ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

Cross-tested Aggregated Hold-out

Joseph Rynkiewicz[1]

Universté de Paris 1 - SAMM
90 rue de Tolbiac - France

Abstract. Model selection is central to machine learning. Recently, Aggregated Hold-Out (Agghoo) has been shown to have optimal theoretical properties. This procedure mixes cross-validation and aggregation and is undoubtedly one of the most popular methods for obtaining state-of-the-art predictors. However, suppose that we have to choose the best family of deep learning models from a wide range of architectures. In that case, it isn't easy to know which one will perform best during aggregation, especially if the sample size is too small. The purpose of this paper is to explore, from both a practical and a theoretical perspective, what can happen with this aggregation method and to provide a method for assessing the performance of the aggregated model.

1 Introduction

The aggregation of hold-out estimates constitutes a principled approach to enhance the reliability and predictive performance of learning algorithms (see Maillard et al. [3] or Maillard [4]). Rather than relying on a single random partition of the data set into training and validation subsets, this method generates multiple hold-out splits. Then it aggregates their corresponding predictive outcomes or performance measures. Agghoo is already popular among practitioners (see, for example, Hoyos-Idrobo et al. [2] or Varoquaux et al. [8])

However, we will see that the performance of the cross-validated criterion may be a poor guess of the performance of aggregated models.

Hence, we suggest a solution to the problem of estimating the aggregated model performance. An initial partition of the data set must be created to generate K-fold test sets, and the aggregated models must be estimated using each complementary set. By averaging the performances on each test set, we obtain a good approximation of the generalization error, even with a small dataset.

This paper is intended to be a study of this procedure. The first section recalls the definition of Agghoo and shows, through simulations, that cross-validation error can mislead about the final test error of the aggregated model. The second section describes how our cross-testing procedure for Agghoo can improve the assessment of the generalization performance of aggregated models. Finally, the last section provides a theoretical justification for this procedure and discusses the advantages and drawbacks of the proposed method.

2 Aggregated Hold-out

Agghoo is explained with details in Mallard et al. [3]. The authors show that this procedure will continuously improve the performance of Hold-out for convex risk

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

measures such as mean squared error (MSE). Here, for clarity, we will consider only the MSE. Moreover, the performance of Hold-out is, up to a constant, optimal (see Blanchard and Massart [1], or van der Vaart et al. [7]). Note that such a procedure is closely related to the super learning strategy (see van der Laan et al. [6]), which is a slight variation of Agghoo.

2.1 Formulation of Aggregated Hold-out

Let $D_n = \{(X_i, Y_i)\}_{i=1}^n$ be a dataset of n i.i.d. observations drawn from an unknown distribution P on $\mathcal{X} \times \mathcal{Y}$. Let $\mathcal{G}$ denote a family of deep learning functions g, $g : \mathcal{X} \to \mathcal{Y}$. Consider validation partition sets of D_n, $V_1, \ldots, V_B$, with $V_k \cap V_l = \emptyset$, if $k \neq l$ and $\cup_{k=1}^B V_k = D_n$, each defining a train, validation split: (V_k^c, V_k). Then we get the agghoo algorithm in pseudo-code.

Algorithm 1 Aggregated Hold-Out (Agghoo) Method

for $b = 1$ to B **do**

- Validation set V_b
- Training set $T_b := V_b^c := \{1, \ldots, n\} \setminus V_b$

for each model $g \in \mathcal{G}$ **do**

Train g on $D_{T_b} = \{(X_i, Y_i) : i \in T_b\}$

Compute validation risk:

$$\widehat{R}_{V_b}(g) = \frac{1}{|V_b|} \sum_{i \in V_b} (Y_i - g(X_i))^2$$

end for

Select best model:

$$\widehat{g}_b = \arg\min_{g \in \mathcal{G}} \widehat{R}_{V_b}(g)$$

end for

Aggregation step:

Define the aggregated predictor:

$$\widehat{f}_{\text{Agghoo}}(x) = \frac{1}{B} \sum_{b=1}^{B} \widehat{g}_b(x)$$

return $\widehat{f}_{\text{Agghoo}}$

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

The aggregated predictor $\widehat{f}_{\text{Agghoo}}$ is used for future predictions. Note that aggregation can only improve the test risk by the convexity of the square function:

$$\left(\frac{1}{n_T}\sum_{i=1}^{n_T} g(X_i) - Y_i\right)^2 \leq \frac{1}{n_T}\sum_{i=1}^{n_T}(g(X_i) - Y_i)^2,$$

where n_T is the size of the test set. Hence, aggregation reduces the variance due to a single data split and provides a more stable approximation to the optimal predictor:

$$f^*(x) = \mathbb{E}[Y \mid X = x]. \tag{1}$$

However, as we will see in the next section, it is difficult to estimate the aggregated model performances solely from the validation error on a small dataset.

2.2 Experimental study: Rethinopathy

The Rethinopathy dataset [1] is from a 2015 Kaggle challenge that aims to predict Diabetic Retinopathy (DR), an eye disease associated with long-standing diabetes. A clinician has rated the presence of diabetic retinopathy in each image on a scale of 0 to 4, according to the following scale:

- 0 - No DR
- 1 - Mild
- 2 - Moderate
- 3 - Severe
- 4 - Proliferative DR

This is our target variable Y. Since its scale is ordinal, it is well-suited to the mean-squared loss. For the experiment, we intentionally create a small learning set from the Rethinopathy dataset, selecting 10% of the data (around 3500 observations) for training and validation of the Agghoo procedure and retaining 90% (around 32000 observations) to obtain a good approximation of the true test error of the aggregated model. Our goal is not to build a competitive model for this set but to understand the behaviour of aggregated models. We use a ConvNext deep model with MSE Loss and 100 epochs for training on each split (5 folds); moreover, we compute the validation error after each epoch and keep the model with the best validation error. By varying the size of the input images, we get the following results:

Image res.	val. error	Mean val.	Agghoo test err.
256×256	$0.497, 0.488, 0.458, 0.471, 0.517$	0.486	0.449
512×512	$0.333, 0.325, 0.272, 0.255, 0.379$	0.313	0.307
1024×1024	$0.256, 0.303, 0.227, 0.219, 0.298$	0.260	0.282

[1] https://www.kaggle.com/datasets/josephrynkiewicz/diabetic-retinopathy-train-unzipped

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

We can see that, for this experimental setup, the mean validation error can exceed the test error, be a fair estimate, or under-estimate the test error. Anything is possible, and this criterion does not allow us to correctly estimate the test error of the aggregated model. All our code is available on Github[2] for replicating the experiments.

3 Cross-tested Hold-Out aggregated models

The idea is to split the training dataset into multiple training/test sets to assess the performance of the aggregated model, and for each training set, apply the Agghoo algorithm.

3.1 Formulation of cross-tested Aggregated Hold-out

With the same framework as the previous section, consider test partition sets of D_n

$$T_1, \ldots, T_A \subset \{1, \ldots, n\},$$

with $T_k \cap T_l = \emptyset$, if $k \neq l$ and $\cup_{k=1}^{A} T_k = D_n$, each defining a train, test split: (T_k^c, T_k). Then we can describe the cross test agghoo by the following pseudo-code:

Algorithm 2 Cross-tested Agghoo Method

1: **for** $a = 1$ to A **do**

- Test set T_a
- Learning set $L_a := T_a^c$

2: For each learning set L_a, compute the Agghoo predictor $\widehat{f}^a_{\text{Agghoo}}$ and compute it's test risk:

$$\widehat{R}_{T_a}(\widehat{f}^a_{\text{Agghoo}}) = \frac{1}{|T_a|} \sum_{i \in T_a} (Y_i - \widehat{f}^a_{\text{Agghoo}}(X_i))^2$$

3: **end for**

4: compute mean test loss:

$$\widehat{R}_T(\widehat{f}) = \frac{1}{A} \sum_{i=1}^{A} \widehat{R}_{T_a}(\widehat{f}^a_{\text{Agghoo}})$$

5: **return** $\widehat{R}_T(\widehat{f})$

[2] https://github.com/JosephRynkiewicz/Cross-tested-Agghoo/tree/main

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

3.2 Experimental study: cross-tested loss

We apply the cross-tested agghoo algorithm with 5-Fold test sets on the same learning set as in section 2.2. Note that we also use 5-fold validation sets for each sub-sampled learning set to create the Agghoo models. Hence, test sets size is about 700 observations and validation sets size is about 560 observations. We get the following results (the "True error" is the Agghoo test error of the previous section):

Image res.	Agghoo test errors	Mean test error	True error
256×256	$0.470, 0.435, 0.436, 0.447, 0.488$	0.455	0.449
512×512	$0.319, 0.320, 0.290, 0.259, 0.372$	0.312	0.307
1024×1024	$0.290, 0.269, 0.280, 0.276, 0.295$	0.282	0.282

By comparing with the results of the section 2.2, we see that we obtain a better approximation of the Agghoo test set error; hence, cross-testing Agghoo is more robust than relying only on the validation error.

4 Theoretical properties of cross-tested Agghoo

4.1 Oracle inequality

The oracle inequality established by van der Vaart et al. [7] applies to validation sets of Agghoo and to test sets of cross-tested Agghoo; we will use it.

Theorem 1. *Let $D_n = \{(X_i, Y_i)\}_{i=1}^n$ be a dataset of n i.i.d. observations drawn from an unknown distribution P on $\mathcal{X}\times\mathcal{Y}$. Let $\mathcal{G}$ denote a family of deep learning functions g, $g : \mathcal{X} \to \mathcal{Y}$. Let f^* be the true conditional expectation, verifying equation (1). Let g^* be the best function of the considered families of models $\mathcal{G}$, $\widehat{g}^*$ be its estimation, and let N be the cardinality of $\mathcal{G}$. Let $\widehat{g}$ be the aggregated function selected by the Agghoo procedures, with a hold-out data set of size np, where p is the proportion of data kept for the hold-out set. Then, for any $\lambda > 0$, a constant $C(\lambda) > 0$ exists such that for the MSE risk: $ER_T(g) = E[(Y - g(X))^2]$, and its estimation on the hold-out set: $\hat{R}_T(g) = \frac{1}{np}\sum_{i=1}^{np}(Y_i - g(X_i))^2$ we get the following oracle inequality:*

$$\mathbb{E}\widehat{R}_T(\widehat{g}) - \mathbb{E}R_T(f^*) \leq (1 + 2\lambda)\left(\mathbb{E}R_T(\widehat{g}^*) - \mathbb{E}R_T(f^*)\right) + 2C(\lambda)\frac{1 + \ln(N)}{np} \quad (2)$$

Note that for a 5-Fold split, the proportion is $p = 0.2$. Moreover, if an optimal model f^* belongs to the family $\mathcal{G}$, then Rynkiewicz [5] shows that for deep learning models with ReLU activation functions, $\mathbb{E}R_T(\widehat{g}^*) - \mathbb{E}R_T(f^*) = O\left(\frac{1}{n}\right)$, and the bound of the oracle inequality has an optimal rate of $O\left(\frac{1}{n}\right)$. Since we maintain the exact sample size np for the validation sets of Agghoo and the test sets for cross-tested Agghoo, the only difference for the bound between the two algorithms will be the number N of models considered. For validation sets in Agghoo, this number is generally big, since we compute the validation error after each epoch and in our experiment $N = 100$. However, for the cross-tested

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

experiment, the test sets are used only once by the aggregated models. So, the ratio of the oracle bounds between Agghoo and cross-tested Agghoo will be $1 + \ln(N) \simeq 5.6$ if $N = 100$, and the theoretical guarantees for the cross-tested Agghoo are much better.

4.2 Computational complexity

We must emphasize that there are many more computations for the cross-tested Agghoo. For example, suppose we use 5-Fold test sets for cross-testing Agghoo. We also have to use a 5-Fold split for the validation sets of the subsampled learning dataset in the cross-tested Agghoo, so we have to estimate 25 models for the cross-tested Agghoo instead of 5 for the raw Agghoo. Hence, such a procedure is indicated when the sample size n is moderate, i.e., when the risk of overfitting and the instability of the splitting are significant.

5 Conclusion

Aggregate Hold-Out is a method well-suited for deep learning. But suppose the dataset is small or moderate, and we need to compare different families of models. In that case, the cross-validation error may be a misleading criterion, since the performance of the aggregated models is unknown. The proposed method provides a better approximation of the test error of aggregated models at the cost of more computations. Finally, we believe this method will be beneficial for more unstable criteria, such as AUC or C-index, even though, to the best of our knowledge, there are no theoretical guarantees for hold-out errors in such cases.

References

[1] G. Blanchard, P. Massart, Discussion: Local Rademacher complexities and oracle inequalities in risk minimization. *The Annals of statistics*, 34:2664-2671, 2006.

[2] A. Hoyos-Idrobo, Y. Schwartz, G. Varoquaux, and B. Thirion, Improving sparse recovery on structured images with bagged clustering. *2015 International Workshop on Pattern Recognition in NeuroImaging*. IEEE, 2015.

[3] G. Maillard, S. Arlot, M. Lerasle, Aggregated Hold-Out. *Journal of Machine Learning Research*, 22:1-55, 2021.

[4] G. Maillard, Aggregated hold out for sparse linear regression with a robust loss function. *Electronic Journal of statistics*, 16(1):935-997, 2022.

[5] J. Rynkiewicz, Asymptotic statistics for multilayer perceptron with ReLU hidden units. *Neurocomputing*, 342:C:16-23, Elsevier, 2019.

[6] M.J. van der Laan, E.C. Polley, A.E. Hubbard, Super Learner, *U.C. Berkeley Division of Biostatistics Working Paper Series.*, Working Paper 222, 2007

[7] A.W. van der Vaart, S. Dudoit, M.J. van der Laan, Oracle inequalities for multi-fold cross validation. *Statistics And Decisions*, 24(3): 351-371, 2006

[8] G. Varoquaux, P.R. Raamana, D.A. Engemann, A. Hoyos-Idrobo, Y. Schwartz, and B. Thirion. Assessing and tuning brain decoders: Cross-validation, caveats, and guidelines. *NeuroImage* 145:166-179, 2017