ReMlX: Resilience for ML Ensembles using XAI at Inference against Faulty Training Data

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

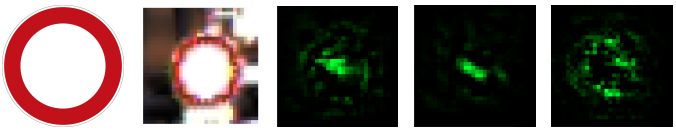
***Abstract*—Safety-critical domains, such as healthcare and autonomous vehicles, employ machine learning (ML), where mispredictions can cause severe repercussions. Training datasets may contain faults, thereby compromising ML accuracy. Ensembles, where multiple ML models vote on predictions, are effective at maintaining predictive capability, and thus resilient against faulty training data because individual models focus on diverse input features. Nevertheless, ensemble diversity varies per input. Hence, weighted ensembles can bolster resilience by assigning unique weights to constituent models. While existing weighted ensembles focus on output-space diversity, we propose leveraging their feature-space diversity to better capture model independence and achieve greater resilience. Therefore, we present ReMlX, which applies explainable artificial intelligence to extract the feature-space diversity of ensemble models, and adjusts their weights to maximize resilience. Compared to its most competitive baseline, ReMlX is 12% more resilient but 15% slower than dynamic weighted ensembles based on stacking.**

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

I. Introduction

Machine learning (ML) systems have been deployed in numerous safety-critical domains, including healthcare [1] and autonomous vehicles (AVs) [2]. Supervised learning, which involves training models on labelled data, is the foundation of many of these systems [3]. To acquire the large volumes of training data for ML, methods such as crowdsourcing [4] and automatic labelling [5] are employed. However, these approaches have led to the emergence of faulty training data, including instances of mislabelling (see Table I). Faulty training data can significantly impair the capacity of ML models to learn effectively and accurately classify test inputs [6], resulting in severe failures during misclassifications. Incorrect predictions made by ML models can have dire repercussions, including injuries and fatalities stemming from AV accidents.

Resilient ML models are those capable of correctly classifying test inputs, *even if a subset of the training data is faulty*. We have shown in our prior work [7] that *ensembles* are effective at providing *resilience*. We define resilience as the ability of an ML component to retain high predictive capability despite be- ing trained with faulty training data. Ensembles are formed by training multiple ML models (typically 3) independently on the same dataset. During inference, predictions from these models are aggregated through a voting mechanism (e.g., simple majority voting). Because the individual models within an ensemble often capture diverse elements of the feature space [8], the ensemble can mitigate the impact of faulty training data during inference by outvoting the model with the wrong prediction. While ensembles have been widely used to improve ML accuracy [9–11], our focus is on its resilience.



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

Fig. 1: **(a)** No vehicles permitted sign. **(b)** Test Image #896 from GTSRB. **(c, d, e)** Feature spaces of different ML models, extracted using Smooth Gradients, while inferring test image.

Consider an ensemble of three models (ConvNet, DeconvNet, VGG11), independently trained on a dataset of German traffic sign images, GTSRB [12] that we injected with randomly mislabelled training images. During inference, the ensemble, under simple majority voting, outputs a correct prediction if at least two of its models correctly predict.

But, what happens when the ensemble mispredicts? When a test image (Fig. 1b) is passed to the ensemble for inference, ConvNet and DeconvNet both mispredict the test image as a '100 km/h speed limit' sign, while VGG11 alone correctly predicts the image as a 'no vehicles permitted' sign. Under simple majority voting, the ensemble would mispredict as ConvNet and DeconvNet would incorrectly outvote VGG11, which will lead to a disastrous outcome if deployed on an AV.

Our objective is to reduce the misclassifications by ensembles under training data faults, thereby boosting their resilience. To address the shortfalls of simple majority voting, *weighted ensembles* [13–16] have been introduced, wherein the predictions of each model do not carry equal weight. Existing approaches for weighted ensembles focus on maximizing the *global* (across the entire test dataset) output-space diversity between ML models to increase the probability of correct predictions. ConvNet, DeconvNet, VGG11 have 85%, 82%, 72% accuracy respectively. If weights were assigned based on accuracy alone, ConvNet and DeconvNet would be assigned higher weights, and again erroneously outvote VGG11.

Each model in an ensemble focuses on a set of features or *feature spaces* to make its prediction. Explainable AI (XAI) techniques can identify such features and their significance to the prediction result [17]. For example, we adopt Smooth

Gradients, an XAI technique that extracts and visualizes the *local* (input-specific) feature spaces of each model, shown in Figs. 1c to 1e. Brighter pixels indicate the features of interest to the ML model. Both ConvNet and DeconvNet focus on the content within objects (*i.e.* the circle's blank interior), while VGG11 focuses on the shape (*i.e.* the circle's thick outline). Because ConvNet and DeconvNet had similar feature spaces, a weighted ensemble should assign greater weight to VGG11 during voting. This is the main idea we explore.

We propose ReMlX[1], a method to boost ML ensemble resilience at inference by *leveraging XAI to maximize feature-space diversity*. ReMlX operates in three steps. First, an ensemble of individual models are trained with the same (faulty) training data. Then, ReMlX uses XAI techniques to generate the local feature spaces separately for each constituent model. Finally, ReMlX generates weights that maximizes the feature-space diversity between models, by applying diversity metrics.

To the best of our knowledge, ReMlX is the first technique to incorporate feature-space diversity to dynamically generate weights at inference time for building resilient ensembles against faulty training data. In summary, we make the following contributions in this paper.

- Developing ReMlX, a novel method to induce ensemble resilience during inference by deriving weights for voting from their XAI-generated local feature-space diversity.
- Systematically shortlisting state-of-the-art XAI techniques (Counterfactual Explanations, Integrated Gradients, LIME, Smooth Gradients, SHAP) and selecting the most efficient, faithful and robust technique against ML models trained with faulty training data. Also, identifying diversity metrics (Coefficient of Determination, Cosine Distance, Frobenius Norm, Wasserstein) that contrast ML models' feature matrices for weight generation.
- Experimentally evaluating the predictive capability of ReMlX across three different image-classification datasets, with varying types and amounts of injected training data faults, against seven baselines: best individual model, uniform majority, uniform average, static weighted, dynamic weighted (stacking), bagging, and boosting. We use balanced accuracy (BA) and F1-score (F_1) as metrics to measure predictive capability across balanced and imbalanced multi-class datasets.

Compared to its most competitive baseline, we find ReMlX is 12% more resilient but only 15% slower than dynamic weighted ensembles based on stacking. We select Smooth Gradients as the best XAI technique, and, Cosine Distance as the most effective feature-space diversity metric for ReMlX to maximize ensemble resilience. ReMlX is more resilient compared to baselines even as the ensemble size increases.

II. Background

A. Fault Model

Following our prior work [8], we consider three categories of faults that frequently arise in training data. Table I shows the prevalence of these faults.

1) *mislabelling faults* – where data is erroneously labelled,
2) *repetition faults* – where input-output pairs are repeated,
3) *removal faults* – where a fraction of data may be deleted.

Reports indicate that mislabelling and removal faults in training data can reach as high as 70%, even within safety-critical datasets [18], including the Lyft [19] and Chest X-Ray datasets [20]. Additionally, repetition faults have been identified in the widely utilized GTZAN music dataset [21]. *Thus, training data faults are widespread in real-world datasets.*

TABLE I: Training data faults found in ML training datasets across different domains. Mis stands for mislabelling.

Dataset	Domain	% Faulty	Fault Type(s)	Source
Udacity [22]	AV	33	Mis, Removal	[23]
Lyft Level 5 [24]	AV	70	Mis, Removal	[19]
ChestX-Ray14 [25]	Medical	20	Mis	[20]
ImageNet [26]	Objects	5.83	Mis	[6]
COCO [27]	Objects	45.5	Mis, Removal	[28]
GTZAN [29]	Music	10.6	Mis, Repetition	[21]

Previous research on training data faults [7, 8] has predominantly concentrated on symmetrically (uniformly) distributed faults across label classes for sake of simplicity. Instead, we wish to study faults that better emulate their real-life distributions in datasets, as done in our prior work [30]. Actual fault distributions observed in datasets like CIFAR-10 are asymmetric [30]. This is because certain label classes can be more easily confused with one another during labelling, *e.g.*, cats resemble dogs rather than trucks.

B. Ensembles

Ensembles are made up of numerous ML models trained independently and inspired by N-version programming [31–33]. During inference, each ML model in the ensemble receives the same input and makes separate predictions. These predictions are merged using voting (*i.e.* simple majority) to generate a single prediction. Ensembles for classification are resilient to faulty training data due to prediction diversity across their constituent models [8, 34]. Compared to alternate mitigation strategies against faulty training data, ensembles have been found to be the most resilient, while requiring no additional hyperparameter tuning [7] (and thus require low effort).

Ensembles can be considered a form of meta-learning, an approach where learning algorithms learn from each other [35] to improve overall performance. Meta-learning divides components into base-learners and meta-learners. Each constituent ML model is a base-learner while the aggregation mechanism (*i.e.* ReMlX) serves as the meta-learner.

C. Post-Hoc XAI Techniques

Explainable AI (XAI) techniques can be applied to obtain a trained ML model's feature space [36]. XAI techniques can be categorized into ante-hoc and post-hoc [36]. Ante-hoc XAI integrates explainability into the design of the ML model itself, ensuring a guaranteed level of explainability at the feature space level. However, this approach requires changes

[1]https://github.com/DependableSystemsLab/Remix

to the architecture of the ML model, which demands expert knowledge. Consequently, we use post-hoc XAI techniques, which do not require changes to the architectures of models.

Post-hoc XAI techniques can be classified as *model-agnostic* and *model-dependent*. Model-agnostic techniques can be applied to any ML model, while model-dependent techniques only work for certain models. Our criteria for choosing XAI techniques is that they must be: (1) local, (2) post-hoc, (3) compatible with image classification and (4) publicly available. Based on these criteria, we select three model-agnostic techniques: SHAP [37], LIME [38], and Counterfactual Explanations [39]. We also select two model-dependent techniques that assume the models are differentiable: Integrated Gradients [40] and Smooth Gradients [41]. We consider only DNNs, which are all differentiable, and hence gradient-based techniques can be applied. The XAI techniques generate explanations as visualizable 2D arrays i.e., feature matrices.

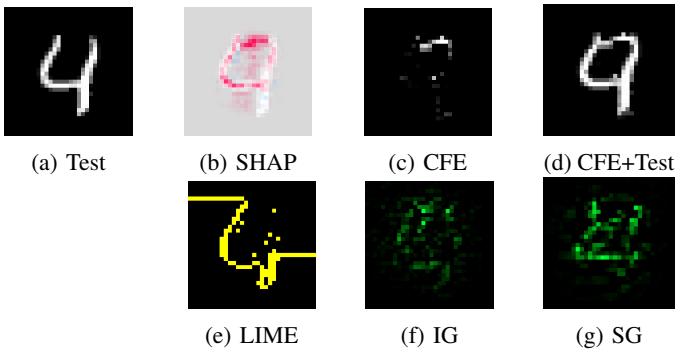
1. SHAP. Shapley values originated from cooperative game theory [42], where values quantify the contribution of each individual player to the final result, with higher values signifying a greater contribution. In image classification, each input pixel is conceptualized as a player, while the inference process represents the game itself. Since calculating Shapley values on N input features requires evaluating 2^N possible input configurations, SHAP [37], SHapley Additive exPlanations, approximates the Shapley value. SHAP identifies input features that have the greatest impact on the final prediction outcome and outputs a matrix of feature importance scores.

2. LIME, which stands for Local Intepretable Model-Agnostic Explanation [38], fits a surrogate model around the perturbed values of test inputs. The surrogate model consists of a lower complexity, more interpetable, ML model (*i.e.* a linear classifier) that mimicks the original model's behaviour. For image classification, LIME produces a segmentation mask over the test image to indicate regions of interest to the model.

3. Counterfactual Explanations (CFE) illustrate the minimal changes in the input feature values necessary to achieve an alternative prediction outcome. CFEs explain ML inference results in a causal framework: *if x occurred, y would not have occurred* [39]. They signify the least modifications needed to reclassify a test input into a different class from the one initially predicted by the ML model. In image classification, CFEs denote the minimal number of pixels that must be modified for a ML model to produce a different classification.

4. Integrated Gradients (IG) [40] demonstrate which pixels contribute more to a prediction, as pixel intensity is gradually increased. Starting with a baseline image (*i.e.* a black image), IG accumulates gradients as pixel intensity is slowly increased at each step, beginning initially with baseline pixels until the pixels resemble the input image. IG produces a matrix of gradients where higher magnitudes indicate greater influence on the prediction.

5. Smooth Gradients (SG) [41] are similar to IG where gradients are computed over the test image. Unlike IG, gradients are computed and averaged across mulitple images, where each image is injected with Gaussian noise. This smooths out noise and sharpens the explanation compared with IG.



(a) Test (b) SHAP (c) CFE (d) CFE+Test

(e) LIME (f) IG (g) SG

Fig. 2: XAI Techniques applied on ConvNet trained with MNIST. **(a)** Test Image. **(b)** SHAP **(c)** CFE **(d)** CFE applied on test image of 4, now resembling 9. **(e)** LIME **(f)** IG **(g)** SG

XAI Examples on MNIST. We show visual examples of the feature matrices after applying various XAI techniques on a test image from the MNIST [43], a dataset for classifying handwritten numeric digits. We train a ConvNet model on the MNIST dataset and ensure that the test image in Fig. 2a is classified as 4. For SHAP, LIME, SG, IG values, Figs. 2b and 2e to 2g show their saliency maps where the coloured areas indicate pixels of greater influence on the model. The CFE is shown in Fig. 2c, exhibiting a small set of pixels highlighted in white. When this CFE is applied to the original test image (Fig. 2d), it resembles a 9, revealing the minimal set of pixel alterations to change the prediction outcome.

XAI Technique Properties. XAI techniques are typically evaluated on five main properties [44]: faithfulness, robustness, randomization, localization, complexity. Faithfulness measures whether explanations change when important features, assigned high relevance scores by the XAI technique, are modified. Robustness assesses the stability of explanations when the test inputs are subjected to minor changes. Randomization determines whether an explanation changes if the model weights are subject to randomized noise. Localization evaluates how well explanations are correlated to user-defined regions of interest. Complexity measures the conciseness of an explanation, by estimating the number of distinct features used. A smaller number of features facilitates human interpretability.

However, the importance of each property differs according on how the XAI technique is applied. Since our objective is to use XAI techniques to determine dynamic weights, we care only about faithfulness and robustness. While randomization emphasizes the importance of obtaining unique explanations on models with different weights, our focus is on achieving explanations from faulty models, which may retain their explanations from the golden model. Localization only applies to object detection tasks, while our focus is on image classification. Complexity is irrelevant for our (non-interactive) automated use of XAI. Since resilience is our goal, not human interpretability, a more complex explanation is acceptable if it is more faithful and robust. We also consider efficiency, as slow XAI techniques would increase inference times.

D. Diversity Metrics

Ensembles are resilient due to the diversity in behaviour of its constituent models [7, 8, 34]. Diversity metrics express the dissimilarity between model behaviour. Kuncheva et al. presented work that extensively studied the relation between ensemble diversity and accuracy, leading to their proposal of ten statistical metrics to represent diversity [34] such as the Q statistic, disagreement measure, entropy, and Kohavi-Wolpert variance. However, their proposed metrics were largely limited to binary classifiers. In contrast, we focus on diversity metrics compatible with multi-class classifiers. We describe metrics used to measure output-space and feature-space diversity.

Output-Space Diversity. The output-space of an ML model is limited to predictions and their confidences (*i.e.* softmax probabilities). We use Shannon entropy (H) [8], an established ensemble diversity metric, to determine output-space diversity. H is calculated on the ensemble prediction of a single test input. If S denotes the number of classes in a dataset, and p_i denotes the ensemble's prediction confidence (*i.e.* the softmax probability) belonging to class i, then H is given by:

$$H = -\left(\sum_{i=1}^{S} p_i \ln p_i\right) \Big/ \ln S. \tag{1}$$

Feature-Space Diversity. The feature representation for image classification ML models is expressed as a $m \times n$ feature matrix (Section II-C). To evaluate the feature-space diversity between two models, we need diversity metrics between their two feature matrices, A and B. We require diversity metrics to work on matrices instead of one dimensional vectors and to be commutative (*i.e.* same result regardless of the ordering of A and B). Hence, we identify four metrics: Coefficient of Determination, Cosine Distance, Frobenius Norm [45], and Wasserstein Distance [46]. We explain how each feature-space diversity metric is calculated on A and B.

1. Coefficient of Determination or R^2, is equal to the squared Pearson correlation obtained from the covariance matrix between A and B. R^2 is computed as follows, where $\overline{A}$ and $\overline{B}$ are the means of their respective matrices, and σ_A and σ_B are their standard deviations.

$$R^2 = \left(\frac{\sum_{i=1}^{m}\sum_{j=1}^{n}(A_{i,j} - \overline{A})(B_{i,j} - \overline{B})}{mn \cdot \sigma_A \sigma_B}\right)^2 \tag{2}$$

R^2 ranges from 0 and 1, where 0 indicates maximal diversity and 1 indicates no diversity.

2. Cosine Distance is based on cosine similarity, which measures the similarity between two vectors, and is widely used to compare text embedding similarity in natural language processing. We first flatten A and B into one dimensional vectors and compute cosine distance as follows: 1 - $\frac{A \cdot B}{||A||||B||}$. It ranges from 0 and 2. Values closest to 0 indicate the lowest diversity, while values closer to 2 indicate the highest diversity.

3. Frobenius Norm [45] is analagous to the elementwise Euclidean norm. We compute the Frobenius norm on A - B. The Frobenius Norm is unbounded, with higher values indicating greater diversity.

$$||A - B||_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}(A_{i,j} - B_{i,j})^2} \tag{3}$$

4. Wasserstein Distance [46] (or Earth mover's distance) measures the distance between two probability distributions (*i.e.* A and B) by estimating the cost required to convert one distribution to match another. The Wasserstein Distance is unbounded, with higher values indicating greater diversity.

$$W(A, B) = \frac{1}{nm}\sum_{i=1}^{m}\sum_{j=1}^{n}|A_{i,j} - B_{i,j}| \tag{4}$$

III. Motivational Case Study

We perform a case study to understand how an ensemble predicts, under training data faults, and compare its output-space and feature-space diversity. We aim to answer three questions: (1) *Is simple majority or uniform average voting sufficient for ensembles?* (2) *How much feature-space diversity is present compared with output-space diversity?* (3) *Should we use dynamic weights based on feature-space diversity?*

We use the German Traffic Sign Recognition Benchmark (GTSRB) [12], a publicly available dataset for autonomous driving, containing 39,209 training and 12,630 testing images belonging to 43 different categories of traffic signs in Germany. We use GTSRB's pre-defined training and testing splits.

Suppose we have seven neural network architectures: ConvNet, DeconvNet, MobileNet, ResNet18, ResNet50, VGG11, VGG16. We build an ensemble of three models, the minimum number required for simple majority voting. Because we wish to evaluate the resilience of ensembles against faulty training data, we inject 30% random mislabelling into GTSRB following its extracted fault pattern (see Section V-B). We pick the three-model ensemble with the highest balanced accuracy (BA) under 30% random mislabelling: ConvNet, ResNet50, VGG11. We also train this ensemble with the GTSRB dataset, without any injected faults - this is the golden dataset.

(a) Golden (b) 30% Mislabelling

Fig. 3: Proportion of correct models by the best ensemble on GTSRB with (a) no faults and (b) 30% random mislabelling.

We evaluate the ensemble on the test dataset, and count the number of constituent models that correctly predict test data. Fig. 3 shows the proportion of test data correctly classified by varying numbers of constituent models, comparing the

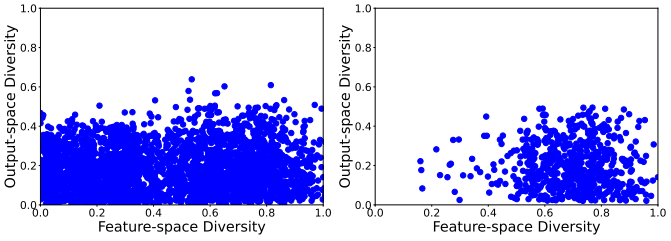
ensemble's predictive capability on GTSRB, when trained with golden and 30% mislabelling respectively. Because ensembles using simple majority voting yield correct predictions as long as there are at least 2 correct models, **we focus on the 1-correct cases** (note that 0-correct cases are unrecoverable). When training data faults are injected into the dataset, the proportion of the 1-correct cases increases from 3% to 12%. The increase in 1-correct cases shows we need an alternative approach to simple majority voting, under training data faults.

We consider using uniform average voting (*i.e.* soft voting [47]). However, since the prediction probabilities are relatively centred around a single class, we observe uniform average voting produce similar results to simple majority voting. This leaves the option of using weighted ensembles.

Motivation 1: Simple majority and uniform average voting for ensembles are insufficient under faulty training data.

Should we use output-space or feature-space diversity? For each input in GTSRB's test dataset, we evaluate the ensemble's output-space and feature-space diversity under 30% mislabelling. The output-space diversity is calculated using the Shannon entropy H (Section II-D), which ranges between 0 (no diversity) and 1 (most diverse). The feature-space diversity is calculated using R^2 as the metric and SHAP as the XAI technique. Since R^2 ranges between 0 (most diverse) and 1 (no diversity), we represent the feature-space diversity by computing $1 - R^2$. We show the scatterplot of the output-space Vs. feature-space diversity in Fig. 4a. We observe that there is a greater range of diversity values in the feature space compared to the output space. This implies that feature-space diversity better captures the differences between models under training data faults compared to output-space diversity.

Motivation 2: Ensembles have a larger range of feature-space diversity compared to output-space diversity.



(a) All ensemble cases (b) 1-correct cases only

Fig. 4: Output-space vs Feature-space Diversity evaluated on the most resilient 3-model ensemble against 30% mislabelled GTSRB. Each point represents a single evaluated test input.

We then show a scatterplot (Fig. 4b) where we only plot the test inputs with the 1-correct constituent models, which is the only case that can be rectified with weighted voting. We observe that 1-correct cases generally have a higher feature-space diversity compared to other cases. Models in the ensemble tend to focus on similar features when encountering 3-correct and 0-correct cases. However, we still observe significant variations in feature-space diversity among test inputs, despite using the same models in the ensemble. This motivates the case for dynamically weighted ensembles using feature-space diversity, where weights can be uniquely adjusted per input.

Motivation 3: Dynamic weights may exploit variations in ensemble feature-space diversity to provide resilience.

IV. Methodology

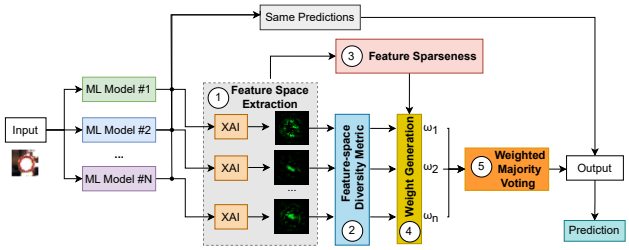


Fig. 5: Workflow diagram of ReMlX when applied on an example road sign image from GTSRB.

As shown in Fig. 5, ReMlX consists of five components: (1) Feature Space Extraction, (2) Feature-space Diversity Metric, (3) Feature Sparseness, (4) Weight Generation and (5) Weighted Majority Voting. ReMlX extracts the feature spaces of the models in the ensemble, calculates their feature-space diversity and their feature sparseness, generates the model weights, and combines the models' predictions through weighted majority voting. Models with greater feature-space diversity are assigned higher weights as diversity is the key component behind resilience in ensembles [8], while using feature sparseness to ensure that the diversity is useful for improving resilience [48]. While ReMlX requires an XAI technique to generate the feature space and a diversity metric to compare the feature matrices, *it does not depend on any specific XAI technique or diversity metric.* For efficiency, when all models predict the same label, ReMlX directly outputs the label since ensembles have no influence under such outcomes. We describe each component in further detail.

(1) Feature Space Extraction. ReMlX applies an XAI technique (*i.e.* Counterfactual Explanations, Integrated Gradients, LIME, SHAP, or Smooth Gradients) to extract the feature space from the ML model, and outputs a 2D feature matrix.

(2) Feature-space Diversity Metric. Consider an ensemble composed of N individual models, where there is a disagreement among predictions. For each individual model, its prediction is passed to the XAI module, which uses an XAI technique to attribute the prediction to its feature space and generates a feature matrix.

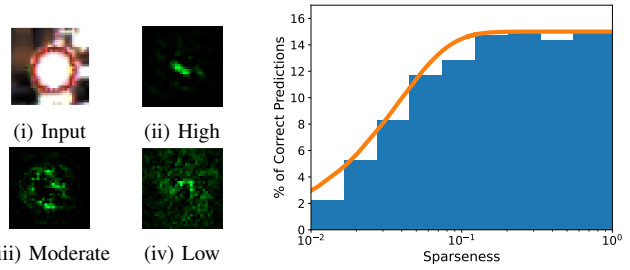
The diversity is calculated pairwise using a diversity metric between the feature matrices of two constituent models. For an ensemble of N models, there are $P = \frac{N(N-1)}{2}$ number of pairs to compute. The pairwise diversities are averaged for each model. For example, in a 3-model ensemble (M_1, M_2, M_3), the feature-space diversity is computed between: (M_1, M_2), (M_1, M_3) and (M_2, M_3). The average diversity δ_i for M_1 is the average of (M_1, M_2) and (M_1, M_3).

For R^2 and Cosine Distance, where higher values indicate less diversity (inverse relationship), we take its reciprocal to determine weights. Because the Frobenius Norm and Wasserstein Distance increase proportionally as diversity increases, we use their diversity values directly for weights.

(3) Feature Sparseness. We hypothesize that the sparseness of feature matrices can determine whether models are focused, and more likely to yield correct predictions. One of the key advantages of feature-space diversity over output-space diversity, is the ability to determine whether a model's diverse prediction is useful [48]. A model could provide a diverse prediction despite not focusing on any specific features in the input. For example, a model predicting the road sign image in Fig. 6a(i) could focus on either specific pixels (Fig. 6a(ii), Fig. 6a(iii)) or erroneously consider every single pixel, including irrelevant pixels in the background (Fig. 6a(iv)).

We quantify a model's focus on features by measuring the sparseness σ of its feature matrix - the proportion of values in a matrix equal to 0 or close to 0 (we count values less than 0.01) over the dimensions of the matrix. The feature sparseness σ ranges from 0 (least sparse) to 1 (most sparse).

We test our hypothesis by evaluating feature sparseness against the GTSRB test dataset. For each test image, we measure an individual model's feature sparseness and note whether it was correctly predicted. We conduct this experiment for the three best individual models for GTSRB (ConvNet, ResNet50, VGG11). We bin sparseness into 10 logarithmic bins between 0.01 and 1, and report the percent of correct predictions per bin, as shown in Fig. 6b. We observe that models having feature matrices with low sparseness are more likely to yield incorrect predictions. Thus, we reduce the weight of models that yield very low sparseness ($\sigma < 0.1$) feature matrices by applying a hyperbolic tangent activation, $f(\sigma) = \tanh(\alpha\sigma)$, over sparseness σ. This follows the trendline over Fig. 6b.



(a) Feature Sparseness Levels (b) Correct Predictions vs. Sparseness

Fig. 6: (a) Sparseness example. (b) Correctly predictions of GTSRB vs Sparseness in log scale. Trendline: $y = \tanh(20x)$.

(4) Weight Generation. Eq. (5) shows how the weight ω_i is calculated for the ith-model in the ensemble, where δ_i is the feature-space diversity value, σ_i is its sparseness, and c_i is its prediction confidence. The weight generation formula, based on the well-established hyperbolic tangent activation function [49], enables models with greater feature-space diversity, sparseness and prediction confidence to be weighed higher than other models. Prediction confidence is commonly used (*i.e.* typically multiplied) for weight generation in dynamic weighted ensembles [50–52]. The weight is a product of the prediction confidence, diversity value, and feature sparseness (shown empirically fitted in Fig. 6b).

$$\omega_i = c_i \delta_i \tanh(\alpha\sigma_i) \tag{5}$$

(5) Weighted Majority Voting. ReMlX combines the predictions from each individual model using weighted majority voting. Votes for each predicted label class are tallied, and the label class that receives at least 50% of the total votes, becomes the final result. In the case of pluralities, where a predicted label class gathers the most votes, yet falls short of the 50% majority threshold, ReMlX considers it a misprediction. In some safety-critical domains such as AVs, disengagement and reversion to manual control is preferred over potential accident-inducing misclassifications [53, 54].

V. Evaluation

A. Research Questions (RQs)

1) How resilient is ReMlX compared with baselines?
2) What is the runtime overhead for ReMlX?
3) Which XAI technique is most faithful, robust, and efficient under training data faults?
4) Which feature-space diversity metric for ReMlX is most effective?
5) How well does ReMlX perform on larger ensembles?

B. Experimental Setup

Datasets. Our evaluation includes three datasets (Table II): CIFAR-10, GTSRB, and Pneumonia - these are all publicly available. CIFAR-10 is comprised of about 50,000 photos organised into ten separate object categories, with 5000 images in each class. The Pneumonia dataset [55] contains 5,863 X-ray images of paediatric patients from China. GTSRB and Pneumonia are both considered safety-critical applications. Pneumonia datasets are smaller than others because of challenges in curating high-quality medical images [56]. Across all datasets, we use the pre-defined training and testing splits.

TABLE II: Image classification datasets and metrics used

Name	Dataset Size		Task (# Classes)	Evaluating
	Training	Test		Metric
CIFAR-10 [57]	50,000	10,000	Objects (10)	BA
GTSRB [12]	39,209	12,630	Traffic signs (43)	BA
Pneumonia [55]	5,239	624	Chest X-rays (2)	F_1

Models. We use 9 different models of varying architectures, outlined in Table III. We selected these 9 models that have varied layer depths and architectural components, such as depthwise convolution layers in MobileNet and residual layers in ResNet. We also use two variants of EfficientNetv2 [58] that perform better on larger images. Prior work have shown that these individual models exhibit high accuracy (above 90%) across the image classification datasets [8]. These models have been adopted in safety-critical applications such as VGG models for COVID detection [59] and MobileNets for distracted

driver detection [60]. Hence, these models serve as a strong foundation for constructing resilient ensembles.

TABLE III: Neural network architectures used

Name	Depth	Architecture Summary
ConvNet	Moderate	3 Conv + 3 FC + Max Pooling
DeconvNet	Moderate	4 Conv + 2 FC w/ 0.5 Dropout
VGG11	Deep	13 Conv + 3 FC + Max Pooling
VGG16	Deep	13 Conv + 3 FC + Max Pooling
ResNet18	Deep	17 Conv + 1 FC + Avg Pooling
MobileNet	Deep	27 Conv + 1 FC + Avg Pooling
ResNet50	Deep	49 Conv + 1 FC + Avg Pooling
EfficientNetv2B0	Deep	5 Fused-MBConv + 16 MBConv + 1 FC
EfficientNetv2B1	Deep	5 Fused-MBConv + 16 MBConv + 1 FC

Fault Injection. We utilize the TF-DM fault injector [61] to inject mislabelling, removal, and repetition faults into training datasets. Mislabelling faults have a greater influence on certain label classes than others, while removal and repetition errors are equally likely to impact any label class. Thus, we employ asymmetric fault distributions for mislabelling faults and symmetric distributions for removal and repetition faults.

Using Cleanlab [62], a tool to detect training data faults, mislabelling fault patterns are extracted from datasets. Mislabelling fault patterns resemble confusion matrices, where each column represent the actual class, each row represents the predicted class, and each element represents the magnitude of confusion. TF-DM performs fault injection based on these extracted fault patterns. Training labels belonging to classes with a higher confusion are more likely to be replaced.

We inject the three types of training data faults (Section II-A), with one fault type and one fault amount per run - we call this a *fault configuration*. The fault amounts (10%-50%) resemble the typical quantities in real datasets.

Ensemble Training. We train each model independently, on the same pre-defined training split of each dataset. Ensembles are constructed by combining the independently trained models. We initially consider ensembles of size 3, which is the minimum number of models required for simple majority voting. Because we have 9 models, and we select ensembles of size 3, we have a total of $\binom{9}{3}$ = 84 models. We choose the most resilient ensemble under each fault configuration and apply ReMlX on that ensemble.

Baselines. First, we compare ReMlX with the best individual model that has the highest resilience for a fault configuration. Then, we compare ReMlX with six state-of-the-art ensembling techniques, as follows:

1) *UMaj*, unweighted simple majority [8]. Ensembles are constructed with individually-trained architecturally-diverse models - the predictions of each individual model are combined with simple majority voting.
2) *UAvg*, uniform average [47]. These are similar to simple majority ensembles. However, their predictive probabilities are averaged instead and the class with the highest probability is chosen as the ensemble prediction.
3) *S-WMaj*, static weighted majority [16]. Models in the ensemble are weighed according to their prediction accuracy, across a validation dataset (*i.e.* a small random subset of the training dataset withheld during training).
4) *D-WMaj*, dynamic weighted majority, using stacking [50]. Weights are assigned to predictions by constituent models during inference. We use stacking, where the predictions of individual models are fed into another classifier (Logistic Regression model), which decides the final prediction. Unlike ReMlX, D-WMaj is based on output-space prediction correctness and prediction confidence to dynamically determine weights.
5) *Bagging* [11]. Ensembles are constructed by training the same ML architecture on subsets of the training dataset that are randomly sampled with replacement. We set the bagging percentage to 63% as recommended [11].
6) *Boosting*, using AdaBoost [10]. We use AdaBoost [10], an adaptive boosting approach where models of the same ML architecture are sequentially sequenced to focus on data instances that are mispredicted by prior models.

ReMlX Configurations. We configure ReMlX to use Smooth Gradient (SG) as the XAI Technique (we show why in Section V-E) and Cosine Distance (we show why in Section V-F). We choose a high value for α (*i.e.* 20), so that only explanations with extremely low sparseness are penalized.

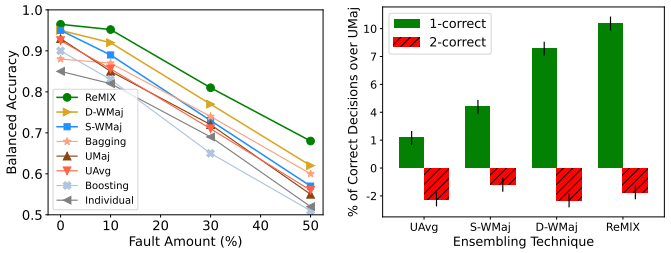
Metrics. We use balanced accuracy (BA) [63] instead of accuracy, as it measures the classification capability for both balanced and imbalanced multiclass datasets. In imbalanced datasets, where the number of samples belonging to each class are unequal, accuracy is a misleading metric as it is biased towards the performance on overrepresented classes. BA eludes this case by calculating the recall on each class separately, and averaging them [63]. For balanced datasets like CIFAR-10, BA converges to the accuracy [64].

To evaluate Pneumonia, a binary dataset, where false positives and negatives (*i.e.* misdiagnosis) are more important than true negatives (*i.e.* benign), we employ the F1 Score (F_1), which reports the harmonic mean of the precision and recall.

Experimental Environment. For our experiments, we used a 64-bit AMD Ryzen Threadripper 3960X 24-Core Processor with 256GB RAM and three NVIDIA RTX 3070 GPUs. In total, training and inference time took 16 days of computation time. Each ensembling approach was run 20 times per fault configuration to reduce its variance and obtain tight bounds within a 95% confidence interval. Models in the ensembles are run in parallel during inference.

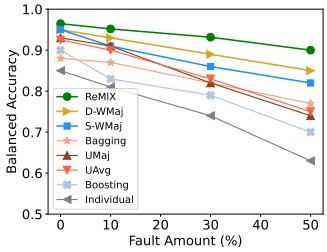
C. RQ1: Resilience of ReMlX vs Baselines

We compare the resilience of ReMlX with the baselines, against randomly injected training data faults using TF-DM. While ReMlX can be applied to ensembles of any size, we focus initially on models of size 3, and apply our technique on the most resilient three model ensemble, as explained in Section V-B. Similarly, all other ensembling baselines utilize three models. Overall, we find that ReMlX is 12% more resilient than DW-Maj, 16% more than bagging, 21% more than S-WMaj, 24% more than UMaj and UAvg, 28% more than boosting and the best individual model.

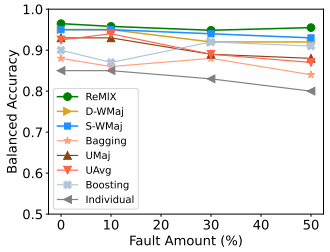


(a) GTSRB, Mislabelling

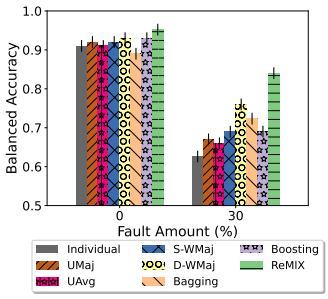
(b) GTSRB, Mislabelling



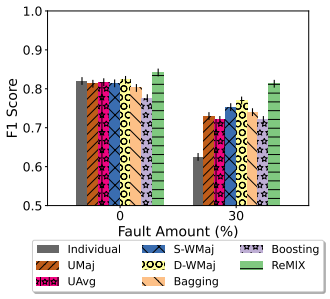
(c) GTSRB, Removal



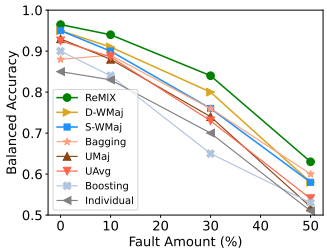
(d) GTSRB, Repetition



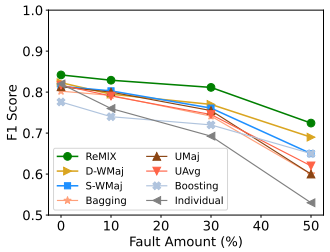
(e) CIFAR-10, Mislabelling



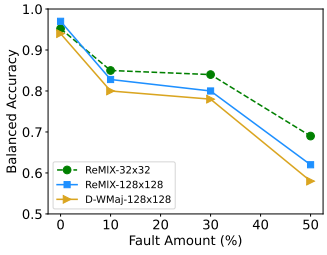
(f) Pneumonia, Mislabelling



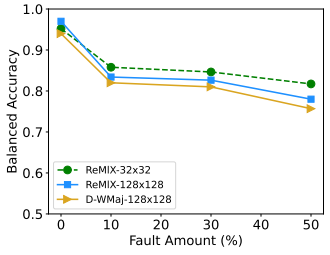
(g) GTSRB, Mislabelling & Removal Combined



(h) Pneumonia, Mislabelling & Removal Combined



(i) CIFAR-10, Effect of Image Size, Mislabelling



(j) CIFAR-10, Effect of Image Size, Removal

Fig. 7: BA, F_1 of ReMIX vs baselines across datasets and fault types. Error bars indicate 95% CI. Y-axis begins at 0.5. ReMIX is more resilient than other baselines across configurations.

By Fault Amount. First, we evaluate the resilience of ReMIX across different fault amounts (0%-50%), which resemble observations in real datasets (Table I). We randomly inject varying amounts of mislabelling faults into the GTSRB training dataset, following their extracted fault distributions. We then retrain the best ensemble on the dataset for each fault amount, and measure the balanced accuracy (BA) of ReMIX along with the baselines. We show the results in Fig. 7a.

We observe that ReMIX has a higher resilience compared to baselines across fault amounts, with the gap in resilience increasing as the fault amount increases. D-WMaj and bagging are the most resilient baseline techniques across fault amounts. While D-WMaj's dynamic weights provides the strongest alternative, ReMIX is still able to outperform it under all fault amounts ($\geq$ 10%). We see that ReMIX is able to correctly predict 1-correct cases that D-WMaj fails to do, owing to D-WMaj's final decision classifier limited to the output-space (*i.e.* constituent model's prediction probabilities). Bagging also performs well under higher fault amounts - its constituent models trained on random subsets of the training dataset yields higher resilience compared to other baselines. However, ReMIX consistently outperforms bagging across all fault amounts, as it has the same limitation as D-WMaj.

Techniques such as UMaj, UAvg and S-WMaj only have high BA at golden and 10% mislabelling. In particular, UMaj and UAvg have similar BA across fault amounts. Since most models tend to predict labels with high prediction probability, averaged voting did not play a deciding factor in most ensemble outcomes. S-WMaj underperforms at high fault amounts as the model weights are statically calibrated during training. These static weights are often suboptimal during inference, as ensemble diversity fluctuates across input instances.

Most of the ensembling baselines outperform the best individual model, with the exception of boosting. Boosting is even outperformed by the best individual model, under higher fault amounts ($\geq$ 30%). This is due to boosting's sequential learning pattern, where subsequent models focus on mispredicted samples by prior models, which increases the ensemble's suspectibility to training data faults.

We further examine how effectively ReMIX provides resilience compared to the baseline ensembling techniques (UAvg, S-WMaj, D-WMaj) in Fig. 7b that use three architecturally-diverse constituent models. First, we count the proportion of 1-correct cases that are correctly classified by each ensembling technique. This shows the improvement over UMaj (simple majority voting), which is unable to correctly classify 1-correct. Then, we count the proportion of 2-correct cases misclassified by each ensembling technique. Ensembling techniques may misclassify 2-correct cases, when simple majority voting is no longer used. We can see ReMIX is able to correctly predict a large proportion of 1-correct cases that are mispredicted by the other ensembling baselines (*i.e.* UAvg), while minimizing mispredictions on 2-correct cases.

By Fault Type. Next, we evaluate the resilience of ReMIX across different fault types. Earlier, we considered mislabelling faults. We randomly inject varying amounts of removal and

repetition faults into the GTSRB training dataset, and repeat the same experiment. We show the results in Figs. 7c and 7d.

We observe that ReMlX outperforms all other baselines for removal and repetition faults as well. ReMlX is most effective against mislabelling and removal faults, where the gap in resilience with baselines is widest. For repetition faults, ReMlX displays similar resilience as two ensembling baselines (S-WMaj and D-WMaj), while still outperforming the other baselines. Unlike mislabelling and removal faults, there is less variation in prediction diversity among test inputs for repetition faults. This is likely due to repetition fault's smaller impact on trained models. However, weighted ensembles continue to yield more correct predictions than unweighted ensembles, owing to the diversity among constituent models.

By Dataset. Finally, we evaluate the resilience of ReMlX across different datasets (CIFAR-10 and Pneumonia) by repeating the previous experiments. We focus our analysis on 30% mislabelling as it represents the average scenario for training data faults. We measure the BA for CIFAR-10 (Fig. 7e) and F_1 for Pneumonia (Fig. 7f), and report the performance of each technique with and without mislabelling.

When examining the results for CIFAR-10 and Pneumonia, we make similar observations in trends to GTSRB. ReMlX continues to outperform other baseline techniques, and D-WMaj and Bagging are the best alternative baselines. There is no significant difference between ReMlX's resilience in GTSRB and CIFAR-10. However, ReMlX has a lower gap in resilience between D-WMaj for Pneumonia. When models are trained under mislabelling, models are less likely to focus on important features, especially in high resolution images found in Pneumonia. The lack of focus yields low feature sparseness, limiting the model's contribution to ensemble diversity.

With Multiple Fault Types. In addition to evaluating ReMlX against individual fault types, we also consider combinations of fault types. Since repetition faults had a smaller impact on resilience, we focus on combinations of mislabelling and removal faults. We inject (0% to 50%) of a combination of mislabelling and removal, with each fault type at equal amounts. For instance, we inject 15% mislabelling and 15% removal for 30% faults. We show the results for GTSRB (Fig. 7g) and Pneumonia (Fig. 7h). ReMlX behaves similarly relative to baselines in GTSRB and Pneumonia when trained with multiple faults compared to mislabelling faults only. A steeper drop in reliability is observed at fault amounts exceeding 30% as the effects of the fault types are compounded.

By Image Size. Because ReMlX relies on the feature space of an input image, we evaluate ReMlX on different image sizes to determine whether the image size has an impact on resilience. We compare the results for CIFAR-10 where images are sized 32×32 to CIFAR-10-128 [65], a dataset of automatically resized CIFAR-10 images of size 128×128. The ensemble with the highest average resilience for CIFAR-10 was {ConvNet, ResNet50, VGG11} whereas for CIFAR-10-128, it was {MobileNet, EfficientNetv2B0, EfficientNetv2B1}.

We compare the resilience of ReMlX under different image sizes against its most competitive baseline, D-WMaj, under mislabelling and removal faults (Figs. 7i and 7j). We excluded repetition faults here due to the limited impact on resilience. We find that the BA drops more quickly when training data faults are present on datasets with larger images due to the increased risk of overfitting. Nevertheless, ReMlX still outperforms D-WMaj because it is able to effectively navigate through ensemble disagreements, which occur more frequently when inferring larger images under training data faults.

Observation 1 *ReMlX offers the highest resilience compared to the baselines across the board, and has the highest effectiveness against mislabelling and removal faults.*

D. RQ2: Runtime Overhead of ReMlX

In this RQ, we measure the runtime (inference) overheads of ReMlX and other ensembling baselines over the best individual model, across each test dataset, averaged across 20 runs each. We focus on the inference overhead over individual test inputs as ReMlX's dynamic weight generation approach is deployed during inference. In certain safety-critical applications such as AVs, there are bounds on the inference time such that it should not exceed the safe disengagement time, in order to enable a timely reversion to manual control if needed [66].

Fig. 8 shows the results. We find that ReMlX incurs $1.15\times$ the overhead of D-WMaj, $2\times$ that of boosting, $4.5\times$ that of UMaj, UAvg, S-WMaj, Bagging, and $6\times$ that of the best individual model. While ReMlX incurs the largest runtime overhead compared to other baselines due to the cost of running the XAI module to extract the feature space, it is the most resilient. For instance, while ReMlX is 15% slower than D-WMaj, it is also 12% more resilient on average. Breaking down ReMlX's overhead in our workflow Fig. 5, we find that ReMlX spends 15% on ensemble prediction, while spending 67% on extracting the features (Step 1) and 18% on computing the diversity and generating weights (Steps 2-5).

We analyze the effect of image size on ReMlX's overhead. ReMlX's overhead for CIFAR-10 (128×128) is $1.63\times$ higher than CIFAR-10 (32×32). Larger images slow down both the ensemble inference (without ReMlX) and the XAI module.

Most importantly, the average and worst-case runtime overhead of ReMlX for time-sensitive safety-critical applications such as GTSRB is 0.07 seconds and 0.32 seconds respectively, which is well within the AV industry standard for disengagement time at 0.83 seconds [66]. Among all datasets, Pneumonia has the highest runtime overhead as it handles the largest image sizes (1024×1024). Its average and worst-case overheads are 0.31 seconds and 0.46 seconds respectively. Overall, ReMlX's overhead is still within 0.5 seconds, which is an acceptable latency for many applications, e.g., in virtual reality (VR) such as telesurgery before VR sickness is encountered [67]. We observed similar trends in runtime overheads across all four datasets, including CIFAR-10-128.

Observation 2 *ReMlX incurs 15% higher performance overhead over its best alternative, D-WMaj, dynamically weighted using stacking.*

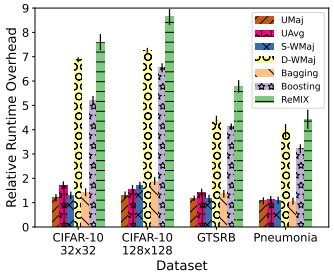
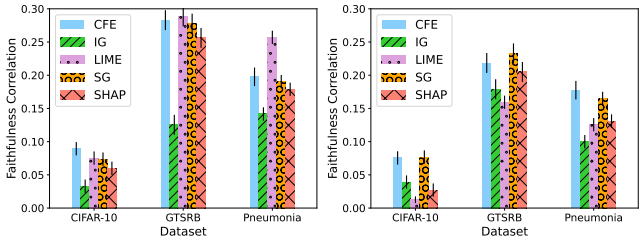


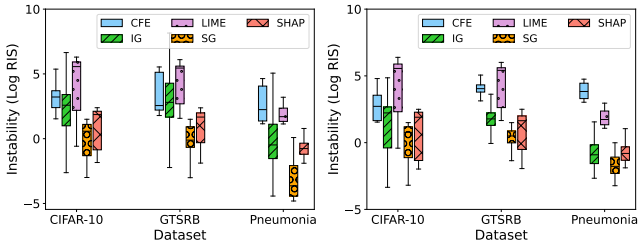
Fig. 8: Average runtime overhead of ReMlX and baselines over the best individual model. Error bars indicate 95% CI.

E. RQ3: Which XAI Technique?

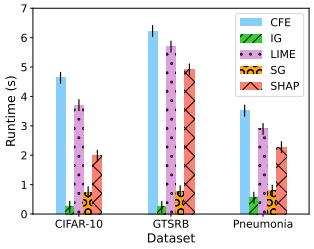
As explained in Section II-C, we seek XAI techniques that are faithful, robust, and efficient. We consider five XAI techniques (Section Section II-C): Counterfactual Explanations (CFE), Integrated Gradients (IG), LIME, Smooth Gradients (SG), and SHAP. We consider both model-agnostic (CFE, LIME, SHAP) and model-dependent techniques (IG and SG).



(a) Faithfulness, Golden (b) Faithfulness, Mislabelling 30%



(c) Robustness, Golden (d) Robustness, Mislabelling 30%



(e) Absolute Runtime per Test Input

Fig. 9: Comparison of XAI Techniques with golden and 30% mislabelling. (a, b) Faithfulness. Higher correlation is better. (c, d) Robustness using Log Relative Input Stability. Lower is better. (e) Runtime. Lower is better. Error bars show 95% CI.

We evaluate each XAI technique to determine which technique yields the best explanations of ML models, when trained with golden and 30% mislabelling - we find similar results for other fault configurations. We apply each XAI technique on the 9 individual models separately. The XAI evaluation metrics are then averaged across all 9 models. We repeat this for the mislabelled cases, across all three datasets.

For faithfulness, we use faithfulness correlation [68], where a higher value indicates higher faithfulness. We calculate the faithfulness correlation for each sample in the test dataset and report the averaged values for golden (Fig. 9a) and mislabelling (Fig. 9b). We find CFE and SG have the highest faithfulness, while IG has the lowest faithfulness. SHAP ranks in the middle among techniques. LIME's faithfulness is most negatively impacted by mislabelling, as it struggles to fit its surrogate model under training data faults.

For robustness, we use Relative Input Stability [69], which is applied inversely, so a lower value indicates greater stability. Since Relative Input Stability is unbounded, we take its log and plot a boxplot of the stability values for golden (Fig. 9c) and mislabelling (Fig. 9d). SG has the lowest Relative Input Stability, indicating its high stability. SG mitigates the noise from background elements. The stability of CFE is most impacted by faulty training data, while other techniques remained stable. Even without faults, CFEs are most unstable as it performs a heuristic search for counterfactuals that can differ per run.

For efficiency, we measure the runtime in seconds, for a single XAI technique to run on a test input. We repeat this for each sample in the test dataset and report the averaged runtime in Fig. 9e. While IG has the lowest overall runtime, SG comes second. Model-dependent techniques (IG, SG) are faster than model-agnostic techniques (CFE, SHAP, LIME). IG and SG use differentiation (quickly computable), while SHAP incurs exponential runtime to generate all feature combinations, CFE relies on heuristic search and LIME needs to fit surrogates.

Overall, SG is the XAI technique with the highest faithfulness and robustness, and incurs the second lowest runtime.

Observation 3 *Smooth Gradients (SG) is the most well-suited XAI technique for ReMlX.*

F. RQ4: Which feature-space diversity metric?

We compare the four feature-space diversity metrics – R^2, Cosine Distance, Frobenius Norm, Wasserstein Distance (shortlisted in Section II-D) – to determine which is most effective for ReMlX. An effective feature-space diversity metric maximizes resilience in ensembles while incurring the least runtime. We perform an experiment where we apply each of the four diversity metrics to ReMlX, with SG as the XAI technique, and measure their BA. Fig. 8 shows the results for GTSRB against varying amounts of mislabelling. We make similar observations for other fault configurations and datasets.

Despite observing small variations (within 5%) in resilience when adapting ReMlX with different diversity metrics, R^2 and Cosine Distance fares generally better than Frobenius Norm and Wasserstein Distance. R^2 and Cosine Distance are scaling-invariant, so they focus on the relative dissimiliarity of the feature matrix elements, and are independent of their absolute

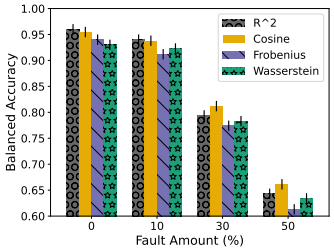
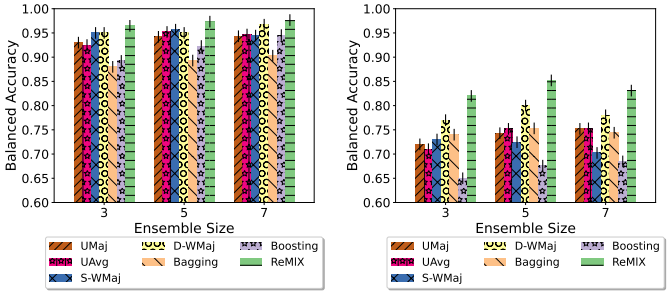


Fig. 10: BA when using different feature-space diversity metrics with ReMlX, against different amounts of mislabelling. Error bars show 95% CI. Y-axis starts at 0.6.



(a) Golden (b) 30% Mislabelling

Fig. 11: BA of ReMlX vs baselines for GTSRB under (a) golden and (b) 30% mislabelling. Error bars indicate 95% CI. Y-axis begins at 0.6.

magnitudes. In particular, Frobenius Norm is the least effective metric for ReMlX as it amplifies elemental magnitude differences due to its squared distance computed between feature matrix elements - a more pronounced observation at higher fault amounts. While R^2 and Cosine Distance are both effective, Cosine Distance's implementation is much faster than R^2. On average, Cosine Distance takes 0.3 milliseconds, while R^2 takes 3 milliseconds, equating to a $10\times$ speedup. This makes Cosine Distance a better metric for ReMlX.

Observation 4 *Cosine Distance is the most effective feature-space diversity metric for ReMlX.*

G. RQ5: ReMlX vs Ensemble Size?

Previously, we focused our evaluation on ensembles of three models only. However, some baselines such as boosting perform better when more constituent models are present [70]. We evaluate and compare the resilience of ReMlX against the ensembling baselines, at different ensemble sizes (3, 5, 7).

We show the results for GTSRB at golden (Fig. 11a) and 30% mislabelling (Fig. 11b). Similar results were found for other datasets and fault configurations. We find ReMlX has the highest resilience when the ensemble size is 5 models, which is also the case for D-WMaj. This is because ensemble resilience tends to saturate at 5 models, as also observed in prior work [8]. Interestingly, for S-WMaj, its resilience drops as the ensemble size increases. This is because static weights are inflexible to handle disagreements between models, even underperforming uniform weights. In larger ensembles, the likelihood of disagreements between constituent models increases, requiring a more dynamic approach for handling diversity (*i.e.* dynamic weights). No significant differences were observed in the resilience of other baselines across ensemble sizes. Overall, ReMlX still maintains a higher resilience compared to the baselines across different ensemble sizes.

Observation 5 *ReMlX has the highest resilience across different ensemble sizes compared to the baselines.*

VI. Discussion

Threat to Internal Validity. We assume that the labels in the test dataset correspond to the ground truth, although this may not consistently hold due to faults in the test data. Similar to software engineering, one does not typically assume that both a program *and* its tests contain faults [71, 72].

Threats to External Validity. We evaluated ReMlX and other approaches separately against each fault type. In practice, datasets may contain multiple fault types. Evaluating ReMlX against multiple fault types is a direction for future work.

We use TF-DM to evaluate ReMlX and the baselines against training data faults. TF-DM relies on Cleanlab [62] to accurately extract mislabelling fault patterns from datasets. Since Cleanlab utitilizes statistical methods to identify mislabelled data, its inferred mislabelling patterns may differ from the actual fault patterns present in the dataset due to false positives and false negatives. We mitigate this risk by raising the reporting confidence threshold in Cleanlab, and manually verifying the samples of reported mislabelling.

Threats to Construct Validity. While we rely on XAI techniques to generate a faithful representation of the feature space, XAI techniques may not always provide a faithful explanation. We mitigate this risk by evaluating the faithfulness and robustness of XAI techniques, and carefully choosing a feasible XAI technique based on this critera. We also manually inspect the visual heatmaps of the feature matrices.

Applicability to Other ML Tasks and Data Modality. While we demonstrate ReMlX's functionality on image classification datasets, ReMlX could also be applied to other ML tasks such as text and tabular data. LIME, SHAP and Integrated Gradients are also applicable to text and tabular data. However, the XAI techniques would generate 1-D vectors of influence scores on the final prediction instead of 2-D vector, which would require new (vector-based) diversity metrics.

Combination with Other Training Data Fault Tolerance Strategies. ReMlX is one solution to tolerate the presence of training data faults, which we evaluate as an improvement over simple majority ensembles. One also has the option of applying data cleaning techniques (*i.e.* using Cleanlab to partially remove mislabelled data), or leveraging robust training techniques on individual models. Evaluating ReMlX in concert with other strategies is an avenue for future work.

Optimizations to Runtime Overhead. Shown in Section V-D, post-hoc XAI is the major source of ReMlX's

overhead. We discuss three optimizations: more efficient post-hoc XAI, ante-hoc techniques and quantized models.

Faster post-hoc XAI techniques such as FusionGrad [73] and AdaptGrad [74] have been studied. These techniques improve upon Smooth Gradients by reducing model parameter perturbations needed to generate robust explanations. However, such optimized techniques sacrifice faithfulness [44].

Ante-hoc techniques [36] avoid a separate XAI step after inference (see Section II-C). Self-Explaining Neural Networks [75] is an example anti-hoc technique, successfully applied for image classification. Vision transformer models (ViTs) [76] intrinsically incorporate attention layers, which can represent the feature space. However, ante-hoc techniques and attention mechanisms require more trained parameters, thus, occupying a higher memory footprint [77]. Further, they slow down inference for all test inputs [78], not only those with disagreements like in ReMlX. Additionally, adapting models for ante-hoc explainability by modifying network architectures can degrade predictive capability [78].

Quantized models can improve the efficiency of ensemble inference and XAI techniques by shortening bit widths to represent model parameters. Shortened bit widths have negligible impact on explainability in quantized models but they can diminish the predictive capability of models [79].

Example of Applying ReMlX on ViTs. Suppose an ensemble of independently trained ViTs is constructed as shown in Fig. 12. A ViT consists of a module to produce patches of the input image, followed by a transformer encoder with a multi-head attention module, and a multi-layer perceptron (MLP) head [76]. To apply ReMlX, we take the attention scores from the (1) multi-head attention layer, and (2) apply diversity metrics (Section II-D) to compare attention scores between ViTs. Since ViTs intrinsicly incorporate attention in their architectures, a separate post-hoc XAI step is unnecessary.

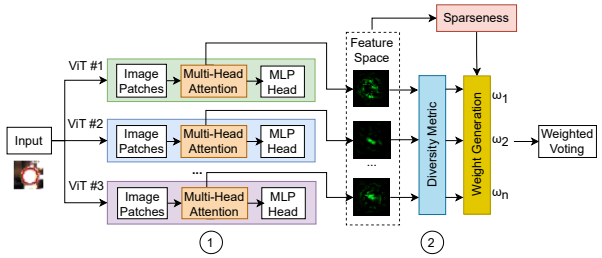


Fig. 12: Workflow of ReMlX applied on an ensemble of VITs.

VII. Related Work

We classify related work into two categories: (1) statically weighted ensembles, and (2) dynamically weighted ensembles.

Statically-weighted Ensembles. Most ensembles that run multiple models in parallel (bagging) are combined through unweighted majority voting [80]. Weighted ensembles can improve accuracy for datasets with small quantities of label noise and unbalanced classes [13].

Iqball et al. [15] introduce a weighted ensemble where weights are calculated based on the predictive capability of each model. Models are assigned a static weight equal to their relative accuracy improvement over the lowest accuracy model. While their work considers a model's global predictive capability, we focus on local classification of inputs.

Gao et al. [81] propose safety-aware weighted ensembles for traffic sign recognition. By conducting a failure modes and effects analysis, they generate a severity matrix and compute misclassification probabilities. The analysis is summarized and passed to a large language model, which generates static safety-aware weights. Unlike their work, we do not require a manual failure analysis or the use of a large language model.

Dynamically-weighted Ensembles (DWEs) have been proposed as an alternative to statically-weighted ensembles to handle unseen inputs, in datasets with label noise [14, 82], imbalanced classes [83], and missing labels [84].

Ren et al. [14] propose DWEs based on test input features during inference. They present an algorithm to calculate dynamic weights based on the eigenvalues of the confusion matrix of each model, and found DWEs outperform statically-weighted ensembles by suppressing misclassifications against certain classes of inputs. In contrast, our technique is customized for each test input rather than label classes.

Zhang et al. [83] apply DWEs on class-imbalanced datasets to improve classification accuracy on minority classes since statically-weighted ensembles bias predictions towards majority classes. Constituent models are trained on distinct combinations of samples from majority and minority classes. At inference, dynamic weights are generated based on input feature similarity to training samples of the minority class. Unlike their work, our solution exploits feature diversity among models rather than training samples.

Catto et al., [84] apply DWEs towards missing value imputation, where training samples with missing labels are approximated by similar samples. DWEs enable the best imputation technique to be applied for each training sample. In contrast, we apply DWEs directly on test predictions, rather than on training data, so even pre-trained models can benefit.

VIII. Conclusions

ML applications require accurate predictions, especially in safety-critical systems. Training data faults can 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 ReMlX, which deploys dynamically-weighted ensembles based on the feature-space diversity between constituent models using local post-hoc explainable AI techniques. ReMlX achieves 12% higher resilience than the best alternative approach, dynamic weighted ensembles using stacking, while incurring 15% higher runtime overhead.

IX. Acknowledgements

We thank Antonio Casimiro, and the DSN'25 reviewers for their invaluable comments. This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and a Four Year Fellowship from 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] C. G. Northcutt, A. Athalye, and J. Mueller, "Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks," 2021, arXiv:2103.14749.

[7] 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.

[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] B. Naderalvojoud and T. Hernandez-Boussard, "Improving machine learning with ensemble learning on observational healthcare data," *AMIA Annual Symposium Proceedings*, 2024.

[10] Y. Freund and R. E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," in *Computational Learning Theory"*, 1995.

[11] L. Breiman, "Stacked regressions," *Machine Learning*, 1996.

[12] 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

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

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

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

[16] G. Chakraborty *et al.*, "A Novel Deep Learning-Based Classification Framework for COVID-19 Assisted with Weighted Average Ensemble Modeling," *Diagnostics*, 2023.

[17] A. Chan, A. Gujarati, K. Pattabiraman, and S. Gopalakrishnan, "Harnessing Explainability to Improve ML Ensemble Resilience," in *Supplementary Proceedings of IEEE/IFIP International Conference on Dependable Systems and Networks (DSN-S)*, 2024.

[18] N. Sambasivan, S. Kapania, H. Highfill, D. Akrong, P. K. Paritosh, and L. M. Aroyo, ""Everyone wants to do the model work, not the data work": Data Cascades in High-Stakes AI," in *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 2021.

[19] D. Kang, N. Arechiga, S. Pillai, P. D. Bailis, and M. Zaharia, "Finding Label and Model Errors in Perception Data With Learned Observation Assertions," in *Proceedings of the International Conference on Management of Data (SIGMOD)*, 2022.

[20] S. Tang, A. Ghorbani, R. Yamashita, S. Rehman, J. A. Dunnmon, J. Zou, and D. L. Rubin, "Data Valuation for Medical Imaging Using Shapley Value: Application on A Large-scale Chest X-ray Dataset," *Scientific Reports*, vol. 11, no. 1, 2021.

[21] B. L. Sturm, "The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use," *CoRR*, 2013, arXiv:1306.1461.

[22] Udacity, "Udacity Self-Driving Car Driving Data 10/3/2016 (dataset-2-2.bag.tar.gz)." [Online]. Available: https://github.com/udacity/self-driving-car

[23] B. Dwyer, "A popular self-driving car dataset is missing labels for hundreds of pedestrians," 2020.

[24] R. Kesten *et al.*, "Lyft Level 5 Perception Dataset 2020," 2020. [Online]. Available: https://level5.lyft.com/dataset/

[25] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "ChestX-Ray8: Hospital-Scale Chest X-Ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases," 2017.

[26] 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.

[27] T.-Y. Lin *et al.*, "Microsoft COCO: Common Objects in Context," 2014, arXiv:1405.0312.

[28] J. Murdoch, "How I found nearly 300,000 errors in MS COCO," https://medium.com/@jamie_34747/how-i-found-nearly-300-000-errors-in-ms-coco-79d382edf22b, accessed: 2022-11-28.

[29] G. Tzanetakis, G. Essl, and P. Cook, "Automatic Musical Genre Classification Of Audio Signals," 2001. [Online]. Available: http://ismir2001.ismir.net/pdf/tzanetakis.pdf

[30] A. Chan, A. Gujarati, K. Pattabiraman, and S. Gopalakrishnan, "D-semble: Efficient Diversity-Guided Search for Resilient ML Ensembles," in *Proceedings of the ACM International Symposium on Applied Computing (SAC)*, 2025.

[31] L. Chen and A. Avizienis, "N-version programming: A fault-tolerance approach to reliability of software operation," in *Proc. of FTCS'78*, 1978.

[32] F. Machida, "N-Version Machine Learning Models for Safety Critical Systems," in *Proc. of DSNW'2019*, 2019.

[33] H. Xu, Z. Chen, W. Wu, Z. Jin, S.-y. Kuo, and M. Lyu, "NV-DNN: Towards Fault-Tolerant DNN Systems with N-Version Programming," in *Proc. of DSNW'19*, 2019.

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

[35] P. Brazdil, J. van Rijn, C. Soares, and J. Vanschoren, *Metalearning: Applications to Automated Machine Learning and Data Mining*, 2022.

[36] S. A. and S. R., "A systematic review of Explainable Artificial Intelligence models and applications: Recent developments and future trends," *Decision Analytics Journal*, 2023.

[37] 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.

[38] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why Should I Trust You?": Explaining the Predictions of Any Classifier," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2016.

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

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

[41] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg,

"SmoothGrad: removing noise by adding noise," in *Proceedings of the Workshop on Visualization for Deep Learning, ICML*, 2017.
[42] L. S. Shapley, "A Value for n-Person Games," *Contributions to the Theory of Games*, 1953.
[43] Y. LeCun, C. Cortes, and C. Burges, "MNIST handwritten digit database," *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, vol. 2, 2010.
[44] P. Bommer, M. Kretschmer, A. Hedström, D. Bareeva, and M. Höhne, "Finding the Right XAI Method-A Guide for the Evaluation and Ranking of Explainable AI Methods in Climate Science," *Artificial Intelligence for the Earth Systems*, 2024.
[45] G. H. Golub and C. F. Van Loan, *Matrix Computations*. The Johns Hopkins University Press, 1996.
[46] S. Kolouri, K. Nadjahi, U. Simsekli, R. Badeau, and G. K. Rohde, "Generalized sliced wasserstein distances," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2019.
[47] K. Cao-Van, T. C. Minh, L. G. Minh, T. T. B. Quyen, and H. M. Tan, "Soft-Voting Ensemble Model: An Efficient Learning Approach for Predictive Prostate Cancer Risk," *Vietnam Journal of Computer Science*, 2024.
[48] P. Popov *et al.*, "Software Diversity as a Measure for Reducing Development Risk," in *Proceedings of European Dependable Computing Conference (EDCC)*, 2014.
[49] B. Kalman and S. Kwasny, "Why tanh: choosing a sigmoidal function," in *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, 1992.
[50] D. H. Wolpert, "Stacked generalization," *Neural Networks*, 1992.
[51] L. Li, L. Mei-Zheng, J.-S. Mi, and B. Xie, "Prediction confidence-based dynamic selection and weighted integration," *Concurrency and Computation: Practice and Experience*, 2018.
[52] D. Do, T. Nguyen, T. Nguyen, V. Luong, A. W.-C. Liew, and J. McCall, *Confidence in Prediction: An Approach for Dynamic Weighted Ensemble*, 2020.
[53] Z. H. Khattak, M. D. Fontaine, and B. L. Smith, "Exploratory Investigation of Disengagements and Crashes in Autonomous Vehicles Under Mixed Traffic: An Endogenous Switching Regime Framework," *IEEE Transactions on Intelligent Transportation Systems*, 2021.
[54] M. Khan, D. LeMaster, and W. G. Najm, "Understanding Safety Challenges of Vehicles Equipped with Automated Driving Systems," *U.S Department of Transportation, Highly Automated Systems Safety Center of Excellence*, 2024.
[55] D. Kermany, K. Zhang, and M. Goldbaum, "Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification," https://data.mendeley.com/datasets/rscbjbr9sj/2, 2018.
[56] Canadian Association of Radiologists (CAR) Artificial Intelligence Working Group, "Canadian Association of Radiologists White Paper on Ethical and Legal Issues Related to Artificial Intelligence in Radiology," *Canadian Association of Radiologists Journal*, 2019.
[57] A. Krizhevsky, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.
[58] M. Tan and Q. V. Le, "Efficientnetv2: Smaller models and faster training," in *Proceedings of International Conference on Machine Learning (ICML)*, 2021.
[59] L. Goyal, A. Dhull, A. Singh, S. Kukreja, and K. K. Singh, "VGG-COVIDNet: A Novel model for COVID detection from X-Ray and CT Scan images," *Procedia Computer Science*, 2023.
[60] M. A. B. Abbass and Y. Ban, "MobileNet-Based Architecture for Distracted Human Driver Detection of Autonomous Cars," *Electronics*, 2024.
[61] N. Narayanan and K. Pattabiraman, "TF-DM: Tool for Studying ML Model Resilience to Data Faults," in *Proceedings of the International Workshop on Deep Learning for Testing and Testing for Deep Learning (DeepTest)*, 2021.
[62] Cleanlab, "Cleanlab," https://github.com/cleanlab/cleanlab, 2024.
[63] V. García, R. A. Mollineda, and J. S. Sánchez, "Index of Balanced Accuracy: A Performance Measure for Skewed Class Distributions," in *Iberian Conference on Pattern Recognition and Image Analysis*, 2009.
[64] P. Thölke *et al.*, "Class imbalance should not throw you off balance: Choosing the right classifiers and performance metrics for brain decoding with imbalanced data," *NeuroImage*, 2023.
[65] J. Schuler, "CIFAR-10 128x128 Resized via CAI Super Resolution," https://www.kaggle.com/datasets/joaopauloschuler/cifar10-128x128-resized-via-cai-super-resolution, accessed: 2025-02-18.
[66] V. Dixit, S. Chand, and D. Nair, "Autonomous Vehicles: Disengagements, Accidents and Reaction Times," *PLOS ONE*, 2016.
[67] K. Brunnström, E. Dima, T. Qureshi, M. Johanson, M. Andersson, and M. Sjöström, "Latency impact on Quality of Experience in a virtual reality simulator for remote control of machines," *Signal Processing: Image Communication*, 2020.
[68] U. Bhatt, A. Weller, and J. M. F. Moura, "Evaluating and Aggregating Feature-based Model Explanations," in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.
[69] C. Agarwal *et al.*, "Rethinking Stability for Attribution-based Explanations," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.
[70] A. Ferrario and R. Hämmerli, "On Boosting: Theory and Applications," *Machine Learning eJournal*, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:199589529
[71] D. Hao *et al.*, "Is This a Bug or an Obsolete Test?" in *Proceedings of the European Conference on Object-Oriented Programming*, 2013.
[72] J. R. Horgan and A. P. Mathur, *Software Testing and Reliability*, 1996.
[73] K. Bykov, A. Hedström, S. Nakajima, and M. M. C. Höhne, "NoiseGrad: Enhancing Explanations by Introducing Stochasticity to Model Weights," 2021.
[74] L. Zhou, C. Ma, Z. Wang, and X. Shi, "Rethinking the Principle of Gradient Smooth Methods in Model Explanation," 2024, arXiv:2410.07711.
[75] D. Alvarez-Melis and T. S. Jaakkola, "Towards Robust Interpretability with Self-Explaining Neural Networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.
[76] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," 2021.
[77] H. Zhao, J. Jia, and V. Koltun, "Exploring Self-attention for Image Recognition," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
[78] A. Sarkar, D. Vijaykeerthy, A. Sarkar, and V. N. Balasubramanian, "A Framework for Learning Ante-hoc Explainable Models via Concepts," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
[79] Y. Tashiro and H. Awano, "Pay Attention via Quantization: Enhancing Explainability of Neural Networks via Quantized Activation," *IEEE Access*, 2023.
[80] L. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Wiley, 2004.
[81] L. Gao, Q. Wen, and F. Machida, "Safety-Aware Weighted Voting for N-version Traffic Sign Recognition System," in *Proceedings of IEEE International Workshop on Reliable and Secure AI for Software Engineering*, 2024.

[82] H. Chen, T. Wang, Y. Zhang, B. Yun, and X. Chen, "Dynamically weighted ensemble of geoscientific models via automated machine-learning-based classification," *Geoscientific Model Development*, 2023.
[83] X. Zhang, Y. Zhou, Z. Lin, and Y. Wang, "Ensemble learning with dynamic weighting for response modeling in direct marketing," *Electronic Commerce Research and Applications*, 2024.
[84] A. Catto, N. Jia, A. Salleb-Aouissi, and A. Raja, "M-DEW: Extending Dynamic Ensemble Weighting to Handle Missing Values," 2024.