BMC Medical Informatics and Decision Making

RESEARCH

Open Access

Addressing label noise for electronic health records: insights from computer vision for tabular data

Jenny Yang[1*], Hagen Triendl[2], Andrew A. S. Soltan[1,3,4], Mangal Prakash[5†] and David A. Clifton[1,6†]

Abstract

The analysis of extensive electronic health records (EHR) datasets often calls for automated solutions, with machine learning (ML) techniques, including deep learning (DL), taking a lead role. One common task involves categorizing EHR data into predefined groups. However, the vulnerability of EHRs to noise and errors stemming from data collection processes, as well as potential human labeling errors, poses a significant risk. This risk is particularly prominent during the training of DL models, where the possibility of overfitting to noisy labels can have serious repercussions in healthcare. Despite the well-documented existence of label noise in EHR data, few studies have tackled this challenge within the EHR domain. Our work addresses this gap by adapting computer vision (CV) algorithms to mitigate the impact of label noise in DL models trained on EHR data. Notably, it remains uncertain whether CV methods, when applied to the EHR domain, will prove effective, given the substantial divergence between the two domains. We present empirical evidence demonstrating that these methods, whether used individually or in combination, can substantially enhance model performance when applied to EHR data, especially in the presence of noisy/incorrect labels. We validate our methods and underscore their practical utility in real-world EHR data, specifically in the context of COVID-19 diagnosis. Our study highlights the effectiveness of CV methods in the EHR domain, making a valuable contribution to the advancement of healthcare analytics and research.

Keywords Machine learning, Electronic health records, Noisy labels

Introduction

In recent years, there has been a substantial surge in digital data within the healthcare sector. Notably, electronic health records (EHRs), which encompass patient health information such as medical history, diagnoses, medications, and lab test results, have played a pivotal role in enhancing patient safety, streamlining care coordination, and improving efficiency. The widespread adoption of EHRs has translated into a substantial increase in the volume of digital data, providing a robust foundation for machine learning (ML) applications in healthcare. Leveraging this wealth of data, ML models can enhance precision, develop personalized treatment strategies, and enable predictive analytics, ultimately resulting in improved patient outcomes and more efficient healthcare delivery.

[†] Mangal Prakash and David A. Clifton shared last authors.

*Correspondence:
Jenny Yang
jenny.yang@eng.ox.ac.uk
[1] Institute of Biomedical Engineering, Dept. Engineering Science, University of Oxford, Oxford, England
[2] Exscientia, Oxford, England
[3] Oxford Cancer & Haematology Centre, Oxford University Hospitals NHS Foundation Trust, Oxford, England
[4] Department of Oncology, University of Oxford, Oxford, England
[5] Work done at Exscientia, Currently Independent Researcher, Reading, United Kingdom
[6] Oxford-Suzhou Centre for Advanced Research (OSCAR), Suzhou, China

However, while EHRs serve as a valuable data source for ML tasks, their utilization can pose challenges due to potential noise and errors. These issues can arise from various common sources, including data entry errors [1–4], incomplete information [1, 2, 4], inconsistencies [2, 5, 6], system errors [7–9], and diagnostic test errors [10–13].

Previous studies have highlighted the need for greater effort to improve the accuracy and completeness of EHR data [10, 14–17]. For example, a survey study published in 2020 asked a total of 136,815 patients at three US healthcare organizations to read their EHR notes and identify any errors [10]. Of the 29,656 patients who provided a response, 1 in 5 reported a mistake, with 40% of these mistakes being perceived by the patient as serious. Among patient-reported serious errors, the most common mistakes included those related to diagnoses, medical history, medications, and test results. Another record-review study published in 2018 investigated the errors and causes of failure in the communication of patients' information between different hospital information/EHR systems [15]. Through the review of 882 hospital records, the study identified 1,256 errors of 41 different types. These errors were classified into system level errors (65%) and operator-dependent errors (35%), and further stratified into four categories: administrative-financial errors (61%), errors related to national codes (23%), clinical errors (9%), and other errors (7%). The presence of errors in EHR data can have serious consequences for patient care and outcomes, as well as for research and analysis that relies on this data. Therefore, ensuring the accuracy and completeness of EHR data is an ongoing challenge for healthcare providers and researchers.

Despite the substantial evidence pointing to the prevalence of noisy and erroneous EHR data, the existing ML models documented in the literature for EHR data analysis have yet to address this concern. These models typically operate under the assumption that the data and labels are free from unwanted noise and corruption, a premise that does not accurately reflect real-world datasets. Additionally, it is widely acknowledged that ML models (especially deep learning [DL] models), are susceptible to overfitting to noisy labels [18–20]. This susceptibility can lead to significant consequences, such as reduced generalization performance on unseen patient EHR records, unreliable predictions, the perpetuation of undesired biases in predictions, and potentially serious repercussions for patient care. Consequently, this could erode trust among healthcare professionals regarding the utilization of ML/DL models. Therefore, it is crucial to acknowledge the inherent imperfect nature of EHR data and devise mechanisms for training ML/DL methods to effectively handle noisy data.

Expanding upon previous research, we acknowledge the presence of noisy data and labels during the training of DL models and aim to tackle this concern within the context of EHR data. Thus, our study is centered on EHR classification, with particular emphasis on situations where only the class labels are affected by noise or errors. Taking inspiration from recent methodologies in computer vision (CV) that account for noisy labels when training DL models [21–23], we undertake an exploration of their suitability for the EHR domain. It is essential to highlight that our tabular EHR data is distinctively different from image data for which these CV methods are originally proposed for. In contrast to images, where pixel values represent visual features, tabular EHR data encompasses a diverse array of patient records, diagnostic codes, timestamps, and various clinical parameters. Additionally, the amount of high quality data available in CV (with benchmarks like ImageNet [24] for training DL models) is typically many magnitudes larger than that available for EHRs. The differences between these two domains calls for ensuring that these methods function effectively and provide meaningful insights when applied to EHR.

In this study, we aim to bridge this gap. With relatively little adaptations tailored to the unique characteristics of EHR data, we found that some of the recently proposed methods in CV domain can substantially mitigate the risks associated with overfitting to noisy labels in EHR data. This finding highlights the adaptability and potential of these techniques, even in the presence of substantial differences in data structure and content. Moreover, our research goes beyond the individual application of these methods. We investigate the synergy of combining multiple approaches, and our results demonstrate that the integration of these techniques not only effectively addresses the issue of noisy labels, but also surpasses the performance of each method independently in many cases. Our findings emphasize the potential to transform how we handle and analyze EHR data, offering new avenues for improved healthcare outcomes and research in the EHR domain.

Related works

Various techniques have been employed in different domains to address the challenges posed by noisy labels for ML tasks. In general, these can roughly be divided into two groups, 1) label correction, which focuses specifically on identifying and rectifying mislabeled data points to improve the quality of the training data, and 2) regularization, which penalises over-confident predictions to prevent overfitting and indirectly reduce the impact of noisy labels.

With respect to label correction, one approach is data cleaning [25], which involves removing data points that are clearly incorrect or inconsistent. This can be achieved through manual inspection or by clustering [26, 27] and outlier detection algorithms [28–30]. However, manual inspection is expensive and time-consuming [25]; and removing samples wastes valuable information that could still provide useful information for training [31]. Conversely, algorithm-driven approaches, including self-training [32–34] and co-training [35, 36], iteratively update and improve labels based on a model's predictions. These methods can be effective in improving the quality of labeled data; however, they typically rely on the initial labeled data. Thus, if the initial labels are noisy or biased, these can be propagated during training, potentially exacerbating the problem. Similarly, these techniques can lead to overconfident predictions on noisy data, especially if the model is uncertain about the true labels, resulting in the inclusion of incorrect labels in the training set. Additionally, these techniques may not be well-suited for very small datasets where the benefits of leveraging unlabeled data might be limited.

Given that these methods rely on the assumption of having access to a limited, dependable set of clean samples, our research will focus on situations where this assumption does not hold, which is frequently encountered in clinical data scenarios. Consequently, we will focus our investigations on regularization methods. These include robust loss functions, label smoothing, taking a convex combination of samples/labels, and using consistency as a metric for evaluation.

Robust loss functions can be utilized to minimize the impact of outliers and noisy labels in the loss function [37, 38]. While employing these outlier-robust loss functions offer advantages, they also exhibit drawbacks, including the potential loss of crucial information, especially notable in real healthcare data, where outliers can yield valuable insights. The loss of outlier information can also introduce bias into a model's predictions, potentially skewing them away from the true underlying data distribution. Furthermore, it's important to acknowledge that within healthcare data, a spectrum of disease severity exists. Consequently, even when incorrect labels are present, the individual data samples themselves may not necessarily qualify as outliers.

Label smoothing [22, 39, 40] is often used in cases where data is imperfectly labeled or contains errors. It aims to improve the generalization and robustness of a model by preventing it from overfitting to the training data. It does this by adding small amounts of uncertainty to the target labels while the model is being trained, encouraging the model to assign lower probabilities to incorrect classes and distribute the probability mass more evenly across all classes. Label smoothing has previously been demonstrated to effectively remove noise from corrupted labels [22].

Mix-up [23] is a data augmentation technique commonly used in CV for tasks such as image classification. It was introduced as a regularization method to improve the generalization and robustness of models, especially in scenarios with limited labeled data. Combining different features and labels with one another prevents the model from becoming overconfident about the relationship between features and their labels, thereby regularizing the model.

Previous studies have also leveraged the concept of generating consistent outputs as a means to constrain training. One such method is bootstrapping [41], whereby the standard prediction objective is enhanced with a term for perceptual consistency. Here, a prediction is defined as consistent when a network produces the same prediction when presented with similar features. Another method is Neighbour Consistency Regularization (NCR) [21]. Similarly, NCR enforces consistency among the predictions of a model on neighboring/similar samples. During training, the model is trained using both a standard supervised loss (such as cross-entropy) and an additional regularization term that encourages the predictions of neighboring samples to be consistent. By doing so, a model learns to be robust to label noise, as the regularization term penalizes inconsistencies in the predictions caused by noisy labels.

Although not specific to addressing noisy labels, ensemble methods offer another possible avenue for mitigating the effects of noisy labels. By training multiple models on different subsets of the data and combining their predictions, ensemble methods can help reduce the influence of individual noisy labels.

In the domain of EHR data analysis, limited studies to date address the problem of label noise. Very recently, Tjandra et al. [42] tackled the problem of instance dependent label noise in EHR data where the authors assume the availability of a small subset of clean data and labels in addition to a larger noisy dataset for learning a two-stage discriminative model. Boughorbel et al. [43] present an alternating loss-correction approach for training models with longitudinal EHR data with noisy labels. This method also requires the availability of separate clean and noisy datasets and alternates the training between these two sets. Unlike these approaches, we do not assume any prior knowledge about which datapoints are labeled correctly or incorrectly, and hence we do not require separate clean and noisy datasets. We show that even for EHR datasets, methods like NCR [21], Mix-up [23] and Label smoothing [22, 39, 40] (which do not need any additional information from a separate clean dataset) can be effectively applied with minor adaptations, thereby offering new avenues for improved EHR data analysis.

Data and methods

An overview of the methods employed is shown in Fig. 1. We focus specifically on a real-world COVID-19 case study where the task is to use patients' EHRs to classify them as being either COVID positive or negative. With government regulation for mandatory testing at the time, there is vast, clinically-rich EHR data available, alongside positive and negative COVID-19 presentations (determined through PCR tests, the gold standard test for diagnosing viral genome targets). Additionally, due to incomplete penetrance of PCR testing during the early stages of the pandemic and imperfect sensitivity, there is uncertainty in the viral status of patients who tested negative, thus making this case study uniquely suitable to test model development in the presence of noisy/incorrect labels.

Datasets and pre-processing

For the purpose of this study, we use the CURIAL datasets [44, 45], which consist of anonymized EHR data for patients presenting to emergency departments (EDs) across four independent United Kingdom (UK) National Health Service (NHS) Trusts. These include Oxford University Hospitals NHS Foundation Trust (OUH), University Hospitals Birmingham NHS Trust (UHB), Portsmouth Hospitals University NHS Trust (PUH), and Bedfordshire Hospitals NHS Foundations Trust (BH). United Kingdom NHS approval via the national oversight/regulatory body, the Health Research Authority (HRA), has been granted for development and validation of artificial intelligence models to detect COVID-19 (CURIAL; NHS HRA IRAS ID: 281832).

Previous studies have shown that ML classification models trained on EHR features could diagnose patients presenting with COVID-19 up to 26% sooner than lateral flow device (LFD) testing and 90% sooner than PCR testing [44] on average, while simultaneously achieving high sensitivities and performing effectively as a rapid test-of-exclusion [31, 44–48]. Similarly, we trained models for the purpose of rapid triage using laboratory blood tests and vital signs, as these are routinely collected during the first hour of patients attending emergency care in hospitals in middle- to high-income countries [45]. The feature sets included are the same as those used in [31, 45–48]. Supplementary Section C summarizes the final features used.

For model development, a training set was used for model training; a validation set was used for continuous validation and threshold adjustment; and after successful development, internal and external test sets were used to evaluate the performance of final models. For training and validation, we used patient presentations from PUH. From PUH, we obtained patient presentations to the ED between March 1, 2020 and February 28, 2021. We created training, validation, and test sets using randomly selected 60%, 20%, and 20% splits, respectively. This resulted in 22,737 (1,182 COVID-19 positive), 7,579 (439 positive), and 7,580 (385 positive) presentations for PUH training, validation, and test sets, respectively.

We additionally curated datasets from three independent hospitals - OUH, UHB, and BH. From OUH, we curated two data extracts. The first extract contains 701 COVID-19 positive cases from the "first wave" of the COVID-19 epidemic in the UK (December 1, 2019 to June 30, 2020), with 91,970 pre-pandemic controls (COVID-free patient presentations from OUH prior to the global COVID-19 outbreak). The second extract contains 22,857 presentations (2,012 positive) from the "second wave" (October 1, 2020 - March 6, 2021). There was one cohort from UHB (presentations between December 1, 2019 and October 29, 2020) and BH (presentations between January 1, 2021 and March 31, 2021), consisting of 10,293 (439 positive) and 1,177 (144 positive), respectively. These cohorts were used to externally validate performance and generalizability, emulating the real-world implementation of such a diagnostic method.

Consistent with previous studies, we addressed the presence of missing values by using population median imputation, then standardized all features in our data to have a mean of 0 and a standard deviation of 1. A summary of the inclusion and exclusion criteria for patient cohorts, summary population statistics, and all training, validation, and test cohort splits can be found in Section C of the Supplementary Material.

For experiments with synthetic noise, we randomly changed the label to the incorrect diagnosis (i.e. increasing the false-negative and false-positive levels). For COVID-19 diagnosis by PCR, sensitivities were estimated to be around 80%-90% during different times of

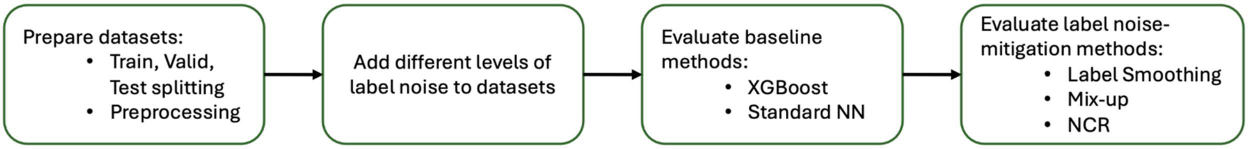

Prepare datasets:
• Train, Valid, Test splitting
• Preprocessing
Add different levels of label noise to datasets
Evaluate baseline methods:
• XGBoost
• Standard NN
Evaluate label noise-mitigation methods:
• Label Smoothing
• Mix-up
• NCR

Fig. 1 Flowchart of methods employed

the pandemic [49–51], while specificity was estimated to be around 98%-100% [49–51]. Thus, to represent more realistic label corruption, we created additional false negatives using 10%, 20%, 30%, and 40% of the COVID-19 positive presentations. Simultaneously, we kept the number of additional false positives consistent, representing 0.5% of the COVID-19 negative cases. This approach was chosen to a) mirror the estimated specificity of PCR testing for COVID-19 and b) address the significant label imbalances prevalent in the training data.

Baselines

As a baseline, we use the same general neural network architecture using cross-entropy loss shown to be successful in [31, 46–48] for the COVID-19 classification task (varying hyperparameters depending on the best results obtained during grid search). Additionally, [44, 45, 47, 48] showed that XGBoost works remarkably well for this task, and serves as another strong baseline.

XGBoost

XGBoost [52] is an ensemble method which combines the predictions of several base estimators (in this case, decision trees) in order to improve generalizability and robustness. The idea is that the weaknesses of one model can be compensated for by the strengths of another, resulting in a more robust ensemble model.

Baseline neural network

Following [31, 46–48], we trained a fully-connected neural network which used the rectified linear unit (ReLU) activation function in the hidden layers and the sigmoid activation function in the output layer. For updating model weights, the Adam optimizer was used during training. Details of the architecture are presented in Section B of the Supplementary Material. This architecture will also be referred as *Baseline NN* going forward.

CV-inspired techniques to address noisy labels

Here we briefly describe three recently proposed methods which were shown to be effective for training neural networks with noisy labels in CV tasks (Fig. 2). We investigated both their individual and combined effectiveness in mitigating the impact of label noise in EHRs when employed in conjunction with the *baseline NN* described in "Baseline neural network".

Label smoothing

Label smoothing [22, 39, 40] is a regularization technique that adds a small amount of noise to the target labels during training. Similar to [22, 40], instead of using 0 or 1 as the correct label, we use a value of $1 - \epsilon$ for the correct label and $\frac{\epsilon}{(C-1)}$ for the other labels. Here, C is the number of classes, and ϵ is sampled uniformly in [0, 1].

Mix-up

Mix-up [23] is an augmentation method that creates new examples as convex combinations of the original training samples. Given a dataset with labeled examples, Mix-up combines pairs of input samples (both the features and labels) by taking a weighted linear combination.

Therefore, for two data points (x_1, y_1) and (x_2, y_2), the mixed data point (x_{mix}, y_{mix}) is computed as follows:

$$x_{mix} = \lambda * x_1 + (1 - \lambda) * x_2 \tag{1}$$

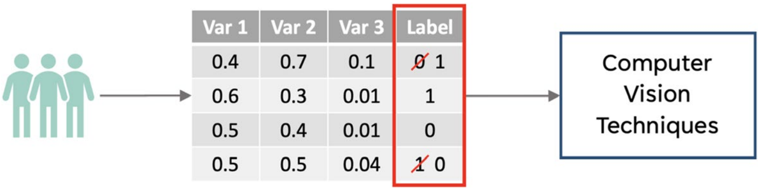
$$y_{mix} = \lambda * y_1 + (1 - \lambda) * y_2 \tag{2}$$

Here, λ denotes a mixing coefficient, which is sampled from a beta distribution, $Beta(\alpha, \alpha)$, where α is a hyperparameter controlling the shape of the distribution, and $\alpha \in (0, \infty)$ [23]. In our implementation, we also set a probability of mix-up per batch.

Neighbour consistency regularization

Neighbour Consistency Regularization (NCR) (Fig. 3) is a regularization technique which relies on enforcing the simple idea that examples from the same class will have similar latent representations and hence should be classified to the same class irrespective of their labels (which may be noisy and different) [21]. As presented in [21], we define the similarity between two examples by the cosine similarity of their feature representations:

$$s_{i,j} = \cos(v_i, v_j) = \frac{v_i^T v_j}{\|v_i\| \|v_j\|} \tag{3}$$



Var 1	Var 2	Var 3	Label
0.4	0.7	0.1	~~0~~ 1
0.6	0.3	0.01	1
0.5	0.4	0.01	0
0.5	0.5	0.04	~~1~~ 0

Fig. 2 Addressing noisy labels using computer vision-inspired techniques

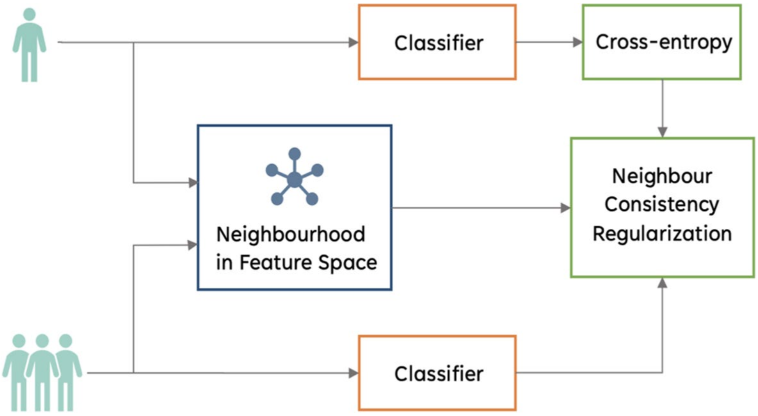

Classifier
Cross-entropy
Neighbourhood in Feature Space
Neighbour Consistency Regularization
Classifier

Fig. 3 Neighbour consistency regularization diagram

Here, the feature representations are non-negative values (obtained after a ReLU transformation) from a specific layer hidden layer. If v_i and v_j have high cosine similarity $s_{i,j}$, then a classifier *f*, is encouraged to predict the same label for $f(v_i)$ and $f(v_j)$, regardless of their labels y_i and y_j. This discourages the model from overfitting to any incorrect mapping (x, y), if either (or both) of y_i and y_j are noisy.

To enforce neighbor consistency regularization, the objective function is formulated to minimize the distance between logits $\mathbf{z_i}$ and $\mathbf{z_j}$, when their corresponding feature representations v_i and v_j are similar. Using 3, the NCR term can be written as:

$$L_{NCR} := \frac{1}{m}\sum_{i=1}^{m} D_{KL}\left(\sigma(\mathbf{z_i}) \middle\| \sum_{j\epsilon NN_k(v_i)} \frac{s_{i,j}}{\sum_k s_{i,k}} \sigma(\mathbf{z_j})\right) \tag{4}$$

In this formulation, the D_{KL} represents the Kullback-Leibler (KL) divergence loss used to measure the dissimilarity between two distributions. The term $NN_k(v_i)$ refers to the set of k nearest neighbors of v_i in the feature space. To ensure that the similarity values form a probability distribution, we normalize them. Additionally, we set the self-similarity $s_{i,i}$ to zero to avoid it dominating the normalized similarity. Gradients are propagated back to all inputs. Thus, this NCR term encourages the output of a classifier to classify x_i in a way which aligns to its latent space neighbors, regardless of the potentially noisy label y_i.

We combine this NCR with the standard supervised classification loss function, namely cross entropy, to form the final objective function that is minimized during training, i.e.,

$$L_{total} := L_{CE} + \alpha L_{NCR}, \tag{5}$$

Here, the hyper-parameter α controls the strength of the NCR term. This differs slightly from the implementation in [21], where the authors vary the NCR term by α and the CE term by $1-\alpha$ (however, both implementations adjust the relative contributions of the CE and NCR loss terms).

As alternatives to KL-divergence, we also investigate the effect of Jensen-Shannon divergence and mean absolute error within the NCR term. Results for these additional metrics can be found in Section F of the Supplementary Material.

Evaluation metrics

We evaluate the trained models with commonly used classification metrics: area under the receiver operator characteristic curve (AUROC), area under the precision recall curve (AUPRC), sensitivity and specificity, alongside 95% confidence intervals (CIs) based on 1,000 bootstrapped samples taken from the test set. Tests of significance (represented by p-values), comparing the accuracies between models, are calculated by evaluating how many times one model performs better than another, across 1, 000 pairs of bootstrapped iterations drawn from the test set.

Hyperparameter optimization and threshold adjustment

Hyperparameter values were chosen through grid search and standard five-fold cross-validation, using the training set (note that the validation and held-out test sets are used in threshold adjustment and final model evaluation, respectively). Grid search was used to determine the number of hidden layers of *baseline NN*, the number of nodes used in each layer, the learning rate, the max depth of XGBoost, the number of nearest neighbours in NCR, the weight of the NCR term, the starting epoch for NCR, and the ϵ and λ values used in Label Smoothing and Mix-up, respectively. Details on the hyperparameter values used in the reported results can be found in Supplementary Table 4.

In ML classification models, the output typically represents the probability of the input belonging to a certain class, where often a threshold on this probability needs to be set to determine a discrete label. We determined this threshold via grid search on the validation split.

For our specific objective, we tuned the threshold to achieve sensitivities of 0.85, ensuring that the model maintains clinically acceptable performance in effectively identifying positive COVID-19 cases. This sensitivity was chosen to exceed lateral flow device (LFD) tests, which achieved a sensitivity of 56.9% (95% confidence interval 51.7%-62.0%) for OUH admissions between December 23, 2021 and March 6, 2021 [44]. Additionally, the gold standard for diagnosing viral genome targets is by real-time PCR (RT-PCR), which had estimated sensitivities around 80%-90% during different times of the pandemic [49–51].

Results

Comparison of methods

In Fig. 4, we compare the performance of all the methods in terms of AUROC for different amounts of label noise. Full numerical results can be found in Supplementary Table 7. The regularizers from CV domain provide significant improvement over the baselines, with Mix-up and NCR emerging as the best performing methods in 14 out of 25 test sets. Since both the Mix-up and NCR methods demonstrated strong performance, we also conducted an assessment of their combined use. The model performances remained fairly consistent across the Mix-up, NCR, and combined Mix-up and NCR methods, with the combined method consistently outperforming the others. NCR or Mix-up method followed as the second-best performers in 18 out of 25 test cases. However, the difference in accuracy between Mix-up (both independently and when combined with NCR) compared to NCR, was not found to be significant ($p > 0.05$, based on 1,000 bootstrapped iterations; exact p-values can be found in the Supplementary Table 6). Amongst the CV methods, label smoothing performed the worst, steadily decreasing in AUROC as the error in the training data increased, across all test sets (AUROC decreased by up to 3% when there was 40% error in COVID-19 positive cases). A more detailed discussion of these results can be found in "Conclusion and discussion" section.

When considering the mean AUROC performances (alongside standard deviation) for each test set (see Supplementary Table 5), across different methods, we found that models trained with NCR (both alone and combined with Mix-up) exhibited lower standard deviations, indicating more consistent classification performance across different noise ratios. Standard deviations were between 0.004-0.008 for NCR and Mix-up+NCR methods, across all test sets, compared to standard devisions of >0.010 for all other methods (except Mix-up, where standard deviation was 0.005-0.006 for three of the five test sets). This implies that the performance drop with increasing noise was reduced with NCR compared to other methods. This is true across all test sets, suggesting that the models trained with NCR are robust and generalizable across independent and unseen cohort distributions. Full numerical values can be found in Supplementary Table 5.

Motivated by these findings, we conducted a more comprehensive assessment of training with NCR. With respect to clinical applications, we chose to focus on NCR since it does not involve altering the data. This is in contrast to Mix-up, which might be considered as using "synthetic data," and may potentially be unsuitable for clinical tasks. NCR also performed closely to that of Mix-up and the combination of Mix-up and NCR. These results fell within confidence intervals, with accuracies that exhibited no significant differences across 1,000 bootstrapped runs.

Extended analysis with neighbour consistency regularization

After performing grid search, using five-fold standard cross validation, we determined the best hyperparameters to use in training (final hyperparameter values used can be found in Supplementary Table 4. The following results shown are for NCR with KL-divergence as mentioned in Eq. 4. This is the same formulation as presented in the original NCR paper [21]; however, we additionally performed similar analyses with other divergence measures. Results for these have been reported in Section F of the Supplementary Material. Our results indicate that these divergence measures produce outcomes similar to those achieved when utilizing KL-divergence.

Ablation study

We further investigate the effect of the key hyperparameters of NCR at different noise levels. Specifically, we analyze the impact of the hidden layer used for extracting the latent representations for calculating the NCR term, the weight of the NCR term, the epoch at which NCR is initialized, the batch size used, and the number of neighbors k. A subset of results for only PUH and BH datasets is shown in Fig. 5 for different noise levels and the full set of results (including the hidden layer and the batch size used) can be found in Supplementary Fig. 4. Figure 5 illustrates that increasing the weight of NCR can improve performance at higher noise ratios, up to a certain extent. Furthermore, it is clear that irrespective of the noise level, providing a *warm-up* period of around 30 epochs

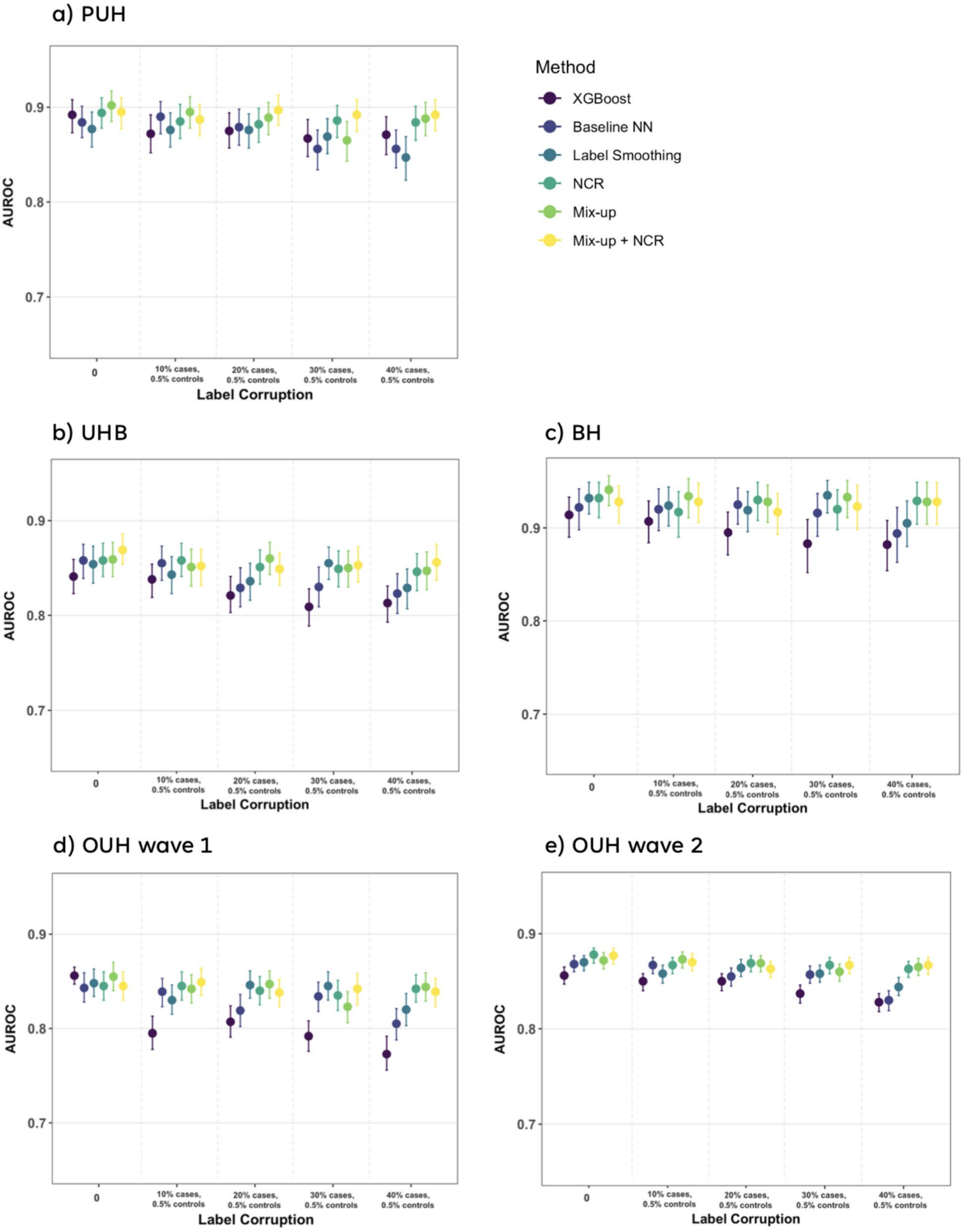


Fig. 4 Change in performance (AUROC) at different training label corruption levels across different methods. Panels (**a**-**e**) show results for different test sets

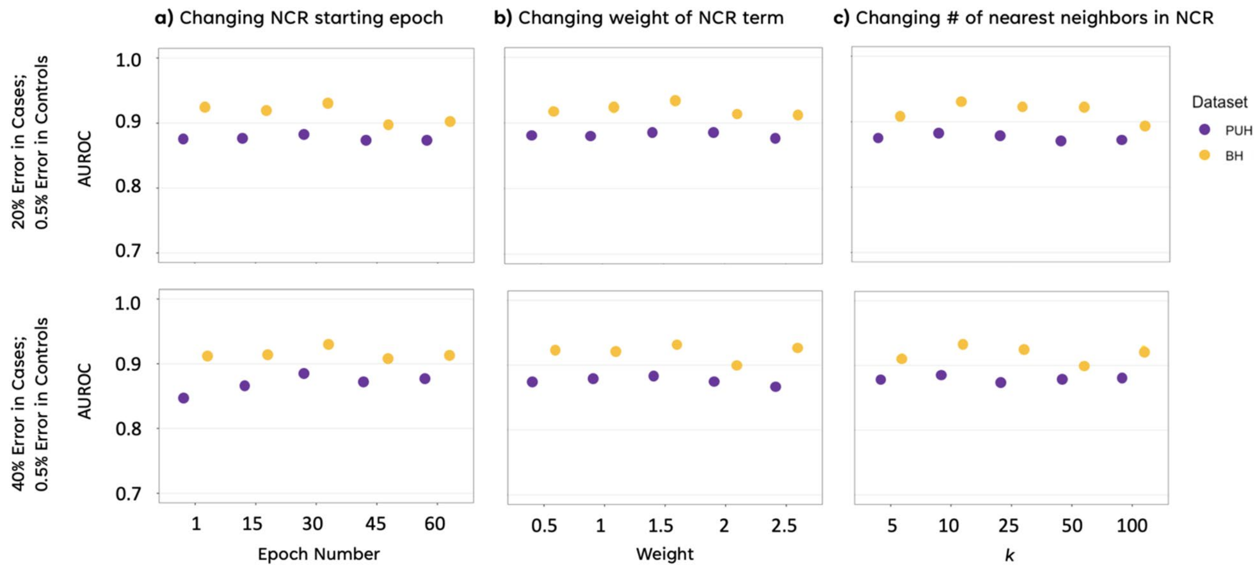


Fig. 5 Ablation study showing model performance with NCR with varying NCR hyperparameters for PUH and BH test sets, across varying **a** NCR starting epochs, **b** NCR weights, and **c** number of nearest neighbors (k). Results presented for 20% error in cases and 0.5% error in controls, and 40% error in cases and 0.5% error in controls

with just the binary cross-entropy loss enabled (before switching on the NCR loss term) helps achieve superior performance than enabling both loss terms (as shown in Eq. 4) from the start of training. A detailed discussion of these results can be found in "Conclusion and discussion" section. Finally, we also observe that considering around 10 nearest neighbors in the NCR loss formulation achieves the best results, with performance saturating with increasing number of neighbors. Hence, we ran all our experiments with 10 neighbors. Complete ablation results including all test sets and hyperparameters, across different noise ratios, can be found Supplementary Figure 5.

In comparison to the *baseline NN* model, NCR achieves a significant improvement in performance, reaching up to a 3.7% increase across all noise ratios for all test sets (Table 1). Surprisingly, the use of NCR achieves performance that is comparable, and in some cases even superior, to that of standard cross entropy when there is 0% added noise ($p > 0.05$), suggesting a general regularization effect of NCR.

As per expectations, the improvement in AUROC is particularly pronounced in the presence of higher levels of label noise within the training set (p-values were generally significant, i.e. < 0.05, at most noise ratios above 10%), providing credibility to the fact that NCR indeed plays a crucial role in constraining the model to learn meaningful patterns, rather than simply memorizing noisy data labels. Detailed numerical results for AUROC, AUPRC, sensitivity, and specificity are shown in Table 1 (results for positive predictive value (PPV) and negative predictive value (NPV) can be found in Supplementary Table 8).

In addition to the considered datasets so far, we present results from two additional case studies conducted using the eICU Collaborative Research Database and the Adult (Census Income) Dataset. Again, NCR proved to be effective in addressing label noise at various noise rates thereby showing its utility in healthcare as well as other tabular data domains. Comprehensive results and analysis for these two tasks can be found in Section H of the Supplementary Material.

Analysis of feature embeddings

By utilizing datasets that contain known noise, we have the opportunity to compare the feature similarity among training examples, considering whether they are correctly or incorrectly labeled as belonging to the same or different classes. In an ideal scenario, the distributions of within-class and between-class similarities for clean examples would not overlap and would perfectly match the true within-class and between-class similarities for mislabeled examples. We perform this comparison for both the *baseline NN* model and the model when trained with NCR. Although the distributions for the baseline model exhibit overlap, they are not identical, indicating that the feature similarities still contain some useful information that NCR can leverage. The utilization of NCR during training results in improved separation of classes in the feature space (Fig. 6).

Table 1 AUROC, AUPRC, Sensitivity, and Specificity comparison between baseline and NCR models, across different amounts of error and test sets. In addition to label error in COVID-19 positive cases, there is also 0.5% label error in the negative controls. 0% error represents the original dataset, without any added label noise. *p*-values shown compare differences in performance between the baseline NN (trained with cross entropy loss only) and the same model trained with NCR in addition to cross entropy loss (denoted by NCR). Bold faced values for each metric denote best performing method across different test sets

	AUROC		**AUPRC**		**Sensitivity**		**Specificity**		
Test Set	**Baseline NN**	**CE+NCR**	**Baseline NN**	**CE+NCR**	**Baseline NN**	**CE+NCR**	**Baseline NN**	**CE+NCR**	***p*-value**
0% error									
PUH	0.884(0.868-0.901)	**0.894(0.878-0.910)**	0.538(0.494-0.583)	**0.579(0.536-0.624)**	**0.841(0.81-0.871)**	0.807(0.774-0.838)	0.722(0.713-0.731)	**0.831(0.824-0.838)**	$p = 0.38$
UHB	0.858(0.839-0.875)	0.858(0.841-0.876)	0.309(0.278-0.340)	**0.387(0.348-0.431)**	**0.861(0.833-0.889)**	0.831(0.800-0.859)	0.638(0.629-0.646)	**0.723(0.716-0.731)**	$p = 0.396$
BH	0.922(0.898-0.942)	**0.932(0.911-0.949)**	0.691(0.627-0.751)	**0.753(0.690-0.803)**	**0.931(0.894-0.961)**	0.903(0.859-0.941)	0.659(0.636-0.684)	**0.770(0.748-0.790)**	$p = 0.346$
OUH "wave 2"	0.868(0.860-0.877)	**0.878(0.869-0.885)**	0.554(0.536-0.573)	**0.635(0.616-0.653)**	**0.875(0.862-0.887)**	0.845(0.831-0.858)	0.604(0.598-0.609)	**0.717(0.712-0.722)**	$p = 0.485$
OUH "wave 1"	0.843(0.828-0.859)	**0.845(0.830-0.860)**	0.087(0.078-0.097)	**0.155(0.133-0.181)**	**0.859(0.837-0.882)**	0.806(0.78-0.83)	0.615(0.613-0.618)	**0.729(0.726-0.731)**	$p = 0.273$
Error in 10% cases									
PUH	**0.890(0.872-0.906)**	0.885(0.867-0.903)	**0.572(0.529-0.617)**	0.553(0.509-0.601)	0.846(0.816-0.875)	**0.875(0.846-0.904)**	**0.745(0.737-0.754)**	0.677(0.668-0.687)	$p = 0.164$
UHB	0.855(0.837-0.873)	**0.858(0.841-0.876)**	**0.353(0.318-0.399)**	0.315(0.285-0.353)	0.854(0.825-0.882)	**0.877(0.852-0.903)**	**0.679(0.671-0.687)**	0.641(0.633-0.649)	$p = 0.014$
BH	**0.920(0.897-0.942)**	0.917(0.890-0.939)	0.693(0.630-0.760)	**0.701(0.635-0.763)**	0.917(0.874-0.954)	**0.924(0.884-0.958)**	**0.734(0.712-0.756)**	0.713(0.689-0.738)	$p = 0.379$
OUH "wave 2"	0.867(0.858-0.875)	0.867(0.858-0.875)	**0.615(0.596-0.634)**	0.584(0.566-0.604)	0.865(0.851-0.877)	**0.888(0.876-0.899)**	**0.647(0.641-0.652)**	0.565(0.56-0.571)	$p = 0.005$
OUH "wave 1"	0.839(0.823-0.853)	**0.845(0.83-0.86)**	**0.124(0.106-0.146)**	0.098(0.087-0.11)	0.850(0.827-0.871)	**0.873(0.852-0.894)**	**0.642(0.639-0.645)**	0.564(0.562-0.567)	$p = 0.325$
Error in 20% cases									
PUH	0.879(0.860-0.898)	**0.882(0.863-0.899)**	0.471(0.427-0.519)	**0.537(0.492-0.584)**	0.854(0.823-0.883)	**0.878(0.850-0.904)**	**0.723(0.715-0.733)**	0.681(0.673-0.691)	$p = 0.491$
UHB	0.829(0.809-0.850)	**0.851(0.833-0.869)**	0.272(0.243-0.305)	**0.361(0.322-0.408)**	0.834(0.803-0.863)	**0.886(0.862-0.910)**	**0.646(0.638-0.654)**	0.608(0.599-0.616)	$p = 0.079$
BH	0.925(0.904-0.943)	**0.930(0.908-0.949)**	0.679(0.614-0.74)	**0.718(0.65-0.792)**	0.910(0.868-0.947)	**0.924(0.885-0.959)**	**0.718(0.695-0.741)**	0.702(0.677-0.723)	$p = 0.01$
OUH "wave 2"	0.855(0.845-0.864)	**0.869(0.86-0.877)**	0.524(0.504-0.542)	**0.618(0.599-0.638)**	0.860(0.847-0.873)	**0.892(0.881-0.903)**	**0.606(0.601-0.612)**	0.548(0.542-0.554)	$p = 0.303$
OUH "wave 1"	0.819(0.802-0.836)	**0.84(0.825-0.855)**	0.065(0.058-0.072)	**0.126(0.110-0.147)**	0.840(0.818-0.863)	**0.876(0.854-0.897)**	**0.607(0.605-0.610)**	0.571(0.568-0.573)	$p = 0.038$
Error in 30% cases									
PUH	0.856(0.834-0.876)	**0.886(0.868-0.902)**	0.493(0.448-0.544)	**0.516(0.471-0.565)**	0.865(0.834-0.891)	**0.888(0.861-0.914)**	0.593(0.583-0.602)	**0.644(0.635-0.653)**	$p = 0.051$
UHB	0.830(0.809-0.851)	**0.849(0.830-0.868)**	0.315(0.281-0.356)	**0.371(0.331-0.416)**	**0.868(0.84-0.892)**	0.863(0.836-0.89)	0.527(0.519-0.536)	**0.612(0.604-0.621)**	$p = 0.005$
BH	0.916(0.891-0.937)	**0.920(0.898-0.941)**	0.675(0.605-0.74)	**0.678(0.609-0.75)**	**0.958(0.928-0.985)**	0.917(0.874-0.952)	0.555(0.530-0.581)	**0.696(0.673-0.721)**	$p = 0.464$
OUH "wave 2"	0.857(0.848-0.866)	**0.867(0.858-0.875)**	**0.581(0.560-0.600)**	0.577(0.557-0.597)	0.891(0.879-0.902)	**0.899(0.888-0.910)**	0.514(0.509-0.520)	**0.520(0.514-0.525)**	$p = 0.001$
OUH "wave 1"	0.834(0.818-0.849)	**0.835(0.819-0.851)**	**0.121(0.104-0.142)**	0.116(0.099-0.136)	**0.880(0.858-0.899)**	0.873(0.852-0.895)	0.511(0.508-0.514)	**0.524(0.522-0.527)**	$p = 0.023$
Error in 40% cases									
PUH	0.856(0.836-0.876)	**0.884(0.865-0.901)**	0.488(0.443-0.543)	**0.553(0.506-0.598)**	0.870(0.841-0.898)	**0.870(0.840-0.898)**	0.548(0.539-0.558)	**0.642(0.634-0.652)**	$p = 0.014$
UHB	0.823(0.802-0.844)	**0.846(0.826-0.865)**	0.352(0.313-0.399)	**0.331(0.295-0.375)**	0.852(0.823-0.879)	**0.875(0.848-0.902)**	0.576(0.568-0.585)	**0.598(0.589-0.606)**	$p = 0.027$

Table 1 (continued)

	AUROC		AUPRC		Sensitivity		Specificity		
Test Set	**Baseline NN**	**CE+NCR**	**Baseline NN**	**CE+NCR**	**Baseline NN**	**CE+NCR**	**Baseline NN**	**CE+NCR**	***p*-value**
BH	0.894(0.863-0.922)	**0.929(0.904-0.949)**	0.701(0.640-0.760)	**0.7(0.631-0.776)**	0.903(0.859-0.941)	**0.938(0.903-0.969)**	0.634(0.612-0.660)	**0.668(0.644-0.693)**	$p = 0.262$
OUH "wave 2"	0.830(0.819-0.84)	**0.863(0.854-0.871)**	0.553(0.532-0.572)	**0.577(0.557-0.598)**	0.852(0.839-0.865)	**0.898(0.886-0.908)**	**0.514(0.508-0.519)**	0.511(0.505-0.516)	$p < 0.001$
OUH "wave 1"	0.805(0.788-0.821)	**0.842(0.828-0.857)**	0.122(0.105-0.144)	**0.09(0.078-0.104)**	0.84(0.818-0.863)	**0.89(0.871-0.911)**	0.497(0.494-0.500)	**0.534(0.531-0.537)**	$p = 0.002$

We further compare the distribution of cosine similarities for training examples that are correctly or incorrectly labeled as the same class or different classes in Fig. 6. The features learned using NCR exhibit noticeably better class separation when compared to the features learned using the *baseline NN* utilizing only cross entropy (CE) loss (top row).

Analysis of prediction confidence

In Fig. 7, additional evidence is provided to support the hypothesis that NCR effectively prevents the memorization of noisy labels. Namely, we assessed the confidence level associated with the predicted label for each training example. Panel b) of the figure demonstrates that the model trained with NCR more frequently assigns higher confidence to correctly labeled samples. In contrast, the model trained without NCR, in panel a), tends to overfit to the noisy labels, resulting in a lower confidence assignment to a larger number of correctly labeled samples.

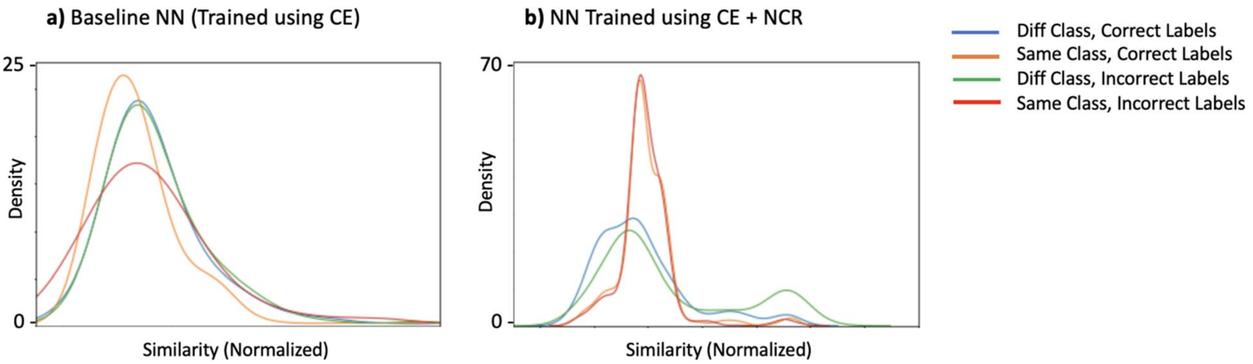


Fig. 6 Feature similarity distributions at the end of training, for samples correctly and incorrectly labelled, across both similar and different classes without and with NCR. Feature similarity is calculated using cosine similarity as described in Eq. 3. Results shown are for models trained on data with 40% noisy labels

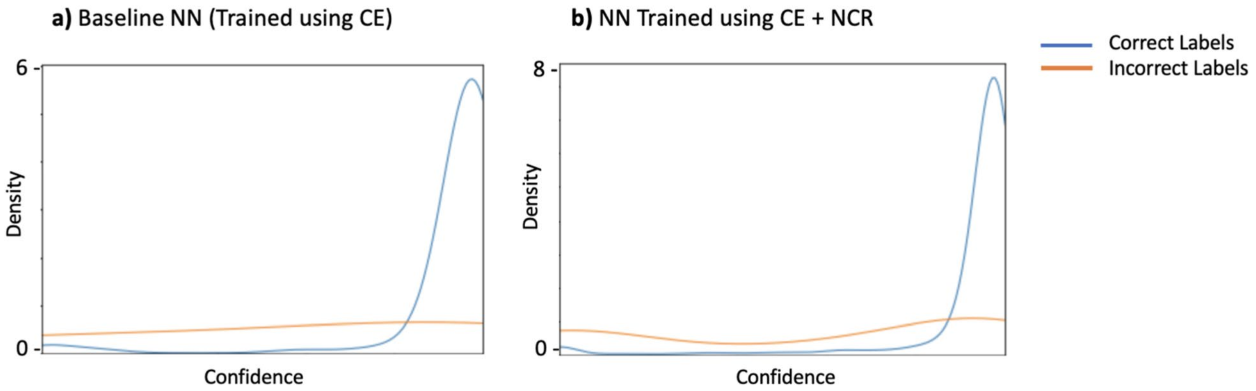


Fig. 7 Predicted confidence of correctly labeled and incorrectly labeled training examples during different stages of training. **a** shows the predicted confidences when using Baseline NN trained using only cross entropy (CE) loss and **b** that of the same model when trained using CE+NCR loss terms

Conclusion and discussion

The results of this research highlight the significant promise of incorporating CV methods to tackle the issues stemming from noisy labels in EHR data. Our investigation underscores that despite the notable differences between CV and EHR datasets, the adaptation and fusion of various CV techniques to address label inaccuracies can offer substantial benefits in healthcare applications. Through the utilization of these recently introduced CV techniques, specifically Label Smoothing, Mix-up, and NCR, we can significantly improve the robustness of DL models when confronted with noisy labels during the analysis of EHR data.

Similar to previous studies [18–20], we noticed that deep neural networks, when trained without these noise-mitigation methods, have the capability to memorize random and noisy labels and exhibit poor generalization on unseen test sets (see Supplementary Table 7 and Fig. 4). In particular, we showed that both Mix-up and NCR, individually as well as when combined, demonstrated strong performance in mitigating overfitting to noisy class labels. This success can be attributed to the fact that neither of these methods solely relies on the initial labels (which can be noisy) for evaluating the model's performance.

We also found that the use of NCR achieves performance that is comparable, and in some cases even superior, to that of standard cross entropy when there is 0% added noise, suggesting a general regularization effect of NCR. This may be because the original training data from the hospitals is already noisy to begin with (even without addition of synthetic noise to the labels), and thus, NCR is able to show improved performance when trained with inherently noisy labels. Improvement in AUROC was generally found to be significant, i.e. $p < 0.05$, across the majority of test sets for noise ratios above 10%. This is reflective of how NCR has greater effect when there is higher levels of label corruption, and less significant effect at low levels of label corruption.

In addition to examining each method in isolation, our findings revealed that combining multiple methods, specifically Mix-up and NCR, exhibited superior performance, outperforming each method when used independently. This improvement stems from the joint effect of Mix-up and NCR in mitigating label noise, with Mix-up also contributing to data augmentation during the training process. This paves the way for future research possibilities, further encouraging the use of multiple CV-based techniques for effectively addressing label noise.

Of the three CV techniques considered, label smoothing performed the worst. We hypothesize that this is because it adds noise to both noisy and clean labels. Thus, as the majority of labels (particularly, COVID-19 negative controls) in our training set are correct, label smoothing can be less effective and actually decrease the accuracy of the model in many cases. Furthermore, previous studies have found that the advantage of label smoothing vanishes when label noise is high [40], which we also observed when the error in COVID-19 positive cases was high.

Our ablation studies with NCR reveal that increasing the weight of NCR can improve performance at higher noise ratios, up to a certain extent. This finding aligns with the expectation that higher noise ratios require a stronger NCR effect to counteract the impact of noise. Additionally, we found that providing a *warm-up* period of few training epochs with just the binary cross-entropy loss enabled (before switching on the NCR loss term) helps achieve superior performance compared to enabling both loss terms from the start of training. We hypothesize that this is because the initial training phase with just the cross entropy loss allows the model to learn the underlying patterns in the noisy data before applying the regularizer, which appears to be more effective when the model has converged to some extent. Also we can view it from a *curriculum learning* perspective, where the model progressively tackles more challenging aspects of the task, i.e., to account for the noise in the data and learning to avoid overfitting to it.

While our efforts were directed at mitigating label noise, it is crucial to emphasize that errors can persist within the features themselves, potentially leading to inaccuracies in the model performance. Feature noise has been a subject of extensive research [53–58]; however, it was not the primary focus of our study. In addition to feature noise, we also encountered missing data concerning the features. To handle this issue, we opted for population median imputation, consistent with similar COVID-19 studies that utilized the same dataset. Nevertheless, it is worth noting that the underlying reasons for and the nature of missing data could carry important information about the source and nature of errors. Therefore, future research should continue exploring alternative approaches for assessing and address missing data.

It should also be noted that the techniques employed can alter decision boundaries and the outcomes of models, potentially affecting the perceived significance of features in classification. For example, mix-up combines features from paired input samples, resulting in smoother decision boundaries within the feature space. Consequently, the importance of individual features may become less pronounced or more diffuse compared to models trained without blending. Additionally, mix-up promotes the learning of robust and generalizable representations by introducing perturbations in the input data space. This can lead to higher importance being assigned to features less affected by mix-up blending, while those highly sensitive to blending may be considered less important. Research has shown the benefits of mix-up in learning rare features compared to standard methods [59].

Similarly, Neighbor Consistency Regularization (NCR) also encourages smoothness and consistency in predictions among neighboring data points. Thus, features contributing to stable predictions and smooth decision boundaries within local neighborhoods are likely to be prioritized by the model. However, an excessive reinforcement of feature significance within local neighborhoods may lead to a focus on local rather than global feature importance.

Furthermore, label smoothing enhances the model's resilience to noisy or incorrectly labeled training data by reducing the influence of individual erroneous labels during training. Consequently, this can result in a more accurate estimation of feature importance by mitigating the effects of label noise on the acquired decision boundaries.

Overall, the impact of such techniques on feature importance may vary depending on the specific characteristics of the dataset, model architecture, and training process. Therefore, future studies may benefit from further investigation into model explainability.

Moreover, probabilities can prove valuable for specific tasks, particularly when binary classification lacks the necessary depth of information. This approach aligns well with the CV techniques we explored, namely Mix-up and Label Smoothing, both of which yield continuous values instead of discrete binary outcomes. Our decision to employ binary classification (COVID-19 positive or negative) was made to align with rapid triaging processes used in hospitals [44, 48].

Finally, we are also aware that the COVID-19 datasets we employed provide a limited perspective compared to the extensive information available in EHR systems. It is worth noting that significant portions of EHR data, such as treatment-related specifics or lifestyle and environmental factors, among others, are not fully represented in the datasets we have utilized in this study. Therefore, further research is required to gain a comprehensive understanding of the consequences of diverse types of label noise and to assess how different noise mitigation techniques influence model performance.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12911-024-02581-5.

Supplementary Material 1.

Acknowledgements

We express our sincere thanks to all patients and staff across the four participating NHS trusts; Oxford University Hospitals NHS Foundation Trust, University Hospitals Birmingham NHS Trust, Bedfordshire Hospitals NHS Foundations Trust, and Portsmouth Hospitals University NHS Trust.

We additionally express our thanks to Artur Speiser and Luca Bertinetto for their comments during this project.

Authors' contributions

JY conceived and ran the experiments, with design input from MP and HT. MP and DAC supervised the work. JY and AAS preprocessed the COVID-19 datasets. JY and MP implemented the code. AAS applied for the ethical approval and co-ordinated data extraction. All authors revised the manuscript.

Funding

This work was supported by the Wellcome Trust/University of Oxford Medical & Life Sciences Translational Fund (Award: 0009350), and the Oxford National Institute of Research (NIHR) Biomedical Research Centre (BRC). JY is a Marie Sklodowska-Curie Fellow, under the European Union's Horizon 2020 research and innovation programme (Grant agreement: 955681, "MOIRA"). AAS is an NIHR Academic Clinical Fellow (Award: ACF-2020-13-015). DAC was supported by a Royal Academy of Engineering Research Chair, an NIHR Research Professorship, the InnoHK Hong Kong Centre for Cerebro-cardiovascular Health Engineering (COCHE), and the Pandemic Sciences Institute at the University of Oxford. The funders of the study had no role in study design, data collection, data analysis, data interpretation, or writing of the manuscript. The views expressed in this publication are those of the authors and not necessarily those of the funders.

Availability of data and materials

Data from OUH studied here are available from the Infections in Oxfordshire Research Database (https://oxfordbrc.nihr.ac.uk/research-themes/modernising-medical-microbiology-and-big-infection-diagnostics/infections-in-oxfordshire-research-database-iord/), subject to an application meeting the ethical and governance requirements of the Database. Data from UHB, PUH and BH are available on reasonable request to the respective trusts, subject to HRA requirements.

Declarations

Ethics approval and consent to participate

United Kingdom National Health Service (NHS) approval via the national oversight/regulatory body, the Health Research Authority (HRA), has been granted for use of routinely collected clinical data and microbiology data from electronic health records (EHRs) to develop and validate artificial intelligence models to detect Covid-19 (CURIAL; NHS HRA IRAS ID: 281832).

As the study was limited to working with deidentified data collected within routine care, and extracted retrospectively, explicit patient consent for use of the data was deemed to not be required, and is covered within the HRA approval. All necessary consent has been obtained and the appropriate institutional forms have been archived.

Consent for publication

NHS Health Research Authority approval was granted for the use of routine clinical and microbiology data from electronic health records (EHRs) for development and validation of artificial intelligence models to detect COVID-19 (CURIAL; IRAS ID 281832). As the study was limited to working with deidentified data collected within routine care, and extracted retrospectively, explicit patient consent for use of the data was deemed to not be required, and is covered within the HRA approval. All necessary consent has been obtained and the appropriate institutional forms have been archived.

Competing interests

DAC reports personal fees from Oxford University Innovation, personal fees from BioBeats, personal fees from Sensyne Health, outside the submitted work. MP and HT are employees at Exscientia.

Received: 26 January 2024 Accepted: 20 June 2024
Published online: 27 June 2024

References

1. Bowman S. Impact of electronic health record systems on information integrity: quality and safety implications. Perspect Health Inf Manag. 2013;10(Fall):1c.
2. Ling Y, An Y, Liu M, Hu X. An error detecting and tagging framework for reducing data entry errors in electronic medical records (EMR) system. In: 2013 IEEE International Conference on Bioinformatics and Biomedicine. IEEE; 2013. pp. 249-254.
3. Roman LC, Ancker JS, Johnson SB, Senathirajah Y. Navigation in the electronic health record: a review of the safety and usability literature. J Biomed Inform. 2017;67:69–79.
4. Wagner MM, Hogan WR. The accuracy of medication data in an outpatient electronic medical record. J Am Med Inform Assoc. 1996;3(3):234–44.
5. Kharrazi H, Ma X, Chang HY, Richards TM, Jung C. Comparing the predictive effects of patient medication adherence indices in electronic health record and claims-based risk stratification models. Popul Health Manag. 2021;24(5):601–9.
6. Winden TJ, Chen ES, Monsen KA, Wang Y, Melton GB. Evaluation of flowsheet documentation in the electronic health record for residence, living situation, and living conditions. AMIA Summits Transl Sci Proc. 2018;2018:236.
7. Kim MO, Coiera E, Magrabi F. Problems with health information technology and their effects on care delivery and patient outcomes: a systematic review. J Am Med Inform Assoc. 2017;24(2):246–50.
8. Menon S, Singh H, Giardina TD, Rayburn WL, Davis BP, Russo EM, Sittig DF. Safety huddles to proactively identify and address electronic health record safety. J Am Med Inform Assoc. 2017;24(2):261–7.
9. Ronquillo JG, Zuckerman DM. Software-related recalls of health information technology and other medical devices: implications for FDA regulation of digital health. Milbank Q. 2017;95(3):535–53.
10. Bell SK, Delbanco T, Elmore JG, Fitzgerald PS, Fossa A, Harcourt K, DesRoches CM. Frequency and types of patient-reported errors in electronic health record ambulatory care notes. JAMA Netw Open. 2020;3(6):e205867–e205867.
11. Graber ML, Siegal D, Riah H, Johnston D, Kenyon K. Electronic health record-related events in medical malpractice claims. J Patient Saf. 2019;15(2):77.
12. Saah AJ, Hoover DR. "Sensitivity" and "specificity" reconsidered: the meaning of these terms in analytical and diagnostic settings. Ann Intern Med. 1997;126(1):91–4.
13. Van Stralen KJ, Stel VS, Reitsma JB, Dekker FW, Zoccali C, Jager KJ. Diagnostic methods I: sensitivity, specificity, and other measures of accuracy. Kidney Int. 2009;75(12):1257–63.
14. Kaboli PJ, McClimon BJ, Hoth AB, Barnett MJ. Assessing the accuracy of computerized medication histories. Am J Manage Care. 2004;10(11 Pt 2):872–7.
15. Khajouei R, Abbasi R, Mirzaee M. Errors and causes of communication failures from hospital information systems to electronic health record: a record-review study. Int J Med Inform. 2018;119:47–53.
16. Staroselsky M, Volk LA, Tsurikova R, Newmark LP, Lippincott M, Litvak I, Bates DW. An effort to improve electronic health record medication list accuracy between visits: patients' and physicians' response. Int J Med Inform. 2008;77(3):153–60.
17. Yadav S, Kazanji N, KC N, Paudel S, Falatko J, Shoichet S, Barnes MA. Comparison of accuracy of physical examination findings in initial progress notes between paper charts and a newly implemented electronic health record. J Am Med Inform Assoc. 2017;24(1):140-4.
18. Song H, Kim M, Park D, Shin Y, Lee JG. Learning from noisy labels with deep neural networks: a survey. IEEE Trans Neural Netw Learn Syst. 2022;34(11):8135–53.
19. Wang Y, Ma X, Chen Z, Luo Y, Yi J, Bailey J. Symmetric cross entropy for robust learning with noisy labels. In: Proceedings of the IEEE/CVF international conference on computer vision. 2019. pp. 322–30.
20. Zhang C, Bengio S, Hardt M, Recht B, Vinyals O. Understanding deep learning (still) requires rethinking generalization. Commun ACM. 2021;64(3):107–15.
21. Iscen A, Valmadre J, Arnab A, Schmid C. Learning with neighbor consistency for noisy labels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022. pp. 4672–81.
22. Lukasik M, Bhojanapalli S, Menon A, Kumar S. Does label smoothing mitigate label noise? In: International Conference on Machine Learning. PMLR; 2020. pp. 6448-6458.
23. Zhang H, Cisse M, Dauphin YN, Lopez-Paz D. mixup: Beyond Empirical Risk Minimization. In: International Conference on Learning Representations. 2018.
24. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L, Imagenet: a large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. IEEE; 2009. pp. 248–55.
25. Chapman AD. Principles and methods of data cleaning. GBIF; 2005.
26. Perez H, Tah JH. Improving the accuracy of convolutional neural networks by identifying and removing outlier images in datasets using t-SNE. Mathematics. 2020;8(5):662.
27. Stolarek I, Samelak-Czajka A, Figlerowicz M, Jackowiak P. Dimensionality reduction by UMAP for visualizing and aiding in classification of imaging flow cytometry data. Iscience. 2022;25(10):105142.
28. Ben-Gal I. Outlier detection. Data mining and knowledge discovery handbook. Springer US. 2005. pp. 131–46.
29. Breunig MM, Kriegel HP, Ng RT, Sander J. LOF: identifying density-based local outliers. In: Proceedings of the 2000 ACM SIGMOD international conference on Management of data. 2000. pp. 93–104.
30. Kriegel HP, Kröger P, Schubert E, Zimek A. Outlier detection in axis-parallel subspaces of high dimensional data. In: Advances in Knowledge Discovery and Data Mining: 13th Pacific-Asia Conference, PAKDD 2009 Bangkok, Thailand, April 27-30, 2009 Proceedings 13 (pp. 831-838). Springer Berlin Heidelberg; 2009.
31. Yang J, Soltan AA, Eyre DW, Yang Y, Clifton DA. An adversarial training framework for mitigating algorithmic biases in clinical machine learning. NPJ Digit Med. 2023;6(1):55.
32. Han J, Luo P, Wang X. Deep self-learning from noisy labels. In: Proceedings of the IEEE/CVF international conference on computer vision. 2019. pp. 5138–47.
33. Lee DH. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: Workshop on challenges in representation learning. ICML. 2013;3(2):896.
34. Wang X, Hua Y, Kodirov E, Clifton DA, Robertson NM. Proselflc: Progressive self label correction for training robust deep neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021. pp. 752-761.
35. Blum A, Mitchell T. Combining labeled and unlabeled data with co-training. In: Proceedings of the eleventh annual conference on Computational learning theory. 1998. pp. 92-100.
36. Nigam K, Ghani R. Analyzing the effectiveness and applicability of co-training. In: Proceedings of the ninth international conference on Information and knowledge management. 2000. pp. 86-93.
37. Ghosh A, Kumar H, Sastry PS. Robust loss functions under label noise for deep neural networks. In: Proceedings of the AAAI conference on artificial intelligence, vol. 31, no. 1. 2017.
38. Xu Y, Cao P, Kong Y, Wang Y. L_dmi: A novel information-theoretic loss function for training deep nets robust to label noise. Adv Neural Inf Process Syst. 2019;32:6225–36.
39. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. pp. 2818-2826.
40. Wei J, Liu H, Liu T, Niu G, Sugiyama M, Liu Y. To Smooth or Not? When Label Smoothing Meets Noisy Labels. In: International Conference on Machine Learning. PMLR; 2022. pp. 23589-23614.
41. Reed S, Lee H, Anguelov D, Szegedy C, Erhan D, Rabinovich A. Training deep neural networks on noisy labels with bootstrapping. 2014. arXiv preprint arXiv:1412.6596.
42. Tjandra D, Wiens J. Leveraging an Alignment Set in Tackling Instance-Dependent Label Noise. In: Conference on Health, Inference, and Learning. 2023. pp.477-497.
43. Boughorbel S, Fethi J, Neethu V, Haithum E. Alternating loss correction for preterm-birth prediction from ehr data with noisy labels. 2018. arXiv preprint arXiv:1811.09782.
44. Soltan AA, Yang J, Pattanshetty R, Novak A, Yang Y, Rohanian O, Muthusami V. Real-world evaluation of rapid and laboratory-free COVID-19 triage for emergency care: external validation and pilot

deployment of artificial intelligence driven screening. Lancet Digit Health. 2022;4(4):e266–78.

45. Soltan AA, Kouchaki S, Zhu T, Kiyasseh D, Taylor T, Hussain ZB, Clifton DA. Rapid triage for COVID-19 using routine clinical data for patients attending hospital: development and prospective validation of an artificial intelligence screening test. Lancet Digit Health. 2021;3(2):e78–87.
46. Yang J, Soltan AA, Clifton DA. Machine learning generalizability across healthcare settings: insights from multi-site COVID-19 screening. NPJ Digit Med. 2022;5(1):69.
47. Yang J, El-Bouri R, O'Donoghue O, Lachapelle AS, Soltan AA, Eyre DW, Clifton DA. Deep reinforcement learning for multi-class imbalanced training: applications in healthcare. Mach Learn. 2023;113:2655–74.
48. Yang J, Soltan AA, Eyre DW, Clifton DA. Algorithmic fairness and bias mitigation for clinical machine learning with deep reinforcement learning. Nat Mach Intel. 2023:1-11.
49. Afzal A. Molecular diagnostic technologies for COVID-19: limitations and challenges. J Adv Res. 2020;26:149–59.
50. Miller TE, Beltran WFG, Bard AZ, Gogakos T, Anahtar MN, Astudillo MG, Lennerz JK. Clinical sensitivity and interpretation of PCR and serological COVID-19 diagnostics for patients presenting to the hospital. FASEB J. 2020;34(10):13877.
51. Williams TC, Wastnedge E, McAllister G, Bhatia R, Cuschieri K, Kefala K, Templeton KE. Sensitivity of RT-PCR testing of upper respiratory tract samples for SARS-CoV-2 in hospitalised patients: a retrospective cohort study. Wellcome Open Res. 2020;5:254.
52. Chen T, Guestrin C. Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016. pp. 785-794.
53. Dong Y, Peng CYJ. Principled missing data methods for researchers. SpringerPlus. 2013;2:1–17.
54. Harel O, Mitchell EM, Perkins NJ, Cole SR, Tchetgen Tchetgen EJ, Sun B, Schisterman EF. Multiple imputation for incomplete data in epidemiologic studies. Am J Epidemiol. 2018;187(3):576–84.
55. Peskoe SB, Arterburn D, Coleman KJ, Herrinton LJ, Daniels MJ, Haneuse S. Adjusting for selection bias due to missing data in electronic health records-based research. Stat Methods Med Res. 2021;30(10):2221–38.
56. Seaman SR, White IR. Review of inverse probability weighting for dealing with missing data. Stat Methods Med Res. 2013;22(3):278–95.
57. Sterne JA, White IR, Carlin JB, Spratt M, Royston P, Kenward MG, Carpenter JR. Multiple imputation for missing data in epidemiological and clinical research: potential and pitfalls. BMJ. 2009;338:b2393.
58. Wells BJ, Chagin KM, Nowacki AS, Kattan MW. Strategies for handling missing data in electronic health record derived data. Egems. 2013;1(3):1035.
59. Zou D, Cao Y, Li Y, Gu Q. The benefits of mixup for feature learning. In: International Conference on Machine Learning. PMLR; 2023. pp. 43423-43479.

Publisher's Note