# Harnessing Explainability to Improve ML Ensemble Resilience

Abraham Chan, Arpan Gujarati, Karthik Pattabiraman, Sathish Gopalakrishnan
The University of British Columbia (UBC), Vancouver, BC, Canada
Email: abrahamc@ece.ubc.ca, arpanbg@cs.ubc.ca, {karthikp, sathish}@ece.ubc.ca

***Abstract*—Safety-critical applications such as healthcare and autonomous vehicles, utilize machine learning (ML), where mispredictions could have disastrous consequences. Training data can contain faults, especially when collected through crowdsourcing. Ensembles, consisting of multiple ML models voting on predictions, have been found to be an effective resilience technique. Ensembles are resilient when their constituent models behave independently during inference, by focusing on different features in an input. However, independence is not observed on every input, resulting in mispredictions. One way to improve ensemble resilience is to dynamically weigh predictions during inference by its constituent models instead of treating each model equally. While previous work on dynamically weighted models in ensembles has relied upon output diversity metrics due to efficiency, we focus on the feature-space of inputs for accuracy. Hence, we propose the use of explainable artificial intelligence (XAI) techniques to dynamically adjust the weight of ensemble models based on local feature-space diversity.**

***Index Terms*—Error resilience, Machine learning, Explainability**

## I. Introduction

Machine learning (ML) systems have been adopted in many safety-critical sectors such as healthcare [1] and autonomous driving [2]. Supervised learning, where models are trained with labelled data, forms the backbone of many such systems due to their high classification accuracy [3].

ML systems require copious amounts of training data. To obtain the training data, crowdsourcing [4] and automatic labelling [5] are utilized. Unfortunately, this has resulted in faulty training data (e.g. mislabelled data). Even frequently used public datasets such as ImageNet [6] have been found to contain faulty training data. For example, Northcutt et al. found that 5.83% of the ImageNet dataset is mislabelled [7]. The prevalence of such training data faults can seriously degrade the ability of ML models to learn effectively and classify test inputs correctly [7], and cause potentially catastrophic failures. Mispredictions by ML models can lead to serious consequences such as injuries and deaths, emanating from AV crashes. *Resilient* ML models are those that can correctly classify test inputs, despite the presence of faulty training data.

Previous work has identified ML resilience techniques [8–12] that improve the resilience of faulty ML models. In particular, Chan et al. [13] have found that *ensembles* provide the most overall resilience with the least practitioner effort. Ensembles are constructed by training multiple ML models independently on the same training data. During inference, their predictions are combined through voting. Since individual models in an ensemble are often able to learn sufficiently diverse aspects of the feature space [8], the ensemble can tolerate the effects of faulty training data during inference.

However, despite their effectiveness, ensembles can still make mispredictions when there is insufficient diversity between their constituent models [8]. To alleviate this issue, instead of each model prediction in an ensemble carrying equal weight, weighted ensembles [14, 15] have been proposed. The weights based on prediction confidence probabilities are assigned dynamically to reduce ensemble mispredictions.

Existing work on ensembles has measured diversity between models using output-space prediction diversity metrics (i.e., Shannon Entropy, Disagreement Metric) [8, 16]. For example, output-space prediction diversity metrics such as the Disagreement Metric are computed over the number of cases where any two models simultaneously predict correctly or mispredict. Unlike output-space prediction diversity, we hypothesize that feature-space diversity can better capture the diversity between models in terms of specific features used for classification.

In this paper, we propose the use of *local post-hoc* explainable AI (XAI) techniques [17] to capture feature-space correlation between models. Post-hoc XAI techniques attempt to explain an ML model's behaviour *after* it has been constructed and trained. Local XAI explains an ML model's behaviour behind individual inputs. For instance, local post-hoc XAI techniques (*e.g.* SHAP) applied on image classification data, return saliency maps indicating which pixels in a test image were used during inference. Because most ML models, especially deep neural networks (DNNs), are black-box models [18], we use post-hoc XAI techniques. Using ante-hoc XAI techniques, where the ML models are purposely designed to be explainable (*i.e.* explainable neural networks), would require architectural modifications.

Thus, we foresee an opportunity to utilize existing XAI techniques to improve ML ensemble reliability. We consider two types of commonly used explainability techniques:

- SHapley Additive exPlanations (SHAP) [19]
- Counterfactual Explanations [20]

We focus on DNNs in our work as they have been shown to exhibit the highest accuracies for classification tasks, while simultaneously, possessing the least explainability among ML algorithms. Each of these XAI techniques attempts to decipher the internal behaviour of an otherwise black-box model.

In summary, we make the following contributions:

- We propose the use of XAI techniques to improve their resilience against faulty training data.
- We evaluate one of the XAI techniques, against faulty training datasets, and present preliminary results.

## II. Background: Explainability Techniques

We provide a background of explainable AI (XAI) techniques used in this work.

### A. Post-Hoc XAI Techniques

XAI techniques are divided into ante-hoc and post-hoc techniques. Ante-hoc XAI incorporates explainability directly into the ML model's design. This results in guaranteed explainability at the feature-space level. However, ante-hoc XAI necessitates changes to an ML model's architecture, which requires expert knowledge.

Therefore, we utilize post-hoc XAI techniques, which do not require any modifications to the ML model architectures. Unlike ante-hoc techniques, post-hoc techniques provide approximations for feature-space explainability without guarantees. We hypothesize that this is sufficient in most cases of image classification. We propose the use of two types of post-hoc XAI techniques that are commonly used in ML and have open-source implementations.

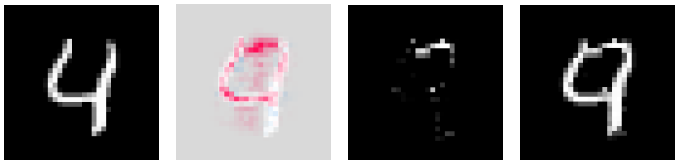
### B. SHAP

Shapley values are based on cooperative game theory [21], where each player contributes to the overall game result. Shapley values determine how much each individual player contributes to the result - higher values indicate more contribution. In image classification, we can model each input pixel as a player, while the inference process is the game.

However, Shapley values must be calculated for each input combination. Suppose there are $N$ input features, and each input feature is either used or not used, this would result in a total of $2^N$ states to be considered. As this is unfeasible, SHAP [19], which stands for SHapley Additive exPlanations, approximates the Shapley value.

SHAP indicates which input features contribute the most to the final prediction result. SHAP is implemented by different explainers, according to the ML task. For instance, we use GradientExplainer [22], which uses integrated gradients to attribute features and calculate the SHAP for image classification. The method for calculating SHAP is model agnostic.

We demonstrate an example of calculating the SHAP values of a test image from MNIST [23], a popular image classification dataset of handwritten numeric digits. Suppose we have a test image of 4 in Fig. 1a and we train an ML model, ConvNet, on MNIST. ConvNet correctly predicts the image as 4, and we calculate the SHAP values and plot its saliency graph in Fig. 1b where red regions represent pixels that positively contributed to the classification. We can see that ConvNet placed more emphasis on certain pixels around the outline of the digit over other regions - this explains how the ML model reached its final prediction.



(a) Test Image (b) SHAP (c) CFE (d) CFE applied

Fig. 1: Example to show SHAP and CFE of ConvNet on MNIST. **(a)** Test Image. **(b)** SHAP plot **(c)** CFE shown. **(d)** CFE applied on original test image of 4, now resembling 9.

### C. Counterfactual Explanation (CFE)

Counterfactual explanations (CFEs) demonstrate the smallest change in input feature values that lead to a different prediction outcome. CFEs explain ML inference results in a causal manner: *if x occurred, y would not have occurred* [20]. CFEs represent the minimal changes required to classify a test input into a different class from the original predicted class by the ML model.

The process of finding CFEs shares similarities with that of adversarial perturbations. Unlike adversarial perturbations, however, CFEs typically involve changes to a small set of input variables, while adversarial perturbations encompass small changes to a large number of input variables. In image classification, CFEs represent the smallest quantity of pixels that are perturbed such that an ML model classifies differently.

We show an example of generating the CFE for an ML model, ConvNet, on a test prediction. Given an MNIST image in Fig. 1a, ConvNet originally predicts the image as 4, as expected. A CFE is generated in Fig. 1c, which shows a small set of pixels in white. If the CFE is applied back on the original test image, as shown in Fig. 1d, ConvNet will predict the new image as 9. The CFE shows the smallest number of pixels to be changed, in order, for the prediction result to also change.

## III. Motivating Example

We demonstrate an example of an ensemble failing to correctly classify a test image, and the potential of utilizing feature-space diversity to rectify the prediction. We use the German Traffic Sign Recognition Benchmark (GTSRB) [24], a publicly available dataset for autonomous driving, containing more than 50,000 images belonging to 43 different categories of traffic signs in Germany. We randomly inject the GTSRB dataset with 30% mislabelling, modifying the labels such that they do not match their image. Suppose we have three ML models: ConvNet, DeconvNet, and VGG11. We independently train each of these ML models with the faulty GTSRB dataset. These models are then combined in an ensemble, where the three models vote on predictions using simple majority voting.

Suppose we pass an input image, shown in Fig. 2b, to the ensemble. ConvNet and DeconvNet misclassify the image as a 'dangerous curve left' while VGG11 correctly classifies the image as a 'slipper road alert'. Following simple majority voting, the ensemble would mispredict the image as ConvNet and DeconvNet outvote VGG11.

(a) Reference (b) Input (c) ConvNet (d) DeconvNet (e) VGG11

Fig. 2: **(a)** Slippery road alert sign. **(b)** Test Image #72 from GTSRB. **(c, d, e)** SHAP using GradientExplainer retrieved from ML models while inferring test image.

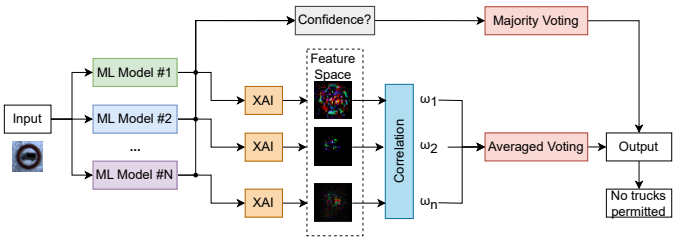



Fig. 3: Weighted ensemble using XAI to extract the feature space, on an example of road sign image from GTSRB.

All three ML models use a softmax activation function in their top layers. ML models take the class with the highest softmax values as their final prediction. However, we observed that the softmax values for the predicted classes are very low (*i.e.* less than 0.5) among all three models, indicating a lack of confidence in their predictions.

Therefore, we apply GradientExplainer on each ML model, to generate a SHAP plot (Figs. 2c to 2e) representing their feature spaces. Upon close inspection, we can observe that ConvNet and DeconvNet (Figs. 2c and 2d) have more similar looking feature spaces compared to VGG11 (Fig. 2e) - the plots show the SHAP values as coloured masks over pixel regions. Red regions represent pixels that positively contributed to the classification, while blue regions represent pixels that negatively contributed (*i.e.* pixels used to predict a different class) to the classification result. However, we can see that it is infeasible to manually inspect the differences across every single pixel across the images - this is why we need an automated and systematic method to determine their similarity.

To complement our visual inspection of the feature spaces, we calculate the $R^2$, the coefficient of determination, between three pairs of SHAP plots: (ConvNet and DeconvNet), (ConvNet and VGG11), (DeconvNet and VGG11). $R^2$ ranges from 0 (no correlation) and 1 (correlation). We find that the $R^2$ values are: 0.72, 0.53, and 0.36.

We propose using the correlations to revise the voting scheme. Instead of simple majority voting, we take a weighted average of the softmax outputs across the three models. Each model is assigned a weight, corresponding to its pairwise feature-space correlation. Because ConvNet is highly correlated with DeconvNet, their weights should be lower than VGG11, which has a lower correlation with the other two models. This enables the correct classification from VGG11 to overcome the misclassifications by ConvNet and DeconvNet.

## IV. Dynamically-Weighted Ensembles using XAI

We demonstrate the workflow of our proposed framework in Fig. 3. Because the application of XAI methods to extract the feature space requires extra overhead during inference, we propose that predictions made with high *confidence* (*i.e.* high softmax outputs) can proceed with unweighted average voting. Only predictions with low confidence need to be revised with weighted voting using XAI.

Ensemble predictions made with low confidence are passed to the XAI module. The XAI module could contain an implementation of either SHAP or CFE. The XAI module attributes each prediction to its feature space.

The pairwise correlation is calculated between the feature spaces. Higher correlation means that the models are less diverse, while lower correlation means the models are more diverse. We propose that the weights ($\omega_n$) for each constituent model be generated based on a combination of the calculated pairwise correlation values and their prediction confidences. $\omega$ is applied to the softmax outputs of each constituent model, and the class with the highest softmax average is selected as the revised prediction.

## V. Preliminary Evaluation

We perform a preliminary evaluation to understand the following research questions.

1) How many ensemble predictions are low confidence and may benefit from dynamically-weighted ensembles?
2) How diverse are ensembles in the feature space, compared to their prediction confidence?

First, we wish to understand how many ensemble predictions could potentially benefit from dynamically-weighted ensembles. We build an ensemble consisting of three models: ConvNet, DeconvNet, and VGG11. Then, we inject varying amounts of random mislabelling faults into the GTSRB training dataset, and train the ensemble on it. We run the ensemble through the GTSRB test dataset, where we combine the predictions made by each model through majority voting, and collect their prediction confidence by averaging the softmax values between the three models. As shown in Fig. 4, we set three confidence thresholds as indicated in the legend, and count the ensemble predictions that fall under each threshold.

We can observe that as the quantity of training data faults increases, more ensemble predictions fall under the confidence thresholds. Even at 10% mislabelling, we see a significant increase in low confidence predictions over the golden case (no injected faults). Inputs with low confidence predictions are candidates for dynamically-weighted ensembles.

Second, we explore the relation between ensemble feature-space diversity with their prediction confidence. We take two models from the ensemble: ConvNet and DeconvNet. Both models are trained on GTSRB with 30% mislabelling. We take a random sample of 30 low confidence (*i.e.* below 0.7) mispredictions, and generate a pair of SHAP plots (one for

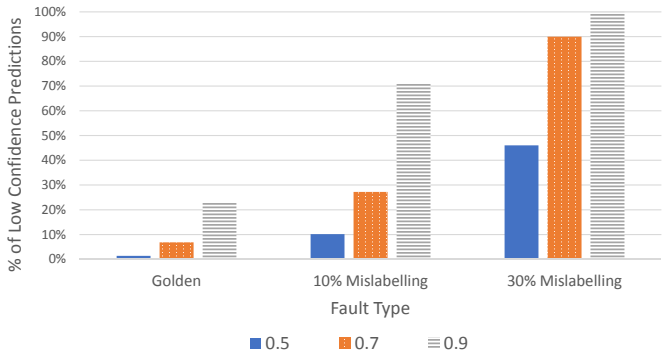



Fig. 4: Percentage of ensemble predictions on GTSRB test images that do not meet the specified confidence threshold. Ensemble is trained with GTSRB, injected with varying levels of mislabelling. Legend indicates three confidence thresholds.

each model). Then, we calculate the $R^2$ correlation between each pair of SHAP plots - we call this the SHAP correlation. In Fig. 5, we plot their SHAP correlation against their prediction confidence. Further, we denote predictions where one of the two models generates a correct prediction with green dots, and predictions where both models are incorrect with red crosses.

We observe that the vast majority of predictions have at least one model predicting correctly. However, the SHAP correlation and prediction confidence do not appear to be linearly separable globally. This shows that while there is an opportunity to increase the number of correct predictions using XAI, more work needs to be done to determine how to dynamically assign weights locally (*i.e.* specific to each input).

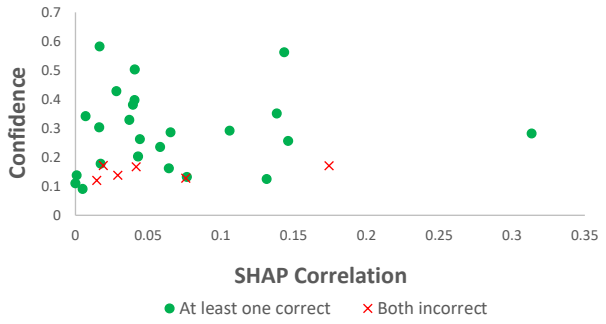



Fig. 5: SHAP Correlation (between ConvNet and DeconvNet) vs. Prediction Confidence. Both models are trained with GTSRB, injected with 30% mislabelling faults. Dots show the predictions where at least one of the models is correct, while crosses show cases where both models are incorrect.

## VI. Related Work

### A. Statically-weighted Ensembles

Most ensembles constructed by training and running multiple models in parallel (bagging) are combined through majority voting [25]. Because models in an ensemble have different classification accuracies, unweighted voting can produce misclassifications in cases preventable by weighted ensembles.

Iqball and Wani [26] introduce a weighted ensemble for image classification where weights are calculated based on the classification ability of each model. All models are initially assigned a weight of 1. Models are then assigned an additional weight equal to their relative accuracy improvement over the lowest accuracy model. Weights are assigned to each model after training, and remain static during inference. While their work considers a model's global classification ability, our proposed method focuses on the local classification of inputs.

Kuncheva and Rodríguez [14] experimented with different voting schemes for ensembles. They assume a class-conditional independence (*i.e.* the classification accuracy of one class is not dependent on another) between models. Ensembles with different voting schemes are evaluated on their output accuracy. While they found that there was no optimal voting scheme overall, weighted ensembles were better for datasets with small quantities of unbalanced classes and with label noise. Unlike their work, we propose ensembles, where weights are derived from feature-space diversity, without assumptions on class-conditional independence or the number of unbalanced classes.

### B. Dynamically-weighted Ensembles

Statically-weighted ensembles have largely focused on classification accuracy, which only optimizes ensembles over a set of test inputs rather than specific inputs. To rectify this, dynamically-weighted ensembles have been proposed.

Ren et al. [15] propose dynamically-weighted ensembles based on test input features during inference. They present an algorithm to calculate the dynamic weights based on the eigenvalues of the confusion matrix of each model, and found that dynamically-weighted ensembles outperform statically-weighted ensembles by suppressing unreliable classifications against certain test inputs. Because the dynamic weights are calculated based on confusion matrices, the ensembles optimize over classes of images rather than individual test images. In contrast, our proposed feature-based ensembles are customized for each test input, which would offer higher classification reliability.

## VII. Conclusions

ML applications require accurate predictions, especially in systems deployed in safety-critical domains. Training data faults have been shown to negatively impact the classification ability of individual ML models. Ensembles have been presented as a promising solution to tolerate the presence of training data faults during inference. However, unweighted ensembles are still prone to misclassifications, where incorrect classifiers outvote the correct classifiers. We propose deploying dynamically-weighted ensembles based on the feature-space diversity between constituent models using local post-hoc XAI techniques. To reduce overhead, dynamically-weighted ensembles are only activated when ensemble predictions are made with low confidence.

## VIII. Acknowledgements

We thank the anonymous reviewers for their valuable comments. This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), a Four Year Fellowship from UBC, and the Institute of Computing, Information and Cognitive Systems (ICICS) at UBC.

## References

[1] J. G. Richens, C. M. Lee, and S. Johri, "Improving the accuracy of medical diagnosis with causal machine learning," *Nature Communications*, 2020.

[2] S. S. Banerjee, S. Jha, J. Cyriac, Z. T. Kalbarczyk, and R. K. Iyer, "Hands Off the Wheel in Autonomous Vehicles?: A Systems Perspective on over a Million Miles of Field Data," in *Proceedings of IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2018.

[3] V. K. Garg and A. Kalai, "Supervising Unsupervised Learning," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2017.

[4] P.-Y. Hsueh, P. Melville, and V. Sindhwani, "Data Quality from Crowdsourcing: A Study of Annotation Selection Criteria ," in *Proceedings of Workshop on Active Learning for Natural Language Processing (NAACL HLT)*, 2009.

[5] M. Hamzah, "Auto-Annotate," https://github.com/mdhmz1/Auto-Annotate, 2020.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[7] C. G. Northcutt, A. Athalye, and J. Mueller, "Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks," 2021, arXiv:2103.14749.

[8] A. Chan, N. Narayananan, A. Gujarati, K. Pattabiraman, and S. Gopalakrishnan, "Understanding the Resilience of Neural Network Ensembles against Faulty Training Data," in *Proceedings of IEEE International Conference on Software Quality, Reliability and Security (QRS)*, 2021.

[9] R. Müller, S. Kornblith, and G. Hinton, "When does label smoothing help?" 2020.

[10] L. Zhang, J. Song, A. Gao, J. Chen, C. Bao, and K. Ma, "Be Your Own Teacher: Improve the Performance of Convolutional Neural Networks via Self Distillation," 2019.

[11] X. Ma, H. Huang, Y. Wang, S. Romano, S. Erfani, and J. Bailey, "Normalized Loss Functions for Deep Learning with Noisy Labels," in *Proceedings of International Conference on Machine Learning (ICML)*, 2020.

[12] G. Zheng, A. H. Awadallah, and S. Dumais, "Meta Label Correction for Noisy Label Learning," in *Proceedings of Association for the Advancement of Artificial Intelligence (AAAI)*, 2021.

[13] A. Chan, A. Gujarati, K. Pattabiraman, and S. Gopalakrishnan, "The Fault in Our Data Stars: Studying Mitigation Techniques against Faulty Training Data in Machine Learning Applications," in *Proceedings of IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2022.

[14] L. Kuncheva and J. Rodríguez, "A weighted voting framework for classifiers ensembles," *Knowledge and Information Systems*, 2014.

[15] F. Ren, Y. Li, and M. Hu, "Multi-classifier ensemble based on dynamic weights," *Multimedia Tools Appl.*, 2018.

[16] L. Kuncheva and C. Whitaker, "Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy," *Machine Learning*, 2003.

[17] A. Barredo Arrieta *et al.*, "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information Fusion*, 2020.

[18] C. Rudin and J. Radin, "Why Are We Using Black Box Models in AI When We Don't Need To? A Lesson From an Explainable AI Competition," *Harvard Data Science Review*, 2019.

[19] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2017.

[20] L. Bottou *et al.*, "Counterfactual Reasoning and Learning Systems: The Example of Computational Advertising," *The Journal of Machine Learning Research*, 2013.

[21] L. S. Shapley, "A Value for n-Person Games," *Contributions to the Theory of Games*, 1953.

[22] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic Attribution for Deep Networks," in *Proceedings of International Conference on Machine Learning (ICML)*, 2017.

[23] Y. LeCun, C. Cortes, and C. Burges, "MNIST handwritten digit database," *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, vol. 2, 2010.

[24] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural Networks*, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0893608012000457

[25] L. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Wiley, 2004.

[26] Iqball,Talib and Wani, M. Arif, "Weighted ensemble model for image classification," *Information Fusion*, 2023.