Label Error Detection in Defect Classification using Area Under the Margin (AUM) Ranking on Tabular Data

Research Paper

Pavlos Rath-Manakidis[1], Kathrin Nauth[2], Henry Huick[1], Miriam Fee Unger[1],
Felix Hoenig[3], Jens Poeppelbuss[2], and Laurenz Wiskott[1]

[1] Institute for Neural Computation, Faculty of Computer Science, Ruhr University Bochum,
Bochum, Germany
{pavlos.rath-manakidis, henry.huick, miriam.unger, laurenz.wiskott}@ruhr-uni-bochum.de
[2] Chair for Industrial Sales and Service Engineering, Ruhr University Bochum,
Bochum, Germany
{kathrin.nauth, jens.poeppelbuss}@isse.rub.de
[3] IMS Messsysteme GmbH, Heiligenhaus, Germany
felix.hoenig@ims-gmbh.de

Abstract. Vision-based automated surface inspection systems (ASIS) in flat steel production identify and classify surface defects to assess quality. Machine learning is used for defect classification, requiring high-quality training data with accurate labels. However, label errors often arise due to annotator mistakes, insufficient domain knowledge, or inconsistent class definitions. We propose a simple and effective method to detect label errors in tabular data using the area under the margin score and gradient-boosted decision tree classifiers. Our approach detects label errors with a single model training run, enabling efficient screening to improve data quality. Validated on multiple datasets, including real-world flat steel defect datasets, our method effectively identifies synthetic and real-world label errors. We demonstrate how to integrate our method into data quality control workflows, improving classification performance and enhancing the reliability of defect detection in industrial applications.

Keywords: Label Error Detection, Automated Surface Inspection System (ASIS), Machine Learning, Gradient Boosting, Data-centric AI.

1 Introduction

Flat steel production involves multiple stages and machines, with quality control ideally taking place after each stage of the production process (Agarwal and Shivpuri, 2014; Neogi et al., 2014). Common surface defects include scratches, cracks, non-metallic inclusions, unevenness, and soiling (Wu et al., 2007). In order to efficiently detect those defects, automated surface inspection systems (ASIS) are deployed as part of the quality control workflows within flat steel production (Neogi et al., 2014; Verein Deutscher Ingenieure e.V., 2023). While attempts have been made to adopt recent

20th International Conference on Wirtschaftsinformatik,
September 2025, Münster, Germany

technologies, i.e., computer vision with Convolutional Neural Networks (CNNs), to the field of ASISs (Bouguettaya and Zarzour, 2024), it is still common practice to derive tabular defect classification features from image data using custom computer vision algorithms (Verein Deutscher Ingenieure e.V., 2023).

Independent of the classifier that is used, labeled defect images are necessary for model training and validation. Selecting and labeling defects requires specialized domain knowledge about the classes of defects that occur in flat steel rolling and their possible appearance, which can be ambiguous (Nauth et al., 2024). Inconsistent labels have a considerable impact on the classifier's performance - A well-known issue in the literature on machine learning models (Vaaras et al., 2025).

Most contemporary methods for the assessment of label quality focus on computer vision settings, i.e., image data, and rely on or presume neural network models, since neural networks are the predominant model type for these tasks (Chen et al., 2019; Han et al., 2018; Pleiss et al., 2020). In contrast, research on tabular data representations is limited and the proposed methods incur a notable computational overhead (Northcutt et al., 2021a). Furthermore, many established methods focus on automatic removal or correction rather than quality-based sorting, limiting their practical benefit since expert oversight is needed to preserve correctly labeled atypical samples, while efficiently directing limited expert time toward investigating potential label errors.

This raises our research question:

How to accelerate the identification of label errors in the data used for ASIS training and validation?

To address this question, we introduce a computationally efficient method for sorting tabular data by label quality. This specific combination - computationally efficient sorting of samples by label quality to detect most likely tabular data label errors - represents a novel contribution. We apply our method to real-world data from two flat steel plants in a case study. Finally, we outline how the method can be integrated into existing classifier training and quality control processes at these plants.

2 Background

2.1 Label Errors

Training classification algorithms requires a sufficient quantity of labeled data with a diverse, yet accurate, representation of each class (Goodfellow et al., 2016). Data samples whose assigned labels do not correctly or meaningfully represent their true class are referred to as label errors, also known as label noise or mislabeled examples (Angluin and Laird, 1988; Frénay and Verleysen, 2013). These errors must be distinguished from hard-to-classify samples, which are legitimately assigned to their designated classes despite either lacking some common characteristics or exhibiting properties that overlap with those of other classes. True label errors, in contrast, reflect incorrect assignments.

Reasons for label errors are diverse. Label errors can be caused by differences in the subjective classification of defects by different annotators, ambiguous defects, or multiple defect classes occurring on the same image snippet, like a water drop near a scratch (Nauth et al., 2024; Wu and Lv, 2021). The distinction between hard-to-classify samples and true label errors is often subtle or subjective, particularly when class definitions are vague. Correct labeling for ASIS, therefore, depends on domain knowledge and consensus class definitions. Label errors in training data may be memorized by machine learning models and impairing generalization, notably when model training lacks sufficient regularization. Label errors in validation data can distort validation performance measures, hindering the reporting of performance and the optimization of model parameters (Northcutt et al., 2021b).

Unlike anomalies, label errors do not violate the distribution of the input domain. However, they distort the joint distribution between inputs and labels, interfering with learning. Detecting label errors is generally more challenging when there is a high rate of label errors in the data, as this complicates modeling the structure in the data (Chen et al., 2019; Zhang et al., 2016). Structured label noise (Chen et al., 2019), such as when certain classes are often confused, also complicates label error detection for the same reason.

2.2 Related Work on Label Error Detection

Most contemporary methods for label error detection focus on computer vision settings, i.e., image data, and depend on neural network models, since neural networks are the predominant model type for these tasks (Chen et al., 2019; Han et al., 2018; C. Northcutt et al., 2021; Pleiss et al., 2020). While the methods proposed by Northcutt et al. (2021a) and Chen et al. (2019) provide effective data-agnostic and model-agnostic approaches, they are computationally expensive due to the requirement of repeated training runs for out-of-sample prediction.

Confident Learning by Northcutt et al. (2021a) exploits patterns in the generalization behavior of classifiers to detect label errors. In this approach, generalization confidence information is utilized to estimate which labels are noisy and to use probabilistic thresholds to estimate the label noise transition matrix of the data.

In contrast, (Pleiss et al., 2020) developed the area under the margin (AUM) ranking method that traces the learning dynamics of individual samples in neural networks over the course of training through the classification logits. Their key observation is that, due to the gradual and iterative nature of neural network training, each training sample contributes only partially to the change in model predictions in each training epoch. Samples that are similar to many other samples of the same class exert a combined effect on model training, resulting in the fast and confident classification of such samples into their assigned class. In contrast, irregular samples or samples with unexpected labels are learned more slowly or not at all.

2.3 Gradient Boosting Trees and Neural Networks

ASIS defect detection typically employs data-driven AI methods. Decision trees are a widely used technology for defect classification in ASIS (Neogi et al., 2014). Decision trees are supervised machine learning algorithms that build models based on the recursive partitioning of the input data space. A popular variant of decision trees is Gradient Boosted Decision Trees (GBDTs). GBDTs build ensembles of models by adding trees sequentially: each new tree is trained to fit the residual errors (pseudo-residuals) of the current ensemble, gradually improving overall accuracy. XGBoost is a widely adopted GBDT implementation that leverages second-order gradient information and built-in regularization to reduce overfitting and accelerate training (Chen and Guestrin, 2016).

Our defect classifier is an XGBoost model. We draw a direct parallel between how GBDTs and neural networks (NNs) optimize the same loss (multiclass cross-entropy) via gradient information. In GBDTs, each tree update follows:

$$F_m(x) = F_{m-1}(x) - \alpha \nabla_{F_{m-1}(x)} L$$

where ∇L are the "pseudo-residuals" (i.e., gradients of the loss with respect to the previous models' predictions $F_{m-1}(x)$). In NNs, training proceeds by updating parameters θ in iteration t as follows:

$$\theta_t = \theta_{t-1} - \alpha \nabla_\theta L(x,y;\theta)$$

This induces changes in the network's output function. Although GBDT gradients operate in function space while NN gradients operate in parameter space, both methods rely on the same cross-entropy gradient:

$$\frac{\partial L}{\partial z_k} = \begin{cases} \hat{p}_k - 1, & \text{if } k = y \\ \hat{p}_k, & \text{if } k \neq y \end{cases}$$

where z_k is the model's output (logit) before applying the softmax function, and $\hat{p}_k$ is the confidence of the sample belonging to class k. Due to this shared gradient structure, both methods increase confidence in the true class early in training: correctly labeled samples produce large negative gradients for the correct class (pushing logits or tree outputs up), while mislabeled samples (which structurally resemble their true class) tend to be "correctly" classified at first and only shift toward the wrong label later when the model begins to memorize noise. By treating the sequence of trees in XGBoost as analogous to a NN weight update trajectory, we exploit a unified gradient-based intuition for detecting mislabeled or atypical samples. Both methods learn exemplary samples of each class early in training through general rules that apply to numerous samples, while increasingly atypical examples and edge cases are learned in later stages. Crucially, mislabeled samples obey the structure of their true class and tend to be assigned to the correct class during early training before eventually being memorized with their incorrect labels. This helps distinguish mislabeled samples from atypical but correctly labeled samples.

3 Methods

To answer the research question, we adapt AUM (Pleiss et al., 2020) for GBDT models (Section 3.1). We explain how our label quality scores enable identification of samples for manual inspection to effectively improve data quality (Section 3.2). We present the state-of-the-art (Northcutt et al., 2021a) against which we compare our approach to assess its competitiveness and discuss its efficiency (Section 3.3) and describe the computational experiment setup for evaluating our method (Section 3.4).

3.1 Algorithm

To exploit the gradient-based learning dynamics discussed in Section 2.4, we adapt the AUM score to GBDT models. With GBDT models, the method requires only a single well-generalizing trained classifier, because we can track the training process through the predictions of each sub-learner.

Definition 1 (AUM for GBDT models).
For each input-label pair in the training dat$a$ $(x, y) \in \mathscr{D}$ and every step $k \leq K$ in the training process, we consider the predicted probability of its assigned class $p_y^{(k)}(x)$ minus the probability of the most likely other class $max_{c\neq y}\, p_c^{(k)}(x)$, where $p_c^{(k)}(x)$ is the predicted probability of x belonging class $c \in \{1,\ldots, C\}$ using the first k estimators, and average over all training steps. Formally,

$$AUM(x,y) = \frac{1}{k}\sum_{k=1}^{K}\left[p_y^{(k)}(x) - \max_{c\neq y} p_c^{(k)}(x)\right].$$

That is, we iteratively add each sub-learner that constitutes the GBDT to the model and measure the probability margin for the sample's assigned class. This way, we measure how the model's confidence in the assigned label evolves during training and how misleading or atypical each sample is for the model at each step.

3.2 Ranking Label Quality

Detecting data points that are most likely mislabeled enables efficient investment of valuable expert time on improving annotation quality. Therefore, a score indicating the relative confidence that a sample's label is wrong is sufficient for most data quality assurance workflows (compare with Northcutt et al. (2021a), where ranking label error likelihood proves crucial). This avoids the need to partition between label errors and correctly labeled samples. Even in approaches where a subset of samples is denoted as potential label errors, in practice, only a fraction of these samples can be manually inspected. This, in turn, requires some measure of *relative* confidence that the labels are corrupt. A common approach, that parallels ours, is to order identified label errors by the so-called Normalized Margin $(p_y(x) - max_{c\neq y}\, p_c(x))$ (Bartlett et al., 2017) computed using out-of-sample probabilities on trained models. If a threshold is required to separate label errors from correctly labeled data, the approach from Pleiss et al., (2020), which uses a set of threshold samples assigned to an additional class to simulate label errors, can be adapted.

3.3 Comparison with Out-Of-Sample Generalization-Based Methods

We compare our method against out-of-sample (OOS) prediction-based methods informed by Confident Learning (Northcutt et al., 2021a) for scoring label quality. To our knowledge, these are the most effective methods in the literature for sorting tabular data by label quality.

We compare AUM against Self-Confidence ($p_y(x)$) and Normalized Margins ($p_y(x) - max_{c\neq y}\, p_c(x)$) (Bartlett et al., 2017) using OOS predictions to score and rank label quality. These methods are robust yet computationally expensive, relying on cross-validation for OOS classification probabilities. Following common practice, we use fivefold class-stratified cross-validation to obtain the OOS classification probabilities. These methods were selected for their reliable performance on the datasets we consider and because Normalized Margins is the method applied to sort label errors by Northcutt et al. (2021a). As an additional baseline, we compare AUM to Confident Learning Method 4: Prune by Noise Rate (CL 4) (Northcutt et al., 2021a).

3.4 Computational Experiment Setup

We conduct experiments on common publicly available datasets for tabular data (Ashwin Srinivasan, 1993; Bator, 2013; D. Campos, 2000; E. Alpaydin, 1998; Jorge Reyes-Ortiz, 2013; Nidula Elgiriyewithana, n.d.; Slate, 1991; UCI Machine Learning Repository, 1981) and one synthetic tabular dataset where each class represents a spiral in two-dimensional space (Malinin et al., 2020). Additionally, we consider two industry datasets with tabular steel strip defect detection data labelled “Industry Dataset A” and “Industry Dataset B”. The input features of the industry datasets are computed by the ASIS provider using computer vision algorithms based on the defect images and thus do not contain plain image data.

Table 1. Datasets, corresponding model configurations, and fivefold cross-validation performance. Configurations and performance on industry data are redacted.

Name	#Samples	#Features	#Classes	#Estimators	Max. Tree Depth	Val. Acc (%)
Cardiotocography	2126	21	3	30	3	94.92
Credit Card Fraud	284807	30	2	50	5	99.96
Digits	1797	64	10	50	5	96.49
Human Activity	10299	562	6	100	5	99.23
Letters	20000	16	26	100	5	96.11
Satellite	6435	36	6	50	5	91.67
Sensorless Drive	58509	48	11	50	5	99.84
Spirals	1500	11	3	50	5	98.93
Mushrooms	8124	117	2	50	3	100
Industry Dataset A	24525	399	46	-	-	-
Industry Dataset B	56047	425	25	-	-	-

The GBDT models were trained using XGBoost (Chen and Guestrin, 2016). The number of estimators (i.e., sub-learners) and the tree depth were chosen to maximize cross-validation performance. All other parameters use XGBoost version 2.0.3 defaults. For the industry datasets, proprietary parameters optimized for the same objective were used. Table 1 summarizes the datasets, model configurations, and cross-validation accuracies of the classifier configuration used for label error detection on these datasets without added label noise.

To simulate diverse label noise conditions of varying strength (5%, 10%, and 20% label flipping probability per sample), we add either uniform or asymmetric synthetic label noise to the data. Asymmetric noise reassigns samples from each class to a specific target class with fixed probability, simulating structured confusion between particular classes. Uniform noise randomly reassigns labels to any other class with equal probability across all classes. Asymmetric noise represents a challenging scenario that introduces systematic bias, while uniform noise provides a baseline without structured corruption (Chen et al., 2019).

We evaluate AUM and the other label error detectors on all samples, irrespective of whether they were originally part of the training or validation data of the underlying classification task. We compare the methods using the area under the receiver operating characteristic curve (AUROC). Accuracy is measured through fivefold class-stratified cross-validation computed on the original validation labels.

We perform 10 trials for each combination of dataset, label noise type, and label noise rate. For each trial we resample the class-stratified folds used in cross-validation, the label noise, including the noise transition matrix in the case of asymmetric noise, and the seed for XGBoost. To compare label error detectors, we compare AUROC across all trials. Qualitatively, we report the frequency with which one detector outperforms another in terms of AUROC across datasets, label noise settings, and trials.

3.5 Ablation Experiments

We consider the extent to which prediction confidence is important to the performance of our method by comparing it to Ablation 1, which, at each training step, only counts whether the model correctly predicts the sample’s label. To assess the importance of using the probability margin at each training step, we compare to Ablation 2, which averages only the model’s probabilities for the assigned class without subtracting the probability for the most likely other class.

3.6 Evaluation of Practical Utility

To evaluate practical utility, we deployed our method with two customers of the ASIS supplier. We conducted semi-structured interviews with the employee responsible for classifier training at one customer firm and the ASIS supplier employee who introduced the label error scores at both test sites. The second customer's employee declined participation, but the supplier employee reported their impressions of this user's experience. The interviews focused on several key aspects, including the process of identifying labeling errors prior to the introduction of the scores, how the scores were introduced (e.g., the type of explanation provided), their visual integration into the software, workflow changes resulting from their use, and potential areas for improvement. To better understand the workflow before and after the introduction of the scores, we asked the employee responsible for classifier training to share his screen and demonstrate how he uses the scores to detect labeling errors in the dataset - an approach aligned with the established Think-Aloud Method (Charters, 2003). This allowed us to verify whether the method was being used as intended and to assess its effectiveness in reducing the time required for label error correction. Both interviews were conducted via video conferencing, recorded, transcribed, and systematically analyzed to generate insights for the optimal integration of the scores into existing processes.

4 Results

We demonstrate that our proposed method is highly competitive while being more efficient than existing solutions for label error detection (Section 4.1). Based on these results, we demonstrate how our method can enhance GBDT performance by simply removing samples by label quality score (Section 4.2). Lastly, we demonstrate how our method facilitates effective data inspection to enhance data quality and can be successfully integrated into existing ASIS quality control workflows (Section 4.3).

4.1 Competitiveness

Comparing AUM to out-of-sample (OOS) generalization-based methods in terms of AUROC across all datasets and label noise conditions shows that AUM equals or outperforms Normalized Margins in 53.0% of trials and equals or outperforms Self-Confidence in 42.0% of trials, while requiring only one training run. The Confident Learning Method 4 (CL 4) approach, which first detects and then ranks label errors, produces worse rankings than AUM in 99.7% of all trials. Figure 1 visualizes the per-trial comparison between AUM and these OOS generalization-based methods.

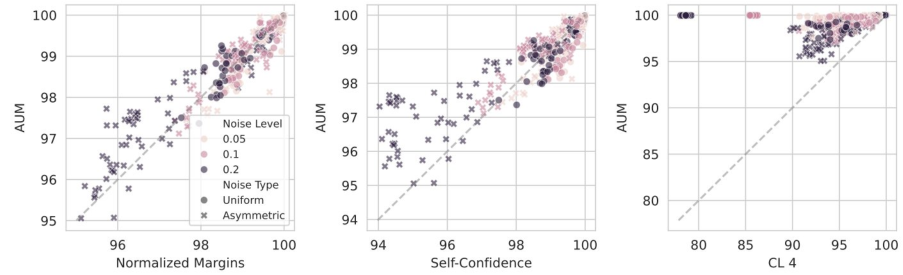

Figure 1. Per-trial AUROC (%) comparison between AUM and OOS generalization-based methods, Normalized Margins, Self-Confidence, and CL 4, across all label noise conditions. We show the trials from all datasets.

Ablation studies reveal that AUM equals or outperforms Ablation 1 in 96.1% of trials and equals or outperforms Ablation 2 in 72.0% of trials, while Ablation 2 equals or outperforms Ablation 1 in 86.2% of trials. Figure 2 visualizes the per-trial comparison between AUM and both ablations. Table 2 presents performance comparisons among all methods under 5% asymmetric label noise for each dataset.

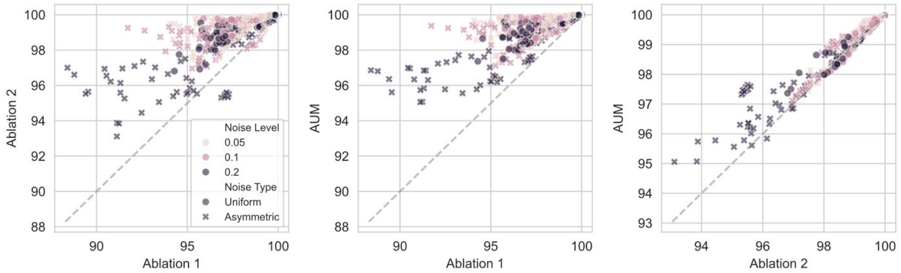

Figure 2. Per-trial AUROC (%) comparison between AUM, Ablation 1, and Ablation 2 across all datasets and label noise conditions. We show the trials from all datasets.

Table 2. Performance comparison of AUM with the OOS prediction-based methods and the two ablations under 5% asymmetric label noise in terms of AUROC (%). Larger values are better. Best values are shown in bold. We report the mean and the standard deviation across 10 trials.

Dataset	Ablation 1	Ablation 2	AUM	Normalized Margins	Self-Confidence	CL 4
Cardiotocography	96.8 ± 1.7	**98.9 ± 0.6**	**98.9 ± 0.6**	**98.9 ± 0.5**	98.8 ± 0.5	93.7 ± 1.4
Credit Card Fraud	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	94.9 ± 0.3
Digits	95.6 ± 1.1	99.5 ± 0.2	**99.6 ± 0.2**	**99.6 ± 0.2**	99.5 ± 0.2	96.2 ± 1.8
Human Activity	99.0 ± 0.4	99.8 ± 0.0	99.8 ± 0.0	**100.0 ± 0.0**	**100.0 ± 0.0**	98.6 ± 0.4
Letters	99.2 ± 0.1	99.1 ± 0.1	99.4 ± 0.1	**99.5 ± 0.0**	99.3 ± 0.1	97.1 ± 0.4
Mushrooms	99.9 ± 0.1	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	99.2 ± 0.7
Satellite	96.3 ± 0.8	98.0 ± 0.3	98.0 ± 0.3	**98.4 ± 0.2**	**98.4 ± 0.3**	94.8 ±1.0
Sensorless Drive	99.9 ± 0.1	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	99.0 ± 0.3
Spirals	97.5 ± 1.6	99.3 ± 0.7	99.6 ± 0.5	**99.7 ± 0.2**	**99.7 ± 0.3**	97.1 ± 1.7
Industry Dataset A	96.5 ± 0.3	96.4 ± 0.3	97.1 ± 0.2	95.9 ± 0.2	**97.8 ± 0.3**	92.7 ± 0.7
Industry Dataset B	96.4 ± 0.5	96.2 ± 0.2	96.7 ± 0.2	96.8 ± 0.2	**97.8 ± 0.3**	93.7 ±0.5

Overall, AUM performs similarly to the more computationally expensive OOS prediction-based techniques for ranking label quality across datasets and label error conditions. Comparing Ablation 1 to Ablation 2 demonstrates that incorporating classifier prediction confidence into the label quality score contributes to our method's effectiveness. The comparison of Ablation 2 with AUM further shows that considering the classifier's confidence for the most likely class not assigned to a sample also contributes to performance.

4.2 Improving Accuracy through Naive Data Cleaning

Figures 3 and 4 illustrate an exemplary process that utilizes AUM scores to enhance models by removing the lowest-scoring samples from training. On industry datasets, our method improves model performance to a similar extent as the OOS prediction-based method, demonstrating that AUM can improve model performance for tabular data under noisy labels even without manual label inspection and correction. Notably, without synthetic noise, AUM generally outperforms Self-Confidence on both industry datasets.

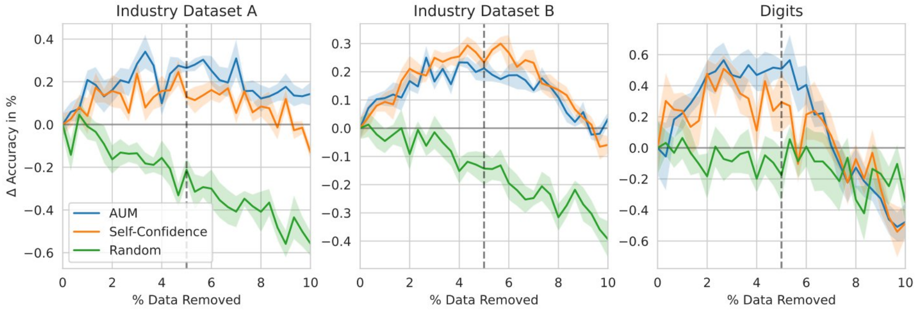

Figure 3. Effect of removing mislabeled samples on model performance with added synthetic label noise. Mislabeled samples are selected either with AUM, Self-Confidence, or at random. We report the change in cross-validation accuracy, compared to retaining all samples. Trials were performed with 5% asymmetric label noise. We report the mean and standard error of the mean over 7 trials. The dashed line corresponds to 5% removed samples.

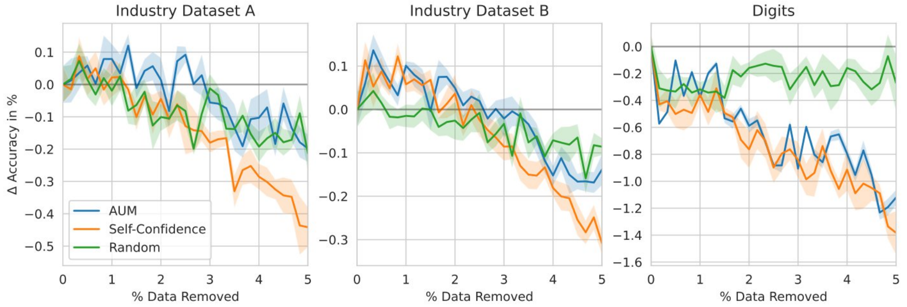

Figure 4. Effect of removing mislabeled samples on model performance without synthetic label noise. Mislabeled samples are selected either with AUM, Self-Confidence, or at random. We report the change in cross-validation accuracy, compared to retaining all samples. We report the mean and standard error of the mean over 7 trials.

4.3 Real-World Application and Process Integration

A preliminary application of AUM for GBDT models (alongside Self-Confidence scores) was conducted on flat steel defect data, which included real-world label errors and data quality issues, provided by two of the ASIS supplier's customers. An expert of the ASIS supplier and the employee responsible for the classifier training at the test customer plants, who can be described as domain expert, examined the samples with the lowest AUM scores. The results confirmed that high scores indicate good training samples in most cases, while low scores identify mislabeled or unusual samples. Out of the 57 samples examined, 24 (42%) were identified as label errors. Additionally, four samples were flagged that were not label errors, but were highly unusual data points that revealed a rare error in the data pipeline.

The method had been integrated into the ASIS software and the classifier training process, accelerating the quality control workflow. ASIS users confirmed that introducing AUM and Self-Confidence scores into the ASIS interface reduced the time spent identifying and correcting low-quality samples and thereby raising the quality of the training and validation data. Previously, the ASIS did not provide workflow assistance for identifying misclassified defects. Users had to check numerous samples randomly or based on unsuitable metrics to identify label errors. With the new scores, significantly fewer labeled samples need to be inspected manually, as low scores from either method reliably indicate label errors. Manual review can result in label correction, sample removal, or the addition of more samples for underrepresented classes. AUM and Self-Confidence may also flag samples as noisy when they belong to underrepresented classes, are outliers with correct labels, or are corrupted by data processing errors. The process can iterate, improving data based on label quality scores and retraining the model, which improves model and label error detection accuracy through progressively better data modeling.

Overall, our method was evaluated as suitable for quickly and effectively identifying problems in the training and validation data:

"I found defects that had just a partial defect or the segmentation is not right, and the [AUM] and [Self-Confidence scores] will call that out to me. And I'm like, yeah, that's right, man, that spot shouldn't even be in my classifier. So I'll throw that out of my classifier." ASIS user

5 Discussion and Conclusion

In this study, we adapted the AUM ranking from Pleiss et al. (2020) to tabular data, particularly real-world industrial flat steel defect data. We demonstrate that the adapted method performs well across diverse datasets while being computationally more efficient than comparable methods and integrates effectively into data quality assurance workflows.

Unlike most research in this field, we evaluate label error detectors primarily on label error detection performance rather than downstream classifier performance. This distinction is critical, as samples flagged as potential label errors, whether truly mislabeled or merely hard-to-classify, may significantly impact model generalization ability. Automatically removing or relabeling these samples risks discarding valuable edge cases. Therefore, using final model performance as a proxy for label error detection efficiency can be misleading. Rather than assuming a fixed methodology for addressing label errors, our approach aims to empower domain experts in quality assurance. By prioritizing the most suspicious samples, our method helps experts use their limited time more efficiently.

Implications for practice - On a practical level, we contribute to improved human-in-the-loop workflows by integrating AUM and Self-Confidence scores into the ASIS quality control processes. This supports the recommendations of Nauth et al. (2024), who advocate for quality control systems that can flag problematic training data. Label quality scores like ours can serve as automated early-warning signals within such systems.

Implications for theory -On a theoretical level, we extend the applicability of AUM-based analysis - originally developed for neural networks - to GBDTs. We demonstrate that monitoring the learning dynamics of individual samples using AUM can effectively highlight anomalous or mislabeled data, even with a GBDT model. This broadens the scope of label error detection techniques to include widely used tabular data models and industrial applications.

Limitations - While our empirical results demonstrate AUM's effectiveness across diverse tabular datasets and noise conditions, future theoretical analysis could further strengthen our understanding of its reliability on GBDT models**.** Despite being faster than multiple-run OOS methods, AUM still requires a single additional training run. Low label quality scores are ambiguous - they may identify corrupted data or samples that represent multiple classes. Additionally, improving data quality using AUM scores remains a manual process that requires expert oversight, as the automatic removal of low-scoring samples can eliminate valuable edge cases and reduce model performance, as shown in Figure 4.

Future research - Future work should focus on integrating label error detection, including AUM, into broader data quality management systems that alert on label quality issues. This could include proposing label corrections based on AUM and identifying classes that are frequently confused during labeling, thereby identifying problems in class ontologies, as in Northcutt et al. (2021a). Further, AUM, Self-Confidence, and potentially other scores could be integrated into a single score, as suggested by an ASIS user responsible for classifier training in an interview: *"So that would make sense to be able to combine all three [scores used for label error detection] into saying this is the true defect here."*

Overall, this research demonstrates that integrating AUM and Self-Confidence scores into classifier training workflows on tabular data is a valid data-centric machine learning practice that optimizes operational efficiency and supports high-quality decision-making, representing a significant advance in data-driven quality control.

6 Acknowledgments

This research and development project is funded by the German Federal Ministry of Research, Technology and Space (BMFTR) within the “The Future of Value Creation-Research on Production, Services and Work” program (02L19C200) and managed by the Project Management Agency Karlsruhe (PTKA). The authors are responsible for the content of this publication.

References

Agarwal, K., Shivpuri, R., 2014. Knowledge discovery in steel bar rolling mills using scheduling data and automated inspection. J Intell Manuf 25, 1289–1299. https://doi.org/10.1007/s10845-013-0730-5

Angluin, D., Laird, P., 1988. Learning from noisy examples. Machine learning 2, 343–370.

Ashwin Srinivasan, 1993. Statlog (Landsat Satellite). https://doi.org/10.24432/C55887

Bartlett, P., Foster, D.J., Telgarsky, M., 2017. Spectrally-normalized margin bounds for neural networks. https://doi.org/10.48550/arXiv.1706.08498

Bator, M., 2013. Dataset for Sensorless Drive Diagnosis. https://doi.org/10.24432/C5VP5F

Bouguettaya, A., Zarzour, H., 2024. CNN-based hot-rolled steel strip surface defects classification: a comparative study between different pre-trained CNN models. Int J Adv Manuf Technol. https://doi.org/10.1007/s00170-024-13341-0

Charters, E., 2003. The use of think-aloud methods in qualitative research an introduction to think-aloud methods. Brock Education Journal 12.

Chen, P., Liao, B., Chen, G., Zhang, S., 2019. Understanding and Utilizing Deep Neural Networks Trained with Noisy Labels. https://doi.org/10.48550/ARXIV.1905.05040

Chen, T., Guestrin, C., 2016. XGBoost: A Scalable Tree Boosting System, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Presented at the KDD '16: The 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, San Francisco California USA, pp. 785–794. https://doi.org/10.1145/2939672.2939785

D. Campos, J.B., 2000. Cardiotocography. https://doi.org/10.24432/C51S4N

E. Alpaydin, C.K., 1998. Optical Recognition of Handwritten Digits. https://doi.org/10.24432/C50P49

Frénay, B., Verleysen, M., 2013. Classification in the presence of label noise: a survey. IEEE transactions on neural networks and learning systems 25, 845–869.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Convolutional networks, in: Deep Learning 2016. pp. 330–372.

Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., Sugiyama, M., 2018. Co-teaching: Robust Training of Deep Neural Networks with Extremely Noisy Labels. https://doi.org/10.48550/ARXIV.1804.06872

Jorge Reyes-Ortiz, D.A., 2013. Human Activity Recognition Using Smartphones. https://doi.org/10.24432/C54S4K

Malinin, A., Prokhorenkova, L., Ustimenko, A., 2020. Uncertainty in Gradient Boosting via Ensembles. https://doi.org/10.48550/ARXIV.2006.10562

Nauth, K., Uenal, E., Meske, C., Poeppelbuss, J., 2024. How to design the interplay between humans and AI-based surface inspection systems. Presented at the

18th CIRP Conference on Intelligent Computation in Manufacturing Engineering, Gulf of Naples, Italy.(Not yet published).

Neogi, N., Mohanta, D.K., Dutta, P.K., 2014. Review of vision-based steel surface inspection systems. J Image Video Proc 2014, 50. https://doi.org/10.1186/1687-5281-2014-50

Nidula Elgiriyewithana, n.d. Credit Card Fraud Detection Dataset 2023. https://doi.org/10.34740/KAGGLE/DSV/6492730

Northcutt, C., Jiang, L., Chuang, I., 2021a. Confident Learning: Estimating Uncertainty in Dataset Labels. jair 70, 1373–1411. https://doi.org/10.1613/jair.1.12125

Northcutt, C., Athalye, A., Mueller, J., 2021b. Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks. https://doi.org/10.48550/ARXIV.2103.14749

Pleiss, G., Zhang, T., Elenberg, E.R., Weinberger, K.Q., 2020. Identifying Mislabeled Data using the Area Under the Margin Ranking. https://doi.org/10.48550/ARXIV.2001.10528

Slate, D., 1991. Letter Recognition. https://doi.org/10.24432/C5ZP40

UCI Machine Learning Repository, 1981. Mushroom. https://doi.org/10.24432/C5959T

Vaaras, E., Airaksinen, M., Räsänen, O., 2025. IAR 2.0: An Algorithm for Refining Inconsistent Annotations for Time-Series Data Using Discriminative Classifiers. IEEE Access 13, 19979–19995. https://doi.org/10.1109/ACCESS.2025.3534637

Verein Deutscher Ingenieure e.V., 2023. VDI/VDE/VDMA 2632 Blatt 4.2.

Wu, G., Zhang, H., Sun, X., Xu, J., Xu, K., 2007. A Bran-new Feature Extraction Method and its application to Surface Defect Recognition of Hot Rolled Strips, in: 2007 IEEE International Conference on Automation and Logistics. Presented at the 2007 IEEE International Conference on Automation and Logistics, IEEE, Jinan, China, pp. 2069–2074. https://doi.org/10.1109/ICAL.2007.4338916

Wu, H., Lv, Q., 2021. Hot-Rolled Steel Strip Surface Inspection Based on Transfer Learning Model. Journal of Sensors 2021, 1–8. https://doi.org/10.1155/2021/6637252

Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O., 2016. Understanding deep learning requires rethinking generalization. https://doi.org/10.48550/ARXIV.1611.03530