Honda Research Institute EU
Honda Research Institute Europe GmbH
https://www.honda-ri.de/

Estimating the electrical power output of industrial devices with end-to-end time-series classification in the presence of label noise

Andrea Castellani, Sebastian Schmitt, Barbara Hammer

2021

Preprint:

This is a post-peer-review, pre-copyedit version of an article published in Machine Learning and Knowledge Discovery in Databases. Research Track. The final authenticated version is available online at: https://doi.org/10.1007/978-3-030-86486-6_29

Estimating the electrical power output of industrial devices with end-to-end time-series classification in the presence of label noise

Andrea Castellani (✉)[1][0000−0003−0476−5978],
Sebastian Schmitt[2][0000−0001−7130−5483], and
Barbara Hammer[1][0000−0002−0935−5591]

[1] Bielefeld University,
{acastellani,bhammer}@techfak.uni-bielefeld.de
[2] Honda Research Institute Europe,
sebastian.schmitt@honda-ri.de

Abstract. In complex industrial settings, it is common practice to monitor the operation of machines in order to detect undesired states, adjust maintenance schedules, optimize system performance or collect usage statistics of individual machines. In this work, we focus on estimating the power output of a Combined Heat and Power (CHP) machine of a medium-sized company facility by analyzing the total facility power consumption. We formulate the problem as a time-series classification problem, where the class label represents the CHP power output. As the facility is fully instrumented and sensor measurements from the CHP are available, we generate the training labels in an automated fashion from the CHP sensor readings. However, sensor failures result in mislabeled training data samples which are hard to detect and remove from the dataset. Therefore, we propose a novel multi-task deep learning approach that jointly trains a classifier and an autoencoder with a shared embedding representation. The proposed approach targets to gradually correct the mislabelled data samples during training in a self-supervised fashion, without any prior assumption on the amount of label noise. We benchmark our approach on several time-series classification datasets and find it to be comparable and sometimes better than state-of-the-art methods. On the real-world use-case of predicting the CHP power output, we thoroughly evaluate the architectural design choices and show that the final architecture considerably increases the robustness of the learning process and consistently beats other recent state-of-the-art algorithms in the presence of unstructured as well as structured label noise.

Keywords: Time-series · Deep Learning · Label noise · Self-supervision · Non-Intrusive Load Monitoring · Time-Series Classification

1 Introduction

It is common to monitor multiple machines in complex industrial settings for many diverse reasons, such as to detect undesired operational states, adjust

maintenance schedules or optimize system performance. In situations where the installation of many sensors for individual devices is not feasible due to cost or technical reasons, Non-Intrusive Load Monitoring (NILM) [18] is able to identify the utilization of individual machines based on the analysis of cumulative electrical load profiles. The problem of the generation of labelled training data sets is a cornerstone of data-driven approaches to NILM. In this context, industry relies mostly on manually annotated data [11] and less often the training labels can be automatically generated from the sensors [13], which is often unreliable because of sensor failure and human misinterpretation. Data cleaning techniques are often hard to implement [42] which unavoidably leads to the presence of wrongly annotated instances in automatically generated datasets i.e. *label noise* [12]. Many machine learning methods, and in particular deep neural networks, are able to overfit training data with noisy labels [48], thus it is challenging to apply data-driven approaches successfully in complex industrial settings.

We consider a medium-sized company facility and target the problem of estimating the electrical power output of a Combined Heat and Power (CHP) machine by only analyzing the facility electrical power consumption. The electrical power output of the CHP is sufficient to supply a substantial share of the total electricity demand of the facility. Therefore, knowing the CHP's electrical power output is very helpful for distributing the electrical energy in the facility, for example when scheduling the charging of electrical vehicles (EVs) or reducing total peak-load [28]. We propose a data-driven deep learning-based approach to this problem, which is modelled as time-series classification challenge in the presence of label noise, where the class label of each time series represents the estimated CHP power output level. As the facility is fully instrumented, and sensor measurements from the CHP are available, we generate the training labels in an automated fashion from the CHP sensor readings. However, these sensors fail from time to time which resulting wrong labels. To tackle this problem, we propose a novel multi-task deep learning approach named Self-Re-Labeling with Embedding Analysis (SREA), which targets the detection and re-labeling of wrongly labeled instances in a self-supervised training fashion.

In the following, after a formal introduction to SREA, we empirically validate it with several benchmarks data sets for time-series classification and compare against state-of-the-art (SotA) algorithms. In order to evaluate the performance, we create a training data set from clean sensor readings and we corrupt it by introducing three types of artificial noise in a controlled fashion. After that, we apply the proposed method to a real-world use-case including real sensor failures and show that those are properly detected and corrected by our algorithm. Finally, we perform an extensive ablation study to investigate the sensitivity of the SREA to its hyper-parameters.

2 Related work

Deep learning based techniques constitute a promising approach to solve the NILM problem [30, 34]. Since the requirement of large annotated data sets is

very challenging, the problem is often addressed as a semi-supervised learning task [20, 4, 46], where only a part of the data is correctly labeled, and the other is left without any label. Here we take a different stance to the problem of energy disaggregation: we assume that labels for all data are given, but not all labels are correct, as is often the case in complex sensor networks [12].

Learning noisy labels is an important topic in machine learning research [38, 16]. Several approaches are based on the fact that deep neural networks tend to first learn clean data statistics during early stages of training [36, 48, 2]. Methods can be based on a loss function which is robust to label noise [50, 40], or they can introduce an explicit [5] or implicit [35] regularization term. Another adaptation of the loss function is achieved by using different bootstrapping methods based on the predicted class labels [16, 33, 41]. Some other common approaches are based on labeling samples with smaller loss as clean ones [1, 15, 23, 27], and fitting a two-component mixture model on the loss distribution, or using cross-validation techniques [7] in order to separate clean and mislabeled samples. Several existing methods [39, 29, 15] require knowledge about the percentage of noisy samples, or estimate this quantity based on a given. The suitability of these approaches is unclear for real-world applications. Since our proposed method, SREA, targets the correction of mislabeled training samples based on their embedding representation, we do not rely on specific assumptions on the amount of label noise in advance. Our modeling approach is based on the observation that self-supervision has proven to provide an effective representation for downstream tasks without requiring labels [17, 19], this leading to an improvement of performance in the main supervised task [22].

Most applications of models which deal with label noise come from the domain of image data, where noise can be induced by e.g. crowd-working annotations [25, 16]. Some approaches consider label noise in other domains such as human activity detection [3], sound event classification [10], or malware detection [14]. An attempt to analyze the effect of noisy data on applications for real-world data sets is made in [44], but the authors do not compare to the SotA. Up to the authors knowledge, we are the first to report a detailed evaluation of time-series classification in the presence of label noise, which is a crucial data characteristics in the domain of NILM.

3 CHP electrical power output estimation

The CHP is a complex industrial machinery which burns natural gas in order to produce heat and electrical power. It is controlled by an algorithm where only some aspects are known, so its behavior is mostly unclear and not well predictable. It is known that important control signals are the ambient outside temperature T_{amb}, the internal water temperature T_{water}, the generated electrical power output P_{CHP}, and the total electricity demand of the facility P_{tot}. Fig. 1 shows examples of recorded data from the CHP as well as P_{tot} of the facility. T_{amb} has a known strong influence as the CHP is off in the summer period and more or less continuously on in winter and cold periods. In the transition

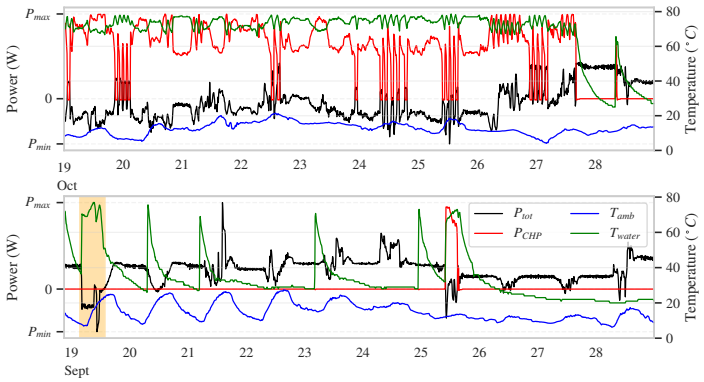


Fig. 1. Sensory data from the CHP and total electrical power. Upper: normal operation. Lower: an example of P_{CHP} sensor malfunction is highlighted in yellow.

seasons (spring and fall), the CHP sometimes turns on (night of 25$^{\text{th}}$ Sept.), sometimes just heats up its internal water (nights of 20$^{\text{st}}$, 21$^{\text{st}}$, and 23$^{\text{rd}}$ Sept.), or exhibits a fast switching behavior (e.g. 20$^{\text{st}}$, and 27$^{\text{th}}$ Oct.). Even though the CHP usually turns on for a couple of hours (e.g. 25$^{\text{th}}$ Sept.) at rare instances it just turns on for a very short time (e.g. 28$^{\text{th}}$ Oct.).

Due to the complicated operational pattern, it is already hard to make a detector for the CHP operational state even with full access to the measurement data. Additionally, the sensors measuring the CHP output power are prone to failure which can be observed in the yellow highlighted area in the bottom side of Fig. 1. During that 10-hour period, the CHP did produce electrical power, even though the sensor reading does not indicate this. The total electrical power drawn from the grid, P_{tot}, provides a much more stable measurement signal. The signature of the CHP is clearly visible in the total power signal and we propose to estimate P_{CHP} from this signal, but many variables also affects the total load, e.g. PV system, changing workloads, etc.

We focus on estimating the power output but where the estimate power value should only be accurate within a certain range. Thus, the problem is formulated as time-series classification instead of regression, where each class represents a certain range of output values. The class labels are calculated directly from the P_{CHP} sensor measurement as the mean power output of a fixed-length sliding window. Due to frequent sensor malfunctioning, as displayed in Fig. 1, the resulting classification problem is subject to label noise.

4 Self-Re-Labeling with Embedding Analysis (SREA)

Architecture and loss function In this work, column vectors are denoted in bold (e.g. $\boldsymbol{x}$). As described in Sec. 3, we treat the challenge to model time series data as a classification problem to predict averaged characteristics of

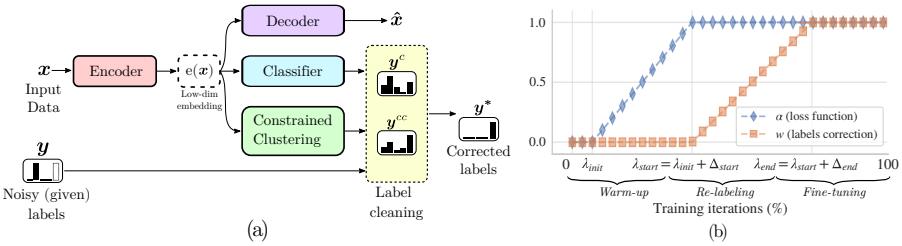


Fig. 2. SREA processing architecture (a) and the dynamics of the parameters α and w during the training epochs (b).

the process using windowing techniques. Hence, we deal with a supervised k-class classification problem setting with a dataset of n training examples $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i), i = 1, ..., n\}$ with $\boldsymbol{y}_i \in \{0, 1\}^k$ being the one-hot encoding label for sample $\boldsymbol{x}_i$. Thereby, **label noise** is present, i.e. we expect that $\boldsymbol{y}_i$ is wrong for a substantial (and possibly skewed) amount of instances in the training set.

The overall processing architecture of the proposed approach is shown in Fig. 2(a). The *autoencoder* (f_{ae}), represented by the *encoder* (e) and the *decoder*, provides a strong surrogate supervisory signal for feature learning [22] that is not affected by the label noise. Two additional components, a *classifier* network (f_c) and a *constrained clustering* module (f_{cc}) are introduced, which independently propose class labels as output. Each of the three processing pipelines share the same embedding representation, created by the encoder, and each output module is associated with one separate contribution to the total loss function.

For the autoencoder, a typical reconstruction loss is utilized:

$$\mathcal{L}_{ae} = \frac{1}{n} \sum_{i=1}^{n} \left(\hat{\boldsymbol{x}}_i - \boldsymbol{x}_i\right)^2, \tag{1}$$

where $\hat{\boldsymbol{x}}_i$ is the output of the autoencoder given the input $\boldsymbol{x}_i$.

Cross entropy is used as loss function for the classification network output,

$$\mathcal{L}_c = -\frac{1}{n} \sum_{i=1}^{n} \boldsymbol{y}_i^T \cdot \log(\boldsymbol{p}_i^c), \tag{2}$$

where $\boldsymbol{p}_i^c$ are the k-class softmax probabilities produced by the model for the training sample i, i.e. $\boldsymbol{p}_i^c = \text{softmax}(f_c(\boldsymbol{x}_i))$.

For the constraint clustering loss, we first initialize the cluster center $\boldsymbol{C} \in \mathbb{R}^{d \times k}$ in the d-dimensional embedding space, with the k-means clustering of the training samples. Then, inspired by the good results achieved in [47], we constrain the embedding space to have small intra-class and large inter-class distances, by iteratively adapting $\boldsymbol{C}$. The resulting clustering loss is given by:

$$\mathcal{L}_{cc} = \frac{1}{n} \sum_{i=1}^{n} \left[\underbrace{\|\boldsymbol{e}(\boldsymbol{x}_i) - \boldsymbol{C}_{\boldsymbol{y}_i}\|_2^2}_{\text{intra-class}} + \underbrace{\log \sum_{j=1}^{k} \exp\left(-\|\boldsymbol{e}(\boldsymbol{x}_i) - \boldsymbol{C}_j\|_2\right)}_{\text{inter-class}} \right] + \ell_{reg}. \tag{3}$$

The entropy regularization $\ell_{reg} = -\sum_i^k \min_{i \neq j} \log \|\boldsymbol{C}_i - \boldsymbol{C}_j\|_2$ is aimed to create well separated embedding for different classes [37].

The final total loss function is given by the sum of those contributions,

$$\mathcal{L} = \mathcal{L}_{ae} + \alpha \left(\mathcal{L}_c + \mathcal{L}_{cc} + \mathcal{L}_\rho \right), \tag{4}$$

where we introduced the dynamic parameter $0 \leq \alpha \leq 1$ which changes during the training (explained below). We also add the regularization loss $\mathcal{L}_\rho$ to prevent the assignment of all labels to a single class, $\mathcal{L}_\rho = \sum_{j=1}^k \boldsymbol{h}_j \cdot \log \frac{\boldsymbol{h}_j}{\boldsymbol{p}_j^\rho}$, where $\boldsymbol{h}_j$ denotes the prior probability distribution for class j, which assume to be uniformly distributed to $1/k$. The term $\boldsymbol{p}_j^\rho$ is the mean softmax probability of the model for class j across all samples in dataset which we approximate using mini-batches as done in previous works [1].

Re-Labeling strategy We do not compute the total loss function in Eq. (4) with the given (noisy) label $\boldsymbol{y}_i$. But, we estimate the true label for each data sample $\boldsymbol{y}_i^*$ by taking the weighted average of the given training label $\boldsymbol{y}_i$, and the pseudo-labels proposed by the classifier $\boldsymbol{y}_i^c$ and the constraint clustering $\boldsymbol{y}_i^{cc}$.

In order to increase robustness of the labels proposed by the classifier ($\boldsymbol{y}_i^c$), we take for each data sample the exponentially averaged probabilities during the last five training epochs, with the weighting factor $\tau_t \sim e^{\frac{t-5}{2}}$:

$$\boldsymbol{y}_i^c = \sum_{\text{last 5 epochs } t} \tau_t \, [\boldsymbol{p}_i^c]_t. \tag{5}$$

The label from constraint clustering ($\boldsymbol{y}_i^{cc}$) is determined by the distances of the samples to the cluster centers in the embedding space:

$$\boldsymbol{y}_i^{cc} = \text{softmin}_j \big(\|\boldsymbol{e}(\boldsymbol{x}_i) - \boldsymbol{C}_j\|_2 \big) \tag{6}$$

Then, the corrected label $\boldsymbol{y}_i^*$ (in one hot-encoding) is produced by selecting the class corresponding to the maximum entry:

$$\boldsymbol{y}_i^* = \text{argmax} \big[(1-w)\, \boldsymbol{y}_i + w \, (\boldsymbol{y}_i^c + \boldsymbol{y}_i^{cc}) \big] \tag{7}$$

where the dynamic weighting factor $0 \leq w \leq 1$ is function of the training epoch t and to be discussed below when explaining the training dynamics.

Training Dynamics A key aspect of our proposed approach is to dynamically change the loss function, as well as the label correction mechanism, during the training. This is achieved by changing the parameters α (loss function) and w (label correction mechanism) as depicted in Fig. 2(b). The training dynamics is completely defined by the three hyper-parameters λ_{init}, Δ_{start} and Δ_{end}.

Initially, we start with $\alpha = 0$ and $w = 0$, and only train the autoencoder from epoch $t = 0$ to epoch $t = \lambda_{init}$. At the training epoch $t = \lambda_{init}$, α is ramped up linearly until it reaches $\alpha = 1$ in training epoch $t = \lambda_{start} = \lambda_{init} + \Delta_{start}$.

Algorithm 1: SREA: Self-Re-Labeling with Embedding Analysis.

Require: Data $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_n$, autoencoder f_{ae}, classifier f_c, constraint clustering f_{cc}, hyper-parameters: λ_{init}, Δ_{start}, Δ_{end}.

Init hyper-parameter ramp-up functions w_t and α_t *// see Fig. 2(b)*
for *training epoch* $t = 0$ **to** t_{end} **do**
  Fetch mini-batch data $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_b$ at current epoch t
  for $i = 1$ **to** b **do**
    if $t == \lambda_{init}$ **then**
      $f_{cc} \leftarrow k$-means$(\boldsymbol{e}(\boldsymbol{x}_i))$ *// Initialize constraint clustering*
    end
    $\hat{\boldsymbol{x}}_i = f_{ae}(\boldsymbol{x}_i)$ *// Auto-encoder forward pass*
    $\boldsymbol{y}_i^c \leftarrow$ Eq.(5) *// Classifier forward pass*
    $\boldsymbol{y}_i^{cc} \leftarrow$ Eq.(6) *// Constraint clustering output*
    Adjust $w \leftarrow w_t$, $\alpha \leftarrow \alpha_t$ *// Label correction and loss parameters*
    $\boldsymbol{y}_i^* \leftarrow$ Eq.(7) *// Re-labeling*
    $\mathcal{L} \leftarrow$ Eq.(4) *// Evaluate loss function*
    Update f_{ae}, f_c, f_{cc} by SGD on $\mathcal{L}$
  end
end

The purpose of this first *warm-up* period is an unsupervised initialization of the embedding space with slowly turning on the supervision of the given labels. The dominant structure of the clean labels is learned, as neural networks tend to learn the true labels, rather than overfit to the noisy ones, at early training stages [48, 2]. Then, we also increase w linearly from zero to one, between epochs $t = \lambda_{start}$ to $t = \lambda_{start} + \Delta_{end} = \lambda_{end}$, thereby turning on the label correction mechanism (*re-labeling*). After training epoch $t = \lambda_{end}$ until the rest of the training, we keep $\alpha = 1$ and $w = 1$ which means we are fully self-supervised where the given training labels do not enter directly anymore (*fine-tuning*). We summarize the SREA and display the pseudo-code in Algorithm 1.

5 Experimental setup

Label Noise True labels are corrupted by a *label transition matrix* T [38], where T_{ij} is the probability of the label i being flipped into label j. For all the experiments, we corrupt the labels with *symmetric* (unstructured) and *asymmetric* (structured) noise with noise ratio $\epsilon \in [0, 1]$. For symmetric noise, a true label is randomly assigned to other labels with equal probability, i.e. $T_{ii} = 1 - \epsilon$ and $T_{ij} = \frac{\epsilon}{k-1}$ $(i \neq j)$, with k the number of classes. For asymmetric noise, a true label is mislabelled by shifting it by one, i.e. $T_{ii} = 1 - \epsilon$ and $T_{(j+1 \bmod k)j} = \epsilon$ $(i \neq j)$. For the estimation of the CHP power, we also analyze another kind of structured noise which we call *flip* noise, where a true label is only flipped to zero, i.e. $T_{ii} = 1 - \epsilon$ and $T_{i0} = \epsilon$. This mimics sensor failures, where a broken sensor produces a constant output regardless of the real value. Note that learning with structured noise is much harder than with unstructured noise [12].

Network architecture Since SREA is model agnostic, we use CNNs in the experiments, as these are currently the SotA deep learning network topology for time-series classification [43, 9]. The encoder and decoder have a symmetric structure with 4 convolutional blocks. Each block is composed by a 1D-conv layer followed by batch normalization [21], a ReLU activation and a dropout layer with probability 0.2. The dimension of the shared embedding space is 32. For the classifier we use a fully connected network with 128 hidden units and *#classes* outputs. We use the Adam optimizer [26] with an initial learning rate of 0.01 for 100 epochs. Such high value for the initial learning rate helps to avoid overfitting of noisy data in the early stages of training [48, 2]. We halve the learning rate every 20% of training (20 epochs). In the experiments, we assume to not have access to any clean data, thus it is not possible to use a validation set, and the models are trained without early stopping. The SREA hyper-parameters $\lambda_{init} = 0$, $\Delta_{start} = 25$ and $\Delta_{end} = 30$ are used if not specified otherwise. Further implementation details are reported in the supplementary material [6], including references to the availability of code and data sets.

Comparative methods In order to make a fair comparison, we use the same neural network topology throughout all the experiments. A baseline method, which does not take in accout any label noise correction criteria, is a CNN classifier [43] trained with cross-entropy loss function of Eq. (2), which we refer to as *CE*. We compare to *MixUp* [49], which is a data augmentation technique that exhibits strong robustness to label noise. In *MixUp-BMM* [1] a two-component beta-mixture model is fitted to the loss distribution and training with bootstrapping loss is implemented. *SIGUA* [15] implements stochastic gradient ascent on likely mislabeled data, thereby trying to reduce the effect of noisy labels. Finally, in *Co-teaching* [39] two networks are simultaneously trained which inform each other about which training examples to keep. The algorithms *SIGUA* and *Co-teaching* assume that the noise level ϵ is known. In our experiments, we use the true value of ϵ for those approaches, in order to create an upper-bound of their performance. All the hyper-parameters of the investigated algorithm are set to their default and recommended values.

Implementation details For the problem of estimating CHP power output, the raw data consists of 78 days of measurement with a sampling rate of 1 sample/minute. As preprocessing, we do a re-sampling to 6 samples/hour. The CHP should have a minimal on-time of one hour, in order to avoid too rapid switching which would damage the machine. However, during normal operation, the CHP is controlled in a way that on- and off-time periods are around 4 to 8 hours. Due to these time-scales, we are interested in the power output on a scale of 6 hours, which means we use a sliding window with a size of 6 hours (36 samples) and a stride of 10 minutes (1 sample). Therefore, the preprocessing of the three input variables P_{tot}, T_{water}, and T_{amb} lead to $\mathbb{R}^{(36\times3)}$-dimensional data samples. For generating the labels, we use 5 power output levels, linearly spaced from 0 to $P_{CHP,max}$, and correspondingly to a five-dimensional one-hot

encoded label vector $\boldsymbol{y}_i \in \mathbb{R}^5$. For every dataset investigated, we normalize the dataset to have zero mean and unit standard deviation, and we randomly divide the total available data in train-set and test-set with a ratio 80:20.

Evaluation measures To evaluate the performance, we report the *averaged* $\mathcal{F}_1$*-score* on the test-set, where the well-known F_1-scores are calculated for each class separately and then averaged via arithmetic mean, $\mathcal{F}_1 = \frac{1}{k}\sum_{j=1}^{k} F_{1,j}$. This formulation of the F_1-score results in a larger penalization when the model do not perform well in the minority class, in cases with class imbalance. Other metrics, such as the accuracy, show qualitatively similar results and are therefore not reported in this manuscript. In order to get performance statistics of the methods, all the experiments have been repeated 10 times with different random initialization. The non-parametric statistical *Mann-Whitney U Test* [32] is used to compare the SREA against the SotA algorithms.

6 Results and discussion

Benchmarks Datasets We evaluate the proposed SREA on publicly available time-series classification datasets from UCR repository [8]. We randomly choose 10 datasets with different size, length, number of classes and dimensions in order to try to avoid bias in the data. A summary of the datasets is given in Table 1.

Table 1. UCR Single-variate and Multivariate dataset description.

Dataset	Size	Length	#classes	#dim
ArrowHead†	211	251	3	1
CBF	930	128	3	1
Epilepsy	275	206	4	3
FaceFour†	112	350	4	1
MelbourneP.	3633	24	10	1
NATOPS	360	51	6	24
OSULeaf†	442	427	6	1
Plane	210	144	7	1
Symbols†	1020	398	6	1
Trace†	200	275	4	1

† results reported in the supplementary material [6].

We show a representative selection of all comparisons of the results in Table 2. Without any label noise, CE is expected to provide very good results. We observe that our SREA gives similar or better $\mathcal{F}_1$ scores than the CE method on 9 out of 10 datasets without label noise. Considering all algorithms and datasets, with symmetric noise we achieve statistically significantly better scores in 62, similar scores in 105, and worse scores in 33 experiments out of a total of 200 experiments[3]. For the more challenging case of asymmetric noise, the SREA-results are 86 times significantly better, 97 times equal, and 17 times worse than SotA algorithms.

CHP power estimation Table 3 shows the results of the estimation of the CHP output power level. Without labels noise, the proposed approach has comparable performance to CE, with an average $\mathcal{F}_1$-score of 0.979. This implies that we

[3] Number of experiments: 10 datasets × 4 noise levels × 5 algorithms = 200. Each experiment consists of 10 independent runs.

Table 2. $\mathcal{F}_1$ test scores on UCR datasets. The best results per noise level are underlined. In parenthesis the results of a Mann–Whitney U test with $\alpha = 0.05$ of SREA against the other approaches: SREA $\mathcal{F}_1$ is significantly higher (+), lower (−) or not significant (≈).

Dataset	Noise	%	CE	MixUp	M-BMM	SIGUA	Co-teach	SREA
CBF	-	0	1.000 (+)	0.970 (+)	0.886 (+)	1.000 (+)	0.997 (+)	1.000
	Symm	15	0.943 (+)	0.923 (+)	0.941 (+)	0.976 (+)	0.923 (+)	1.000
		30	0.780 (+)	0.799 (+)	0.932 (+)	0.923 (+)	0.833 (+)	0.998
	Asymm	10	0.973 (+)	0.956 (+)	0.920 (+)	0.989 (+)	0.963 (+)	1.000
		20	0.905 (+)	0.897 (+)	0.949 (+)	0.980 (+)	0.900 (+)	1.000
Epilepsy	-	0	0.974 (≈)	0.955 (+)	0.926 (+)	0.978 (≈)	0.971 (+)	0.973
	Symm	15	0.890 (≈)	0.913 (≈)	0.899 (≈)	0.884 (≈)	0.861 (≈)	0.861
		30	0.784 (−)	0.823 (−)	0.805 (−)	0.741 (≈)	0.744 (≈)	0.708
	Asymm	10	0.919 (≈)	0.930 (≈)	0.847 (≈)	0.905 (≈)	0.919 (≈)	0.888
		20	0.861 (≈)	0.894 (−)	0.891 (−)	0.826 (≈)	0.863 (≈)	0.825
Melbourne	-	0	0.923 (≈)	0.879 (+)	0.773 (+)	0.918 (≈)	0.913 (≈)	0.911
	Symm	15	0.869 (+)	0.870 (+)	0.856 (+)	0.883 (≈)	0.886 (≈)	0.883
		30	0.826 (+)	0.858 (≈)	0.870 (≈)	0.855 (≈)	0.876 (−)	0.862
	Asymm	10	0.898 (+)	0.877 (+)	0.860 (+)	0.899 (+)	0.897 (+)	0.911
		20	0.865 (+)	0.861 (+)	0.851 (+)	0.858 (+)	0.893 (≈)	0.903
NATOPS	-	0	0.858 (≈)	0.801 (≈)	0.711 (+)	0.848 (≈)	0.835 (≈)	0.866
	Symm	15	0.779 (≈)	0.718 (+)	0.702 (+)	0.754 (≈)	0.761 (≈)	0.796
		30	0.587 (≈)	0.580 (+)	0.602 (+)	0.593 (+)	0.673 (+)	0.670
	Asymm	10	0.798 (+)	0.822 (≈)	0.756 (+)	0.764 (+)	0.790 (+)	0.829
		20	0.703 (≈)	0.763 (≈)	0.762 (≈)	0.698 (≈)	0.733 (≈)	0.762
Plane	-	0	0.995 (≈)	0.962 (+)	0.577 (+)	0.981 (+)	0.990 (≈)	0.998
	Symm	15	0.930 (+)	0.953 (+)	0.873 (+)	0.971 (≈)	0.981 (≈)	0.983
		30	0.887 (+)	0.902 (+)	0.943 (≈)	0.862 (+)	0.941 (≈)	0.944
	Asymm	10	0.981 (≈)	0.986 (≈)	0.648 (+)	0.986 (≈)	0.990 (≈)	0.976
		20	0.952 (≈)	0.923 (≈)	0.751 (+)	0.976 (≈)	0.990 (≈)	0.966

successfully solve the CHP power estimating problem by analyzing the total load with an error rate less than 2%. When the training labels are corrupted with low level of symmetric label noise, $\epsilon \leq 0.4$, SREA consistently outperforms the other algorithms. With higher level of symmetric noise we achieve a comparable performance to the other algorithms. Under the presence of asymmetric label noise, SREA shows a performance comparable to the other SotA algorithms. Only for a highly unrealistic asymmetric noise level of 40%, the performance is significantly worse than the SotA. This indicates that, during the warm-up and relabelling phase, the network is not able to learn the true labels but also learns the wrong labels induced by the structured noise. During the fine-tuning phase, the feedback of the wrongly labeled instances is amplified and the wrong labels a reinforced. For flip noise, which reflects sensors failures, SREA retains a high performance and outperforms all other SotA algorithms up to noise levels of 40%. For low noise levels up to 20%, SREA has similar performance to Co-teaching, but without the need to know the amount of noise.

In Fig. 3 (left) we show the label confusion matrix (with in-class percentage in parentheses) of SREA for the resulting corrected labels for the case of 30% of

Table 3. $\mathcal{F}_1$ test scores of the CHP power estimation. Same notation as Table 2.

Noise	%	CE	MixUp	M-BMM	SIGUA	Co-teach	SREA
-	0	0.980 (≈)	0.957 (+)	0.882 (+)	0.979 (≈)	0.974 (≈)	0.979
Symmetric	15	0.931 (+)	0.934 (+)	0.903 (+)	0.954 (≈)	0.950 (≈)	0.960
	30	0.856 (+)	0.910 (+)	0.897 (+)	0.912 (+)	0.920 (+)	0.938
	45	0.763 (+)	0.883 (+)	0.895 (+)	0.867 (+)	0.886 (+)	0.918
	60	0.661 (+)	0.761 (≈)	0.692 (+)	0.817 (≈)	0.839 (≈)	0.800
Asymmetric	10	0.954 (≈)	0.945 (+)	0.893 (+)	0.959 (≈)	0.964 (≈)	0.961
	20	0.924 (+)	0.925 (+)	0.899 (+)	0.935 (≈)	0.938 (≈)	0.946
	30	0.895 (≈)	0.909 (≈)	0.873 (+)	0.916 (≈)	0.923 (≈)	0.919
	40	0.807 (−)	0.848 (−)	0.784 (−)	0.836 (−)	0.876 (−)	0.287
Flip	10	0.970 (≈)	0.950 (+)	0.860 (+)	0.965 (+)	0.973 (≈)	0.971
	20	0.942 (+)	0.942 (+)	0.860 (+)	0.945 (+)	0.962 (≈)	0.963
	30	0.908 (+)	0.919 (+)	0.880 (+)	0.923 (+)	0.792 (+)	0.956
	40	0.868 (+)	0.791 (+)	0.696 (+)	0.779 (+)	0.623 (+)	0.945

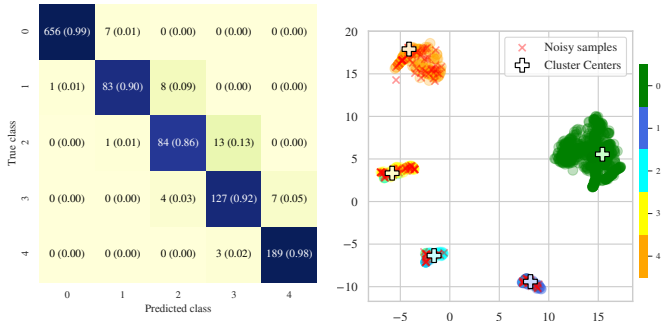


Fig. 3. Confusion matrix of the corrected labels (left) and embedding space of the train-set (right) of the CHP power estimation, corrupted with 30% flip noise.

flip label noise. The corrected labels have 99% and 98% accuracy for the fully off- (0) and on-state (4), respectively. The intermediate power values' accuracies are 90% for state 1, 86% for state 2 and 92% for state 3. As an example, a visualization of the 32 dimensional embedding using the UMAP [31] dimension reduction technique is shown in Fig. 3 for the cases of 30% flip noise. The clusters representing the classes are very well separated, and we can see that the majority of the noisy label samples have been corrected and assigned to their correct class. Similar plots for other noise types as well as critical distance plots can be found in the supplementary material [6].

Finally, we run SREA on a different real-world test-set which includes a sensor failure, and the corresponding noisy label, as shown in Fig. 1. The method was able to correctly re-label the period of the sensor failure.

6.1 Ablation studies

Hyper-parameter sensitivity We investigate the effect of the hyper-parameters of SREA on both, benchmarks and CHP datasets. The observed trends were similar in all datasets, and therefore we only report the result for the CHP dataset.

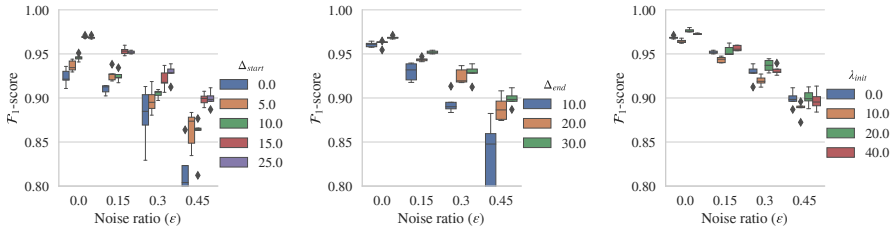


Fig. 4. SREA sensitivity to hyper-parameters Δ_{start} (left), Δ_{end} (middle), λ_{init} (right) for the CHP data and symmetric noise.

Table 4. Ablation studies on loss function components of SREA.

Noise	%	$\mathcal{L}_c$	$\mathcal{L}_c + \mathcal{L}_{ae}$	$\mathcal{L}_c + \mathcal{L}_{cc}$	$\mathcal{L}_c + \mathcal{L}_{ae} + \mathcal{L}_{cc}$
-	0	0.472±0.060	0.504±0.012	0.974±0.003	0.980±0.003
Symmetric	15	0.388±0.027	0.429±0.023	0.943±0.004	0.957±0.007
	30	0.355±0.038	0.366±0.041	0.919±0.006	0.930±0.008
	45	0.290±0.014	0.318±0.010	0.892±0.009	0.902±0.008
Asymmetric	10	0.400±0.026	0.407±0.012	0.949±0.003	0.957±0.003
	20	0.348±0.003	0.358±0.003	0.930±0.009	0.941±0.009
	30	0.342±0.002	0.349±0.004	0.901±0.007	0.922±0.010
Flip	10	0.460±0.030	0.468±0.030	0.961±0.009	0.973±0.006
	20	0.415±0.028	0.424±0.037	0.951±0.008	0.962±0.005
	30	0.405±0.030	0.411±0.040	0.943±0.007	0.957±0.003

The effect of the three hyper-parameters related to the training dynamics (λ_{init}, Δ_{start}, and Δ_{end}) are reported in Fig. 4 for the cases of unstructured symmetric label noise (results for the other noise types can be found in the supplementary material [6]). For the variation of Δ_{start} and Δ_{end} there is a clear pattern for every noise level as the performance increases with the values of the hyper-parameters, and best performance is achieved by $\Delta_{start} = 25$ and $\Delta_{end} = 30$. This shows that both the *warm-up* and *re-labeling* periods should be rather long and last about 55% of the training time, before fully self-supervised training. The effect of λ_{init} is not as clear, but it seems that either a random initialization of the cluster centers (λ_{init} close to zero) or an extended period of unsupervised training of the autoencoder (λ_{init} between 20 and 40) is beneficial.

Loss function components We study what effect each of the major loss function components of Eq. 4 has on the performance of SREA and report the results in Table 4. It can be observed, that not including the constrained clustering, i.e. using only $\mathcal{L}_c$ or $\mathcal{L}_c + \mathcal{L}_{ae}$, gives rather poor performance for all noise types and levels. This is explicitly observed as the performance decreases again during training in the self-supervision phase without $\mathcal{L}_{cc}$ (not shown). This seems understandable as the label correction method repeatedly bootstraps itself by using only the labels provided by the classifier, without any anchor to preserve the information from the training labels. This emphasizes the necessity of constraining the data in the embedding space during self-supervision.

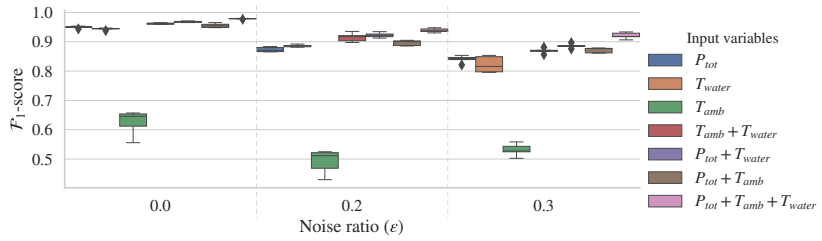


Fig. 5. SREA sensitivity to the input variables for CHP data and asymmetric noise.

Input variables We investigate the influence of the selection of the input variables on the estimation of the CHP power level. The results for all possible combination of the input signals are reported in Fig 5. Unsurprisingly, using only the ambient temperature gives by far the worst results, while utilizing all available inputs results in the highest scores. Without the inclusion of the P_{tot}, we still achieve a $\mathcal{F}_1$-score above 0.96 with only using the T_{water} as input signal.

7 Conclusion and future work

In this work, we presented the problem of estimating the electrical power output of a Combined Heat and Power (CHP) machine by analyzing the total electrical power consumption of a medium size company facility. We presented an approach to estimate the CHP power output by analyzing the total load, the ambient temperature and the water temperature of the CHP, all of which are known to be control variables of the CHP. The training dataset for the deep-learning based approach was automatically derived from sensor measurements of the CHP power output, and sensor failures create noisy samples in the generated class labels. The proposed Self-Re-Labeling with Embedding Analysis (SREA) incorporates an autoencoder, a classifier and a constraint clustering which all share and operate on a common low-dimensional embedding representation. During the network training, the loss function and the label correction mechanism are adjusted in a way that a robust relabeling of noisy training labels is possible. We compare SREA to five SotA label noise correction approaches on ten time-series classification benchmarks and observe mostly comparable or better performance for various noise levels and types. We also observe superior performance on the CHP use-case for a wide range of noise levels and all studied noise types. We thoroughly analyzed the dependence of the proposed methods on the (hyper-)parameters and architecture choices.

The proposed approach is straight-forward to realize without any (hyper-)parameter tuning, as there are clear insights on how to set the parameters and the method is not sensitive to details. It also has the strong benefit that the amount of label noise need not be known or guessed. We used CNNs as building blocks of the proposed algorithm, but since SREA is model agnostic, it is possible to utilize other structures, such as recurrent neural networks [24]

or transformers [45], which would also utilize the time-structure of the problem. The application of such dynamic models is left for future work.

Estimating the CHP output as shown in this work will be used in the future in energy optimization scenarios to arrive at more reliable and robust EV charging schedules. But, due to the robustness of the proposed method and the ability to exchange the neural networks with arbitrary other machine learning modules, we see a high potential for this architecture to be used for label noise correction in other application domains. We also see a high potential for an application in anomaly detection scenarios where sensor failures need to be detected. A thorough evaluation in these application areas is left for future work.

References

1. Arazo, E., Ortego, D., Albert, P., O'Connor, N., McGuinness, K.: Unsupervised label noise modeling and loss correction. In: International Conference on Machine Learning. pp. 312–321 (2019)
2. Arpit, D., Jastrzebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M.S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., et al.: A closer look at memorization in deep networks. In: International Conference on Machine Learning. pp. 233–242 (2017)
3. Atkinson, G., Metsis, V.: Identifying label noise in time-series datasets. In: Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers. pp. 238–243 (2020)
4. Barsim, K.S., Yang, B.: Toward a semi-supervised non-intrusive load monitoring system for event-based energy disaggregation. In: 2015 IEEE global conference on signal and information processing (GlobalSIP). pp. 58–62 (2015)
5. Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., Raffel, C.: Mixmatch: A holistic approach to semi-supervised learning. arXiv:1905.02249 (2019)
6. Castellani, A., Schmitt, S., Hammer, B.: Supplementary material for: Estimating the electrical power output of industrial devices with end-to-end time-series classification in the presence of label noise. arXiv:2105.00349 (2021)
7. Chen, P., Liao, B.B., Chen, G., Zhang, S.: Understanding and utilizing deep neural networks trained with noisy labels. In: International Conference on Machine Learning. pp. 1062–1070 (2019)
8. Dau, H.A., Keogh, E., Kamgar, K., Yeh, C.C.M., Zhu, Y., Gharghabi, S., Ratanamahatana, C.A., Yanping, Hu, B., Begum, N., Bagnall, A., Mueen, A., Batista, G., Hexagon-ML: The ucr time series classification archive (2018)
9. Fawaz, H.I., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A.: Deep learning for time series classification: a review. Data Mining and Knowledge Discovery **33**(4), 917–963 (2019)
10. Fonseca, E., Plakal, M., Ellis, D.P., Font, F., Favory, X., Serra, X.: Learning sound event classifiers from web audio with noisy labels. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 21–25 (2019)
11. Fredriksson, T., Mattos, D.I., Bosch, J., Olsson, H.H.: Data labeling: An empirical investigation into industrial challenges and mitigation strategies. In: PROFES (2020)

12. Frénay, B., Verleysen, M.: Classification in the presence of label noise: a survey. IEEE transactions on neural networks and learning systems **25**(5), 845–869 (2013)
13. Gan, O.P.: Automatic labeling for personalized iot wearable monitoring. IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society pp. 2861–2866 (2018)
14. Gavrilut, D., Ciortuz, L.: Dealing with class noise in large training datasets for malware detection. 2011 13th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing pp. 401–407 (2011)
15. Han, B., Niu, G., Yu, X., Yao, Q., Xu, M., Tsang, I., Sugiyama, M.: Sigua: Forgetting may make learning with noisy labels more robust. In: International Conference on Machine Learning. pp. 4006–4016 (2020)
16. Han, B., Yao, Q., Liu, T., Niu, G., Tsang, I.W., Kwok, J.T., Sugiyama, M.: A survey of label-noise representation learning: Past, present and future. arXiv:2011.04406 (2020)
17. Hendrycks, D., Mazeika, M., Kadavath, S., Song, D.: Using self-supervised learning can improve model robustness and uncertainty. arXiv:1906.12340 (2019)
18. Holmegaard, E., Kjærgaard, M.B.: Nilm in an industrial setting: A load characterization and algorithm evaluation. 2016 IEEE SMARTCOMP pp. 1–8 (2016)
19. Huang, L., Zhang, C., Zhang, H.: Self-adaptive training: Bridging the supervised and self-supervised learning. arXiv:2101.08732 (2021)
20. Humala, B., Nambi, A.S.U., Prasad, V.R.: Universalnilm: A semi-supervised energy disaggregation framework using general appliance models. In: Proceedings of the Ninth International Conference on Future Energy Systems. pp. 223–229 (2018)
21. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. pp. 448–456 (2015)
22. Jawed, S., Grabocka, J., Schmidt-Thieme, L.: Self-supervised learning for semi-supervised time series classification. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. pp. 499–511 (2020)
23. Jiang, L., Zhou, Z., Leung, T., Li, L.J., Fei-Fei, L.: Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In: International Conference on Machine Learning. pp. 2304–2313 (2018)
24. Karim, F., Majumdar, S., Darabi, H., Chen, S.: Lstm fully convolutional networks for time series classification. IEEE Access **6**, 1662–1669 (2018)
25. Karimi, D., Dou, H., Warfield, S., Gholipour, A.: Deep learning with noisy labels: exploring techniques and remedies in medical image analysis. Medical image analysis **65**, 101759 (2020)
26. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv:1412.6980 (2014)
27. Li, J., Socher, R., Hoi, S.C.: Dividemix: Learning with noisy labels as semi-supervised learning. arXiv:2002.07394 (2020)
28. Limmer, S.: Evaluation of optimization-based ev charging scheduling with load limit in a realistic scenario. Energies **12**(24) (2019)
29. Mandal, D., Bharadwaj, S., Biswas, S.: A novel self-supervised re-labeling approach for training with noisy labels. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1381–1390 (2020)
30. Massidda, L., Marrocu, M., Manca, S.: Non-intrusive load disaggregation by convolutional neural network and multilabel classification. Applied Sciences **10**(4), 1454 (2020)
31. McInnes, L., Healy, J.: Umap: Uniform manifold approximation and projection for dimension reduction. arXiv:1802.03426 (2018)

32. McKnight, P.E., Najab, J.: Mann-whitney u test. The Corsini encyclopedia of psychology p. 1 (2010)
33. Nguyen, D.T., Mummadi, C.K., Ngo, T.P.N., Nguyen, T.H.P., Beggel, L., Brox, T.: Self: Learning to filter noisy labels with self-ensembling. In: International Conference on Learning Representations (2019)
34. Paresh, S., Thokala, N., Majumdar, A., Chandra, M.: Multi-label auto-encoder based electrical load disaggregation. 2020 International Joint Conference on Neural Networks (IJCNN) pp. 1–6 (2020)
35. Reed, S.E., Lee, H., Anguelov, D., Szegedy, C., Erhan, D., Rabinovich, A.: Training deep neural networks on noisy labels with bootstrapping. In: ICLR (2015)
36. Rolnick, D., Veit, A., Belongie, S., Shavit, N.: Deep learning is robust to massive label noise. arXiv:1705.10694 (2017)
37. Sablayrolles, A., Douze, M., Schmid, C., Jégou, H.: Spreading vectors for similarity search. In: ICLR 2019-7th International Conference on Learning Representations. pp. 1–13 (2019)
38. Song, H., Kim, M., Park, D., Lee, J.G.: Learning from noisy labels with deep neural networks: A survey. arXiv:2007.08199 (2020)
39. Sugiyama, M.: Co-teaching: Robust training of deep neural networks with extremely noisy labels. In: NeurIPS (2018)
40. Van Rooyen, B., Menon, A.K., Williamson, R.C.: Learning with symmetric label noise: The importance of being unhinged. arXiv:1505.07634 (2015)
41. Wang, J., Ma, Y., Gao, S.: Self-semi-supervised learning to learn from noisylabeled data. arXiv:2011.01429 (2020)
42. Wang, X., Wang, C.: Time series data cleaning: A survey. IEEE Access **8**, 1866–1881 (2020)
43. Wang, Z., Yan, W., Oates, T.: Time series classification from scratch with deep neural networks: A strong baseline. In: 2017 International joint conference on neural networks (IJCNN). pp. 1578–1585 (2017)
44. Wang, Z., yi Luo, X., Liang, J.: A label noise robust stacked auto-encoder algorithm for inaccurate supervised classification problems. Mathematical Problems in Engineering **2019**, 1–19 (2019)
45. Wu, N., Green, B., Ben, X., O'Banion, S.: Deep transformer models for time series forecasting: The influenza prevalence case. arXiv:2001.08317 (2020)
46. Yang, Y., Zhong, J., Li, W., Gulliver, T.A., Li, S.: Semisupervised multilabel deep learning based nonintrusive load monitoring in smart grids. IEEE Transactions on Industrial Informatics **16**(11), 6892–6902 (2019)
47. Zeghidour, N., Grangier, D.: Wavesplit: End-to-end speech separation by speaker clustering. arXiv:2002.08933 (2020)
48. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning requires rethinking generalization. arXiv:1611.03530 (2016)
49. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. arXiv:1710.09412 (2017)
50. Zhang, Z., Sabuncu, M.R.: Generalized cross entropy loss for training deep neural networks with noisy labels. arXiv:1805.07836 (2018)