Label Noise Correction via Fuzzy Learning Machine

Jiye Liang*, Yixiao Li, Junbiao Cui

Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, School of Computer and Information Technology, Shanxi University, Taiyuan, 030006, Shanxi, China
ljy@sxu.edu.cn, 1095766055@qq.com, cjb@sxu.edu.cn

Abstract

The ubiquitous and unavoidable label noise brings great challenges to the generalization performance of learning methods. Label noise correction aims to detect and correct label noise in the data, which is one of the most potential methods to address this challenge. Current methodologies for noise filtering that utilize primitive features primarily concentrate on identifying noise, which often limits their capacity to adaptively learn features crucial for specific tasks, thereby resulting in a higher rate of noise identification within the noise recognition process. On the other hand, deep neural networks, endowed with robust feature extraction capabilities, typically exhibit lower noise identification, as they are prone to fitting noise patterns during the recognition process, potentially undermining their overall efficacy. Moreover, Fuzzy Learning Machine (FLM) excels not only in feature extraction but also in noise tolerance, adeptly navigating data uncertainties. FLM enhances label accuracy by calculating the membership degrees of samples across categories and determining their fuzzy memberships. The introduction of a two-stage FLM-based framework, which employs a secondary learning mechanism for precise noise filtering and correction, has shown substantial improvements in noise correction across various large-scale noisy datasets, thereby significantly enhancing sample quality and boosting the generalization capabilities of classifiers.

Introduction

Classification is one of the most critical issues in the field of machine learning. Numerous studies have shown that the generalization performance of training classifiers heavily depends on the quality of labels in the training samples (Bi and Jeske 2010), and that high-quality labels represent accurate and meaningful annotations, which helps to create robust and reliable models. With the rapid development of AI technology, there is a growing demand for high-quality labelled data in many application domains, such as medical imaging, autonomous driving, and security surveillance (Esteva et al. 2017; Kermany, Goldbaum et al. 2018; Grigorescu, Trasnea et al. 2020). However, in the real world, data is growing explosively in terms of variety and quantity, using automated tools and 'crowdsourcing' labeling has gradually become the mainstream method for obtaining large-scale labels. However, these low-cost labeling methods inevitably produce erroneous labels, known as label noise (Breve, Zhao, and Quiles 2015).

The quality of a data set can usually be characterized by two information sources: (1) attributes, and (2) class labels. Depending on the data mining, data noise for supervised learning is mainly categorized into two types: attribute noise and class noise (Zhu and Wu 2004). Attribute noise refers to data where the observed features are in error from the true features, and class noise refers to data where the observed instance markings are inconsistent with the true markings, e.g. in medical diagnosis problems, the inconsistency in expert labeling can lead to erroneous labeling of case data (Khoshgoftaar and Van Hulse 2005). Both attribute noise and class noise affect the generalization performance of the model, but it has been shown that class noise has a more severe impact compared to attribute noise due to the uniqueness of the labels (Frénay and Verleysen 2013). Therefore, filtering and handling label noise is crucial for building efficient and accurate machine learning models. Properly managing label noise not only improves data quality but also enhances model accuracy, ensuring the reliability and robustness of the model in practical applications.

The issue of label noise in classification tasks can be addressed from both the model level and the data level. At the model level, solutions typically involve constructing robust loss functions (Ghosh, Kumar, and Sastry 2017) and introducing regularization (Tanno, Saeedi et al. 2019) to reduce the impact of label noise. However, robust models cannot achieve complete robustness. Data level processing is mainly to improve data quality by labeling noise filtering. Existing noise filtering methods based on raw features primarily focus on noise filtering and struggle to adaptively learn task-relevant features, making them ineffective for managing noise in large-scale datasets. Deep neural networks, while powerful in feature extraction, display a low tolerance for label noise, leading to susceptibility to disruptions. This study focuses on correcting label noise, aiming to refine labels post noise filtering, thus enhancing data quality and elevating model generalization capabilities.

Fuzzy Learning Machine (FLM) (Cui and Liang 2022) integrates the advantages of deep neural networks in feature extraction and the advantages of fuzzy set theory in han-

*Corresponding author

dling uncertainty, making it a potent method for handling label noise. In this paper, we fully leverage the potentiality of FLM to handle label noise. We utilizes deep neural networks to extract features pertinent to label noise filtering, and employs fuzzy permissible loss to mitigate the risk of neural networks overfitting label noise. A computational method is proposed to assess the fuzzy membership of samples to their labels, and a two-stage FLM-based noise correction framework is developed to identify and correct label noise.

The main contributions of this paper are as follows:

(1) Utilizing exemplar theory, this method develops a technique for calculating the fuzzy membership degree of samples with respect to their own labels, thereby assessing the cleanliness of sample labels. This method facilitates preliminary noise filtering.

(2) We have developed a label noise correction framework based on the two-stage FLM. In the First-Learning-Stage, the model is trained on all samples to obtain exemplar sets for each concept. Subsequently, based on the fuzzy membership degrees of each sample, we differentiate between high-confidence and low-confidence sample sets, facilitating the initial screening of noisy samples. During the Re-Learning-Stage, the model is retrained using the high-confidence sample set, utilizing the exemplar set to predict the latent labels of samples. The filtering and correction of label noise are achieved by comparing the predicted labels with the observed labels, ensuring that the noise filtering and correction processes are more accurate and stable.

(3) Experiments on various datasets demonstrate that the proposed method not only successfully identifies and rectifies noise compared to existing label noise filtering methods but also significantly improves data quality and generalization capabilities of the model.

Notations and Related Works

Problem Formalization

This article focuses on classification problems with label noise. For a classification problem, let $\mathcal{X}$, $\mathcal{Y}$, and $\varphi : \mathcal{X} \to \mathcal{Y}$ be the input space, class label space, and unknown classification function, respectively. For the convenience of discussion, we denote $\mathcal{Y}$ as $\{1, 2, \cdots, |\mathcal{Y}|\}$.

Given a noisy training set $D = \{(x_i, y_i) \mid x_i \in \mathcal{X}, y_i \in \mathcal{Y}, i = 1, 2, \ldots, n\}$, where n is the number of training samples, x_i is the i-th training sample and y_i is the observed class label of x_i. In the label noise scenario, the class labels of some training samples are incorrect, i.e., $\exists(x_i, y_i) \in D$, such that $y_i \neq \varphi(x_i)$. Let $D_{\text{noise}} = \{(x_i, y_i) | (x_i, y_i) \in D, y_i \neq \varphi(x_i)\}$ be the set of training samples with incorrect class labels. Furthermore, the noise rate (NR) of the training set is defined as $\frac{|D_{\text{noise}}|}{|D|}$.

Given noisy training set D, the goal of label noise filtering is to find the D_{noise}, and the goal of label noise correction is to find the D_{noise} and the true class labels of samples in D_{noise}.

Label Noise Learning

Model-level methods to label noise learning can be categorized into two main strategies: designing robust models and developing robust loss functions.

(1) Robust model structures. Lee et al. (Lee, Yun et al. 2019) enhanced decision boundaries' robustness by introducing a generative classifier in the hidden feature space of a pre-trained deep model. They used a minimum covariance determinant estimator to accurately estimate the parameters of the generative classifier, thereby achieving more reliable classification under noisy conditions. Tanno et al. (Tanno, Saeedi et al. 2019) addressed the challenge of robust regularization by jointly estimating annotator noise and true labeling distributions from noisy datasets. They introduced a regularization term to the cross-entropy loss function, effectively mitigating the impact of overfitting caused by label noise.

(2) Robust loss functions. A loss function is deemed noise-resistant if it enables the model to achieve consistent predictions on both clean data and data contaminated with label noise. Ghosh et al. (Ghosh, Kumar, and Sastry 2017) demonstrated through both theoretical derivation and empirical experiments that loss functions with symmetric properties exhibit greater resistance to noise. They found that Mean Absolute Error (MAE) loss is particularly robust to noise, whereas the commonly used cross entropy loss is more susceptible to the detrimental effects of noise. Wang et al. (Wang, Ma et al. 2019) investigated the inherent issues of cross entropy loss, specifically its tendency to overfit easily-acquired categories while making it difficult to learn hard-to-acquire categories. Drawing inspiration from symmetric Kullback-Leibler divergence, they proposed the symmetric cross entropy loss, which aims to address these challenges.

Data-level methods Two primary methods to address label noise at the data level are statistical methods and machine learning-based noise filtering methods.

(1) Statistical methods. Such methods calculate a statistical measure for each sample to assess its noise intensity, and then identify noisy samples based on a predetermined threshold. For example, Xia et al. (Xia, Xiong et al. 2014) proposed a filtering method based on Relative Density (RD), which uses the relative density of samples to evaluate their noise intensity, and proposed the Voting Mechanism Based on Relative Density (vRD) (Xia, Chen et al. 2021). Most of these methods rely on learning from raw features, which limits their ability to adaptively learn task-relevant features, making them less effective in filtering label noise.

(2) Machine learning-based methods. These methods typically involves training a learner to filter noisy samples. The process usually begins by training a model, and samples whose predictions from the learner are inconsistent with their original labels are considered to be noisy.

Neighbor-based filtering methods often utilize K-Nearest Neighbor (KNN) models, such as the Full Nearest Neighbor filter (Barandela and Gasca 2000) and Mutual Nearest Neighbor filter (Liu and Zhang 2012). These methods are typically highly sensitive to the choice of the parameter k.

Numerous filtering methods based on ensemble learning principles focus on filtering noise by evaluating the correctness of predictions from multiple base classifiers in combination. Examples include the Majority Vote Filter (MVF)

(Brodley and Friedl 1999), the Dynamic Integration Filter (Sánchez et al. 2003), and Random Forest with High Consistency (RF) (Sluban, Gamberger, and Lavra 2010). Another method is the Complete Random Forest (CRF) (Xia et al. 2018), which evaluated the extent to which a sample is surrounded by samples from the same class by constructing fully randomized trees, thereby determining the noise intensity of the sample. Xia et al. have also proposed methods such as the Adaptive Complete Random Forest filter (Adap_mCRF) (Huang, Shao, and Peng 2022). This method utilized an adaptive voting strategy, leveraging classification accuracy from randomly partitioned test sets as an adaptive index. However, their filtering performance significantly declines when the proportion of label noise is high. These methods aim to construct more effective learners, potentially filtering a larger number of noisy samples.

Deep learning-based methods. These methods focuses on distinguishing clean samples from noisy ones during the training process, selecting samples with minimal training losses as clean to update the network parameters. Jiang et al. (Jiang et al. 2018) proposed MentorNet, which involves pre-training an auxiliary network to identify clean instances, which then guides the training of the primary network. MentorNet effectively mitigates overfitting to corrupted labels. However, it is prone to error accumulation over time. Han et al. (Han, Yao et al. 2018) introduced the Co-teaching mechanism, which enhances model robustness by training two neural networks concurrently. Each network alternates in selecting samples deemed clean by the other for parameter updates, thus mitigating the impact of noisy labels. Building on this, Yu et al. (Yu et al. 2019) proposed Co-teaching+, where each network independently evaluates small batches of data, identifies instances with divergent predictions, and then selects low-loss samples from these instances. Each network back-propagates and updates parameters using the low-loss data chosen by the peer network, thereby refining the training process. Tan et al. (Tan et al. 2021) proposed Co-learning, which leverages a single model with a shared backbone network, employing two distinct subnetworks for self-supervised and supervised learning. One subnetwork utilizes feature-related information for self-supervised learning through intrinsic similarity, while the other focuses on supervised learning using labeling-related information. Despite the strong representational capabilities of deep neural networks, but these networks often overfit to noise, typically exhibiting, lower noise filtering, thereby reducing their effectiveness in handling labeled noise.

This paper focuses on the issues of label noise filtering and correction. Current noise filtering methods suffer from high computational complexity and inadequacies in handling the burgeoning size and complexity of vast datasets. Predominantly, methods that rely on raw feature recognition fail to learn task-specific features adaptively, impairing their efficacy in accurate noise detection. Despite the profound feature extraction capabilities of deep neural networks, their low tolerance for label noise makes them vulnerable to its interference. Thus, we utilize FLM as an effective instrument for the filtering and correction of label noise.

Fuzzy Learning Machine

Fuzzy Learning Machine (FLM) (Cui and Liang 2022), serves as an effective machine learning method, fundamentally focusing on the insightful understanding of human concept cognition. By capturing the inherent fuzziness in concept cognition and utilizing the knowledge of fuzzy set theory, it reformulates the classification problem into a problem of solving for fuzzy similarity, specifically:

Given a $(\mathcal{X}, \mathcal{Y}, \varphi)$-classification problem, (1) let $\mathcal{X}\times\mathcal{X}$ be the new input space, (2) let $[0, 1]$ be the new output space, (3) let $\varphi^\dagger : \mathcal{X} \times \mathcal{X} \to [0, 1]$ be the new target function, where $\varphi^\dagger$ is the fuzzy equivalence relation (FER) on $\mathcal{X}$, and $\forall(x_i, x_j) \in \mathcal{X} \times \mathcal{X}, \varphi^\dagger((x_i, x_j))$ be the degree that x_i and x_j belong to the same concept. This process can be formally described as: $\mathrm{f}^* = \arg\min_{\mathrm{f}} \mathcal{L}(D, \mathrm{f}) + \rho\mathcal{R}(\mathrm{f})$ s.t. $\mathrm{f} \in \{\mathrm{g} \mid \mathrm{g}$ is an FER on $\mathcal{X}\}$, where $\mathcal{L}$ is the loss function that measures how well the model f^* fits the training data set D, $\mathcal{R}$ is the regularization term, and $\rho > 0$ is the tradeoff parameter.

FLM includes three components: (1) fuzzy similarity relation network, (2) fuzziness permissible loss, and (3) stochastic gradient descent based optimizer.

First, FLM employs a multi-layer neural network with nonlinear activation function as the feature extraction module. This setup is formally described as follows:

$$\forall x \in \mathbb{R}^d,\ h(x; \Theta) \in \mathbb{R}_+^{d_h}, \tag{1}$$

where $\mathbb{R}_+$ is the non-negative real numbers, d_h is the dimension of the latent space, and Θ is the set of learnable parameters.This endows FLM with robust feature extraction capabilities, enabling it to effectively capture the essential characteristics of the samples.

Second, the cosine similarity is used as the skeleton of the binary fuzzy relation network, i.e.

$$\forall h_i, h_j \in \mathbb{R}^{d_h}, \quad g(h_i, h_j) = \frac{h_i^T h_j}{\|h_i\|_2 \|h_j\|_2}, \tag{2}$$

where $f(x_i, x_j; \Theta) = g(h(x_i; \Theta), h(x_j; \Theta)), \forall x_i, x_j \in \mathbb{R}^d$, be the composite of h and g.

Third, FLM employs the fuzziness permissible loss (FPL) to complete the learning process. For a pair of training samples (x_i, y_i) and (x_j, y_j), let $t_{ij} = f(x_i, x_j; \Theta)$ denotes the fuzzy similarity between samples x_i and x_j. The fuzziness permissible loss is constructed as follows:

$$L_{\alpha,\beta}(t_{ij}, y_i, y_j) = \begin{cases} \max\{t_{ij} - \alpha, 0\}, & \text{if } y_i \neq y_j \\ \max\{\beta - t_{ij}, 0\}, & \text{if } y_i = y_j \end{cases}, \tag{3}$$

where α and β are fuzzy parameters with $\alpha \in [0, 0.5)$ and $\beta \in (0.5, 1]$ which control the fuzziness of the concept. According to this formula, the similarity between samples belonging to the same class should exceed β, while for samples from different classes, their similarity should be less than α.

Compared with commonly used classification losses, e.g., cross entropy loss, the FPL has unique advantages in dealing with label noise. First, the FPL effectively models the uncertainty of the class label to which the sample belongs, which makes it to tolerate label noise effectively. Second, the FPL is defined on sample pairs, which makes it not only considers

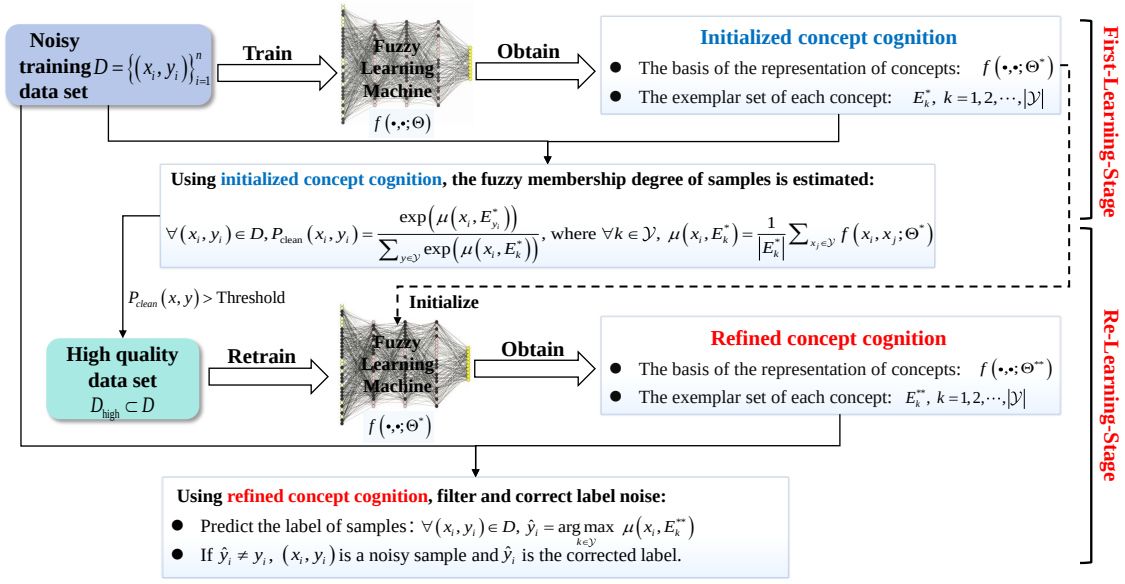

Figure 1: The overall design of FLM-LNC

the matching relationship between samples and class labels, but also takes into account the inherent similarity between samples. When the class labels of the samples are incorrect, the model can utilize the inherent similarity between samples to reduce the harmful effects caused by incorrect class labels.

Based on the above observations, we design a noise filtering and correction framework based on FLM.

Methodology

Overall framework FLM, as a machine learning method known for its superior feature extraction and fuzziness tolerance capabilities, is applied to the problem of label noise filtering and correction. We use FLM for label noise filtering called FLM-LNF, and use FLM for label noise correction called FLM-LNC.

The label noise correction framework based on the two-stage FLM is illustrated in Figure 1. The two-stage learning framework is divided into two stages: the First-Learning-Stage and the Re-Learning-Stage. In the First-Learning-Stage, training the model is conducted using all samples, which produces a preliminary model capable of distinguishing between high-confidence and low-confidence samples. By filtering out the low-confidence samples, the Re-Learning-Stage operates in a cleaner data environment, effectively filtering and correctly correcting label noise.

First-Learning-Stage During the First-Learning-Stage, we initially construct FLM. Throughout the training process, we iteratively update model parameters by minimizing the fuzziness permissible loss on D, capturing the latent representations of all samples. The formalization of this process is as follows:

$$\Theta^* = \underset{\Theta}{arg\,min} \sum_{(x_i,y_i),(x_j,y_j)\in D} L_{\alpha,\beta}(f(x_i,x_j;\Theta),y_i,y_j). \tag{4}$$

After several iterations and training epochs, Θ^* becomes local optimum of the model $f(\cdot,\cdot;\Theta)$.

Next, we complete the representation of concepts based on exemplar theory.

$\forall k \in \mathcal{Y}$, let $X_k = \{x_i \mid (x_i,y_i) \in D, y_i = k\}$. And $\forall x_i \in X_k$, let $\mu(x_i,X_k) = \frac{1}{|X_k|}\sum_{j\in X_k} f(x_i,x_j;\Theta^*)$. Let $U_k = \{\mu(x_i,X_k)|x_i \in X_k\}$. Then, the exemplar set of class k is defined as

$$E_k^* = \left\{ \begin{matrix} x \mid x \in X_k, \mu(x,X_k) \text{ is the top-}n_k^{\text{exe}} \\ \text{largest value in } U_k \end{matrix} \right\}, \tag{5}$$

where n_k^{exe} is a manually specified parameter.

After the exemplar set of each class has been established, it is employed to calculate the fuzzy membership degree of each sample, in order to calculate the probability that the sample (x_i,y_i) belongs to its own category. The fuzzy membership degree of the sample (x_i,y_i) is defined as follows:

$$P_{\text{clean}}(x_i,y_i) = \frac{\exp\left(\mu\left(x_i,E_{y_i}^*\right)\right)}{\sum_{k\in\mathcal{Y}}\exp\left(\mu\left(x_i,E_k^*\right)\right)}, \tag{6}$$

where $\forall k \in \mathcal{Y}$, $\mu(x_i,E_k^*) = \frac{1}{|E_k^*|}\sum_{x_j\in E_k^*} f(x_i,x_j;\Theta^*)$. Based on the fuzzy membership degrees of each sample, a high-confidence subset D_{high} of the training set D can be selected as follows:

$$D_{\text{high}} = \left\{ \begin{matrix} (x,y) \mid (x,y) \in D, P_{\text{clean}}(x,y) \text{ is} \\ \text{the top } \gamma * n \text{ largest element in } U_p \end{matrix} \right\}, \tag{7}$$

where $U_p = \{P_{\text{clean}}(x_i, y_i) \mid (x_i, y_i) \in D\}$, and γ is the manually specified threshold.

This filtering method relies on the model's assessment of the fuzzy membership degree of each sample. Samples with a fuzzy membership degree that exceeds the predefined fuzzy threshold γ are classified as high-confidence samples. This method not only effectively filters out the noise from the data, but also provides a cleaner data environment for subsequent learning processes.

Re-Learning-Stage To provide a cleaner data environment for the subsequent learning process, we enter the Re-Learning-Stage. During this stage, we retrain the model $f(\cdot, \cdot; \Theta)$ using the high-confidence sample set D_{high}. The formalization of this process is as follows:

$$\Theta^{**} = \underset{\Theta}{arg\,min} \sum_{(x_i,y_i),(x_j,y_j)\in D_{\text{high}}} L_{\alpha,\beta}(f(x_i, x_j; \Theta), y_i, y_j). \tag{8}$$

To accelerate convergence, the local optimal solution Θ^* of formula (4) can be used as the initial point. Let Θ^{**} be the local optimum obtained after retraining, we update the exemplar set of each class on D using the retrained model $f(\cdot, \cdot; \Theta^{**})$. Subsequently, we leverage the exemplar set of each class to infer the labels of samples on D. The prediction process is outlined as follows:

$$\hat{y} = \underset{k\in\mathcal{Y}}{arg\,max}\ \mu\left(x, E_k^{**}\right), \quad \forall x \in \mathcal{X}, \tag{9}$$

where $\mu\left(x, E_k^{**}\right) = \frac{1}{|E_k^{**}|} \sum_{x_j \in E_k^{**}} f\left(x, x_j; \Theta^{**}\right)$.

For any sample $(x_i, y_i) \in D$, whenever the predicted label $\hat{y}_i$ differs from the observed label y_i, i.e., $\hat{y}_i \neq y_i$, we consider the sample to be a noisy sample. Additionally, $\hat{y}_i$ becomes the corrected label for the sample x_i.

Through the secondary learning stage, we ultimately obtain the noisy data set D_{noise} and the corrected label $\hat{y}$ for each noise sample.

The two-stage framework not only significantly mitigates noise interference within the training data but also ensures that the model is developed on a foundation of precise and superior-quality data, thereby substantially enhancing the model's generalization capabilities. This strategy capitalizes on the strength of FLM to furnish a robust method for elevating data quality.

Experiment

Experimental Settings

Datasets The experiments were conducted on three commonly used datasets, including MNIST, CIFAR-10, SVHN. Among them, the MNIST dataset (Lecun et al. 1998) is a classic handwritten digit dataset widely used in machine learning and pattern recognition. The CIFAR-10 dataset (Krizhevsky, Hinton et al. 2009) is a well-known image dataset extensively used in the field of computer vision. It contains 60,000 32×32 color images divided into 10 different categories. The SVHN (Street View House Numbers) dataset (Netzer et al. 2011) is a publicly available large-scale dataset for digit recognition. Unlike the handwritten digits in the MNIST dataset, the digits in SVHN come from house numbers in Google Street View images.

Comparison methods We compared proposed method with mainstream filtering methods, including statistical methods and machine learning-based methods. Statistical methods include Relative Density Filter (RD) (Xia, Xiong et al. 2014) and Voting Mechanism Based on Relative Density (vRD) (Xia, Chen et al. 2021). Machine learning-based methods include Mutual Nearest Neighbor Filter (MNN) (Liu and Zhang 2012), Majority Voting Filter (MVF) (Brodley and Friedl 1999), Complete Random Forest Filter (CRF) (Xia et al. 2018) and Adaptive Complete Random Forest Filter (Adap_mCRF) (Huang, Shao, and Peng 2022).

Noise addition methods and experimental framework We perform extensive experiments on the MNIST, CIFAR-10, and SVHN datasets to demonstrate the effectiveness of our method. To validate the filtering performance of the proposed method, we first added completely random noise at proportions of 5%, 10%, 15%, and 20% to each training set. Subsequently, we applied the proposed filtering method as well as mainstream filtering methods to remove label noise and compared various noise filtering metrics. Finally, we trained classifiers using the denoised training sets and evaluated their generalization performance on the test sets. To reduce the randomness of the experimental results, each scenario with different random noise levels was tested 10 times, and the average results were taken as the final experimental outcomes.

	True	
	Normal	**Noise**
Predicted Normal	TP	FP
Predicted Noise	FN	TN

Table 1: The confusion matrix

Evaluation metrics To validate the effectiveness of the proposed method, multiple evaluation metrics are used to measure the method's noise filtering capability and classification generalization ability. These metrics include Accuracy (Acc), Noise filtering accuracy (NfAcc), Precision (Pre), Recall (Re), $F1$ score, and Classification Accuracy (PreAcc). The definitions of these metrics are as follows:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

$$NfAcc = \frac{TN}{FN + TN} \tag{11}$$

$$Pre = \frac{TP}{TP + FP} \tag{12}$$

$$Re = \frac{TP}{TP + FN} \tag{13}$$

$$F1 = \frac{2TP}{2TP + FP + FN} \tag{14}$$

NR	Methods	MNIST					CIFAR-10					SVHN				
		Acc	NfAcc	Pre	Re	F1	Acc	NfAcc	Pre	Re	F1	Acc	NfAcc	Pre	Re	F1
5%	RD	49.8	5.1	94.9	49.8	64.2	30.6	5.0	94.9	28.5	43.8	28.7	5.5	96.6	25.9	40.8
	vRD	76.3	17.3	99.9	75.1	85.8	30.1	5.6	96.5	27.3	42.6	28.8	5.5	96.3	26.1	41.0
	MNN	59.6	11.1	**100.0**	57.5	73.0	17.0	5.7	**99.8**	12.6	22.4	24.3	6.2	**99.7**	20.3	33.8
	MVF	35.9	6.8	98.8	33.0	49.4	33.9	6.6	98.7	30.8	46.9	41.3	7.4	99.0	38.6	55.5
	CRF	75.8	6.5	95.4	78.3	86.0	73.9	5.5	95.1	76.4	84.7	74.5	7.3	95.7	76.6	85.1
	Adap_mCRF	67.9	13.5	**100.0**	66.2	79.6	56.8	6.5	96.1	56.9	71.4	54.4	5.0	95.0	54.8	69.5
	FLM-LNF	**99.2**	**86.4**	**100.0**	**99.1**	**99.6**	**98.0**	**82.1**	98.8	**99.1**	**98.9**	**98.8**	**84.9**	99.5	**99.1**	**99.3**
10%	RD	49.7	10.2	89.9	49.6	63.2	33.2	10.0	89.8	29.1	43.4	34.2	11.1	92.5	29.2	44.3
	vRD	76.4	29.5	99.7	74.0	84.9	32.5	11.0	92.7	27.2	42.0	31.5	11.0	92.2	25.9	40.5
	MNN	58.1	19.3	**99.9**	53.5	69.7	20.7	11.2	**99.4**	11.9	21.2	27.1	12.0	**99.3**	19.1	32.0
	MVF	35.5	12.7	97.0	29.2	44.9	36.3	12.8	97.1	30.1	46.0	42.8	14.2	97.8	37.3	54.0
	CRF	74.0	13.1	90.8	79.1	84.6	71.2	11.0	90.3	76.2	82.7	72.5	13.6	91.1	77.0	83.4
	Adap_mCRF	73.9	27.6	99.8	71.1	83.0	56.3	12.5	92.0	56.4	69.9	53.3	10.1	90.2	54.0	67.5
	FLM-LNF	**96.5**	**87.1**	97.4	**98.8**	**98.1**	**96.6**	**88.2**	97.4	**98.9**	**98.1**	**98.3**	**92.9**	98.8	**99.2**	**99.0**
15%	RD	54.1	15.9	85.1	55.5	65.9	34.7	15.0	84.8	28.3	41.7	34.6	16.2	87.9	26.6	40.7
	vRD	76.2	38.5	99.3	72.5	83.8	35.0	16.3	88.6	26.9	41.3	33.7	15.8	88.1	25.6	39.7
	MNN	57.0	25.8	99.8	49.5	66.2	24.4	16.5	**99.0**	11.2	20.1	30.1	17.6	**98.8**	17.9	30.3
	MVF	39.3	18.9	95.6	30.0	45.6	38.9	18.8	95.4	29.5	45.1	44.2	20.3	96.4	35.6	52.0
	CRF	71.6	19.2	86.4	79.0	82.6	68.5	16.2	85.4	76.0	80.4	70.3	19.3	86.2	77.3	81.5
	Adap_mCRF	78.2	40.5	99.1	75.0	85.4	55.8	18.2	87.7	55.9	68.3	54.2	15.2	85.0	56.0	67.5
	FLM-LNF	**99.1**	**94.8**	**100.0**	**99.0**	**99.5**	**94.9**	**90.3**	95.6	**98.6**	**97.1**	**97.5**	**94.7**	97.9	**99.1**	**98.5**
20%	RD	52.8	20.8	79.9	54.4	63.4	37.2	20.0	79.4	28.9	41.2	39.2	21.5	83.5	29.8	43.7
	vRD	76.0	45.4	98.8	70.9	82.5	37.4	21.5	84.2	26.7	40.5	36.5	21.1	84.1	25.6	39.3
	MNN	56.4	31.4	99.5	45.7	62.6	28.2	21.7	**98.2**	10.4	18.8	33.2	22.9	**98.0**	16.8	28.7
	MVF	41.0	24.3	93.2	28.3	43.4	41.5	24.5	93.6	28.9	44.1	44.9	25.6	94.5	33.1	49.0
	CRF	70.3	26.2	81.3	81.6	81.4	66.1	21.6	80.5	76.0	78.2	68.1	25.3	81.6	77.6	79.5
	Adap_mCRF	80.1	50.2	97.9	76.8	86.1	57.3	24.1	83.1	58.5	68.6	53.1	20.1	80.0	55.2	65.3
	FLM-LNF	**99.4**	**96.9**	**100.0**	**99.3**	**99.6**	**93.6**	**88.8**	94.6	**97.5**	**96.0**	**96.8**	**95.8**	97.0	**99.0**	**98.0**

Table 2: Noise filtering experimental results (%)

$$\mathrm{PreAcc} = \frac{\text{The number of samples correctly classified}}{\text{Total number of samples}} \tag{15}$$

Among them, TP (True Positive), FP (False Positive), FN (False Negative) and TN (True Negative) are defined as shown in Table 1. Among the six evaluation metrics, the first five metrics are used to evaluate the method's noise recognition performance, in which the higher values of Acc, NfAcc, Pre, Re and F1 indicate that the method's noise recognition performance is better. The classification accuracy (PreAcc) is used to measure the generalization performance of the classifier.

Label Noise filtering Experiments

Setting In the experiments, four noise ratios were added to the training sets of the MNIST, CIFAR-10, and SVHN datasets. For the MNIST dataset, the FLM employed a 7-layer convolutional neural network as the feature extraction network. For the CIFAR-10 and SVHN datasets, ResNet-18 (Fang, Yu et al. 2021) was used as the feature extraction network. The fuzzy parameters were set to $\alpha = 0.2$ and $\beta = 0.8$. The model was optimized using the Adam optimizer with a learning rate of 0.001. The threshold γ was set to 0.7, and the batch-size was set to 2048. Iterative training continued until there was no significant change in the loss value for 10 consecutive epochs.

Experimental results and analysis The results of the six metrics of the noise recognition experiments of each method under four noise ratios are shown in Table 2.

From the Acc and the NfAcc, it is evident that the noise recognition capability of our method significantly surpasses that of other methods across noise ratios ranging from 5% to 20%. Other methods tend to filter out noise samples more aggressively, sometimes resulting in the removal of clean samples and leading to over-filtering in severe cases. Moreover, our method's recall rate is markedly higher than that of other methods. This is attributed to our method's ability to retain genuine samples as much as possible, minimizing the erroneous deletion of true samples. When comparing the average F1 scores, which reflect a balance between precision and recall, our method consistently achieves significantly higher F1 values than other methods, maintaining a stable performance above 0.95. This underscores the comprehensive effectiveness of our method. In summary, our method shows a clear superiority in all five indicators under noise ratios of 5%-20%, demonstrating strong noise recognition capabilities. The relatively low performance of other meth-

Data Set	NR	Methods							
		NOF	RD	vRD	MNN	MVF	CRF	Adap_mCRF	**FLM-LNF**
MNIST	5%	93.4	85.6	84.5	96.3	57.8	92.4	64.6	**97.6**
	10%	89.1	84.7	81.2	95.4	55.4	92.3	61.6	**97.6**
	15%	84.9	81.4	67.6	96.1	55.7	91.5	58.6	**97.5**
	20%	78.6	71.8	62.9	95.3	56.4	90.6	54.9	**97.5**
CIFAR-10	5%	61.4	29.2	59.2	49.1	48.0	52.0	61.5	**66.1**
	10%	58.2	26.5	55.5	48.4	47.2	49.2	52.7	**64.7**
	15%	53.9	24.0	52.7	47.1	46.0	45.5	49.6	**63.0**
	20%	51.1	20.5	48.2	46.6	44.8	45.0	49.6	**62.9**
SVHN	5%	88.0	29.6	86.3	76.7	77.3	76.5	84.7	**88.0**
	10%	85.8	46.9	83.0	73.5	77.2	70.5	81.4	**90.7**
	15%	81.8	45.0	81.8	73.3	72.0	66.3	76.9	**89.7**
	20%	76.3	42.7	78.6	67.1	70.5	61.0	61.5	**89.3**

Table 3: Generalization performance of classifiers (%) after noise filtering

ods indicates their disadvantages are more pronounced on large-scale datasets.

Noise Correction Experiments

We filter noise by comparing the consistency between predicted and observed labels. The final predicted labels derived from our two-stage filtering framework are utilized as corrected labels. When the predicted label matches the observed label, it is considered a correct correction. Moreover, we have experimentally validated the efficacy of the two-stage FLM in rectifying label noise, demonstrating our method's effective correction capabilities. The noise correction accuracy of our method at various noise ratios is shown in the following Table 4.

Data Set	NR			
	5%	10%	15%	20%
MNIST	65.4	72.6	97.2	98.0
CIFAR-10	60.5	62.2	63.5	63.8
SVHN	86.4	86.8	87.6	88.3

Table 4: Label noise correction accuracy (%)

Experiments on the Generalization Performance of Classifiers

Setting The classifier used to evaluate the generalization performance on the MNIST dataset employs a three-layer neural network structure. This network includes two hidden layers with 500 and 300 neurons, respectively, designed to process the 784-dimensional input data and perform classification across 10 categories. For the CIFAR-10 and SVHN datasets, the ResNet-18 architecture was utilized as the classifier to validate generalization performance.

Experimental results and analysis Table 3 presents the classification accuracy results of training a classifier on datasets filtered using various methods and then testing on a clean dataset. A higher PreAcc indicates that the dataset filtered by the method improves the classification ability of the trained classifier. The data in the table demonstrate that our method achieves better classification accuracy compared to other methods in most cases and significantly outperforms results obtained without any filtering. Additionally, the classification accuracy of other filtering methods on the CIFAR-10 and SVHN datasets is slightly lower than the unfiltered dataset, suggesting that these methods may have mistakenly removed a substantial number of samples.

Conclusions

This paper proposes using Fuzzy Learning Machine (FLM) for the task of label noise correction. By fully leveraging its capabilities in feature representation and noise tolerance, the FLM robustly represents concepts through exemplar sets, thereby providing a method to assess the cleanliness of label for samples. Compared to existing methods, this method is better equipped to handle large datasets and the challenges posed by insufficient representation capabilities in the original sample space. It filters out noise that significantly impacts classifiers more precisely, while minimizing information loss, thus enhancing the generalization performance of classifiers. Furthermore, the proposed two-stage learning framework based on the FLM ensures outstanding robustness. The numerous experiments show that the proposed method not only has significant advantages in label noise filtering, but also can correct incorrect labels effectively, making it a promising method to address the challenge of label noise.

Acknowledgements

This work is supported by the National Science and Technology Major Project (2020AAA0106102), National Natural Science Foundation of China (62376141), and UK_China Joint Laboratory of Security and Control on Smart Energy (202104041101020).

References

Barandela, R.; and Gasca, E. 2000. Decontamination of training samples for supervised pattern recognition methods. In *Proceedings of the Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition and Structural and Syntactic Pattern Recognition*, 621–630.

Bi, Y.; and Jeske, D. R. 2010. The efficiency of logistic regression compared to normal discriminant analysis under class-conditional classification noise. *Journal of Multivariate Analysis*, 101(7): 1622–1637.

Breve, F. A.; Zhao, L.; and Quiles, M. G. 2015. Particle competition and cooperation for semi-supervised learning with label noise. *Neurocomputing*, 160: 63–72.

Brodley, C. E.; and Friedl, M. A. 1999. Identifying mislabeled training data. *Journal of Artificial Intelligence Research*, 11: 131–167.

Cui, J.; and Liang, J. 2022. Fuzzy learning machine. In *Proceedings of the Annual Conference on Neural Information Processing Systems*, 36693–36705.

Esteva, A.; Kuprel, B.; Novoa, R. A.; et al. 2017. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639): 115–118.

Fang, W.; Yu, Z.; et al. 2021. Deep residual learning in spiking neural networks. In *Proceedings of the Annual Conference on Neural Information Processing Systems*, 21056–21069.

Frénay, B.; and Verleysen, M. 2013. Classification in the presence of label noise: a survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5): 845–869.

Ghosh, A.; Kumar, H.; and Sastry, P. S. 2017. Robust loss functions under label noise for deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 1919–1925.

Grigorescu, S.; Trasnea; et al. 2020. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3): 362–386.

Han, B.; Yao, Q.; et al. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Proceedings of the Annual Conference on Neural Information Processing Systems*, 8536–8546.

Huang, L.; Shao, Y.; and Peng, J. 2022. An adaptive voting mechanism based on relative density for filtering label noises. In *Proceedings of the IEEE International Conference on Electronics Technology*, 1327–1331.

Jiang, L.; Zhou, Z.; Leung, T.; et al. 2018. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *Proceedings of the International Conference on Machine Learning*, 2304–2313.

Kermany, D. S.; Goldbaum; et al. 2018. Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*, 172(5): 1122–1131.

Khoshgoftaar, T. M.; and Van Hulse, J. 2005. Identifying noisy features with the pairwise attribute noise detection algorithm. *Intelligent Data Analysis*, 9(6): 589–602.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Lecun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Lee, K.; Yun, S.; et al. 2019. Robust inference via generative classifiers for handling noisy labels. In *Proceedings of International Conference on Machine Learning*, 3763–3772.

Liu, H.; and Zhang, S. 2012. Noisy data elimination using mutual k-nearest neighbor for classification mining. *Journal of Systems and Software*, 85(5): 1067–1074.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; Ng, A. Y.; et al. 2011. Reading digits in natural images with unsupervised feature learning. In *Proceedings of the Annual Conference on Neural Information Processing Systems Workshop on Deep Learning and Unsupervised Feature Learning*.

Sánchez, J. S.; Barandela, R.; Marqués, A. I.; Alejo, R.; and Badenas, J. 2003. Analysis of new techniques to obtain quality training sets. *Pattern Recognition Letters*, 24(7): 1015–1022.

Sluban, B.; Gamberger, D.; and Lavra, N. 2010. Advances in class noise detection. In *Proceedings of the European Conference on Artificial Intelligence*, 1105–1106.

Tan, C.; Xia, J.; Wu, L.; and Li, S. Z. 2021. Co-learning: Learning from noisy labels with self-supervision. In *Proceedings of the ACM International Conference on Multimedia*, 1405–1413.

Tanno, R.; Saeedi, A.; et al. 2019. Learning from noisy labels by regularized estimation of annotator confusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11244–11253.

Wang, Y.; Ma, X.; et al. 2019. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 322–330.

Xia, S.; Chen, B.; et al. 2021. mCRF and mRD: Two classification methods based on a novel multiclass label noise filtering learning framework. *IEEE Transactions on Neural Networks and Learning Systems*, 33(7): 2916–2930.

Xia, S.; Wang, G.; Chen, Z.; Duan, Y.; et al. 2018. Complete random forest based class noise filtering learning for improving the generalizability of classifiers. *IEEE Transactions on Knowledge and Data Engineering*, 31(11): 2063–2078.

Xia, S.; Xiong, Z. y.; et al. 2014. Relative density-based classification noise detection. *Optik*, 125(22): 6829–6834.

Yu, X.; Han, B.; Yao, J.; Niu, G.; Tsang, I.; and Sugiyama, M. 2019. How does disagreement help generalization against label corruption? In *Proceedings of the International Conference on Machine Learning*, 7164–7173.

Zhu, X.; and Wu, X. 2004. Class noise vs. attribute noise: A quantitative study. *Artificial Intelligence Review*, 22: 177–210.