-entropy. Note that our approach is model-agnostic and other models, including any with, for instance, higher accuracy, could also be used instead.

For the training process, we used Adam’s optimizer [18]. We utilized the default parameters ( $\alpha = 0.001$ ,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ ) and employed a batch size of 32. For properly training the system we employed early stopping [19] based on the loss on the validation set, stopping the training if the loss did not decrease more than 0.0001 for more than 3 epochs.

**Case Study.** Following the structure shown in Figure 1, we have created an ensemble for a case study based on the German Road Sign dataset [20], for which we are taking 35,209 training and 12,630 testing images of 43 classes of traffic signals. Since this set is imbalanced, there is great probability of bias towards the most common classes, fact that is reinforced due to similarities between some signals.

The first model,  $h_1$ , is trained to predict accurately all classes, and the first meta-classifier,  $\gamma_1$ , determines whether  $h_1$  should be used, or it is better to defer classification to subsequent models. The rest of the meta-classifiers and models follow an analogous approach, with models  $h_2$  and  $h_3$  trained with those images rejected by all previous meta-classifiers (i.e.  $h_2$  is trained with images rejected by  $\gamma_1$ , and  $h_3$  with images rejected by  $\gamma_1$  and  $\gamma_2$ ).

Table I presents the results using the 12,630 images provided for inference in this dataset. The top part of the table shows the combination of results given by  $\gamma_1$  and  $h_1$ . In particular, green and red cells (values 6658 and 700) correspond to the correct and erroneous classifications made by  $h_1$  for those inputs that  $\gamma_1$  allows  $h_1$  to classify. The white cells right below

TABLE I  
CONFUSION MATRICES OF THE THREE STAGES OF THE GERMAN ROAD SIGN EXAMPLE. WHITE CELLS ARE DEFERRED TO THE SUBSEQUENT MODEL. GRAY CELLS REPRESENT INSTANCES FOR WHICH THERE IS A LACK OF CONFIDENCE AND ARE DEFERRED TO AN EXTERNAL EXPERT.

		Accuracy $h_1$							
		1		0					
$\gamma_1$	1	6658		700					
	0	1234		4038					
		Accuracy $h_2$							
		1		0					
$\gamma_2$	1	152	19	996	142				
	0	201	862	429	2471				
		1		0		1		0	
$\gamma_3$	1	24	5	72	25	52	15	140	97
	0	37	135	113	652	75	287	187	2047

(values 1234 and 4038) correspond to the inputs that  $\gamma_1$  defers for subsequent models, being those on the left the ones that  $h_1$  would have classified correctly if it had to classify them (1234), and those on the right the ones that  $h_1$  would have been erroneous (4038). Overall, if instead of our ensemble we had used only  $h_1$  to make all predictions, we would have obtained 7892 (6658+1234) correct classifications and 4738 erroneous ones (so 62.5% correct and 37.5% erroneous). Instead, with our ensemble, at this first level we obtain 52.7% correct classifications, 5.6% erroneous ones, and 41.7% inputs are deferred for classification by subsequent models.

Results right below the label *Accuracy  $h_2$*  correspond to the second model ( $h_2$ ). Here,  $\gamma_2$  receives the 5272 (1234+4038) inputs rejected by  $\gamma_1$ , and selects which of those should be classified by  $h_2$ . This information is shown separately for rejections that would have been properly classified by  $h_1$  on the left, and those that would have been erroneously classified on the right. For instance, out of the 4038 rejections of  $\gamma_1$  that would have been misclassified by  $h_1$ ,  $\gamma_2$  allows  $h_2$  predicting 1138, out of which 996 are correctly classified and 142 erroneously classified. The remaining 2900 are rejected and passed to the subsequent model.

Overall, the correct predictions of the ensemble correspond to the addition of the green cells, and errors to the addition of red cells. Inputs rejected by all meta-classifiers correspond to the addition of all values in the bottom gray row. Hence, the ensemble provides 64.1% correct predictions, only 7.9% erroneous ones, and is able to reject 28.0% inputs for which the ensemble itself detects that its confidence predicting those is too low. These values compare against 62.5%, 37.5% and 0.0% that using a single model only would provide.

If we enforce our ensemble to always classify images, hence ignoring  $\gamma_3$  and making  $h_3$  classify all inputs rejected by  $\gamma_2$ , then the accuracy of the ensemble would be 67.3%, which would still be better than the original 62.5%. However, in the context of safety-related systems, it is far more useful being able to identify scenarios with low confidence where safety measures can be taken such as, for instance, triggering system level safety measures (e.g., turning on/off lights pointing to the road sign, decreasing driving speed, etc.).

#### V. CONCLUSIONS

This paper presents a flexible ensemble framework allowing to raise global accuracy levels and to identify when predictions would be given with too low confidence to be trusted. Our evaluation for a road sign classification case study shows the strengths of our approach, providing higher global accuracy, and rejecting to predict inputs that would be mispredicted in their vast majority.

## REFERENCES

- [1] A. Hevelke and J. Nida-Rümelin, "Responsibility for crashes of autonomous vehicles: an ethical analysis," *Science and engineering ethics*, vol. 21, no. 3, pp. 619–630, 2015.
- [2] J. Perez-Cerrolaza, J. Abella, M. Borg, C. Donzella, J. Cerquides, F. J. Cazorla, C. Englund, M. Tauber, G. Nikolakopoulos, and J. L. Flores, "Artificial intelligence for safety-critical systems in industrial and transportation domains: A survey," *ACM Comput. Surv.*, vol. 56, no. 7, apr 2024. [Online]. Available: <https://doi.org/10.1145/3626314>
- [3] A. Brando, I. Serra, E. Mezzetti, F. J. Cazorla, J. Pérez-Cerrolaza, and J. Abella, "On neural networks redundancy and diversity for their use in safety-critical systems," *Computer*, vol. 56, no. 5, pp. 41–50, 2023. [Online]. Available: <https://doi.org/10.1109/MC.2023.3236523>
- [4] D. Kondratyuk, M. Tan, M. Brown, and B. Gong, "When ensembling smaller models is more efficient than single large models," *arXiv preprint arXiv:2005.00570*, 2020.
- [5] B. Biggio, I. Corona, G. Fumera, G. Giacinto, and F. Roli, "Bagging classifiers for fighting poisoning attacks in adversarial classification tasks," in *Multiple Classifier Systems: 10th International Workshop, MCS 2011, Naples, Italy, June 15-17, 2011. Proceedings 10*. Springer, 2011, pp. 350–359.
- [6] S. Sinha, H. Bharadhwaj, A. Goyal, H. Larochelle, A. Garg, and F. Shkurti, "Diversity inducing information bottleneck in model ensembles," *arXiv preprint arXiv:2003.04514*, 2020.
- [7] L. A. Ortega, R. Cabañas, and A. Masegosa, "Diversity and generalization in neural network ensembles," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2022, pp. 11 720–11 743.
- [8] D. Wood, T. Mu, A. Webb, H. Reeve, M. Lujan, and G. Brown, "A unified theory of diversity in ensemble learning," *arXiv preprint arXiv:2301.03962*, 2023.
- [9] R. E. Schapire, "The strength of weak learnability," *Machine learning*, vol. 5, pp. 197–227, 1990.
- [10] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *European conference on computational learning theory*. Springer, 1995, pp. 23–37.
- [11] D. Madras, T. Pitassi, and R. Zemel, "Predict responsibly: improving fairness and accuracy by learning to defer," *Advances in Neural Information Processing Systems*, vol. 31, 2018.
- [12] L. Breiman, "The little bootstrap and other methods for dimensionality selection in regression: X-fixed prediction error," *Journal of the American Statistical Association*, vol. 87, no. 419, pp. 738–754, 1992.
- [13] —, "Bagging predictors," *Machine learning*, vol. 24, pp. 123–140, 1996.
- [14] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *Journal of artificial intelligence research*, vol. 11, pp. 169–198, 1999.
- [15] C. Zhang and Y. Ma, *Ensemble machine learning: methods and applications*. Springer, 2012.
- [16] N. García-Pedrajas, D. Ortiz-Boyer, R. del Castillo-Gomariz, and C. Hervás-Martínez, "Cascade ensembles," in *Computational Intelligence and Bioinspired Systems: 8th International Work-Conference on Artificial Neural Networks, IWANN 2005, Vilanova i la Geltrú, Barcelona, Spain, June 8-10, 2005. Proceedings 8*. Springer, 2005, pp. 598–603.
- [17] C. Cortes, G. DeSalvo, and M. Mohri, "Learning with rejection," in *Algorithmic Learning Theory: 27th International Conference, ALT 2016, Bari, Italy, October 19-21, 2016, Proceedings 27*. Springer, 2016, pp. 67–82.
- [18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.
- [19] F. Chollet *et al.*, "Keras," <https://keras.io>, 2015.
- [20] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural Networks*, no. 0, pp. –, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0893608012000457>