Training Dataset Curation by L_1-Norm Principal-Component Analysis for Support Vector Machines

Shruti Shukla, *Graduate Student Member, IEEE*, Dimitris A. Pados, *Senior Member, IEEE*, George Sklivanitis, *Member, IEEE*, Elizabeth Serena Bentley, *Member, IEEE*, and Michael J. Medley, *Senior Member, IEEE*

***Abstract*—Support vector machines (SVMs) have been the learning model of choice in numerous classification applications. While SVMs are widely successful in real-world deployments, they remain susceptible to mislabeled examples in training datasets where the presence of few faults can severely affect decision boundaries, thereby affecting the model's performance on unseen data. In this brief, we develop and describe in implementation detail a novel method based on L_1-norm principal-component data analysis and geometry that aims to filter out atypical data instances on a class-by-class basis before the training phase of SVMs and thus provide the classifier with robust support-vector candidates for making classification boundaries. The proposed dataset curation method is entirely data-driven (touch-free), unsupervised, and computationally efficient. Extensive experimental studies on real datasets included in this brief illustrate the L_1-norm curation method and demonstrate its efficacy in protecting SVM models from data faults during learning.**

***Index Terms*—L_1-norm, dataset curation, faulty data, mislabeled data, outlier resistance, principal-component analysis (PCA), rank selection, support vector machines (SVMs).**

I. Introduction

Support vector machines (SVMs) are introduced [1] as a learning classification algorithm designed to maximize the margin between class training examples and decision boundaries. SVMs attain broadly high generalization performance by matching their adjustable parameters with the size of the available training set preventing over- and underfitting [2]. The derived classification function considers just few data points from a linearly separable training dataset referred to as support vectors. The support vectors are the training examples closest to the decision boundary (usually a small subset of the training data) and have direct bearing on its location. Depending on the number of the input data features (dimensionality), a decision boundary can be a point (1-D datasets), a line (2-D datasets), a plane (3-D datasets), or a hyperplane (four or more data dimensions) [3], [4]. SVM classifiers can also deal effectively with nonlinearly separable datasets. When linear boundaries are not deemed appropriate to separate the patterns, input data are mapped to a new subspace (generally of higher dimension) using a kernel function to make the dataset linearly separable. Kernel SVMs have produced highly popular, widely used classifiers in various applications such as pattern recognition, image classification, face detection, text categorization, and time-series data analysis including medical diagnosis and prognosis [5], [6], [7], [8].

The performance of all the data-driven machine learning classifiers including SVMs is tightly regulated by the quality of the training data. In particular, the presence of a few atypical/faulty/noisy data in the training dataset can affect the decision boundaries created by the model and drastically inhibit the performance of the classifier on unseen data [9], [10], [11], [12]. Multiple comparative studies have revealed that depending upon the type and level of noise contamination, different machine learning classification algorithms can have different sensitivity to the irregularities present in training datasets [13], [14], [15]. Noise in a dataset can manifest itself in various ways. Discrepancies due to measurement technique inconsistencies and sensor hardware limitations or during automated or human-expert labeling are few examples. Correspondingly, dataset noise can be present in the form of feature noise, which refers to noise in the values of features/attributes of the training samples, or label noise which refers to class/label faults of the samples. It has been pointed out that label noise can have indelible impact on learning in the models and prove disastrous to classification algorithms as accidental (or intentional) flip in the label training values can induce model bias and lower drastically the generalization power of the classifiers [16], [17], [18]. Eliminating/suppressing label noise instances is expected to enhance significantly classification accuracy [19]. SVMs are particularly susceptible to mislabeling during the training phase as decision boundaries produced by SVMs come directly from small subsets of training samples. When data points belonging to the small subset of support vectors are fallacious, the decision boundaries can become severely flawed and lead to significant increase in misclassification of operational input data.

Several methods have been developed to deal with noisy training data labels. These methods can be organized into three main categories [20]. The first category involves building classifiers that are robust against noise without taking into consideration the underlying nature and model of noise. In [21], a methodology is presented where clustering analysis is used to create classification boundaries for robust SVM operation. Another technique based on the flipping probability of label noise and a logistic regression classifier that uses both noisy and auxiliary less-noisy labels to learn a classifier is presented in [22]. The second category of methods involves building a noise model along with a classification model to develop noise-tolerant classifiers [20]. The third approach is removing the outliers and noisy data from the datasets. To combat the adverse effects of label noise, it has been demonstrated that curating the training data and successfully eliminating likely mislabeled instances produces, with high probability, classifiers of superior predictive accuracy [17], [19]. It can be noted that label noise reduction is closely related to the process of outlier detection and excision. The occurrences of mislabeled data for a class or label can be considered as outlying data as they have low odds of existence in that class [16]. Various distance-based, density-based, and clustering-based outlier detection

Received 13 February 2024; revised 27 September 2024 and 26 January 2025; accepted 6 May 2025. This work was supported in part by the National Science Foundation under Grant CNS-2117822 and Grant EEC-2133516, in part by the Air Force Research Laboratory under Grant FA8750-21-F-1012, and in part by the Air Force Office of Scientific Research under Grant W911NF-20-1-0283. Distribution A. Approved for public release: Distribution Unlimited: AFRL-2024-1028 on 26 Feb 2024. *(Corresponding author: Dimitris A. Pados.)*

Shruti Shukla, Dimitris A. Pados, and George Sklivanitis are with the Center for Connected Autonomy and AI and the Department of Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL 33431 USA (e-mail: sshukla2020@fau.edu; dpados@fau.edu; gsklivanitis@fau.edu).

Elizabeth Serena Bentley is with the Air Force Research Laboratory, AFRL/RI, Rome, NY 13441 USA (e-mail: elizabeth.bentley.3@us.af.mil).

Michael J. Medley is with the SUNY Polytechnic Institute, Utica, NY 13502 USA (e-mail: michael.medley@sunypoly.edu).

Digital Object Identifier 10.1109/TNNLS.2025.3568694

methods have been used as label noise detection techniques [23]. Data undersampling (a form of indirect excision) has also been used effectively to deal with imbalanced classification problems [24], [25].

In this brief, we focus specifically on SVM classifiers and propose a novel training dataset curation method that is built on robust L_1-norm subspace representation (summarization) [26], [27] of individual classes and L_1-norm data-point distances. In contrast to conventional L_2-norm subspace representation methods (i.e., singular-vector-decomposition-based) that place square emphasis on the amplitude of data points, the L_1-norm methods operate on the absolute value of data and are known to be inherently robust in the presence of outliers [28]. The developed method identifies and eliminates on a class-by-class basis data in training sets of SVMs that do not seem to be conforming with the rest and may not be suitable contenders to become a support vector for the class. The technical novelty of the method is summarized as follows.

1) For the first time in the literature, class datasets are characterized by joint L_1-norm maximum projection computed subspaces, and
2) distance of individual class data points from a class subspace is L_1-norm-computed.
3) For the first time in the literature, optimal class subspace rank selection is embedded in the outlier excision process.

The operational highlights of the developed procedure for dataset curation are summarized as follows.

1) Robust subspace summarization of individual classes and identification of outlying data points class by class.
2) Binary and multiclass classification training dataset curation.
3) Zero-touch dataset curation with no tunable parameters or human operator.
4) Seamless integration as pretraining step across all forms of SVMs including recent and future advanced variants.

Extensive tests on real and synthetically contaminated data (with different levels of label noise) presented in this brief illustrate the theoretical developments and operational properties and demonstrate consistently notable SVM classification improvement.

The rest of this brief is organized as follows. Section II introduces the general classification model and notation. Section III presents in algorithmic implementation detail the proposed training dataset curation method. Section IV is devoted to extensive experimental studies and comparisons. Finally, Section V summarizes the scientific findings and discusses possible future work.

Notation: In this brief, matrices are denoted by upper case bold letters, column vectors by lower case bold letters, and scalars by lower case plain-font letters. The transpose operation is represented by the superscript $^{\mathrm{T}}$.

II. Data Model and Notation

We consider a general classification problem where we collect D-dimensional real-valued data samples $\mathbf{x} \in \mathbb{R}^D$ and we want to decide their class of origin among $L \geq 2$ different class alternatives. The only guidance that we have is one set of examples from each class that we organize in the form of individual matrices

$$\mathbf{X}^{(l)}_{D\times N_l} = \left[\mathbf{x}^{(l)}_1 \quad \mathbf{x}^{(l)}_2 \quad \cdots \quad \mathbf{x}^{(l)}_{N_l}\right], \quad l = 1, \ldots, L \tag{1}$$

where N_l is the number of available examples from class l (sample support of class l). Collectively, the matrices $\mathbf{X}^{(l)}, l = 1, \ldots, L$, constitute our complete training dataset; if $\mathbf{x} \in \mathbf{X}^{(l)}$, we say that the label of $\mathbf{x}$ is l. Without loss of generality and for simplicity in our treatment, we assume that $N_l \geq D$ for each class $l = 1, \ldots, L$.

In broad mathematical notation, every machine learning classifier uses the dataset $\mathbf{X}^{(l)}, l = 1, \ldots, L$, to build a parametrically described function $f(\cdot)$ from $\mathbb{R}^D$ to $\{1, 2, \ldots, L\}$ such that every unseen $\mathbf{x}$ is classified to $f(\mathbf{x}; \mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(L)})$. For example, linear support vector machines (SVMs) solve binary hypothesis testing problems by looking at functions of the form $f(\mathbf{x}) = \mathrm{sign}(\mathbf{w}^{\mathrm{T}}\mathbf{x} + b)$ from $\mathbb{R}^D$ to $\{-1, 1\}$ where the weight vector $\mathbf{w}$ and the bias term b are chosen according to the given data examples $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$. Multiple hypothesis testing problems ($L > 2$) are usually solved by SVMs as a series of one-to-one or one-to-rest tests. Nonlinear SVM classifiers use "kernel" transforms $K(\mathbf{x})$ from $\mathbb{R}^D$ to $\mathbb{R}^{D'}$ with $D' > D$ and design classifiers of the form $f'(\mathbf{x}) = \mathrm{sign}(\mathbf{w}^{\mathrm{T}}K(\mathbf{x}) + b)$.

In this work, we deal exclusively with SVM classifiers and we are concerned with cases where the available training dataset of matrices $\mathbf{X}^{(l)}, l = 1, \ldots, L$, may be contaminated with faulty label entries, i.e., there are columns $\mathbf{x} \in \mathbb{R}^D$ of $\mathbf{X}^{(l)}$ where $\mathbf{x}$ are not truly coming from class l due to annotation or sensing error or other reason. In Section III, we describe a purely data-driven method (zero human touch) that curates the SVM training dataset $\mathbf{X}^{(l)}, l = 1, \ldots, L$, i.e., removes from each individual class examples that are not conforming with general class data characteristics and therefore are not good candidates to serve as support vectors. Conformity is evaluated by new robust L_1-norm principal-component analysis (PCA) (feature extraction) algorithms and L_1-norm distance geometry.

III. Algorithm for Training Dataset Curation

In Sections I and II, we discussed the importance of training machine learning classifiers with correctly labeled data and underlined the vulnerability of SVMs that presuppose correctly labeled support-vector selection to position decision boundaries. In this section, we propose a novel solution to filter out atypical data instances on a class-by-class basis before the training phase of SVMs and thus provide the classifier with robust support-vector candidates for making classification boundaries. The complete flow of the process is summarized in Fig. 1. The proposed algorithmic method occupies Steps 1–4 before training; the core mathematical developments fall under Steps 2 and 3.

In the sequel, we describe Steps 2 and 3 of Fig. 1 in complete implementation detail in the form of four data-driven optimization operations: 1) class-by-class L_1-norm data feature extraction; 2) excision threshold optimization; 3) optimal rank selection, and finally and 4) data curation.

A. L_1-Norm Data Feature Extraction

The first operation in the proposed method to curate training datasets for SVMs involves feature extraction on a class-by-class basis.

PCA has been proven instrumental in moving datasets into lower dimensions and encapsulating the information in the data with few projection vectors referred to as principal components (PCs) [29], [30]. PCA in its conventional equivalent forms of L_2-norm error minimization and L_2-norm projection maximization (i.e., L_2-norm PCA executed by the singular-vector decomposition algorithm) is known to be sensitive to the presence of outliers [28]. As a remedy, several robust PCA methods have been created and studied [31], [32], [33]. One example is to attempt direct L_1-norm PCA by L_1-norm projection maximization. This is a discrete mathematics (combinatorics) problem that was recently solved in [26] and [27].

In this context, for each class label $l = 1, 2, \ldots, L$, we consider the corresponding available dataset of N_l examples $\mathbf{X}^{(l)}_{D\times N_l}$ where D is the data sample dimension (without loss of generality $D \leq N_l$.) We are interested in the "summarization" (or feature extraction or PCA) of $\mathbf{X}^{(l)}_{D\times N_l}$ by a size-K orthonormal basis $\{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_K\}$, $1 \leq K \leq D-1$, calculated as follows:

$$\mathbf{Q}^{(l),(K)}_{L_1} = \underset{\substack{\mathbf{Q}\in\mathbb{R}^{D\times K}\\ \mathbf{Q}^{\mathrm{T}}\mathbf{Q}=\mathbf{I}_K}}{\arg\max} \left\|\mathbf{X}^{(l)^{\mathrm{T}}}\mathbf{Q}\right\|_1 \tag{2}$$

where $\mathbf{Q}^{(l),(K)}_{L_1} = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_K]$ is the rank-K basis in matrix form and $\mathbf{I}_K$ is the size K identity matrix. The analysis problem in (2) was solved: 1) exactly (optimally) by an exhaustive search algorithm of complexity $O(2^{N_l K})$ in [26]; 2) exactly (optimally) by a polynomial algorithm of complexity $O(N_l^{DK})$ in [26]; and 3) approximately with low complexity $O(N_l D^2 + N_l{}^2 K^2 (K^2 + D))$ by the bit-flipping algorithm in [27].

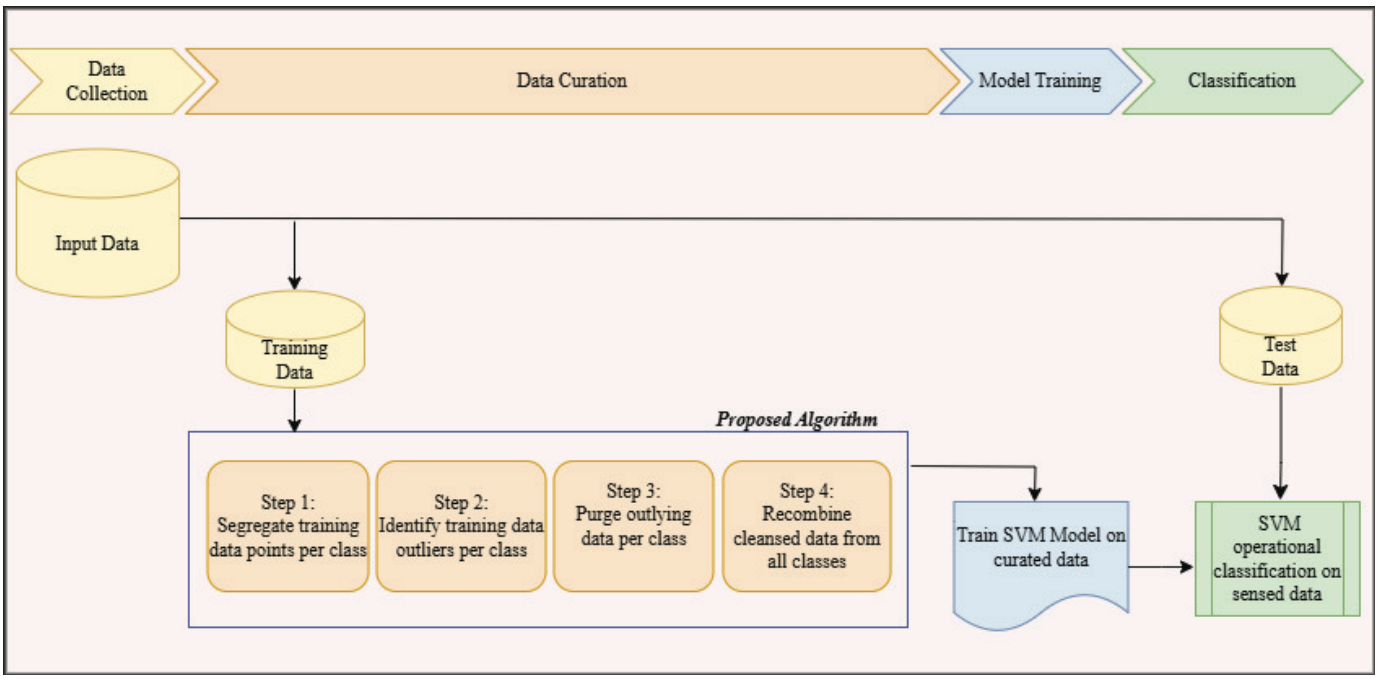

Fig. 1. Flowchart of the proposed SVM training dataset curation process.

Having calculated the rank K subspace representation of class l data, $\mathbf{Q}_{L_1}^{(l),(K)}$, we can now take each data instance $\mathbf{x}_n^{(l)}$, $n = 1, \ldots, N_l$, and measure its distance from $\mathbf{Q}_{L_1}^{(l),(K)}$ under the L_1-norm metric

$$d_n^{(l),(K)} = \left\| \mathbf{x}_n^{(l)} - \mathbf{Q}_{L_1}^{(l),(K)} \mathbf{Q}_{L_1}^{(l),(K)^{\mathrm{T}}} \mathbf{x}_n^{(l)} \right\|_1, \quad n = 1, \ldots, N_l. \tag{3}$$

Intuitively, data entries $\mathbf{x}_n^{(l)}$ with small distance value $d_n^{(l),(K)}$ are in the core of the examples from label l, while data entries with large distance value are peripheral/outlying examples that are suspect to be faulty labeled. Furthermore, we min–max normalize the calculated distance values by defining $d_{\min}^{(l),(K)} \triangleq \min_n d_n^{(l),(K)}$, $d_{\max}^{(l),(K)} \triangleq \max_n d_n^{(l),(K)}$, and

$$d_n^{(l),(K)'} \triangleq \frac{d_n^{(l),(K)} - d_{\min}^{(l),(K)}}{d_{\max}^{(l),(K)} - d_{\min}^{(l),(K)}} \tag{4}$$

and reindex in ascending order

$$\mathbf{d}^{(l),(K)'} = \begin{bmatrix} d_1^{(l),(K)'} & d_2^{(l),(K)'} & \cdots & d_{N_l}^{(l),(K)'} \end{bmatrix}^{\mathrm{T}}$$
$$0 \le d_1^{(l),(K)'} \le d_2^{(l),(K)'} \le \cdots \le d_{N_l}^{(l),(K)'} \le 1. \tag{5}$$

With this arrangement, suspect label examples are toward the bottom of the column vector $\mathbf{d}^{(l),(K)'} \in [0,1]^{N_l \times 1}$ with increasing likelihood. To exploit this property, we turn to the problem of selecting in a data-driven manner a distance threshold $\lambda^{(l),(K)} \in [0,1]$ above which examples with $d_n^{(l),(K)'} > \lambda^{(l),(K)}$ are to be excised from the class l dataset.

B. Excision Threshold Optimization

Given the L_1-norm rank K representation of class l data and their corresponding ordered L_1-norm min–max normalized distance sequence $0 \le d_1^{(l),(K)'} \le d_2^{(l),(K)'} \le \cdots \le d_{N_l}^{(l),(K)'} \le 1$, we propose a two-line-fit method [34] across the ordered distance points to determine the excision threshold. In particular, for every index value $p = 2, 3, \ldots, N_l - 1$, we fit one line on the data points to the left (L) of p (i.e., from 1 to p) and one line to the right (R) of p (i.e., from p to N_l) of the form

$$m_{L,p}^{(l),(K)} x + b_{L,p}^{(l),(K)} \tag{6}$$

and

$$m_{R,p}^{(l),(K)} x + b_{R,p}^{(l),(K)} \tag{7}$$

where, by standard least-squares linear regression,

$$m_{L,p}^{(l),(K)} = \frac{\sum_{i=1}^{p} i d_i^{(l),(K)'} - (1/p) \sum_{i=1}^{p} i \sum_{i=1}^{p} d_i^{(l),(K)'}}{\sum_{i=1}^{p} i^2 - (1/p) \left(\sum_{i=1}^{p} i \right)^2} \tag{8}$$

$$b_{L,p}^{(l),(K)} = \frac{\sum_{i=1}^{p} d_i^{(l),(K)'}}{p} - m_{L,p} \frac{\sum_{i=1}^{p} i}{p} \tag{9}$$

and, similarly, for the $N_l - p + 1$ number of data points to the right side of breakpoint p

$$m_{R,p}^{(l),(K)} = \frac{\sum_{i=p}^{N_l} i d_i^{(l),(K)'} - \frac{1}{N_l - p + 1} \sum_{i=p}^{N_l} i \sum_{i=p}^{N_l} d_i^{(l),(K)'}}{\sum_{i=p}^{N_l} i^2 - \frac{1}{N_l - p + 1} \left(\sum_{i=p}^{N_l} i \right)^2} \tag{10}$$

$$b_{R,p}^{(l),(K)} = \frac{\sum_{i=p}^{N_l} d_i^{(l),(K)'}}{N_l - p + 1} - m_{R,p} \frac{\sum_{i=p}^{N_l} i}{N_l - p + 1}. \tag{11}$$

We will now seek the breakpoint p that gives the best two-line fit in the sense of smallest sum of absolute-value errors $e_p^{(l),(K)}$ defined as

$$e_p^{(l),(K)} = \sum_{i=1}^{p} \left| \left(d_i^{(l),(K)'} - \left(m_{L,p} \cdot i + b_{L,p} \right) \right) \right| + \sum_{i=p}^{N_l} \left| \left(d_i^{(l),(K)'} - \left(m_{R,p} \cdot i + b_{R,p} \right) \right) \right|. \tag{12}$$

The selected breakpoint is found as

$$p_{\min}^{(l),(K)} = \underset{2 \le p \le N_l - 1}{\arg\min} \, e_p^{(l),(K)} \tag{13}$$

and the threshold value is set to

$$\lambda^{(l),(K)} = d_{p_{\min}}^{(l),(K)'}. \tag{14}$$

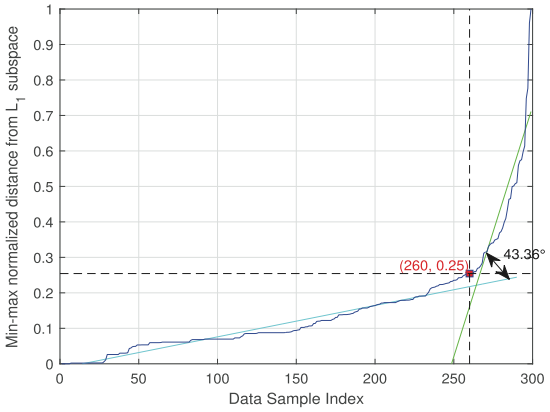

Fig. 2. Example of ordered min–max-normalized distance sequence of data points from their L_1-norm principal-component representation ("Breast Cancer Wisconsin" dataset [38], class "Benign," $D = 9$, $N_l = 299$, rank $K = 1$, breakpoint $p_{\min} = 260$, threshold value $\lambda = 0.25$, and angle $\theta = 43.36^o$).

C. Optimal Rank Selection

Once the threshold for each L_1-norm rank of interest $K = 1, 2, \ldots, D-1$ is determined for a class l, the best rank is identified as the one for which the breakpoint $p_{\min}^{(l),(K)}$ calculated above in (13) gives the maximum acute angle between the two least-square regression lines fit in (6) and (7). This selection is founded on the geometric notion that in the curve of $\mathbf{d}^{(l),(K)'}$, the larger the acute angle formed between the two fit regression lines corresponding to breakpoint $p_{\min}^{(l),(K)}$ for a given rank K, the steeper the change in the value of $d_{p_{\min}}^{(l),(K)'}$ at the elbow and, hence, the sharper the distinction between the outlying and conforming data. The acute angle formed between the two lines fit at breakpoint $p_{\min}^{(l),(K)}$ for rank K is

$$\theta^{(l),(K)} = \tan^{-1}\left|\frac{m_{R,p_{\min}}^{(l),(K)} - m_{L,p_{\min}}^{(l),(K)}}{1 + m_{R,p_{\min}}^{(l),(K)} \cdot m_{L,p_{\min}}^{(l),(K)}}\right| \in [0^o, 90^o]. \tag{15}$$

Fig. 2 offers a visual illustration of a min–max-normalized ordered distance sequence example in (5), its $p_{\min}$ breakpoint calculated by (13), and its angle θ measured by (15). Given $\theta^{(l),(K)}$ for ranks $K = 1, 2, \ldots, D-1$, we calculate the optimal rank for class l by

$$K_{\text{opt}}^{(l)} = \underset{1 \le K \le D-1}{\arg\max}\ \theta^{(l),(K)}, \quad l = 1, 2, \ldots, L. \tag{16}$$

D. Dataset Curation

Having obtained the optimized rank value $K_{\text{opt}}^{(l)}$ for every class $l = 1, 2, \ldots, L$ by (16), we curate each dataset $\mathbf{X}_{D \times N_l}^{(l)}$ by purging samples with min–max-normalized distance value greater than the threshold value $\lambda^{(l),(K_{\text{opt}})}$ in (14). That is, we remove samples with indexing in ascending distance value greater than or equal to $p_{\min}^{(l),(K_{\text{opt}})}$ in (13).

Fig. 3 offers a visual illustration of the excision process for a given dataset class and highlights comparatively the importance of applying the proposed robust L_1-norm PCs methodology over potentially conventional L_2-norm PCA (i.e., singular/eigenvector analysis).

Fig. 4 offers a visual illustration of the effect of the proposed dataset curation method on the selection of support vectors and the subsequently created decision boundaries.

The developed multistep procedure described above to curate training datasets of SVMs is summarized in Table I for easy reference. Complete coded implementation can be found and downloaded for execution in [36]. If L_1-norm PCA is carried out by bit-flipping [27], the overall worst case computational complexity of the dataset curation algorithm is dominated by the term $\sum_{K=1}^{D-1}[ND^2 + N^2K^2(K^2 + D)]$ where N is the maximum training sample size across classes, i.e., $N \triangleq \arg\max_{1 \le l \le L} N_l$, which simplifies to $O(ND^3 + 2N^2D^4 + N^2D^5)$. If we upper bound the rank optimization search in (16) to $1 \le K \le T$ for some $1 \le T < D-1$, then the computational complexity of the dataset curation algorithm reduces to $O(ND^3 + N^2(DT^3 + T^4 + T^5))$.

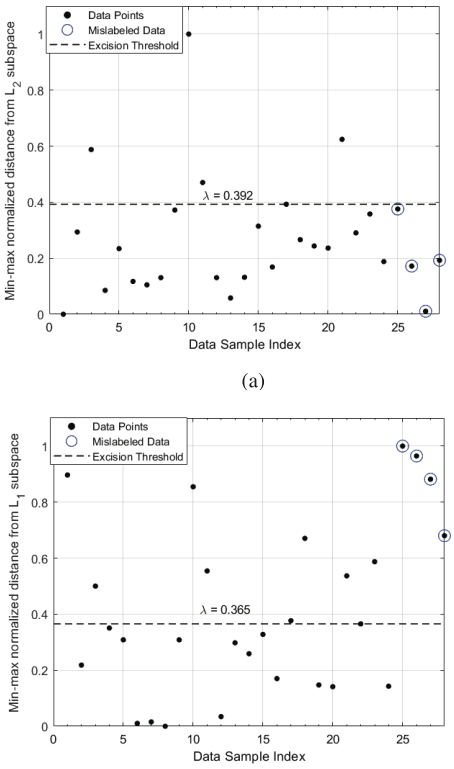

Fig. 3. Data point excision under (a) L_2-norm and (b) L_1-norm PCA ("Iris" dataset [37], class "Virginica" with 10% contamination from class "Versicolor."). L_1-norm-PCA-equipped data curation correctly removes all four mislabeled data points, while L_2-norm data curation fails.

TABLE I
TRAINING DATASET CURATION FOR SVM CLASSIFIERS BY L_1-NORM PCA

for $l = 1, 2, \ldots, L$ **do**
  for $K = 1, \ldots, D-1$ **do**
    Find $\mathbf{Q}_{L_1}^{(l),(K)} \in \mathbb{R}^{D \times K}$ by (2).
    Compute $d_n^{(l),(K)}$, $n = 1, 2, \ldots, N_l$, by (3) and min-max normalize to $d_n^{(l),(K)'}$ by (4).
    Re-index $d_n^{(l),(K)'}$ in ascending order to form $\mathbf{d}^{(l),(K)'}$ in (5).
    Fit left-and-right least-square regression lines around $\mathbf{d}^{(l),(K)'}[p]$, $p = 2, 3, \ldots, N_l - 1$, by (6), (7).
    Compute sum of absolute error $e_p^{(l),(K)}$, $p = 2, 3, \ldots, N_l - 1$, in (12).
    Obtain breakpoint $p_{\min}^{(l),(K)} = \underset{2 \le p \le N_l - 1}{\arg\min}\ e_p^{(l),(K)}$
    Determine threshold $\lambda^{(l),(K)} = d_{p_{\min}}^{(l),(K)'}$.
    Calculate $\theta^{(l),(K)}$ by (15).
  end for
  Calculate optimal rank $K_{\text{opt}}^{(l)}$ by (16).
  Remove samples $\mathbf{x}_n^{(l)}$ from $\mathbf{X}_{D \times N_l}^{(l)}$ for which $d_n^{(l),(K_{opt})'} > \lambda^{(l),(K_{opt})}$, $n = 1, 2, \ldots, N_l$.
end for
Output: Curated training dataset $\mathbf{X}^{(l)}$ for each class l

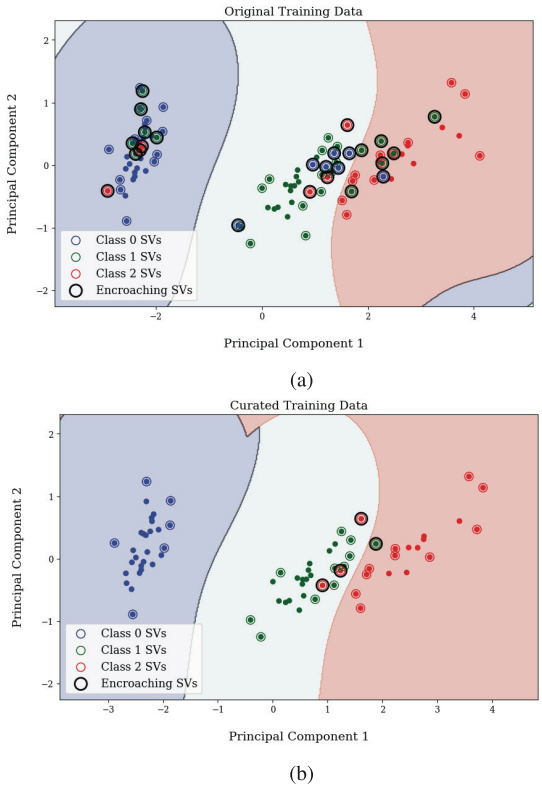

Fig. 4. Comparison of support vectors selected (a) before and (b) after the proposed training dataset curation visualized in standard 2-D-feature space (three-class "Iris" dataset [37] with 25% cross-class mislabeling, radial basis function SVM). Misclassification error was reduced from 7.55% to 1.89%.

IV. Experimental Studies and Comparisons

In this section, we apply the developed training dataset curation algorithm to real-world datasets. In all the experiments, we execute the code in [36] using MATLAB R2023b and Python 3.10.12 on a system equipped with an Intel64 Family 6 Model 166 Stepping 0 processor operating at clock speed of 1105 MHz with 16 GB of RAM.

A. Raw Versus Curated Training Data

We consider four publicly available datasets, the MNIST Database of Handwritten Digits [35], the Iris Dataset [37], the Breast Cancer Wisconsin Dataset [38], and the Wine Dataset [39]. We evaluate and compare the performance of SVM classification models when trained on raw data and curated data. As performance evaluation metrics, we use the SVM model's class confusion matrix as well as class-specific power probability and false alarm rate values when relevant. Below, we describe the four dataset experiments and the obtained results.

1) Mnist Database of Handwritten Digits: MNIST is a database of 70 000 examples of handwritten digits $0, 1, \ldots, 9$ [35]. In vector representation, each data point has dimensionality $D = 784$ (number of pixels). For our study, we isolated about 13 000 available records combined of handwritten digits "1" and "7" to carry out SVM design with polynomial kernel under 70%/30% training/testing split. All the classification performance results presented below are averages over ten independent training/testing splits. L_1-norm rank selection in class data curation [K_{opt} in (16)] was sought among ranks $K = 1, 2, 3$ (i.e., $T = 3$) for rapid execution.

In Table II, we present side by side the binary classification confusion matrices of the SVMs when trained on raw or curated data. It is interesting to observe that although MNIST is a high quality dataset with no known label faults/errors, the developed training dataset curation algorithm improved correct classification of "seven" from 0.97 to 0.98.

TABLE II
Confusion Matrix for the MNIST Dataset Without Induced Label Noise

	Raw Training Data		Curated Training Data	
	'One'	'Seven'	'One'	'Seven'
'One'	0.99	0.01	0.99	0.01
'Seven'	0.03	0.97	0.02	0.98

TABLE III
Confusion Matrix for the MNIST Dataset With 20% Induced Label Noise

	Raw Training Data		Curated Training Data	
	'One'	'Seven'	'One'	'Seven'
'One'	0.82	0.18	0.84	0.16
'Seven'	0.20	0.80	0.06	0.94

TABLE IV
Confusion Matrix for the Iris Dataset Without Induced Label Noise

	Raw Training Data			Curated Training Data		
	Sentosa	Versicolor	Virginica	Sentosa	Versicolor	Virginica
Sentosa	1	0	0	1	0	0
Versicolor	0.01	0.93	0.06	0	0.94	0.06
Virginica	0	0.04	0.96	0	0.02	0.98

To examine the performance of the developed curation dataset method in the presence of faulty training data labels, in Table III we repeat the study of Table II under 20% label noise contamination of the training set, i.e., 20% of the data samples in each class of the training dataset are randomly changed to the label of the other class. The effect of the training data curation algorithm is significant. Under data curation, the probability of correct classification of "seven" rises from 0.80 to 0.94, while the probability of correct classification of "one" increases from 0.82 to 0.84.

The studies demonstrate the strong impact of the proposed data curation method on model performance under training with corrupted data and underscore the ability of the proposed algorithm to lead to gains in model accuracy even when initial data quality is presumed to be high.

2) Iris Dataset: The Iris Dataset [37], [40] categorizes Iris flowers into three subspecies classes: Sentosa, Versicolor, and Virginica. The dimensionality of each data point is $D = 4$ where the four attributes identify length and width of sepals and petals in centimeters. The dataset consists of combined samples from the three subspecies of Iris totaling 150 data points. We carry out SVM design using a radial basis function kernel and evaluate experiments under 70%/30% training/testing splits. All the results presented below are averages over ten independent training/testing splits. L_1-norm rank selection in class data curation [K_{opt} in (16)] is sought among all ranks $K = 1, 2, 3$ (i.e., $T = 3$.)

In Table IV, we present side by- side the classification confusion matrices of the SVMs when trained on raw or curated data. Under raw (and curated) data, "Sentosas" are always classified correctly. Training dataset curation improves correct classification of "Versicolors" from 0.93 to 0.94 and "Virginicas" from 0.96 to 0.98. Dataset curation eliminates misclassification of "Versicolors" as "Sentosas" and reduces the probability of misclassifying "Virginicas" as "Versicolors" from 0.04 to 0.02.

In Table V, we repeat the same study of Table IV under 10% label noise contamination, that is, the labels of 10% of data in each class

TABLE V

CONFUSION MATRIX FOR THE IRIS DATASET WITH 10% INDUCED LABEL NOISE

	Raw Training Data			Curated Training Data		
	Sentosa	Versicolor	Virginica	Sentosa	Versicolor	Virginica
Sentosa	1	0	0	1	0	0
Versicolor	0.02	0.94	0.04	0	0.96	0.04
Virginica	0	0.05	0.95	0	0.02	0.98

TABLE VI

CONFUSION MATRIX FOR THE BREAST CANCER WISCONSIN DATASET

	Raw Training Data		Curated Training Data	
	Benign	Malignant	Benign	Malignant
Benign	0.96	0.04	0.96	0.04
Malignant	0.04	0.96	0.02	0.98

TABLE VII

CONFUSION MATRIX FOR THE WINE DATASET WITHOUT INDUCED LABEL NOISE

	Raw Training Data			Curated Training Data		
	Cultivar1	Cultivar2	Cultivar3	Cultivar1	Cultivar2	Cultivar3
Cultivar1	0.99	0.01	0	0.99	0.01	0
Cultivar2	0.01	0.98	0.01	0	0.99	0.01
Cultivar3	0	0.03	0.97	0	0.03	0.97

are randomly changed to labels of the two other classes. Algorithmic curation of the faulty (label noisy) training dataset offers across the board classification performance improvement, reinstating or even exceeding precontamination performance levels.

3) Breast Cancer Wisconsin Dataset: The Breast Cancer Wisconsin Dataset in the UCI repository [38], [40] has data dimensionality $D = 9$ (the index attribute "sample code number" is disregarded). Each data point represents breast cell sample measurements and is labeled as benign or malignant.

There are 683 complete (no missing values) data points in the dataset. In our study, we use the 683 data points to carry out SVM design with radial basis function kernel under 70%/30% training/testing split and performance averaging over ten independent training/testing splits. L_1-norm rank selection in class data curation [K_{opt} in (16)] is sought among all ranks $K = 1, 2, \ldots, D-1 = 8$ (i.e., $T = 8$.)

In Table VI, we present side by side the binary classification confusion matrices of the SVM when trained on raw data and when trained on curated data. The developed training dataset curation algorithm improves the power probability of the classifier (the probability of correctly identifying malignant cells) from 0.96 to 0.98. The false alarm rate of the classifier (probability of identifying benign cells as malignant) remains 0.04. A 2% improvement in probability of cancer detection maintaining the same false alarm rate over this celebrated dataset is arguably remarkable.

4) Wine Dataset: The Wine Dataset is created with samples of wines that come from three different cultivars in Italy. Each data point in the set consists of $D = 13$ chemical composition measurements such as alcohol content, color intensity, hue, flavonoids, and total phenols [39]. The objective of the experiment is to detect the cultivar of a wine sample from the chemical composition data provided. The dataset has a total of 178 data points and is label balanced. We carry out again SVM design using radial basis function kernel under 70%/30% training/testing splits and performance averaging over ten independent training/testing splits. L_1-norm rank selection in class data curation [K_{opt} in (16)] is sought among all ranks $K = 1, 2, \ldots, D-1 = 12$ (i.e., $T = 12$.)

TABLE VIII

CONFUSION MATRIX FOR THE WINE DATASET WITH 10% INDUCED LABEL NOISE

	Raw Training Data			Curated Training Data		
	Cultivar1	Cultivar2	Cultivar3	Cultivar1	Cultivar2	Cultivar3
Cultivar1	0.91	0.04	0.05	0.96	0.03	0.01
Cultivar2	0.05	0.91	0.04	0.03	0.94	0.03
Cultivar3	0.02	0.08	0.90	0.02	0.05	0.93

Table VII presents side by side the classification confusion matrices of the SVM when trained on raw or curated data. Training dataset curation of the database improves the probability of correct classification of class "Cultivar 2" from 0.98 to 0.99 eliminating misclassification of "Cultivar 2" as "Cultivar 1."

In Table VIII, we repeat the same study of Table VII under artificial 10% label noise contamination; that is, the label of 10% of data in each class is randomly changed to the label of another class. Dataset curation of the faulty (label noisy) training dataset offers, as anticipated, significant classification performance improvement across the classes. For the class "Cultivar 1," correct classification improves from 0.91 to 0.96, for "Cultivar 2" from 0.91 to 0.94, and for "Cultivar 3" from 0.90 to 0.93.

B. Proposed Algorithm Versus Benchmark Methods

In this section, we compare the proposed data-driven training dataset curation algorithm against leading data correction methods and robust SVM classifiers, such as the scikit-learn implementation of one-class SVM (ocSVM) [41], [42], [43], the scikit-clean ensemble [41], [44], [45], robust SVM (RSVM) with rescaled hinged loss [46], SVM via $L_{0/1}$ soft margin loss [47], SVM with maximum minimum margin (M^3SVM) [48], and the wave loss function SVM (Wave-SVM) [49].

In Table IX, we present the misclassification error of these six methods alongside our proposed algorithm on the datasets Wine [39], Iris [37], Breast Cancer Wisconsin [38], Palmer Penguins [50], Pima Indian Diabetes [51], Glioma [52], and Heart Disease [53]. Label noise in each dataset is varied from 0% to 25% to study and compare the relative behavior of the algorithms. The misclassification error presented for each algorithm on each dataset at each label noise level is the average over ten independent experiments. Multiclass classification is implemented throughout by one-over-all binary classification extension.

It is remarkable to observe that the proposed dataset curation method is nearly universally superior and maintains near stable performance as label contamination increases from 0% to 25%. $L_{0/1}$ and M^3SVM exhibit competitive performance on the Glioma dataset only; scikit-clean is in general the second best performing scheme across datasets.

To investigate the statistical significance of these experimental findings, we implemented the Wilcoxon signed-rank test [54] on the misclassification error rates of the proposed algorithm (propAlgo) versus each of the other models (ocSVM, scikit-clean, rsvm, L0/1, M^3SVM, and Wave-SVM.) In Table X, the positive rank (Pos Rank Sum) points to the total ranks where the proposed algorithm performed better than the competing method and the negative rank (Neg Rank Sum) shows the sum of ranks when the opposite is true. We can see that for all the scenarios, the proposed model strongly outperforms each other model. The proposed model's statistically significant advantage is confirmed by the high positive rank sums and significant p-values observed in all the comparisons, indicating its robustness and efficacy in managing datasets with different degrees of label noise. Table XI summarizes the findings and formalizes that the proposed algorithm (propAlgo) performance is statistically distinguishable from every other method.

As a final study in this section, Table XII lists expended computation time in seconds and establishes that the proposed algorithm is in general the fastest, with only exception its execution over the

TABLE IX

MISCLASSIFICATION RATES ACROSS DIFFERENT DATASETS AND METHODS AT VARYING LEVELS OF MISLABELING

		Misclassification Error						
	% Mislabel	ocSVM	scikitClean	rsvm	L0/1	M^3SVM	Wave-SVM	propAlgo
Wine	0	0.021	0.021	0.072	0.059	0.038	0.264	0.019
	5	0.028	0.023	0.058	0.051	0.057	0.321	0.023
	10	0.036	0.026	0.062	0.074	0.038	0.302	0.019
	15	0.087	0.025	0.070	0.051	0.057	0.340	0.017
	20	0.049	0.028	0.087	0.083	0.038	0.340	0.023
	25	0.126	0.030	0.089	0.096	0.057	0.396	0.034
Iris	0	0.066	0.058	0.108	0.357	0.113	0.076	0.040
	5	0.060	0.057	0.157	0.362	0.113	0.132	0.043
	10	0.064	0.049	0.170	0.368	0.151	0.038	0.036
	15	0.049	0.058	0.245	0.370	0.094	0.151	0.042
	20	0.064	0.068	0.213	0.357	0.132	0.189	0.043
	25	0.064	0.070	0.257	0.406	0.340	0.350	0.042
Cancer	0	0.035	0.033	0.076	0.037	0.044	0.029	0.027
	5	0.037	0.031	0.053	0.046	0.044	0.045	0.028
	10	0.038	0.031	0.042	0.046	0.039	0.098	0.029
	15	0.042	0.035	0.039	0.042	0.039	0.088	0.035
	20	0.050	0.034	0.037	0.042	0.039	0.102	0.033
	25	0.073	0.038	0.031	0.039	0.044	0.059	0.033
Penguin	0	0.006	0.007	0.018	0.043	0.040	0.270	0.007
	5	0.010	0.007	0.019	0.070	0.030	0.330	0.005
	10	0.009	0.007	0.019	0.123	0.020	0.330	0.005
	15	0.017	0.008	0.017	0.132	0.050	0.310	0.008
	20	0.009	0.009	0.017	0.083	0.160	0.290	0.007
	25	0.026	0.014	0.019	0.201	0.180	0.330	0.008
Pima	0	0.341	0.233	0.292	0.256	0.239	0.274	0.234
	5	0.376	0.242	0.300	0.249	0.256	0.326	0.239
	10	0.341	0.241	0.291	0.267	0.261	0.291	0.240
	15	0.341	0.249	0.306	0.293	0.261	0.322	0.249
	20	0.341	0.250	0.323	0.270	0.261	0.357	0.251
	25	0.371	0.261	0.327	0.311	0.270	0.283	0.254
Glioma	0	0.425	0.145	0.271	0.159	0.147	0.286	0.143
	5	0.425	0.146	0.264	0.140	0.159	0.242	0.144
	10	0.425	0.156	0.269	0.150	0.143	0.290	0.146
	15	0.425	0.160	0.352	0.150	0.143	0.234	0.145
	20	0.471	0.165	0.271	0.154	0.173	0.298	0.163
	25	0.475	0.180	0.364	0.180	0.179	0.444	0.176
Heart	0	0.192	0.182	0.227	0.222	0.220	0.341	0.179
	5	0.216	0.198	0.243	0.234	0.220	0.451	0.197
	10	0.220	0.197	0.249	0.229	0.209	0.341	0.196
	15	0.279	0.211	0.256	0.236	0.220	0.407	0.211
	20	0.275	0.221	0.253	0.257	0.242	0.451	0.212
	25	0.302	0.236	0.284	0.288	0.231	0.473	0.234

TABLE X

WILCOXON SIGNED-RANK TEST OF PROPALGO VERSUS OTHER METHODS

Method	Pos Rank Sum	Neg Rank Sum	Ties	Mean Diff	W-Stat	p-value
propAlgo vs ocSVM	902	1	0	0.0735	1.0	< 0.00001
propAlgo vs scikitClean	839	43	6	0.0053	26.5	< 0.00001
propAlgo vs rsvm	902	1	0	0.0690	1.0	< 0.00001
propAlgo vs L0/1	894	9	0	0.0801	9.0	< 0.00001
propAlgo vs M^3SVM	853	8	0	0.0373	8.0	< 0.00001
propAlgo vs Wave-SVM	903	0	0	0.1697	0.0	< 0.00001

TABLE XI

SIGNIFICANCE TEST RESULTS

Significance	ocSVM	scikitClean	rsvm	L0/1	M^3SVM	Wave-SVM
propAlgo	Yes	Yes	Yes	Yes	Yes	Yes

Pima dataset. As anticipated, the $L_{0/1}$ and M^3SVM methods are considerably slower by one and two orders of magnitude, respectively.

C. Application to New SVM Variants

The field of SVM classification enjoyed significant advances in recent years, for example, in the domain of Twin-SVMs [55] and their follow-up variants which address jointly matters of computational complexity, class imbalance, and robust operation [56], [57], [58]. The data-driven, hands-free, training dataset curation method that we developed in this brief can be directly applied as a preprocessing step to any SVM system of interest.

In this section, we consider the widely successful robust classification method with fuzzy class-membership values known as FULSTSVM [59], as well as the neo-twin SVMs for pattern classification (Neo-TSVM) introduced in [60]. There, the quadratic optimization problem of Twin-SVMs is reformulated as an unconstrained minimization problem and maximum separability between nonparallel hyperplanes is pursued by maximizing the generalized angle between hyperplanes.

We execute FULSTSVM and Neo-TSVM classification on the datasets Breast Cancer, Glioma, and Pima under 15%, 20%, and 25% cross-label contamination, with or without the proposed dataset curation preprocessing. Table XIII presents misclassification error averages over ten independent experiments. Remarkably, the data curation method almost universally improves the performance of the already well-performing, outlier-resistant SVMs, even in the highest performing test cases such as FULSTSVM on the Cancer dataset.

TABLE XII

COMPUTATION TIME IN SECONDS ACROSS DATASETS

	Computation Time (secs)						
Dataset	ocSVM	scikitClean	rsvm	L0/1	M^3SVM	Wave-SVM	propAlgo
Wine	0.378	0.540	0.795	49.850	392.230	1.590	0.282
Iris	0.533	0.660	0.728	34.620	205.079	1.230	0.409
Cancer	0.428	0.510	0.610	14.060	202.300	1.320	0.420
Penguin	0.473	0.584	0.773	45.370	201.000	1.560	0.341
Pima	0.990	0.977	1.028	27.108	337.570	1.770	2.453
Glioma	0.953	2.860	0.716	38.580	206.700	1.580	0.699
Heart	0.577	0.634	0.601	15.360	144.760	1.460	0.560

TABLE XIII

TWIN-SVM MISCLASSIFICATION ERROR WITH/WITHOUT DATA CURATION

Dataset	% Mislabel	FULSTSVM [59] (raw data)	Proposed Algo + FULSTSVM	Neo-TSVM [60] (raw data)	Proposed Algo + Neo-TSVM
Cancer	15%	0.039	0.034	0.083	0.054
	20%	0.039	0.029	0.068	0.068
	25%	0.044	0.024	0.146	0.117
Glioma	15%	0.143	0.135	0.171	0.163
	20%	0.159	0.147	0.159	0.135
	25%	0.179	0.163	0.151	0.143
Pima	15%	0.270	0.248	0.287	0.274
	20%	0.309	0.274	0.270	0.257
	25%	0.287	0.261	0.278	0.265

V. CONCLUSION AND FUTURE WORK

Label noise (faults) present in training datasets can deteriorate the generalization ability of machine learning algorithms. SVMs in particular, which rely on a small subset of the training dataset (support vectors) to draw decision boundaries, can be severely affected by erroneous labels.

This brief proposed a novel method based on robust L_1-norm PCA and L_1-norm geometry to curate the training datasets of SVM classifiers. The approach is entirely data-driven and touch-free, including the inherent well-known challenge of rank selection. As a plug-and-play computationally efficient dataset curator, the method can become the front part of any preferred SVM classification system.

Extensive experimental studies on raw and curated datasets across multiple databases and SVM classifier systems demonstrated consistent robustness against label noise and notable classification performance improvement even on high-quality training datasets with

no known label faults (for example, the Wisconsin Breast Cancer dataset.)

Future work can be directed toward: 1) further lowering the computational complexity of data curation (in particular, the rank selection step) and 2) generalized training dataset curation for non-SVM classifiers. An interesting, yet challenging, question is whether the mathematical foundation of this presented work can be extended to address issues of completeness and fairness (bias) in training datasets.

REFERENCES

[1] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. 5th Annu. Workshop Comput. Learn. Theory*, Pittsburgh, PA, USA, 1992, pp. 144–152.

[2] G. N. Karystinos and D. A. Pados, "On overfitting, generalization, and randomly expanded training sets," *IEEE Trans. Neural Netw.*, vol. 11, no. 5, pp. 1050–1057, Sep. 2000.

[3] D. Nasien, S. S. Yuhaniz, and H. Haron, "Statistical learning theory and support vector machines," in *Proc. 2nd Int. Conf. Comput. Res. Develop.*, Kuala Lumpur, Malaysia, May 2010, pp. 760–764.

[4] Q. Wang, "Support vector machine algorithm in machine learning," in *Proc. IEEE Int. Conf. Artif. Intell. Comput. Appl. (ICAICA)*, Dalian, China, Jun. 2022, pp. 750–756.

[5] C. Cortes and V. Vapnik, "Support vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Oct. 1995.

[6] S. Ghosh, A. Dasgupta, and A. Swetapadma, "A study on support vector machine based linear and non-linear pattern classification," in *Proc. Int. Conf. Intell. Sustain. Syst. (ICISS)*, Palladam, India, Feb. 2019, pp. 24–28.

[7] U. Maulik and D. Chakraborty, "Remote sensing image classification: A survey of support-vector-machine-based advanced techniques," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 1, pp. 33–52, Mar. 2017.

[8] P. Wetchasit, P. Phaisakamas, S. Pongsuwan, P. Sapphaphab, O. Rinthon, and S. Pechprasarn, "Using machine learning to predict heart failure: Evaluating model performance on clinical data," in *Proc. 15th Biomed. Eng. Int. Conf. (BMEiCON)*, Tokyo, Japan, Oct. 2023, pp. 1–5.

[9] M. Dundar, B. Krishnapuram, J. Bi, and R. B. Rao, "Learning classifiers when the training data is not IID," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2007, pp. 756–761.

[10] P. Li, X. Rao, J. Blase, Y. Zhang, X. Chu, and C. Zhang, "CleanML: A study for evaluating the impact of data cleaning on ML classification tasks," in *Proc. IEEE 37th Int. Conf. Data Eng. (ICDE)*, Chania, Greece, Apr. 2021, pp. 13–24.

[11] A. A. Gharawi, J. Alsubhi, and L. Ramaswamy, "Impact of labeling noise on machine learning: A cost-aware empirical study," in *Proc. 21st IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Nassau, Bahamas, Dec. 2022, pp. 936–939.

[12] N. Natarajan, I. S. Dhillon, P. Ravikumar, and A. Tewari, "Learning with noisy labels," in *Adv. Neural Inf. Process. Syst.*, vol. 26, Dec. 2013, pp. 1196–1204.

[13] D. F. Nettleton, A. Orriols-Puig, and A. Fornells, "A study of the effect of different types of noise on the precision of supervised learning techniques," *Artif. Intell. Rev.*, vol. 33, no. 4, pp. 275–306, Apr. 2010.

[14] R. Bodo, M. Bertocco, and A. Bianchi, "Impact of noise on machine learning-based condition monitoring applications: A case study," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf. (I2MTC)*, Dubrovnik, Croatia, May 2020, pp. 1–6.

[15] H. Yin and H. Dong, "The problem of noise in classification: Past, current and future work," in *Proc. IEEE 3rd Int. Conf. Commun. Softw. Netw.*, Xi'an, China, May 2011, pp. 412–416.

[16] B. Frenay and M. Verleysen, "Classification in the presence of label noise: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 5, pp. 845–869, May 2014.

[17] C. E. Brodley and M. A. Friedl, "Identifying and eliminating mislabeled training instances," in *Proc. Nat. Conf. Artif. Intell.*, Aug. 1996, pp. 799–805.

[18] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee, "Learning from noisy labels with deep neural networks: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 11, pp. 8135–8153, Mar. 2022.

[19] L. P. F. Garcia, A. C. Lorena, and A. C. P. L. F. Carvalho, "A study on class noise detection and elimination," in *Proc. Brazilian Symp. Neural Netw.*, Curitiba, Brazil, Oct. 2012, pp. 13–18.

[20] J. Nalepa and M. Kawulok, "Selecting training sets for support vector machines: A review," *Artif. Intell. Rev.*, vol. 52, no. 2, pp. 857–900, Aug. 2019.

[21] V. Blanco, A. Japón, and J. Puerto, "A mathematical programming approach to SVM-based classification with label noise," *Comput. Ind. Eng.*, vol. 172, Oct. 2022, Art. no. 108611.

[22] Y. Duan and O. Wu, "Learning with auxiliary less-noisy labels," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 7, pp. 1716–1721, Jul. 2017.

[23] H. Xiong, G. Pandey, M. Steinbach, and V. Kumar, "Enhancing data analysis with noise removal," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 3, pp. 304–319, Mar. 2006.

[24] Q. Kang, X. Chen, S. Li, and M. Zhou, "A noise-filtered under-sampling scheme for imbalanced classification," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4263–4274, Dec. 2017.

[25] Q. Kang, L. Shi, M. Zhou, X. Wang, Q. Wu, and Z. Wei, "A distance-based weighted undersampling scheme for support vector machines and its application to imbalanced classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 9, pp. 4152–4165, Sep. 2018.

[26] P. P. Markopoulos, G. N. Karystinos, and D. A. Pados, "Optimal algorithms for L_1-subspace signal processing," *IEEE Trans. Signal Process.*, vol. 62, no. 19, pp. 5046–5058, Oct. 2014.

[27] P. P. Markopoulos, S. Kundu, S. Chamadia, and D. A. Pados, "Efficient L1-norm principal-component analysis via bit flipping," *IEEE Trans. Signal Process.*, vol. 65, no. 16, pp. 4252–4264, Aug. 2017.

[28] G. R. Naik, *Advances in Principal Component Analysis*. Cham, Switzerland: Springer, 2018.

[29] S. Sehgal, H. Singh, M. Agarwal, V. Bhasker, and Shantanu, "Data analysis using principal component analysis," in *Proc. Int. Conf. Med. Imag., M-Health Emerg. Commun. Syst. (MedCom)*, Greater Noida, India, Nov. 2014, pp. 45–48.

[30] I. T. Jolliffe and J. Cadima, "Principal component analysis: A review and recent developments," *Phil. Trans. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 374, no. 2065, Apr. 2016, Art. no. 20150202.

[31] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *J. ACM*, vol. 58, no. 3, pp. 1–37, 2011.

[32] L. Yu, M. Zhang, and C. Ding, "An efficient algorithm for L_1-norm principal component analysis," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Kyoto, Japan, Mar. 2012, pp. 1377–1380.

[33] Q. Ke and T. Kanade, "Robust L1 norm factorization in the presence of outliers and missing data by alternative convex programming," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, San Diego, CA, USA, Jun. 2005, pp. 739–746.

[34] D. Kaplan. *Knee Point: MATLAB Central File Exchange*. Accessed: Feb. 24, 2023. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/35094-knee-point

[35] Y. LeCun, C. Cortes, and C. J. C. Burges. *The MNIST Database of Handwritten Digits*. [Online]. Available: https://www.kaggle.com/datasets/oddrationale/mnist-in-csv

[36] *MATLAB Implementation of Training Dataset Curation By L_1-Norm Principal-component Analysis for Support Vector Machines*. Accessed: Sep. 24, 2024. [Online]. Available: https://github.com/C2A2-at-Florida-Atlantic-University/Training-Dataset-Curation-for-SVMs/tree/main

[37] R. A. Fisher. *Iris, UCI Machine Learning Repository*. Accessed: Feb. 24, 2024. [Online]. Available: https://doi.org/10.24432/C56C76

[38] W. H. Wolberg, W. N. Street, and O. L. Mangasarian. *Breast Cancer Wisconsin (Diagnostic) Data Set*. UCI Mach. Learn. repository. Accessed: Feb. 24, 2024. [Online]. Available: https://doi.org/10.24432/C5HP4Z

[39] S. Aeberhard and M. Forina, "Wine," *UCI Machine Learning Repository*. Accessed: Feb. 24, 2024. [Online]. Available: https://doi.org/10.24432/C5PC7J

[40] *UCI Machine Learning Repository*. Accessed: Feb. 24, 2024. [Online]. Available: https://archive.ics.uci.edu/ml/index.php

[41] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011.

[42] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, Jul. 2001.

[43] *Sklearn One-Class SVM*. Accessed: Sep. 22, 2024. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.SVM.OneClassSVM.html

[44] S. S. Khan, N. T. Niloy, M. A. Azmain, and A. Kabir, "Impact of label noise and efficacy of noise filters in software defect prediction," in *Proc. SEKE*, Jan. 2020, pp. 347–352.

[45] *Scikit Clean ML Library in Python*. Accessed: Sep. 20, 2024. [Online]. Available: https://scikit-clean.readthedocs.io/en/latest/intro.html

[46] G. Xu, Z. Cao, B.-G. Hu, and J. C. Principe, "Robust support vector machines based on the rescaled Hinge loss function," *Pattern Recognit.*, vol. 63, pp. 139–148, Mar. 2017.

[47] H. Wang, Y. Shao, S. Zhou, C. Zhang, and N. Xiu, "Support vector machine classifier via soft-margin loss," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 7253–7265, Oct. 2022.

[48] F. Nie, Z. Hao, and R. Wang, "Multi-class support vector machine with maximizing minimum margin," in *Proc. AAAI Conf. Artif. Intell.*, Vancouver, BC, Canada, Jan. 2023, pp. 14466–14473.

[49] M. Akhtar, M. Tanveer, and M. Arshad, "Advancing supervised learning with the wave loss function: A robust and smooth approach," *Pattern Recognit.*, vol. 155, Nov. 2024, Art. no. 110637.

[50] D. K. Gorman. *Palmer Penguins*. UCI Mach. Learn. repository. Accessed: Sep. 20, 2024. [Online]. Available: https://doi.org/10.24432/C5R89W

[51] J. W. Smith, J. E. Everhart, W. C. Dickson, W. C. Knowler, and R. S. Johannes. *Pima Indians Diabetes*. Accessed: Sep. 20, 2024. [Online]. Available: https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

[52] E. Tasci, Y. Zhuge, H. Kaur, K. Camphausen, and A. V. Krauze. *Glioma Grading Clinical and Mutation Features*. UCI Mach. Learn. repository. Accessed: Sep. 20, 2024. [Online]. Available: https://doi.org/10.24432/C5R62J

[53] A. Janosi, W. Steinbrunn, M. Pfisterer, and R. Detrano. *Heart Disease*. UCI Mach. Learn. repository. Accessed: Sep. 20, 2024. [Online]. Available: https://doi.org/10.24432/C52P4X

[54] S. M. Taheri and G. Hesamian, "A generalization of the Wilcoxon signed-rank test and its applications," *Statist. Papers*, vol. 54, no. 2, pp. 457–470, May 2013.

[55] R. Khemchandani and S. Chandra, "Twin support vector machines for pattern classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 5, pp. 905–910, May 2007.

[56] B. B. Hazarika, D. Gupta, and P. Borah, "Fuzzy twin support vector machine based on affinity and class probability for class imbalance learning," *Knowl. Inf. Syst.*, vol. 65, no. 12, pp. 5259–5288, Jun. 2023.

[57] B. B. Hazarika and D. Gupta, "Affinity based fuzzy kernel ridge regression classifier for binary class imbalance learning," *Eng. Appl. Artif. Intell.*, vol. 117, Jan. 2023, Art. no. 105544.

[58] D. Gupta, U. Gupta, and H. J. Sarma, "Functional iterative approach for universum-based primal twin bounded support vector machine to EEG classification (FUPTBSVM)," *Multimedia Tools Appl.*, vol. 83, no. 8, pp. 22119–22151, Aug. 2023.

[59] B. Richhariya and M. Tanveer, "A fuzzy universum least squares twin support vector machine (FULSTSVM)," *Neural Comput. Appl.*, vol. 34, no. 14, pp. 11411–11422, Jul. 2022.

[60] S. Jain, S. S. Roy, and R. Rastogi, "Neo-twin support vector machines for pattern classification," in *Proc. Int. Conf. Decis. Aid Sci. Appl. (DASA)*, Chiangrai, Thailand, Mar. 2022, pp. 347–351.