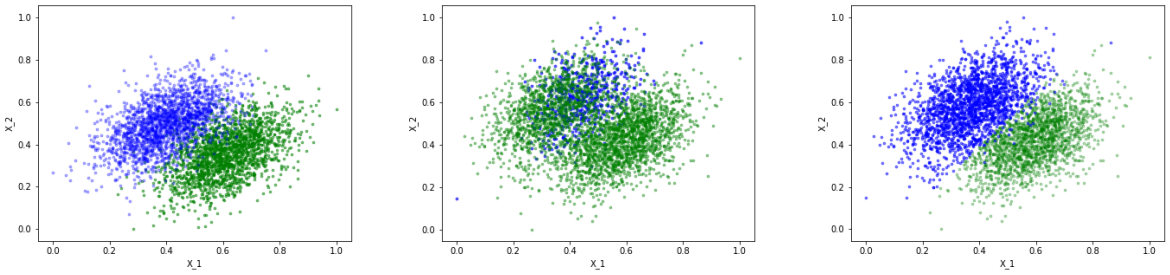
Addressing Feature-Dependent Label Noise: A Generative Framework



(a) Clean Data **(b) Noisy Data** **(c) Corrected Data**

Figure 1: Framework performance on feature-dependent label noise correction task on simulated 2D data (a). Equipped with solely the noisy data (b), the proposed framework recovers the original classes to produce the corrected data in (c). The use of an energy-based interpretation of class separation is key in accomplishing this.

ABSTRACT

We propose a framework for correcting mislabeled training examples in the context of binary classification. In particular, we focus on correcting *feature-dependent label noise*. Despite its ubiquity in real-world data, this type of noise is difficult to model. This difficulty has resulted in a relative dearth of literature when compared to research on random and class-dependent label noise. Two elements distinguish our approach from others: 1) instead of relying on the original feature space, we employ an autoencoder to learn a discriminative representation and 2) we introduce an energy-based formalism for the label correction problem. We demonstrate that using an energy-based model trained on a contrastive loss function is more resilient to feature-dependent label noise than existing methods. We demonstrate the technique's state-of-the-art label correction performance across eight datasets prevalent in the label correction literature, spanning synthetic and realistic settings. Furthermore, we derive analytical expressions relating the gradients of empirical risk to the label noise model.

KEYWORDS

label noise, binary classification

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

ACM Reference format:
. 2019. Addressing Feature-Dependent Label Noise: A Generative Framework. In *Proceedings of ACM Conference, Washington, DC, USA, July 2017 (Conference'17),* 9 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Machine learning algorithms depend on reliable training labels and sufficiently robust learning models in order to produce generalizable predictions [Frénay and Verleysen 2014; Zhu and Wu 2004]. Many datasets suffer from training label corruption, which can result from annotation error, human bias, or a noisy process for generating labels [Brodley and Friedl 1999; Smyth 1996]. In practice, it can be costly to obtain accurate labels; hence, research on reducing the effects of label noise on learning has received considerable attention.

Most work focuses on label noise that is independent of the input features (e.g., random label noise or class conditional random noise) [Brodley and Friedl 1999; Natarajan et al. 2013]. Although work on addresing label noise has yielded impressive results and improved generalization [Natarajan et al. 2013; Patrini et al. 2017; Rebbapragada and Brodley 2007a; Ren et al. 2018; Rolnick et al. 2017], the simplicity of the underlying noise assumptions fails to capture crucial mislabeling processes that arise in practice. The introduction of feature dependency significantly complicates mathematical analysis [Liu and Tao 2014; Natarajan et al. 2013; Northcutt et al. 2017]. This is especially common in domains where experts disagree, such as patient diagnosis. In this work, we propose a semi-supervised framework for addressing the effects of feature dependent label noise on supervised learning algorithms.

Label noise processes can be categorized into three types. Type I assumes any label in the training data is incorrect with probability γ. Type II assumes the probability of label corruption is class-conditional, such that we have different noise rate γ_i for each class. Finally, Type III noise assumes that label noise depends explicitly on input features.

Type III noise, or equivalently feature-dependent noise, is ubiquitous in empirical datasets [Frénay and Verleysen 2014; Lachenbruch 1974; Schafer and Graham 2002]. Type III noise often also results in unreliable labels for training instances in low density regions of feature space [Denoeux 1995, 1997, 2000] or near classification boundaries [Beigman and Klebanov 2009; Beigman Klebanov and Beigman 2009; Chhikara and McKeon 1984; Cohen 1997; Kolcz and Cormack 2009; Lachenbruch 1966, 1974]. As an example of Type III noise, consider the diagnostic labels for Alzheimer's disease. In this setting, the probability of label noise depends on both age and sex (i.e., younger male patients are harder to diagnose) [Khachaturian 1985; Murray et al. 2016].

In this paper, we present a practical framework for correcting Type III label noise. We accomplish this by employing an generative framework, in which we learn the distribution $p(x|y = i)$ for a given class i. Learning this manifold in a feature-dependent manner - in our case, through an autoencoder - allows the method to identify labels perturbed by feature-dependent label noise. We train this autoencoder on a small subset of the training data with reliable labels, either manually annotated or automatically inferred. This model is used to identify and correct mislabeled examples - as in **??** - based on the whether the assigned value of a given point is compatible with its original label. We empirically demonstrate our framework's ability to handle input-dependent label noise on both simulated and real datasets. In simulation, we test across six commonly used datasets and three Type III noise models. Regardless of the feature-dependent noise specification, the proposed approach achieves a higher average rank than all existing methods. This result remains true on a real dataset as well, involving the algorithmic annotation of 50,000 arrhythmias.

2 RELATED WORK

The full body of work on the problem of label noise is too extensive to review here; we direct interested readers to the detailed review by [Frénay and Verleysen 2014]. In this section, we highlight key contributions in the label noise literature. We then briefly discuss existing theoretical work.

2.1 Type I and Type II Methods

Existing work in label noise correction is principally designed for Type I and Type II noise but may be applied to Type III noise. This work falls into three categories: relabeling, learning procedures, and loss functions. We expand on each area below.

Relabeling: Earlier approaches to label noise focus on relabeling the noisy training set. [Brodley and Friedl 1999] use the output of an ensemble of classifiers to identify mislabeled training examples. [Sun et al. 2007] identify mislabeled instances based on the entropy of class probabilities output by a Bayesian classifier. These approaches are robust to Type I and Type II noise models, but, as demonstrated in experiments to follow, do not generalize to Type III.

Learning Procedures: More recent approaches suggest learning procedures that are resilient to label noise in training datasets. The Perceptron Algorithm with Margin (PAM) [Frénay and Verleysen 2014] aims to accomplish this using a margin, such that mislabeled examples hold less influence over the final model. [Crammer and Lee 2010] use a velocity-based learning procedure to learn the weight vector distribution, termed gaussian herding (NHERD). [Sukhbaatar et al. 2014] introduces a noise layer to a neural network architecture to learn the noise function, while [Rebbapragada and Brodley 2007b] weights each example by class confidence in the training procedure. These methods commonly suffer from overfitting to the noise.

Loss Functions: [Long and Servedio 2010] shows that classification algorithms that optimize a convex potential over a linear class are not robust to random label noise. This has led to work that aims to modify convex loss functions in order to make them noise-tolerant in the presence of Type I and Type II noise. [Ghosh et al. 2015] derived sufficient conditions for classification losses that render them robust to random noise; namely, the components of a loss (where each component is defined over a given class) must sum to a constant value. [Rooyen et al. 2015] proposed a convex loss that avoids the negative result of [Long and Servedio 2010] for type I noise by virtue of being negatively unbounded. [Natarajan et al. 2013] shows that risk minimization over the corrupted data is consistent with risk minimization over clean data provided that the standard convex loss (e.g., binary cross entropy) is modified appropriately to produce an unbiased loss estimator (ULE). Each of these loss functions is only presented for and validated on Type I and Type II noise.

2.2 Type III Methods

Literature on methods built for Type III noise is limited. [Cheng et al. 2017] propose a method catered towards instance and label-dependent noise. While broad in its definition of instance-dependent noise, the method necessitates manual labeling. [Menon et al. 2016] presents Isotron and demonstrates its validity under boundary-consistent instance-dependent label noise assumptions. In this paper, we do not adhere to the boundary-consistent noise model and focus on a broader set of feature-dependent label noise assumptions.

2.3 Theoretical Guarantees

[Blum et al. 1998; Blum and Mitchell 1998; Bylander 1994] offer guarantees for hypothesis generalization in the face of Type I and Type II noise. [Angluin and Laird 1988; Bylander 1997, 1998; Servedio 1999] present guarantees for Type III noise, where the probability of error depends on the distance to the margin. We introduce a gradient-based interpretation of label noise.

3 CLASSIFICATION WITH NOISY LABELS

3.1 Problem Statement

Let P denote the true distribution from which n training examples $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)$ have been drawn i.i.d., where $(\boldsymbol{x}_i, y_i) \in \mathbb{R}^d \times \{0, 1\}$. The clean and corrupted training datasets

are denoted as:

$$\mathcal{T} = \left\{(\boldsymbol{x}_i, y_i) \text{ for } i = 1, 2, \ldots, n\right\} \tag{1}$$

$$\tilde{\mathcal{T}} = \left\{(\boldsymbol{x}_i, \tilde{y}_i) \text{ for } i = 1, 2, \ldots, n\right\} \tag{2}$$

respectively, where due to some label noise process perturbs y_i to $\tilde{y}_i$. We have access to the noisy data $\tilde{\mathcal{T}}$ during training, but not to the clean data $\mathcal{T}$. It can be assumed that the corrupted samples $(\boldsymbol{x}_i, \tilde{y}_i)$ are drawn from distribution $\tilde{P}$ over the corrupted labels.

In supervised binary classification, we aim to learn a discriminator $f : \mathbb{R}^d \rightarrow \mathbb{R}$ that minimizes the *risk* with respect to a given loss function[1]. Formally, we want to minimize the empirical risk $\hat{R}\left[\ell, f, \mathcal{T}\right] = \frac{1}{n}\sum_{i=1}^{n} \ell\left(f(\boldsymbol{x}_i, \boldsymbol{\theta}), y_i\right)$, where the n, ℓ, $\boldsymbol{\theta}$ denote the training set size, loss function, and the model parameters respectively. Gradient based learning algorithms compute $\nabla_{\boldsymbol{\theta}}(\hat{R}\left[\ell, f, \mathcal{T}\right])$ and update model parameters. For example, as in mini-batch gradient descent,

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \nabla_{\boldsymbol{\theta}} \left(\hat{R}\left[\ell, f, \mathcal{T}\right]\right), \tag{3}$$

where η is the learning rate. In practice, we do not have access to the clean data $\mathcal{T}$. Instead we must learn the discriminator using the noisy data $\tilde{\mathcal{T}}$. Thus, the second term in Eq. 3 becomes $\nabla_{\boldsymbol{\theta}}(\hat{R}[\ell, f, \tilde{\mathcal{T}}])$, which denotes the risk with respect to the noisy training data.

3.2 Effect on Risk Minimization

In this section, we examine the effect of Type III noise on empirical risk. Specifically, we look at how the gradient of the empirical risk responds to Type III noise, as opposed to Types I and II.

Gradient-based optimization of risk is common across machine learning literature. Thus, we focus on the gradient of the empirical risk with respect to θ. We refer to the empirical risk over the clean training set as $\hat{R}\left[\ell, \Phi, \mathcal{T}\right]$.

$$\hat{R}\left[\ell, \Phi, \mathcal{T}\right] = \frac{1}{n}\sum_{i=1}^{n} \ell\left(\Phi(\mathbf{X}_i, \boldsymbol{\theta}), Y_i\right), \tag{4}$$

Similarly, we refer to the empirical risk over the corrupted training set as $\hat{R}\left[\ell, \Phi, \tilde{\mathcal{T}}\right]$. Below, we break the gradient of the empirical risk into two class-dependent terms and introduce short-hand to represent each.

$$\begin{aligned}
&\mathbb{E}_{(\mathbf{X}, Y)\sim P}\left[\nabla_{\boldsymbol{\theta}} \hat{R}[\ell, \mathcal{T}]\right] = \\
&\quad = \nabla_{\boldsymbol{\theta}} \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{(\mathbf{X}, Y)\sim P}\left[\ell\left(\Phi(\mathbf{X}_i, \boldsymbol{\theta}), Y_i\right)\right] \\
&\quad = \nabla_{\boldsymbol{\theta}} \left[\frac{1}{n}\sum_{i=1}^{n} \int p(\mathbf{x}_i, y_i)\ell\left(\Phi(\mathbf{x}_i, \boldsymbol{\theta}), y_i\right)\, d\mathbf{x}_i\, dy_i\right] \\
&\quad = p(y_i = 0)\nabla_{\boldsymbol{\theta}} \left[\frac{1}{n}\sum_{i=1}^{n} \int p(\mathbf{x}_i|0)\ell_0\left(\Phi(\mathbf{x}_i, \boldsymbol{\theta})\right)\, d\mathbf{x}_i\right] \\
&\quad + p(y_i = 1)\nabla_{\boldsymbol{\theta}} \left[\frac{1}{n}\sum_{i=1}^{n} \int p(\mathbf{x}_i|1)\ell_1\left(\Phi(\mathbf{x}_i, \boldsymbol{\theta})\right)\, d\mathbf{x}_i\right]
\end{aligned} \tag{5}$$

[1]In practice the loss function ℓ is usually *classification-calibrated* to minimize the 0-1 loss given that the sample size is sufficiently large [Bartlett et al. 2006].

We define shorthand based on Equation (5):

$$\boldsymbol{\beta}_0 = \nabla_{\boldsymbol{\theta}} \left[\frac{1}{n}\sum_{i=1}^{n} \int p(\mathbf{x}_i|y_i == 0)\ell_0\left(\Phi(\mathbf{x}_i, \boldsymbol{\theta})\right)\, d\mathbf{x}_i\right] \tag{6}$$

$$\boldsymbol{\beta}_1 = \nabla_{\boldsymbol{\theta}} \left[\frac{1}{n}\sum_{i=1}^{n} \int p(\mathbf{x}_i|1)\ell_1\left(\Phi(\mathbf{x}_i, \boldsymbol{\theta})\right)\, d\mathbf{x}_i\right] \tag{7}$$

β_0 represents the gradient of the empirical risk condition on class 0, while β_1 represents the same for class 1. Rewriting the original equation in terms of the above, we arrive at the following form for empirical risk:

$$\mathbb{E}_{(\mathbf{X}, Y)\sim P}\left[\nabla_{\boldsymbol{\theta}} \hat{R}[\ell, \mathcal{T}]\right] = p(y = 0)\boldsymbol{\beta}_0 + p(y = 1)\boldsymbol{\beta}_1 \tag{8}$$

We revisit this equation in the following sections to interpret empirical risk under varying noise models.

Types I & II: In Type I noise and Type II noise, $\boldsymbol{\beta}_0$ and $\boldsymbol{\beta}_1$ are the same in both the true empirical risk (based on the true label distribution (P)) and the noisy empirical risk (based on the corrupted label distribution ($\tilde{P}$) for Type I and Type II noise). This is because the only terms affected are $p(y = 0)$ and $p(y = 1)$. This lends credence to existing methods which cleverly re-weight examples to negate the effect that noisy values for $p(y = 0)$ and $p(y = 1)$ have on the resulting separator.

Perturbations to these coefficients may also be negated by increasing the amount of training data proportionately with the altered class percentages, as discussed by [Rolnick et al. 2017]. We can suppress the effects of random noise (Type I) or class conditional random noise (Type II) by increasing the number of training examples or by utilizing a robust loss function [Natarajan et al. 2013].

Type III: In contrast, Type III noise is substantially more difficult because it alters terms within $\boldsymbol{\beta}_0$ and $\boldsymbol{\beta}_1$: $p(x_i|y = 0)$ and $p(x_i|y = 1)$. This is hard to address because the available training data $\tilde{\mathcal{T}}$ can have optimal discriminators that are significantly different from the true discriminator, corresponding to the clean data $\mathcal{T}$. Existing methods to address Type III label noise approximate $p(y|x)$ to decide whether a give instance has been mislabelled. Instead, we focus on learning the generative model for $p(x_i|y = 1)$ by using an autoencoder. In Figure 2 we show how a gradient based approach performs when faced with Types I, II and III noise over three different datasets. We see that the gradient optimizer is more resilient to Types I and II than Type III across three datasets.

4 PROPOSED FRAMEWORK

Type III noise is particularly challenging because noise can depend on the input features in an arbitrarily complex way. Modeling this requires significant information about the noise process, which can be difficult to obtain in practice. In order to deal with this challenge we avoid training directly on the mislabeled instances. Instead, we start our training procedure by assuming there exists a small subset of $\tilde{\mathcal{T}}$ that is noise-free. This does *not* assume annotation of this subset, but merely its existence. Such a set may be obtained via annotation or inference - we test both scenarios below.

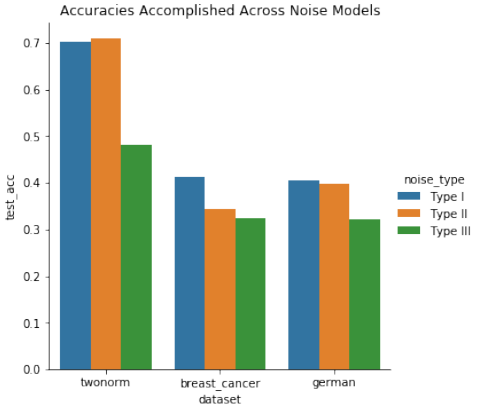


Figure 2: Accuracies (averaged over 5 runs) over three datasets perturbed by the three noise models: Type I noise (blue), Type II noise (orange) and Type III noise (green). We train a one-layer 10 unit neural network paired with a gradient-based optimizer for 500 epochs. We see that the model suffers most when faced with Type III noise.

Our objective is to correct mislabeled training instances such that the corrected data leads to the improved generalization of a trained model. We will demonstrate in Section 5 that our framework extends beyond Type I and Type II noise to Type III noise and performs well in correcting algorithmically mis-assigned labels. The approach has three steps:

(1) *Obtain known labels*: There are two ways of identifying instances that have correct labels. The first approach is to use domain knowledge and subsequently use this clean subset for Step 2. More often than not, however, this clean set is unknown or too small. In this scenario, we can infer which examples are likely to be correctly labeled, as is done in [Ding et al. 2018]. We report results on both approaches in Section 5.

(2) *Train semi-supervised model*: The framework uses the filtered data from the previous step to train a model that learns to classify instances based on similarity between features. In contrast with supervised learning, where the goal is to learn $p(y|\boldsymbol{x})$, we want to learn which features are most representative of a given class: i.e., $p(\boldsymbol{x}|y)$. This ties back to the term that is perturbed by feature-dependent label noise in Equation 8. Thus, we train a generative model (e.g. an autoencoder) to identify which instances belong to a class based on a certain function that maps $p(\boldsymbol{x}|y)$ to a decision value. In line with [Zhao et al. 2016], we refer to this as an energy function and the resulting outputs as energy values. We aim for instances with similar features to achieve similar energy values.

(3) *Identify and correct mislabeled instances*: The features of the full training data are fed through the semi-supervised model resulting in each instance being assigned an energy value based on the output of the model. The energy corresponding to each instance serves as a proxy for class assignment (i.e., low energy corresponds to $y = 0$ whereas high energy corresponds to $y = 1$). Contradictions between energy assignment and training labels are used to correct the training data such that all labels are compatible with their assigned energy.

We first elaborate on our implementation of this framework and subsequently motivate the design decisions involved. Namely, we explore our choice of contrastive divergence as a loss function and empirically justify our use of an autoencoder.

4.1 Model

In Step 1, we infer which examples are to be trusted using a logistic regression model and identifying trustworthy examples for of each class. In our experiments, we use the top 10% of examples from each class as determined by the logistic regression output. Although a fixed parameter, we opt for 10% for two reasons: 1) 10% offers a conservative estimate for the number of trustworthy examples in a training set and 2) We demonstrate resilience to this parameter choice through experiments across a range of datasets.

We implement Step 2 of our framework by using an energy-based autoencoder (similar to [Zhao et al. 2016]). We train the autoencoder on the trustworthy subset of the training data, as identified by Step 1, using a contrastive loss:

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{|T_0|} \sum_{\boldsymbol{x}_0 \in T_0} \|\boldsymbol{\theta}(\boldsymbol{x}) - \boldsymbol{x}\|^2 - \frac{1}{|T_1|} \sum_{\boldsymbol{x}_1 \in T_1} \|\boldsymbol{\theta}(\boldsymbol{x}) - \boldsymbol{x}\|^2, \quad (9)$$

This means that the model minimizes reconstruction loss for class 0 and maximizes reconstruction loss for class 1. In order to ensure stable training and to prevent the second term of the contrastive loss from diverging, we balance our mini-batch to have an equal number of positive and negative class samples. Optimizing the contrastive loss function directly optimizes for class separation when comparing reconstruction losses.

The Step 3 implementation involves two parameters, a and b. We first compute the average energy within each class - $\bar{E}_0$ and $\bar{E}_1$. All examples within a of $\bar{E}_1$ are relabeled to class 1. Similarly, all examples within b of $\bar{E}_0$ are relabeled to class 0. Thus, a and b are tuneable hyperparameters that determine how aggressively we want to change $0 \rightarrow 1$ and $1 \rightarrow 0$, respectively. The rationale is to modify samples where the original label $\tilde{y}$ contradicts the energy assignment.

4.2 Motivation

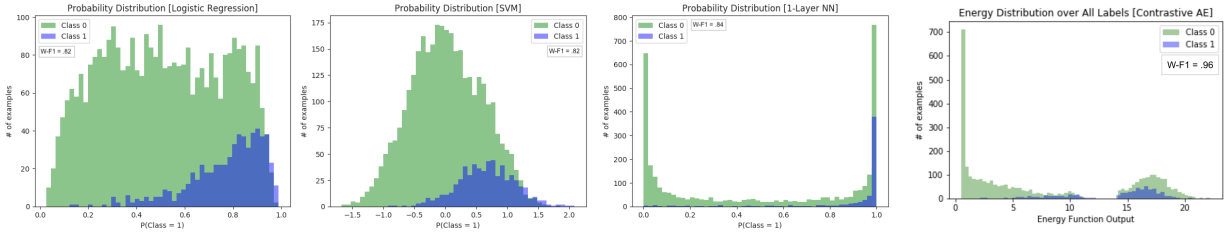
Within this framework, we make two key choices: the contrastive loss function and the autoencoder model.

Contrastive Loss In our work, we approximate contrastive divergence with a contrastive energy-based loss. Contrastive divergence can be described as:

$$\mathrm{CD}(\boldsymbol{\theta}) = \mathrm{KL}\left[p_0(\boldsymbol{x}) \| q(\boldsymbol{x};\boldsymbol{\theta})\right] - \mathrm{KL}\left[p_1(\boldsymbol{x}) \| q(\boldsymbol{x};\boldsymbol{\theta})\right]. \quad (10)$$

where $p_0(\boldsymbol{x})$ and $p_1(\boldsymbol{x})$ denote the target probability distributions conditioned on class 0 and 1 respectively and $q(\boldsymbol{x};\theta)$ denotes the learned model. The contrastive loss we use to approximate this divergence is:

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{|T_0|} \sum_{\boldsymbol{x}_0 \in T_0} E(\boldsymbol{x};\boldsymbol{\theta}) - \frac{1}{|T_1|} \sum_{\boldsymbol{x}_1 \in T_1} E(\boldsymbol{x};\boldsymbol{\theta}), \quad (11)$$



(a) Logistic Regression **(b) SVM** **(c) FCNN** **(c) Autoencoder**

Figure 3: Distributions achieved using different loss functions. We are looking for two traits in this distribution: 1) Distinct clusters of energies and 2) distinctly mislabeled instances within each cluster. In other words, we want clusters with large separation that contain both blue and green examples, such that blue examples in the green cluster are evidently mislabeled and vice versa.

where the first term refers to the average energy ($E(\boldsymbol{x};\boldsymbol{\theta})$) over class 0 and the second term refers to the average energy over class 1. As described in section 2.2 of [LeCun et al. 2006], the negative log-likelihood function, $E(\boldsymbol{x};\boldsymbol{\theta}) = -\log q(\boldsymbol{x};\boldsymbol{\theta})$, constitutes a valid energy which has been used in numerous energy-based models [e.g. [Bengio et al. 2003; Berthelot et al. 2017; Zhao et al. 2016]]. The contrastive loss has been applied successfully in adversarial training, resulting in faster and improved stability of training, enhanced generator quality, and improved generator diversity [Berthelot et al. 2017; Zhao et al. 2016].

Autoencoder The obvious approach to separate data into groups is a binary classifier. However, as discussed in [Berthelot et al. 2017], binary classification provides a relatively weak training signal that largely ignores the intricacies of the input feature distribution. Further, traditional unsupervised training regimes typically use a maximum likelihood formulation (i.e., forward KL divergence) which is prone to distributing probability mass broadly (the so-called "mean-seeking behaviour" [Murphy 2012]) and further limits the discriminative ability of the learned model. This autoencoder choice is further supported by its defined ability to learn an energy manifold in the absence of negative examples [Zhao et al. 2016]. As a result, an autoencoder can still learn an informative manifold from a heavily imbalanced dataset.

We empirically evaluate our use of an autoencoder to generate this energy distribution in Figure 3 by comparing its class separation to standard binary classifiers. We have two aims with respect to the energy distribution: 1) We would like to see two distinct clusters - a low energy cluster and a higher energy cluster - and 2) within these clusters, it should be clear which examples have been mislabeled. The low energy cluster corresponds to one class, while the high energy cluster corresponds to all other classes. We see that using the autoencoder's reconstruction loss results in increased separation between the two clusters. Due to the distinct space between clusters that fails to appear with the logistic regression, SVM, and FCNN, we are able to identify a small mass of blue examples towards the left to relabel green and a small mass of green examples to the right to relabel blue.

5 EXPERIMENTS

We conduct experiments on both simulated and real-world data. Below, we detail three experiments that validate the proposed framework across commonly used benchmarks, images, and algorithmically assigned labels. We report the results for our autoencoder architecture under *AE (Learned)* and *AE (Known)*. In *AE (Learned)*, trustworthy examples are inferred from the data in Step 1. In *AE (Known)*, however, the framework is equipped with knowledge on which examples to trust. We explore both for fair comparison to methods based on differing assumptions.

5.1 UCI Benchmarks

We measure the proposed framework's performance using the classification accuracy of logistic regression model trained on the corrected data set. The following noise robust algorithms serve as our baselines: NHERD [Crammer and Lee 2010], PAM [Frénay and Verleysen 2014], and ULE [Natarajan et al. 2013]. We use the accuracy of logistic regression trained on the noisy data (LR-N) as a lower-bound baseline for performance. We also run logistic regression on the clean data (LR-C) to demonstrate the best case scenario. Below, we detail the datasets and noise models we consider. Results from these experiments are found in Tables 2, 3, and 4.

Datasets Literature on label noise correction frequently relies on synthetically modified datasets. This is because $\tilde{y}$ and y are rarely available. We test on five commonly used UCI datasets, as described in Table 1. Lin-Sep and TwoNorm are contain synthetic examples, while the remaining four (Diabetes, German, Image and BreastCancer) consist of real examples with synthetically flipped labels. We elaborate on the process by which the labels are flipped in the following section.

Noise Models We control the amount of noise added with α and add noise based on a certain feature $x[i]$. We select $x[i]$ if it has sufficient variance to make the noise model distinct from random noise. This is important because adding label noise dependent on a feature that is more or less constant across classes would have no effect. More concretely, we experiment over the following Type III

Table 1: UCI Datasets

Name	# Features	# Instances
Lin-Sep	2	10000
TwoNorm	20	7400
Diabetes	8	768
German	20	700
Image	18	2086
BreastCancer	9	267

noise models:

- *Linear noise* ($p_{\text{error}} \sim \alpha x_i$): the probability of an error occurring depends linearly on a feature (Table 2).
- *Quadratic noise* ($p_{\text{error}} \sim \alpha x_i^2$): the probability of an error occurring depends quadratically on the feature value (Table 3).
- *Boundary noise*: the probability of error depends on the distance from the class boundary, as determined by the noise-free data (Table 4).

Results Tables 2, 3, and 4 demonstrate that our method outperforms existing methods on both real and synthetic datasets. The average rank of our method is better than that of existing methods under each of the feature-dependent label noise models.

The binary classification tasks across datasets range in difficulty - even with the entirely clean BreastCancer dataset, logistic regression achieves an F1 score of only .70. Furthermore, AE's performance is consistent as evidenced by the low standard deviation across runs. This is on par with existing methods, which also demonstrate high consistency.

In the linearly dependent noise setting, AE ranks second to NHERD on both the German and BreastCancer dataset. We hypothesize that this is a result of dataset size because AE assumes 10% of the data is clean. This results in an extremely small training set of 100 examples for both the German and BreastCancer datasets.

This trend continues with the quadratically dependent noise model. AE places below NHERD and PAM for both Diabetes and German, two of the smaller datasets.

AE outperforms competing methods with increasing margins as the noise models become more complex. This is best demonstrated by average ranks. To calculate average rank, we order all methods by the F1 score for each dataset. This produces a method rank per dataset, which we then average across the datasets. With linearly dependent label noise, the framework achieves a rank of 1.33, with the next highest rank being NHERD at a rank of 2. In the quadratically dependent noise experiment, our framework achieves a rank of 1.5, with the next highest rank at 3.17. Finally, boundary dependence results in a rank of 1 with NHERD following at 2.83.

Testing across different feature-dependent noise models allows us to identify the strengths and weaknesses of not only our framework, but also existing methods. NHERD performs particularly well with linearly dependent label noise, but performs poorly under quadratic and boundary-dependent label noise models.

5.2 MNIST

In this experiment, we validate the framework by testing the label noise correction process in images. Additionally, we compare to a method that relies on a clean subset of the data to learn a re-weighting scheme [Ren et al. 2018]. We demonstrate that our method generalizes better in the presence of Type III noise, even with only 1% of the data annotated as clean. We report the following accuracies: the performance of the standard LeNet architecture on the dataset corrected by our method (Ours), the noisy dataset (SLN), and the learned re-weighting model (LRW) in Figure 4.

Dataset We equip both methods with the same percentage of clean data for each noise setting. The task is to distinguish "3" (class 0) from the other digits (class 1) using the MNIST dataset, which consists of 60,000 28x28 black and white images. We test method performance with 1, 5, and 10% of the original dataset annotated as clean.

Noise Model We use a different noise model with images because introducing label noise dependent on one feature - in this case, a single pixel value - is unrealistic. Instead, we use α to designate how aggressively the labels for "4", "5", and "6" examples are flipped to label "3". Thus, label noise is implicitly input dependent based on features shared by these digit classes.

Results The framework outperforms [Ren et al. 2018] when equipped with as little as 1% of the original data. Furthermore, increases in noise incur a lower loss in accuracy than that of the competing method. Comparing the results in which $\alpha = .1$ to the experiment where $\alpha = .8$ highlights our method's resilience to Type III noise. This increase in noise rate corresponds to a 0.2% decrease in F1 score for our framework and a 2.5% decrease in F1 score for [Ren et al. 2018].

5.3 Algorithmically Assigned Labels

Algorithmically-assigned labels (AALs) are prevalent in domains with abundant unlabeled data and high labeling costs. Common applications include web page annotation, but the value of this approach extends to automatic annotation of images and natural language. As demonstrated in Table 6, the proposed technique achieves a higher F1 score correcting the AALs than competing methods. The success of the framework in this setting suggests its robustness to feature-dependent noise.

Dataset We use the MIT-BIH Arrhythmia dataset [Goldberger et al. 2000] to evaluate the proposed method's ability to correct algorithmically assigned labels. We employ the data preprocessing procedure described by [Mondéjar-Guerra et al. 2019], where 59 features represent each arrhythmia. The dataset consists of 50,000 instances. We divide the dataset into three parts: T_1, T_2 and T_3. T_1 and T_2 serve as training sets and T_3 is the test set. We train a classifier on T_1 and subsequently use that classifier to generate AALs for T_2. Thus, the predictions of the trained classifier become $\tilde{y}$ and the original expert labels remain y.

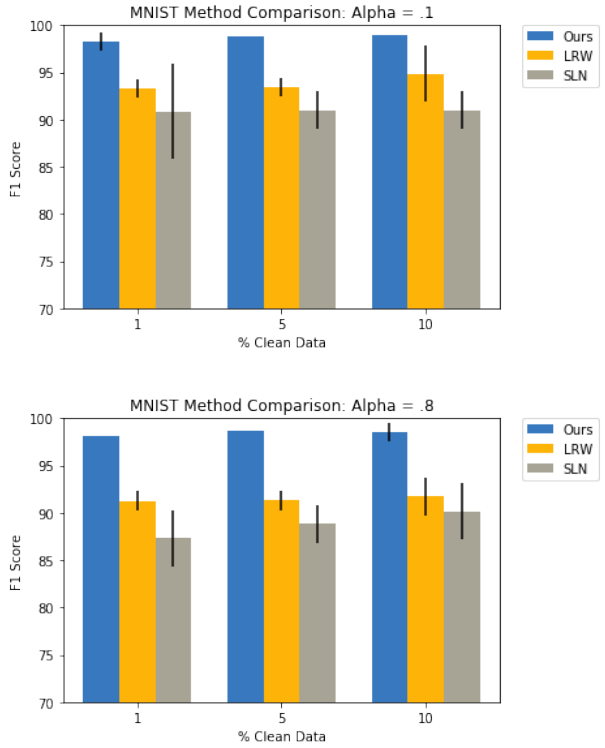


Figure 4: Performance on label noise correction task when given an annotated clean subset of the training data. We test across three sizes (1%, 5% and 10%) for this clean subset. Moreover, we test across both low feature-dependent label noise (top) and high feature-dependent label noise (bottom), as parameterized by α, where α dictates the probability that instances of "4", "5" or "6" are flipped to label "3". F1 scores for our method (blue), SRW (gold) and SLN (gray) are included. While both LRW and our method outperform the baseline, our method outperforms LRW in all three settings.

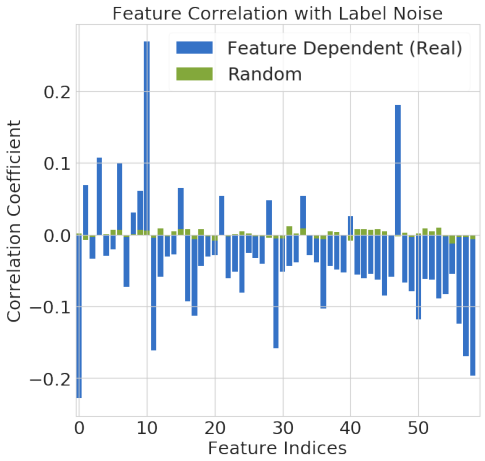


Figure 5: Feature-dependence of label noise in AAL dataset. We compare the correlation of label noise introduced by the T_1 trained classifier to the correlation of each feature to random label noise. It is evident that several features demonstrate higher correlation to an incorrect label.

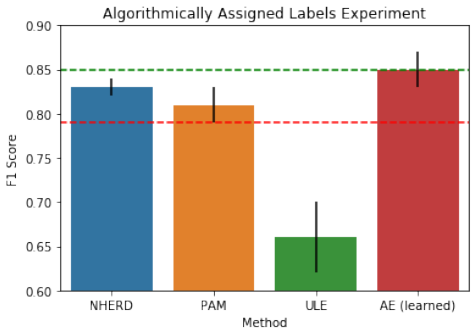


Figure 6: Performance of label noise methods on algorithmically assigned labels. The red dashed line indicates the accuracy of regularized logistic regression trained on the algorithmically assigned labels. The green dashed line denotes the accuracy of logistic regression trained on the entirely correct labels. We see that ULE performs worse than training a simple model on the noisy labels, while the AE model matches the performance of logistic regression trained on the clean labels.

Noise Model It is important to show that the noise introduced by the labels supplied by the classifier trained on T_1 is truly feature dependent. In Figure 5, we visualize the correlation of features with incorrect labels. In green, we show this correlation in the random noise scenario. In blue, we show the same in the AAL scenario. Although there is no concise description of this noise model, it is a good example of the way mistakes are made by algorithmic annotation.

Results Our method outperforms all existing methods in learning from our algorithmically assigned labels, as shown in Figure 6. Surprisingly, the proposed approach achieves the same accuracy as a model trained on the clean labels. ULE ultimately performs worse than logistic regression trained with the noisy labels. Note that the performance ranks in the face of algorithmically assigned labels mirror ranks on synthetically perturbed datasets. The success of our framework in this setting suggests its superior robustness to feature-dependent noise.

6 DISCUSSION

In this paper, we propose an energy-based framework to correct mislabeled training instances. The framework is designed to address feature-dependent label noise. We explore the performance of existing methods under a variety of feature-dependent label noise assumptions.

Table 2: Linearly feature-dependent label noise

Dataset	Noise Parameters (col, α)	LR-N	NHERD	PAM	ULE	AE (learned)	LR-C
Lin-Sep	**1, 1.2**	0.86 ± 0.03	0.91 ± 0.01	0.76 ± 0.01	0.90 ± 0.02	**0.94 ± 0.01**	0.96 ± 0.01
Diabetes	5 , 1.0	0.60 ± 0.03	0.50 ± 0.01	0.48 ± 0.03	0.60 ± 0.03	**0.64±0.01**	0.72 ± 0.02
German	1 , 1.2	0.67 ± 0.03	**0.72 ± 0.01**	0.49 ± 0.03	0.63 ± 0.01	0.67 ± 0.03	0.76 ± 0.02
Image	1 , 0.7	0.61 ± 0.03	0.63 ± 0.01	0.54 ± 0.01	0.61 ± 0.02	**0.67±0.01**	0.77 ± 0.01
Twonorm	1 , 1.2	0.68 ± 0.04	0.50 ± 0.02	0.43 ± 0.03	0.82 ± 0.04	**0.88±0.01**	0.98 ± 0.01
Breast Cancer	5 , 1.0	0.66 ± 0.02	**0.67 ±0.02**	0.52 ± 0.03	0.60 ± 0.03	0.60 ± 0.02	0.70 ± 0.01

Table 3: Quadratically feature-dependent label noise

Dataset	Noise Parameters (col, α)	LR-N	NHERD	PAM	ULE	AE (learned)	LR-C
Lin-Sep	1, 1.2	0.40 ± 0.01	0.53 ± 0.04	0.62 ± 0.03	0.61 ± 0.02	**0.93±0.01**	0.96 ± 0.01
Diabetes	5 , 1.2	0.62 ± 0.01	0.59 ± 0.01	0.67 ± 0.01	**0.71±0.01**	0.66 ± 0.03	0.72 ± 0.02
German	1 , 1.2	0.60 ± 0.02	**0.72± 0.01**	0.60 ± 0.01	0.67 ± 0.01	0.68 ± 0.03	0.76 ± 0.02
Image	1 , 0.7	0.55 ± 0.02	0.64 ± 0.01	0.57 ± 0.01	0.60 ± 0.03	**0.74±0.01**	0.77 ± 0.01
Twonorm	1 , 1.2	0.90 ± 0.03	0.55 ± 0.02	0.86 ± 0.01	0.94 ± 0.01	**0.96±0.01**	0.98 ± 0.01
Breast Cancer	5 , 1.0	0.60 ± 0.02	0.60 ± 0.02	0.63 ± 0.02	0.63 ± 0.03	**0.65±0.03**	0.70 ± 0.01

Table 4: Boundary-based feature-dependent label noise

Dataset	Noise Parameters (α)	LR-N	NHERD	PAM	ULE	AE (learned)	LR-C
Lin-Sep	0.7	0.39 ± 0.01	0.41 ± 0.01	0.53 ± 0.02	0.85 ± 0.01	**0.94±0.01**	0.96 ± 0.01
Diabetes	0.7	0.56 ± 0.01	0.53 ± 0.03	0.50 ± 0.02	0.55 ± 0.02	**0.68±0.02**	0.72 ± 0.02
German	0.7	0.57 ± 0.01	0.70 ± 0.01	0.47 ± 0.01	0.60 ± 0.01	**0.72±0.01**	0.76 ± 0.02
Image	0.7	0.43 ± 0.01	0.61 ± 0.01	0.45 ± 0.02	0.43 ± 0.01	**0.74±0.03**	0.77 ± 0.01
Twonorm	0.5	0.52 ± 0.03	0.51 ± 0.03	0.52 ± 0.03	0.51 ± 0.04	**0.88±0.03**	0.98 ± 0.01
Breast Cancer	0.7	0.62 ± 0.02	0.62 ± 0.02	0.46 ± 0.02	0.63 ± 0.02	**0.66±0.04**	0.70 ± 0.01

We report the class weighted mean f1 score on the noise-free test set along with the standard error over 5 runs. In Tables 3 and 4, *col* indicates the feature upon which noise depends on and α signifies the noise rate. In Table 5, label noise depends linearly on an instance's distance from the class boundary multiplied by α.

We validate a natural implementation of this framework using an autoencoder and a contrastive loss function. Together, the autoencoder and the contrastive loss produce an energy function that distinguishes classes based on feature-level information. Importantly, the contrastive loss optimizes the separation of energies for each class. This ultimately allows the method to identify mislabeled instances based on their energy output. We evaluate the proposed method across six datasets, varied in both difficulty and dimension, and find that the proposed method outperforms existing work. Furthermore, we explore three feature-dependent label noise models to demonstrate the method's empirical value under varying assumptions. In each feature-dependent label noise scenario, the proposed method outperforms existing work. The method is further validated on the weakly supervised label correction task, in which a small, clean, annotated subset of the training data is available. Using only 1% of MNIST, the method recovers from a scenario in which 9600 labels - 80% of the instances labeled "4", "5" and "6" - are flipped to "3"s.

Feature-dependent label noise is prevalent across applications, including bias and algorithmically assigned labels. The explosion of available data is accompanied by an explosion of flawed annotation techniques. This necessitates solutions catered towards Type III noise and the proposed method addresses this niche. Future directions include validation on the multi-class label correction problem and a rigorous exploration of alternate feature-dependent label noise models.

REFERENCES

Dana Angluin and Philip Laird. 1988. Learning from noisy examples. *Machine Learning* 2, 4 (1988), 343–370.

Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. 2006. Convexity, classification, and risk bounds. *J. Amer. Statist. Assoc.* 101, 473 (2006), 138–156.

Eyal Beigman and Beata Beigman Klebanov. 2009. Learning with Annotation Noise. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1 (ACL '09)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 280–287. http://dl.acm.org/citation.cfm?id=1687878.1687919

Beata Beigman Klebanov and Eyal Beigman. 2009. From Annotator Agreement to Noise Models. *Comput. Linguist.* 35, 4 (Dec. 2009), 495–503. https://doi.org/10.1162/coli.2009.35.4.35402

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research* 3, Feb (2003), 1137–1155.

David Berthelot, Thomas Schumm, and Luke Metz. 2017. BEGAN: boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717* (2017).

A. Blum, A. Frieze, R. Kannan, and S. Vempala. 1998. A Polynomial-Time Algorithm for Learning Noisy Linear Threshold Functions. *Algorithmica* 22, 1 (01 Sep 1998), 35–52. https://doi.org/10.1007/PL00013833

Avrim Blum and Tom Mitchell. 1998. Combining Labeled and Unlabeled Data with Co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory (COLT' 98)*. ACM, New York, NY, USA, 92–100. https://doi.org/10.1145/279943.279962

Carla E. Brodley and Mark A. Friedl. 1999. Identifying Mislabeled Training Data. *J. Artif. Int. Res.* 11, 1 (July 1999), 131–167. http://dl.acm.org/citation.cfm?id=3013545.3013548

Tom Bylander. 1994. Learning Linear Threshold Functions in the Presence of Classification Noise. In *In Proceedings of the Seventh Annual Workshop on Computational Learning Theory*. ACM Press, 340–347.

Tom Bylander. 1997. Learning Probabilistically Consistent Linear Threshold Functions. In *Proceedings of the Tenth Annual Conference on Computational Learning Theory (COLT '97)*. ACM, New York, NY, USA, 62–71. https://doi.org/10.1145/267460.267479

Tom Bylander. 1998. Learning noisy linear threshold functions. *Submitted for journal publication* (1998).

Jiacheng Cheng, Tongliang Liu, Kotagiri Ramamohanarao, and Dacheng Tao. 2017. Learning with Bounded Instance-and Label-dependent Label Noise. *arXiv preprint arXiv:1709.03768* (2017).

Raj S. Chhikara and Jim McKeon. 1984. Linear Discriminant Analysis with Misallocation in Training Samples. *J. Amer. Statist. Assoc.* 79, 388 (1984), 899–906. http://www.jstor.org/stable/2288722

E. Cohen. 1997. Learning noisy perceptrons by a perceptron in polynomial time. In *Proceedings 38th Annual Symposium on Foundations of Computer Science*. 514–523. https://doi.org/10.1109/SFCS.1997.646140

Koby Crammer and Daniel D Lee. 2010. Learning via gaussian herding. In *Advances in neural information processing systems*. 451–459.

Thierry Denoeux. 1995. A k-nearest neighbor classification rule based on Dempster-Shafer theory. *IEEE Transactions on Systems, Man, and Cybernetics* 25, 5 (May 1995), 804–813. https://doi.org/10.1109/21.376493

Thierry Denoeux. 1997. Analysis of Evidence-theoretic Decision Rules for Pattern Classification. *Pattern Recogn.* 30, 7 (July 1997), 1095–1107. https://doi.org/10.1016/S0031-3203(96)00137-9

Thierry Denoeux. 2000. A neural network classifier based on Dempster-Shafer theory. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 30, 2 (Mar 2000), 131–150. https://doi.org/10.1109/3468.833094

Yifan Ding, Liqiang Wang, Deliang Fan, and Boqing Gong. 2018. A Semi-Supervised Two-Stage Approach to Learning from Noisy Labels. *arXiv preprint arXiv:1802.02679* (2018).

B. Frénay and M. Verleysen. 2014. Classification in the Presence of Label Noise: A Survey. *IEEE Transactions on Neural Networks and Learning Systems* 25, 5 (May 2014), 845–869. https://doi.org/10.1109/TNNLS.2013.2292894

Benoît Frénay and Michel Verleysen. 2014. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems* 25, 5 (2014), 845–869.

Aritra Ghosh, Naresh Manwani, and P.S. Sastry. 2015. Making risk minimization tolerant to label noise. *Neurocomputing* 160 (2015), 93 – 107. https://doi.org/10.1016/j.neucom.2014.09.081

Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. 2000. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. *Circulation* 101, 23 (2000), e215–e220.

Zaven S. Khachaturian. 1985. Diagnosis of Alzheimer's disease. *Archives of Neurology* 42, 11 (1985), 1097–1105. https://doi.org/10.1001/archneur.1985.04060100083029 arXiv:/data/journals/neur/16038/archneur$_4$2$_1$1$_0$29.*pdf*

Aleksander Kolcz and Gordon V. Cormack. 2009. Genre-based Decomposition of Email Class Noise. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '09)*. ACM, New York, NY, USA, 427–436. https://doi.org/10.1145/1557019.1557070

Peter A. Lachenbruch. 1966. Discriminant Analysis When the Initial Samples Are Misclassified. *Technometrics* 8, 4 (1966), 657–662. http://www.jstor.org/stable/1266637

Peter A. Lachenbruch. 1974. Discriminant Analysis When the Initial Samples Are Misclassified II: Non-Random Misclassification Models. *Technometrics* 16, 3 (1974), 419–424. http://www.jstor.org/stable/1267672

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. 2006. A tutorial on energy-based learning. *Predicting structured data* 1, 0 (2006).

Tongliang Liu and Dacheng Tao. 2014. Classification with Noisy Labels by Importance Reweighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38 (11 2014).

Philip M. Long and Rocco A. Servedio. 2010. Random classification noise defeats all convex potential boosters. *Machine Learning* 78, 3 (01 Mar 2010), 287–304. https://doi.org/10.1007/s10994-009-5165-z

Aditya Krishna Menon, Brendan Van Rooyen, and Nagarajan Natarajan. 2016. Learning from binary labels with instance-dependent corruption. *arXiv preprint arXiv:1605.00751* (2016).

V Mondéjar-Guerra, J Novo, J Rouco, MG Penedo, and M Ortega. 2019. Heartbeat classification fusing temporal and morphological information of ECGs via ensemble of classifiers. *Biomedical Signal Processing and Control* 47 (2019), 41–48.

Kevin P Murphy. 2012. *Machine learning: a probabilistic perspective*. The MIT Press, Cambridge, MA.

Melissa E. Murray, Adel Aziz, Owen A. Ross, Ranjan Duara, Dennis W. Dickson, and Neill R. Graff-Radford. 2016. Alzheimer's disease may not be more common in women; men may be more commonly misdiagnosed. *Alzheimer's & Dementia: The Journal of the Alzheimer's Association* 12, 7 (2016), 1097–1105. https://doi.org/doi:10.1016/j.jalz.2016.06.527

Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. 2013. Learning with noisy labels. In *Advances in neural information processing systems*. 1196–1204.

C. G. Northcutt, T. Wu, and I. L. Chuang. 2017. Learning with Confident Examples: Rank Pruning for Robust Classification with Noisy Labels. In *Uncertainty in Artificial Intelligence*. arXiv:stat.ML/1705.01936

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. 2017. Making deep neural networks robust to label noise: A loss correction approach. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.(CVPR)*. 2233–2241.

Umaa Rebbapragada and Carla Brodley. 2007a. Class Noise Mitigation Through Instance Weighting. In *Machine Learning: ECML 2007: 18th European Conference on Machine Learning*. 708–715.

Umaa Rebbapragada and Carla E Brodley. 2007b. Class noise mitigation through instance weighting. In *European Conference on Machine Learning*. Springer, 708–715.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. 2018. Learning to Reweight Examples for Robust Deep Learning. *CoRR* abs/1803.09050 (2018). arXiv:1803.09050 http://arxiv.org/abs/1803.09050

David Rolnick, Andreas Veit, Serge Belongie, and Nir Shavit. 2017. Deep learning is robust to massive label noise. *arXiv preprint arXiv:1705.10694* (2017).

Brendan van Rooyen, Aditya Krishna Menon, and Robert C. Williamson. 2015. Learning with Symmetric Label Noise: The Importance of Being Unhinged. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1 (NIPS'15)*. MIT Press, Cambridge, MA, USA, 10–18. http://dl.acm.org/citation.cfm?id=2969239.2969241

Joseph L. Schafer and John W. Graham. 2002. Missing data: our view of the state of the art. *Psychological Methods* 7 2 (2002), 147–77.

Rocco A Servedio. 1999. On PAC learning using Winnow, Perceptron, and a Perceptron-like algorithm. In *Proceedings of the twelfth annual conference on Computational learning theory*. ACM, 296–307.

Padhraic Smyth. 1996. Bounds on the mean classification error rate of multiple experts. *Pattern Recognition Letters* 17, 12 (1996), 1253 – 1257. https://doi.org/10.1016/0167-8655(96)00105-5

Sainbayar Sukhbaatar, Joan Bruna, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. 2014. Training convolutional networks with noisy labels. *arXiv preprint arXiv:1406.2080* (2014).

Jiang-wen Sun, Feng-ying Zhao, Chong-jun Wang, and Shi-fu Chen. 2007. Identifying and correcting mislabeled training instances. In *Future generation communication and networking (FGCN 2007)*, Vol. 1. IEEE, 244–250.

Junbo Zhao, Michael Mathieu, and Yann LeCun. 2016. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126* (2016).

Xingquan Zhu and Xindong Wu. 2004. Class Noise vs. Attribute Noise: A Quantitative Study. *Artificial Intelligence Review* 22, 3 (01 Nov 2004), 177–210. https://doi.org/10.1007/s10462-004-0751-8