Int. J. Mach. Learn. & Cyber. (2018) 9:1039–1052
DOI 10.1007/s13042-016-0629-5

ORIGINAL ARTICLE

# Data sanitization against adversarial label contamination based on data complexity

**Patrick P. K. Chan[1] · Zhi-Min He[2] · Hongjiang Li[1] · Chien-Chang Hsu[3]**

Received: 13 November 2016 / Accepted: 27 December 2016 / Published online: 24 January 2017


**Abstract** Machine learning techniques may suffer from adversarial attack in which an attacker misleads a learning process by manipulating training samples. Data sanitization is one of countermeasures against poisoning attack. It is a data pre-processing method which filters suspect samples before learning. Recently, a number of data sanitization methods are devised for label flip attack, but their flexibility is limited due to specific assumptions. It is observed that abrupt label flip caused by attack changes complexity of classification. A data sanitization method based on data complexity, which is a measure of the difficulty of classification on a dataset, is proposed in this paper. Our method measures the data complexity of a training set after removing a sample and its nearest samples. Contaminated samples are then distinguished from untainted samples according to their data complexity values. Experimental results support the idea that data complexity can be used to identify attack samples. The proposed method achieves a better result than the current sanitization method in terms of detection accuracy for well known security application problems.

**Keywords** Adversarial learning · Poisoning attack · Data sanitization · Data complexity

✉ Zhi-Min He
zhmihe@gmail.com

Patrick P. K. Chan
patrickchan@ieee.org

Hongjiang Li
hjli@scut.edu.cn

Chien-Chang Hsu
cch@csie.fju.edu.tw

[1] School of Computer Science and Engineering, South China University of Technology, Guangzhou, China

[2] School of Electronic and Information Engineering, Foshan University, Foshan 528000, China

[3] Computer Science and Information Engineering, Fu Jen Catholic University, New Taipei City, Taiwan

## 1 Introduction

Machine learning technique has been increasingly applied to various security problems, i.e., spam filtering [18, 47], intrusion detection [37, 41] and malware detection [2, 35] due to their satisfying performances. However, many studies show that these techniques are vulnerable to adversarial attack since they do not consider the existence of an attack recently [8, 11–13, 22, 46]. Poisoning attack, sometimes called causative attack, is a well known adversarial attack which misleads a learning process by manipulating training samples [1, 3, 26, 44]. An adversary is able to affect a training set in security applications. For example, collaborative spam filtering updates a classifier based on the feedback from end-users [24]. Training samples of some applications are labeled through an online system, e.g., Amazon's Mechanical Turk [14, 16], in order to reduce the cost and time of the labeling process. In this case, an adversary can mislead the decision of a system by giving well-crafted labels to its training samples.

There are two kinds of methods, i.e., data sanitization [15, 17, 31] and robust learning [6–9, 33, 38], to defend against poisoning attack. Data sanitization removes contaminated samples from a training data set before training a classifier while robust learning focuses on increasing the robustness of a learning algorithm to reduce the influence of contaminated samples. However, previous works mainly focused on robust learning. Only a few data sanitization

methods have been proposed. Reject on Negative Impact (RONI) [3, 36] which removes an contaminated sample with significant negative impact on classification accuracy iteratively has achieved satisfying performance on dictionary attacks in the Spam filter application. For each sample $x$, training and test sets are sampled randomly from an untainted dataset. A sample $x$ is said to have a negative impact if the accuracy of a classifier trained using the training set with $x$ is lower than that of the one trained without $x$ on the test set. However, using impact of one sample on classification accuracy as an evaluation criterion may not be efficient in some attacks, e.g., focused attack, which mislead a decision boundary by a cluster of contaminated samples. In addition, RONI relies on a small untainted dataset which may be difficult to obtain [36]. Another data sanitization method is micro-model [15]. A number of micro-models (i.e., classifiers) are trained using a dataset collected in different periods of time. A sample is removed if most micro-models classify it wrongly. However, this method assumes poisoning attack appearing in a short period of time, which may not hold in long-lasting attack. Furthermore, this method may only be suitable in a time series problem in which new samples can be collected periodically.

The limitations mentioned above decrease the flexibility of the current data sanitization methods. In this paper, similar to the existing data sanitization methods, we focus on label flip attack, which is a kind of poisoning attack. A data sanitization method is proposed based on data complexity [21], which measures difficulty of classification based on a dataset. Since label flip attack misleads learning by flipping the labels of samples, the difficulty of classification usually increases by these abrupt label changes. Removing a contaminated sample but not an untainted sample should reduce the difficulty of a problem. By this observation, we examine the influence of a number of samples to geometric characteristics of a dataset using data complexity measure. Contaminated and untainted samples of each class are clustered into two groups according to their data complexity measures. One advantage of our method is that no clean dataset will be required. This is different from RONI and micro-model which only evaluate one sample each time, and the influence of a number of samples is examined. Moreover, our method uses a number of data complexity measures to capture the influence of samples on geometric characteristics of a dataset. This provides more information than RONI and micro-model which only use the classification error.

This paper is organized as follows. Section 2 introduces the related work briefly. The proposed method is described in Sect. 3. And then in Sect. 4, we first analyzed a toy example to illustrate the difference between clean data and attacked data on data complexity and then showed the experimental result on two applications. Finally, the conclusion and future work are given in Sect. 5.

## 2 Related work

Related concepts including poisoning attack and data complexity in this paper are introduced briefly in this section.

### 2.1 Poisoning attack

A training set is contaminated by introducing feature noise [28, 31] or flipping labels [9, 43, 45] to training samples in a poisoning attack. The contaminated data is carefully crafted based on a target classifier which an attacker intends to attack. We focus on label flip attack in this study. Current studies focus on the worst scenario in which an adversary obtains complete knowledge of the targeted classifier, i.e., the most efficient attack strategy can be proposed.

Most poisoning attacks focus on SVM at present. Nearest-first and furthest-first attacks are two intuitive label flip attacks. Nearest(furthest)-first attack reverses the labels of samples which are closest (furthest) to the decision hyperplane of support vector machines (SVM) trained on the untainted dataset. Generally, nearest-first attack slightly changes a class boundary. One of the strength of the nearest-first attack is that it is difficult to detect its contaminated samples without additional information besides the training set as they are located near the boundary. However, due to the same reason, the influence of nearest-first attack to a class boundary is also small. On the other hand, although the furthest-first attack usually changes a boundary significantly, the contaminated samples are easier to detect in comparison with nearest-first attack since the samples are surrounded by samples from other classes.

An advanced adversarial label flip attack is proposed for SVMs in [9]. The samples, which are located furthest to both the hyperplane trained by the untainted training set and the hyperplane generated randomly, are attacked. Different from the furthest-first attack, this adversarial label flip attack considers an artificial hyperplane in order to cause the rotation of the original hyperplane. Another adversarial label flip attack [43] aims to maximize classification error of a SVM trained using tainted data on the untainted data. That is, this method generates attack samples in order to yield the maximal classification error on the untainted data. As both methods formulate label flip attack as an optimization problem which maximizes the error under some constraint, their performance is usually better than nearest- and furthest-first attacks [43, 45]. However, the time complexity is relatively high. Instead of manipulating the feature or label of the samples in the training set, a poisoning attack by inserting crafted attack samples is

proposed to maximize the classification error of SVM on a validation dataset [10].

Alfeld et al. formulated poisoning attack against autoregressive models by a mathematical framework [1]. Another poisoning attack against collaborative filtering systems is proposed in [26]. It manipulates the training data based on the attacker's malicious objectives while mimicking the behavior of a legitimate user so as not to arouse suspicion.

### 2.2 Data complexity

Data complexity indicates the difficulty of a classification problem by measuring its geometrical characteristics. Previous studies showed that the data complexity measures had superior performances to describe the difficulty of a classification problem [5, 20, 29, 34]. Different measures [21, 39], listed in Table 1, describe the difficulty of classification in different aspects. "↑" ("↓") in the third column indicates that a higher (lower) data complexity value denotes a more difficult classification problem. According to the taxonomy proposed in [21], data complexity measures are divided into three categories: measures of overlap of individual feature values, measures of separability of classes and measures of geometry, topology, and density of manifolds.

*Measures of overlap of individual feature values* describe the difficulty of a classification problem according to overlapping between different classes. Fisher's discriminant ratio ($F1$) applies the ratio of the square of the difference between the centers of two classes to the summation of the variance of each class. A large $F1$ means samples of different classes are well-separated. Directional-vector Fisher's discriminant ratio ($F1v$) is an extension of $F1$, which calculates the two-class Fisher's discriminant ratio after the instances are projected. Volume of overlapping region ($F2$) calculates the product of ratio between the width of overlapping region and the range of values spanned by both classes for each feature. A dataset with a large value of $F2$ indicates that the overlapping area of classes is large, i.e., a more difficult classification problem. The maximum separating ability of a feature is considered by Maximum (individual) feature efficiency ($F3$). The largest fraction of points lying outside the overlapping region for each individual feature is calculated. Large value of $F3$ indicates that at least one feature is able to discriminate samples. Different from $F3$, collective feature efficiency ($F4$) calculates the discriminative power of all attributes instead of the maximum one.

*Measures of separability of classes* calculate the linear separability ($L1$ and $L2$) and the complexity of class boundary ($N1$, $N2$ and $N3$) of a dataset. Minimized error by linear programming ($L1$) is the sum of distances of error points to the separating hyperplane of a linear classifier, i.e., the value of the objective function in a linear programming formulation [40]. Similarly, *error rate of linear classifier by linear programming (L2)* calculates the training error rate of the linear classifier defined in $L1$. Fraction of points on class boundary ($N1$) relies on construction of a minimum spanning tree (MST) connecting all samples regardless of a class label. $N1$ is the ratio of samples connected to samples of other classes in MST to the total number of samples in the dataset. A large value of $N1$ indicates that many samples are located near the class boundary and the shape of the class boundary is complicated. Ratio of average intra/inter class NN distance ($N2$) is the ratio of the total distance between all samples and their nearest samples within the same class to the total distance between all samples and their nearest samples of other classes. Large $N2$ shows samples are closer to other classes than their own class, i.e., a more difficult classification problem. Error rate of 1-NN classifier ($N3$) is the error rate of the nearest-neighbor classifier estimated by leave-one-out cross validation. $N3$

**Table 1** 13 data complexity measures adopted in this paper

| Measure | Description | Tendency |
|---|---|---|
| F1 | Fisher's discriminant ratio | ↓ |
| F1v | Directional-vector Fisher's discriminant ratio | ↓ |
| F2 | Volume of overlapping region | ↑ |
| F3 | Maximum (individual) feature efficiency | ↓ |
| F4 | Collective feature efficiency | ↓ |
| L1 | Minimized error by linear programming | ↑ |
| L2 | Error rate of linear classifier by linear programming | ↑ |
| L3 | Nonlinearity of linear classifier by linear programming | ↑ |
| N1 | Fraction of points on class boundary | ↑ |
| N2 | Ratio of average intra/inter class NN distance | ↑ |
| N3 | Error rate of 1-NN classifier | ↑ |
| N4 | Nonlinearity of the one-nearest neighbor classifier | ↑ |
| T1 | Fraction of maximum covering spheres | ↑ |

describes how many samples are closer to the sample from other class than the one from the same class. A large value of $N3$ means samples from different classes are highly interleaved.

*Measures of geometry, topology, and density of manifolds* assume that a class is constructed by a single or multiple manifolds. These measures describe the class separability indirectly according to the shape, position, and interconnectedness of manifolds. Nonlinearity of linear classifier by linear programming ($L3$) and nonlinearity of 1NN classifier ($N4$) is the error rate of a linear classifier and nearest-neighbor classifier trained on an original dataset and tested on an artificial dataset generated by linear interpolation with random coefficients between two samples randomly drawn from the same class. $L3$ and $N4$ are large when the classes are interleaved. Fraction of maximum covering spheres ($T1$) measures the shape of class manifolds with adherence subsets. An adherence subset is a hyper-sphere centered at each training sample and expanded until it touches a sample from another class. $T1$ counts the number of hyper-spheres required to cover each class after removing the redundant hyper-spheres lying completely in other hyper-spheres. A large value of $T1$ represents the samples of different classes are close to each other, i.e., the shape of decision boundary is complicated.

## 3 Data complexity based data sanitization

Our data complexity based data sanitization approach is presented in this section. The underlying rationale of our proposed method is to distinguish contaminated samples from legitimate ones based on their influences on data complexity measures which can well capture the geometric characteristics of a dataset.

Given a training set $D = \{(x_1, \hat{y}_1), (x_2, \hat{y}_2), ..., (x_n, \hat{y}_n)\}$ which may potentially be contaminated by label flip attack of a binary classification problem, where $n$ is the number of training samples, $x_i$ is the feature vector and $\hat{y}_i \in \{+1, -1\}$ is the given class label of the sample $i$. Let $y_i$ be the true label of the sample $i$. If the sample $i$ is untainted, $\hat{y}_i = y_i$; otherwise, $\hat{y}_i = -y_i$. The aim of a data sanitization method is to identify contaminated samples (i.e., $\{x_i | \hat{y}_i \neq y_i$ and $x_i \in D\}$).

The distribution of samples is changed by label flip attack since the attack aims to mislead the learning of a system in order to increase its classification error by flipping labels of training samples. The abrupt label flip changes complication of a classification problem which is captured by data complexity measures in our model. We define a data complexity vector ($M_s$) consisting of $p$ chosen data complexity measures of a dataset $d$ as:

$$M_d = \{m_1, m_2, \ldots, m_p\} \tag{1}$$

where $m_i$ denotes the $i$th data complexity measure value of a dataset $d$. Let $K_i$ be the set of samples containing $x_i$ and its $k$ nearest-neighborhoods. The influence score ($S_i$) which represents the influence of removing $x_i$ on the geometric characteristics of a dataset $D$ is defined as:

$$S_i = \mathbf{E}[M_{(D-K_j)} | x_i \in K_j] - M_D \tag{2}$$

where $M_{(D-K_j)}$ denotes the data complexity vector on $D$ after removing $x_j$ and its $k$ nearest samples. $\mathbf{E}[M_{(D-K_j)} | x_i \in K_j]$ denotes the conditional expected value on data complexity vectors on a dataset without $x_i$. As a result, the influence score is the difference between the expected vector and the data complexity vector of the original $D$. All data complexity measures are linearly normalized by $a'_i = (a_i - \min(a))/(\max(a) - \min(a))$, where $a_i$ and $a'_i$ are the original and normalized value of the $i$th element of the feature $a$. $\max(a)$ and $\min(a)$ denotes the maximum and minimum value of the feature $a$.

We assume that any class in $D$ can be contaminated. Training samples of each class in $D$ are then clustered into two groups, denoted as $C_1^+$ and $C_2^+$ ( $C_1^-$ and $C_2^-$) for class $+1$ ($-1$), based on $S$, i.e., the influence score of removing the sample, by $k$-means separately. As the influence of an untainted and contaminated sample to a dataset is different, untainted and contaminated samples are expected to be separated into two groups for each class. Then, clusters containing contaminated samples are identified according to accuracy of a classifier trained with samples in a cluster of each class, i.e., $C_i^+ \cup C_j^-$ where $i, j \in \{1, 2\}$. Let $Err_{i,j}$ be the classification error of the classifier trained using $C_i^+ \cup C_j^-$ on $D$. For example, if $Err_{1,l} > Err_{2,l}$ where $l \in \{1, 2\}$, the samples in cluster $C_1^+$ have negative impact on the classification accuracy in comparison with the ones in $C_2^+$. Thus, the samples in cluster $C_1^+$ are more likely to be contaminated than those in $C_2^+$. If the classification errors of both cluster 1 and 2 are the same, no contaminated cluster is determined. Similarly, the contaminated cluster for $C_1^-$ and $C_2^-$ can be identified. Finally, all samples in contaminated cluster(s) are removed. The detailed procedure and the architecture of the proposed method are described in Algorithm 1 and Fig. 1.

One of the advantages of the proposed method is that the impact of a sample and its nearest neighbors to a training set is measured in comparison with RONI. The influence of a single contaminated sample to a training set may not be measurable. However, a number of contaminated samples may significantly affect a decision boundary [43, 45]. In addition, the classification error is used as a selection criterion in both RONI and micro-model. Classification error may be insensitive to the change of a decision boundary, i.e., a movement of decision boundary may not affect its accuracy. In contrast, our method uses a number of data complexity measures, which capture the geometric

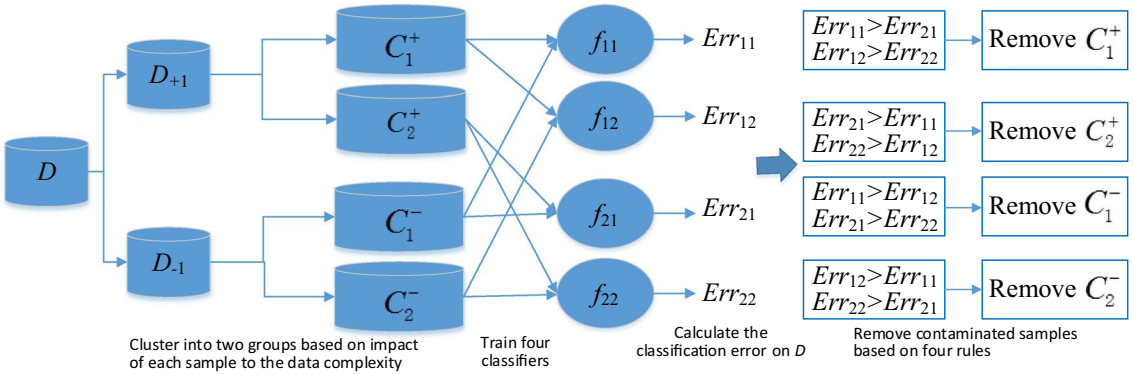



**Fig. 1** The architecture of the proposed method

characteristics of a dataset in different angles. A contaminated sample may be more easily identified by our method than RONI or micro-model. Therefore, our method may identify contaminated samples more accurately.

**Algorithm 1** Data Complexity Based Data Sanitization

**Require:**
Training set $D = \{(x_1, \hat{y}_1), (x_2, \hat{y}_2), \cdots, (x_n, \hat{y}_n)\}$ where $n$ is the number of training samples; the parameter $k$;
**Ensure:**
the sanitized dataset $D'$;
1: calculate the data complexity vector ($M_D$) containing $p$ chosen data complexity measures ($m_i$) of the training set ($D$);
2: **for** each sample $j$ in $D$ **do**
3: calculate data complexity vector ($M_{(D-K_j)}$) of dataset $D - K_j$ which is generated by removing samples in $K_j$ (*i.e.,* Sample $j$ and its $k$-nearest neighbors) from $D$;
4: **end for**
5: calculate the average impact of each sample $i$ in set $D$ to the data complexity: $S_i = \mathbf{E}[M_{(D-K_j)}|x_i \in K_j] - M_D$;
6: training samples of each class in $D$ are then clustered into two groups, denoted as $C_1^+$ and $C_2^+$ ( $C_1^-$ and $C_2^-$) for class $+1$ ($-1$), based on $S_i$ by $k$-means separately
7: train four classifiers by $C_i^+ \cup C_j^-$, where $i, j \in \{1, 2\}$.
8: calculate the classification error $Err_{i,j}$ of the classifier trained using $C_i^+ \cup C_j^-$ on $D$.
9: $D' \leftarrow D$
10: if $Err_{11} > Err_{21}$ and $Err_{12} > Err_{22}$, then remove the samples in cluster $C_1^+$ from $D'$;
11: if $Err_{21} > Err_{11}$ and $Err_{22} > Err_{12}$, then remove the samples in cluster $C_2^+$ from $D'$;
12: if $Err_{11} > Err_{12}$ and $Err_{21} > Err_{22}$, then remove the samples in cluster $C_1^-$ from $D'$;
13: if $Err_{12} > Err_{11}$ and $Err_{22} > Err_{21}$, then remove the samples in cluster $C_2^-$ from $D'$;
14: **return** the remaining dataset $D'$;

## 4 Experiment

In this section, the performance of data complexity based data sanitization is evaluated and also compared with RONI which is the most popular data sanitization method. Our experiment does not consider micro-model method because our method makes no assumption that samples are collected in accordance with the time series and attack appears in a short period of time. Since most of the existing label flip attacks focus on a SVM classifier, SVMs with the linear kernel and the RBF kernel are chosen as the classifiers in this study. We focus on four poisoning label flip attacks including the furthest-first attack (furthest), the nearest-first attack (nearest), the one maximizing classification error (maxErr) [43], and the one maximizing the rotation degree (far-rotate) [9] to a SVM classifier. Attack strength is defined as the percentage of samples contaminated in a dataset. Therefore, the valid range of attack strength is from 0 to 100%.

A toy dataset with two dimensional features is firstly applied to visualize the decision boundary of classifiers trained by the contaminated dataset and the dataset after sanitizing by RONI and our method. Experimental evaluation of the methods in two real applications, i.e., spam filtering and letter recognition, is reported. Finally, we offer an explanation why data complexity measures are useful in data sanitization by analyzing the feature values of legitimate and contaminated samples in our method.

### 4.1 Visualization of classifiers on contaminated dataset after data sanitization

This section aims to illustrate the performance of our method and RONI by visualizing classifiers trained by the contaminated dataset with and without data sanitization. An artificial two-dimensional dataset with two classes is constructed. 100 samples are randomly generated for each class following a Gaussian distribution with different mean values ($(-1, 0)$ for "+" class and $(1, 0)$ for "−" class) and the same covariance matrix [0.2 0; 0 0.2].

The untainted dataset and the SVM trained on this dataset are shown in Fig. 2. Red "+" and green "*" denote

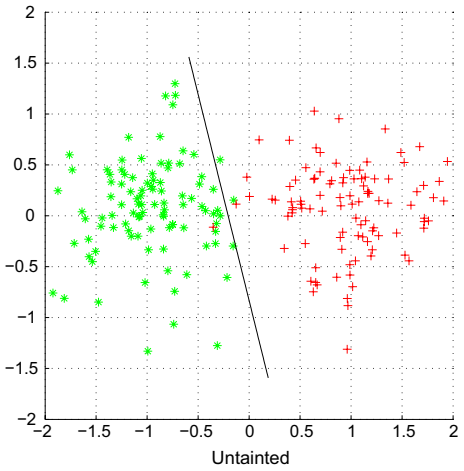



**Fig. 2** The artificial dataset

samples of "+" class and "−" class respectively. The decision hyperplane denoted by a black solid line accurately separates samples from the two classes.

Figures 3a–d show the datasets contaminated by different label flip attacks with 10% attack strength, and also the classifiers trained with these datasets. The contaminated samples are highlighted with blue circles. In maxErr attack, the attack samples concentrate on the left-bottom corners. In both furthest and far-rotate attacks, the samples far from the decision hyperplane are attacked, while the nearest attack flips the labels of samples which are near the decision hyperplane. The decision hyperplanes trained with the contaminated dataset are different from the one trained with the untainted dataset, especially in maxErr and far-rotate attack, i.e., the label flip attacks mislead the learning process.

The third (e–h) and the fourth (i–l) row of Fig. 3 show the contaminated samples sanitized by RONI and the proposed method. The classifiers are trained on the dataset after removing the suspected samples identified by the data santization methods. The suspected samples are highlighted by blue circles.

The recall rates, i.e., the number of contaminated samples detected divided by the total number of existing contaminated samples, of our method are higher than the ones of RONI for all attacks. Almost all the contaminated samples of furthest, far-rotate and maxErr attacks are successfully detected by the proposed method. In contrast, RONI misses some contaminated samples which are located as a cluster. Even worse, the legitimate samples around the contaminated samples are identified as suspected samples by RONI in the dataset under furthest attack. This indicates that only the influence of one sample to a learning process may not provide sufficient information to determine whether the sample is contaminated.

### 4.2 Performance in real security-related applications

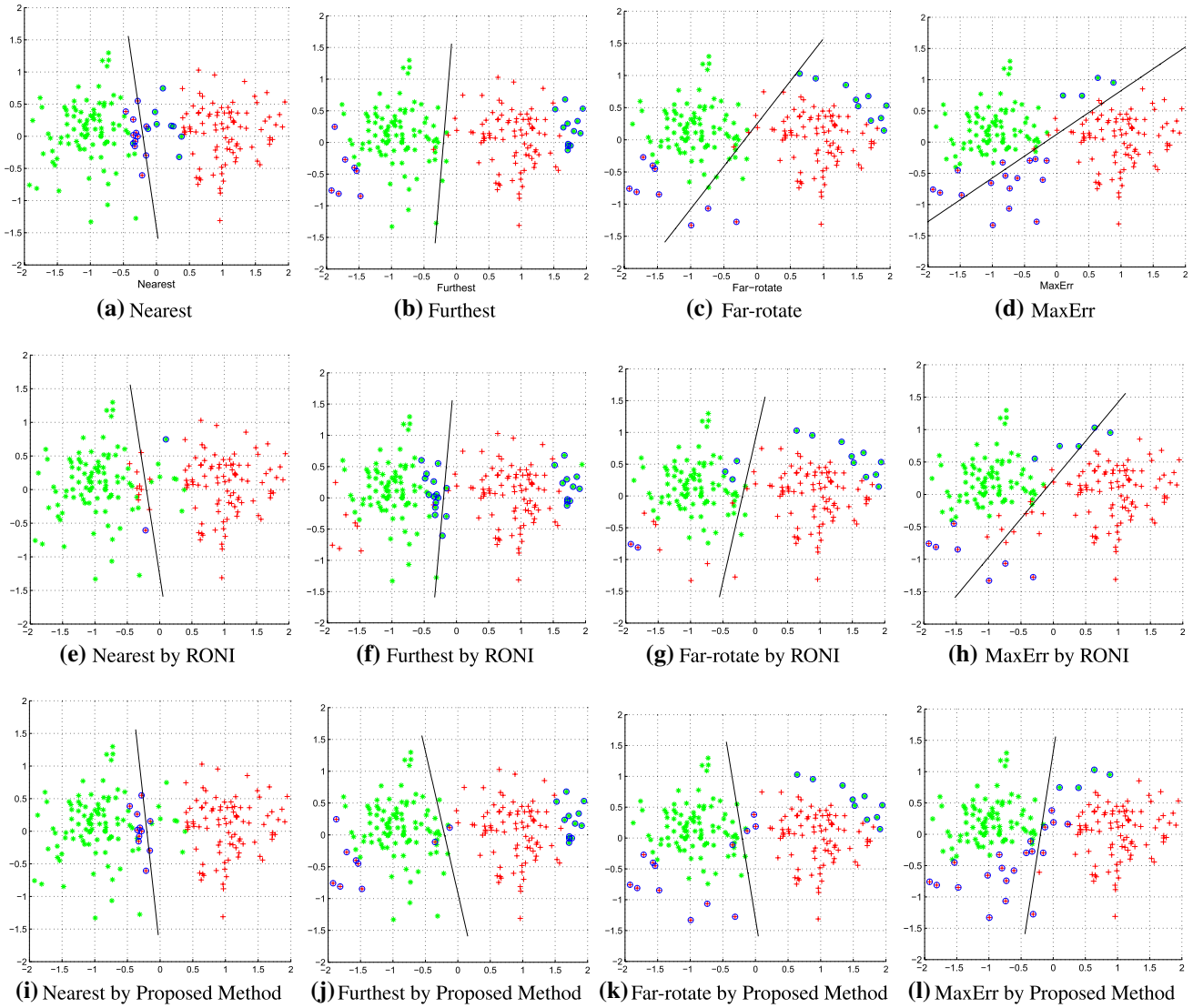
Two real applications, i.e., spam filtering and letter recognition, are considered in this section in order to evaluate the performance of RONI and our data sanitization method. In each experiment, a training set is contaminated by four kinds of label flip attacks (i.e., nearest, furthest, far-rotate and maxErr attack) with different attack strengths (5, 10, 15 and 20%), respectively. SVM with the linear kernel is trained using the contaminated dataset after the suspected samples identified by the data sanitization methods are removed. The performance of the data sanitization methods are evaluated by 1) test accuracy of the SVM on the untainted test sets, and 2) detection rate of contaminated samples in terms of precision rate ($P$), recall rate ($R$) and F-score ($F_1$). $P$ is the fraction of detected instances that are contaminated, while $R$ is the fraction of contaminated instances that are detected. $F_1$ is defined as the harmonic mean of precision and recall: $2 * (P * R)/(P + R)$.

The parameters of RONI are set according to [4], while the $k$ parameter of the proposed method is set to 5 determined by a 5-fold cross-validation on the training set to minimize the classification error. Additional samples are prepared for RONI besides the training set, while our method only identifies contaminated samples according to the training set. Therefore, more prior knowledge have been assumed by RONI. Each experiment is executed 10 times independently.

#### *4.2.1 Spam filtering*

Spam filtering is one of the most popular application examples considered in adversarial learning [19, 23, 25, 31, 32, 42]. We use the spam dataset from UCI machine learning repository [27]. This dataset contains two classes, i.e., spam and legitimate emails, with 57 features. Most of the features describe the percentage of some sensitive words in the e-mail. The first 1000 spam emails and 1000 legitimate emails are selected from the dataset for this experiment. 500 training and 500 test samples are randomly selected from the dataset. The rest of the samples are reserved for RONI as the additional candidate training set and the calibration set.

Table 2 shows the average accuracies with standard deviations of the linear SVMs trained on the training set contaminated by label flip attacks without data sanitization (denoted by No Sanitization), sanitizing by RONI and the proposed method. If no sanitization is applied, the accuracies sharply drop when the training set is contaminated by label flip attacks even though the attack ratio is small. For example, the accuracy of the SVM without sanitization deceases from 92.20 to 85.00% due to Far-rotate attack with 5% attack ratio. It indicates that traditional classification




**(a)** Nearest **(b)** Furthest **(c)** Far-rotate **(d)** MaxErr

**(e)** Nearest by RONI **(f)** Furthest by RONI **(g)** Far-rotate by RONI **(h)** MaxErr by RONI

**(i)** Nearest by Proposed Method **(j)** Furthest by Proposed Method **(k)** Far-rotate by Proposed Method **(l)** MaxErr by Proposed Method

**Fig. 3** Contaminated datasets by label flip attacks (i.e., nearest, furthest, far-rotate and maxErr attacks), and detected samples by RONI and our proposed method. The *first row* **a**–**d** shows the contaminated datasets and SVMs trained on these dataset. The contaminated samples are marked by *blue circles*. The *second row* **e**–**h** and the *third row* **i**–**l** show the contaminated datasets sanitized by RONI and our proposed method, and also SVMs trained on these dataset. The suspected samples are marked by *blue circles*

methods are sensitive to poisoning attacks since their accuracy can significantly decrease by a few contaminated samples. Furthermore, nearest attack is the least effective label flip attack method in this study. All attacks except the nearest attack reduce the accuracies of the SVMs to less than 87 and 69% with 5 and 20% attack ratio respectively.

Experimental results of using the SVMs with the RBF kernel, shown in Table 3, are similar to those obtained with the linear one. MaxErr and nearest is the most and least efficient attack among the label flip attacks. Compared with the linear SVMs, the SVMs with the RBF kernel are more robust to label flip attacks, i.e., accuracies of the nonlinear SVMs are higher than the linear ones without sanitization in all cases. The reason is that a simpler classifier can be misled more easily by label flip attack with the same attack strength.

The results also indicate that both RONI and our proposed method are able to reduce the attack effect. The performance of the SVMs trained on the sanitized dataset is better than the one without sanitization in general. Our method is 3.76% (2.32%) more accurate than RONI under furthest, far-rotate and maxErr attacks in average for the linear (nonlinear) SVMs. In some cases, our method achieves 5% better than RONI, for example, in far-rotate attack against the linear SVM with 15% attack ratio. Furthermore, the proposed method is better than RONI with a statistical significance at 95% levels in most of the cases. The superiority of our method increases with larger attack

**Table 2** Accuracy (%) of SVM with the linear kernel trained on the spam datasets under different label flip attacks with attack strength (0.05, 0.1, 0.15 and 0.2) without sanitization, sanitized by RONI and the proposed method

| Attack method | Attack ratio | No sanitization | RONI | Our method |
|---|---|---|---|---|
| No attack | 0 | **92.20** | 92.07 | 90.90 |
| Nearest | 0.05 | 91.56 | **91.60** | 89.90 |
| | 0.1 | 90.96 | **90.98** | 90.16 |
| | 0.15 | 89.76 | **90.06** | 88.14 |
| | 0.2 | 89.12 | **89.25** | 86.88 |
| Furthest | 0.05 | 85.76[a] | 87.72[a] | **89.92** |
| | 0.1 | 80.46[a] | 84.63[a] | **87.52** |
| | 0.15 | 74.26[a] | 79.98[a] | **85.02** |
| | 0.2 | 67.62[a] | 74.24[a] | **81.46** |
| Far-rotate | 0.05 | 85.00[a] | 87.45[a] | **90.02** |
| | 0.1 | 80.16[a] | 84.39[a] | **87.58** |
| | 0.15 | 73.82[a] | 79.34[a] | **84.38** |
| | 0.2 | 68.04[a] | 73.38[a] | **79.46** |
| MaxErr | 0.05 | 86.52[a] | 87.56 | **88.74** |
| | 0.1 | 78.52[a] | 82.44[a] | **85.66** |
| | 0.15 | 74.48[a] | 75.02[a] | **79.62** |
| | 0.2 | 67.90[a] | 71.59 | **73.44** |

[a]The proposed method outperforms the corresponding method with a statistical significance at 95% levels

**Table 3** Accuracy (%) of SVM with the RBF kernel trained on the spam datasets under different label flip attacks with attack strength (0.05, 0.1, 0.15 and 0.2) without sanitization, sanitized by RONI and the proposed method

| Attack method | Attack ratio | No sanitization | RONI | Our method |
|---|---|---|---|---|
| No attack | 0 | **92.52** | 92.42 | 90.82 |
| Nearest | 0.05 | **91.72** | 91.66 | 90.92 |
| | 0.1 | 90.98 | **91.20** | 90.44 |
| | 0.15 | 90.74 | **90.84** | 90.04 |
| | 0.2 | 89.40 | **89.75** | 87.40 |
| Furthest | 0.05 | 87.72[a] | 89.32 | **90.60** |
| | 0.1 | 83.68[a] | 85.68[a] | **89.36** |
| | 0.15 | 78.84[a] | 81.67[a] | **85.44** |
| | 0.2 | 73.46[a] | 77.34 | **79.48** |
| Far-rotate | 0.05 | 87.76[a] | 88.63 | **89.92** |
| | 0.1 | 83.26[a] | 85.50[a] | **87.72** |
| | 0.15 | 78.62[a] | 81.75 | **83.22** |
| | 0.2 | 72.32[a] | 76.94[a] | **79.92** |
| MaxErr | 0.05 | 87.58[a] | 88.92 | **89.60** |
| | 0.1 | 82.68[a] | 83.76[a] | **87.14** |
| | 0.15 | 77.44[a] | 79.61 | **81.88** |
| | 0.2 | 72.00[a] | 74.60[a] | **77.28** |

[a]The proposed method outperforms the corresponding method with a statistical significance at 95% levels

ratio in comparison with RONI. However, RONI and our method do not perform satisfactorily under nearest attack. As the contaminated samples of nearest attack locate near to the class boundary, removing these samples may have little impact on the performance of a classifier. It explains why RONI does not achieve higher accuracy than no sanitization. Similarly, as removing the contaminated samples near to the class boundary also affects the data complexity measures only slightly, our proposed method also does not achieve a satisfying performance.

The performance of the proposed method is slightly lower than those without using any defence and RONI when there is no attack in a training set. The reason might be that samples overlapping with another class may also be removed by our method, which affects the learning process.

Tables 4 and 5 show recall rate, precision rate and F-score of RONI and our method under different label flip attack against linear and nonlinear SVMs respectively. The results show that the difficulty of contaminated sample identification increases with the increase of the attack ratio. As more contaminated samples form a cluster, it is difficult to determine whether the cluster is generated by an attack. Recall rates of the proposed method are much higher than that of RONI, which indicates that more contaminated samples can be detected by our method. On the other hand, more clean samples are removed by our method (i.e., lower precision rates) than that of RONI. These removed samples may not be helpful in improving the generalization ability of the classifier since the accuracies of our method are higher than the one of RONI. In data sanitization, recall rate may be a more important indicator than precision rate. It is because a dataset containing a contaminated sample affects more significantly on learning than the one without a clean sample. Furthermore, in most of the cases, the proposed method achieves higher F-score than RONI, which means the sanitization results using our method is generally more satisfying than that of RONI.

#### *4.2.2 Letter recognition*

A well-known dataset, i.e., letter recognition, from the Statlog collection [30], which contains 20,000 samples, 26 classes and 16 features, is used in this section. These features are the primitive numerical attributes, i.e., statistical moments and edge count, which are integers from 0 to 15. Two similar classes, "E" and "F", contain 768 and 775 samples respectively. Similar to the experimental setting previously, 500 training and 500 test samples are randomly selected from the dataset, and the rest of the dataset is reserved for RONI.

The results on the letter recognition dataset shown in Tables 6 and 7 are consistent with the ones on the spam dataset. Linear and nolinear SVMs achieve 97.73 and

**Table 4** Recall rate (R), precision rate (P) and F-score ($F_1$) of RONI and the proposed method under different label flip attacks with attack strength (0.05, 0.1, 0.15 and 0.2) on spam filtering against SVM with the linear kernel

| Attack method | Attack ratio | RONI (%) | | | Our method (%) | | |
|---|---|---|---|---|---|---|---|
| | | R | P | $F_1$ | R | P | $F_1$ |
| Nearest | 0.05 | 6.01 | 18.25 | 9.04 | 25.41 | 23.28 | 24.30 |
| | 0.1 | 7.31 | 39.72 | 12.35 | 8.80 | 73.08 | 15.71 |
| | 0.15 | 7.35 | 56.20 | 13.01 | 16.27 | 81.81 | 27.14 |
| | 0.2 | 6.02 | 57.58 | 10.90 | 14.50 | 84.16 | 24.74 |
| Furthest | 0.05 | 36.48 | 53.14 | 43.26 | 87.20 | 34.15 | 49.08 |
| | 0.1 | 36.12 | 71.11 | 47.91 | 73.80 | 45.97 | 56.65 |
| | 0.15 | 29.73 | 74.84 | 42.56 | 69.47 | 49.96 | 58.12 |
| | 0.2 | 25.58 | 79.23 | 38.67 | 66.00 | 48.85 | 56.14 |
| Far-rotate | 0.05 | 29.12 | 53.48 | 37.71 | 88.00 | 33.41 | 48.43 |
| | 0.1 | 31.36 | 66.09 | 42.54 | 81.20 | 40.71 | 54.23 |
| | 0.15 | 29.28 | 73.54 | 41.88 | 72.27 | 45.69 | 55.98 |
| | 0.2 | 29.50 | 80.68 | 43.20 | 65.40 | 46.64 | 54.45 |
| MaxErr | 0.05 | 36.88 | 50.35 | 42.57 | 73.60 | 25.52 | 37.90 |
| | 0.1 | 30.12 | 69.39 | 42.01 | 64.00 | 43.82 | 52.02 |
| | 0.15 | 31.55 | 74.36 | 44.30 | 38.80 | 39.73 | 39.26 |
| | 0.2 | 27.10 | 79.52 | 40.42 | 43.30 | 39.22 | 41.16 |

**Table 5** Recall rate (R), precision rate (P) and F-score ($F_1$) of RONI and the proposed method under different label flip attacks with attack strength (0.05, 0.1, 0.15 and 0.2) on spam filtering against SVM with the RBF kernel

| Attack method | Attack ratio | RONI (%) | | | Our method (%) | | |
|---|---|---|---|---|---|---|---|
| | | R | P | $F_1$ | R | P | $F_1$ |
| Nearest | 0.05 | 4.80 | 39.67 | 8.56 | 11.60 | 19.25 | 14.48 |
| | 0.1 | 4.12 | 45.56 | 7.56 | 14.20 | 73.96 | 23.83 |
| | 0.15 | 3.95 | 55.30 | 7.37 | 9.87 | 86.75 | 17.72 |
| | 0.2 | 4.82 | 43.75 | 8.68 | 10.70 | 87.00 | 19.06 |
| Furthest | 0.05 | 7.92 | 36.65 | 13.03 | 91.60 | 35.37 | 51.04 |
| | 0.1 | 7.28 | 48.10 | 12.65 | 88.40 | 47.91 | 62.14 |
| | 0.15 | 5.39 | 51.45 | 9.75 | 76.80 | 50.56 | 60.98 |
| | 0.2 | 4.30 | 48.05 | 7.89 | 55.10 | 47.85 | 51.22 |
| Far-rotate | 0.05 | 5.36 | 26.25 | 8.90 | 90.80 | 35.55 | 51.10 |
| | 0.1 | 4.72 | 36.43 | 8.36 | 80.60 | 45.41 | 58.09 |
| | 0.15 | 2.64 | 40.84 | 4.96 | 60.80 | 43.44 | 50.68 |
| | 0.2 | 4.46 | 50.32 | 8.19 | 59.80 | 46.44 | 52.28 |
| MaxErr | 0.05 | 7.12 | 29.53 | 11.47 | 78.00 | 39.83 | 52.73 |
| | 0.1 | 6.64 | 33.81 | 11.10 | 70.40 | 41.20 | 51.98 |
| | 0.15 | 7.60 | 47.37 | 13.10 | 47.33 | 41.19 | 44.05 |
| | 0.2 | 6.12 | 50.41 | 10.91 | 43.20 | 60.24 | 50.32 |

99.56% accuracies without attack. However, the accuracies drop dramatically under the label-flip attacks, especially the advanced attack strategies, i.e., far-rotate and maxErr attack. In most cases of furthest, far-rotate and maxErr attack, SVMs trained by the dataset sanitized by our method achieve more accurate prediction with a statistical significance at 95% levels than the ones without sanitization and with RONI. However, with the same reason discussed in Sect. 4.2.1, the performance of our method is slightly worse than RONI under no attack and nearest attack.

Tables 8 and 9 show recall rate, precision rate and F-score of RONI and the proposed method under different label flip attacks against SVMs with the linear kernel and the RBF kernel. Recall rate and F-score of the proposed method are always much higher than those of RONI. In furthest attack with 0.05 and 0.1 attack ratio and maxErr attack with 0.05 attack ratio, the recall rates are 100%, which means that all the contaminated samples are identified by our method as shown in Table 8. In most of the cases, the precision rates of the proposed method are also higher than those of RONI. Thus, the proposed method

**Table 6** Accuracy (%) of SVM with the linear kernel trained on the letter recognition datasets under different label flip attacks with attack strength (0.05, 0.1, 0.15 and 0.2) without sanitization, sanitized by RONI and the proposed method

| Attack method | Attack ratio | No sanitization | RONI | Our method |
|---|---|---|---|---|
| No attack | 0 | **97.73** | 97.71 | 96.70 |
| Nearest | 0.05 | **97.16** | 97.07 | 96.04 |
| | 0.1 | 95.30 | **96.20** | 94.80 |
| | 0.15 | 95.30 | **95.43** | 94.08 |
| | 0.2 | 93.84 | **94.02** | 91.64 |
| Furthest | 0.05 | 96.12 | **96.33** | 96.18 |
| | 0.1 | 91.32[a] | 92.88[a] | **95.24** |
| | 0.15 | 81.86[a] | 85.02[a] | **93.80** |
| | 0.2 | 76.40[a] | 79.85[a] | **92.14** |
| Far-rotate | 0.05 | 94.92[a] | 95.67 | **96.42** |
| | 0.1 | 84.72[a] | 87.25[a] | **93.74** |
| | 0.15 | 78.90[a] | 82.27[a] | **92.66** |
| | 0.2 | 68.16[a] | 75.28[a] | **85.58** |
| MaxErr | 0.05 | 93.50[a] | 93.65[a] | **96.04** |
| | 0.1 | 84.40[a] | 86.05[a] | **94.00** |
| | 0.15 | 78.76 | 80.69 | **85.64** |
| | 0.2 | 74.72[a] | 78.01[a] | **84.90** |

[a]The proposed method outperforms the corresponding method with a statistical significance at 95% levels

**Table 7** Accuracy (%) of nonlinear SVM with the RBF kernel trained on the letter recognition datasets under different label flip attacks with attack strength (0.05, 0.1, 0.15 and 0.2) without sanitization, sanitized by RONI and the proposed method

| Attack method | Attack ratio | No sanitization | RONI | Our method |
|---|---|---|---|---|
| No attack | 0 | **99.56** | 99.55 | 96.70 |
| Nearest | 0.05 | 97.34 | **97.46** | 95.18 |
| | 0.1 | 95.82 | **96.01** | 94.58 |
| | 0.15 | 93.86 | 94.20 | **94.36** |
| | 0.2 | 91.16 | 91.68 | **92.20** |
| Furthest | 0.05 | 99.06 | **99.27** | 96.38 |
| | 0.1 | 96.40 | **96.92** | 96.18 |
| | 0.15 | 92.52[a] | 93.57 | **96.24** |
| | 0.2 | 84.88[a] | 85.82[a] | **92.38** |
| Far-rotate | 0.05 | 96.12 | **96.43** | 95.20 |
| | 0.1 | 91.66[a] | 92.06[a] | **95.60** |
| | 0.15 | 87.40[a] | 87.66[a] | **94.74** |
| | 0.2 | 81.38[a] | 82.14[a] | **89.70** |
| MaxErr | 0.05 | 95.96 | **96.04** | 95.16 |
| | 0.1 | 90.58 | 90.98 | **91.46** |
| | 0.15 | 85.32[a] | 86.04[a] | **90.94** |
| | 0.2 | 79.44 | 80.16 | **84.36** |

[a]The proposed method outperforms the corresponding method with a statistical significance at 95% levels

detects contaminated samples more efficiently than RONI in this letter recognition dataset.

### 4.3 Discriminant ability of data complexity measure on label flip attacks

This section analyzes the values of features in our proposed method in order to explain why data complexity is useful in data sanitization. Our method calculates the average change of data complexity measures ($S_i$) of the dataset caused by removing the sample $x_i$. $S_i$ has been used to separate the legitimate samples from the contaminated ones by the clustering method. We discuss the values of $S_i$ for legitimate and contaminated samples of the Spam filtering and letter recognition datasets used in previous sections.

The values of 13 data complexity measures in $S_i$ of legitimate and contaminated samples are displayed in box plot in Figs. 4 and 5. The horizontal and the vertical axis denote the complexity measure and the normalized value of data complexity measure. The 25th and 75th percentiles of values are denoted by the box, while the line inside the box indicates the median. The range of values is represented by the dotted line. The figures in blue and red represent the legitimate and contaminated samples respectively.

Discriminant ability of a data complexity relies on the overlapping between the values of contaminated and legitimate samples. Large overlapping region indicates that the data complexity cannot separate the two classes accurately. Figures 4 and 5 show that the discriminant ability of the data complexity measure varies in applications and label flip attacks. For example, although contaminated and legitimate samples can be separated by F1 in most of the cases, the samples in Spam dataset under nearest dataset are not separated well. Although there is no best data complexity measure for all scenarios, some of them offer satisfying discriminant ability in each case. These data complexity may be complement to each other in order to increase the accuracy of distinguishing contaminated from legitimate samples. The results also suggest that considering only one measurement may not be able to detect contaminated samples successfully. This may explain why the performance of our method is much better than RONI in previous sections.

## 5 Conclusion and future work

Many studies show that label flip attack can dramatically decrease the accuracy of a classification system. Data sanitization is an important technique to defend against the poisoning attack. However, current methods may not perform well due to their inappropriate assumptions. Data complexity based data sanitization method against label flip attacks has been proposed in this study. The influence of a sample

**Table 8** Recall rate (R), precision rate (P) and F-score ($F_1$) of RONI and the proposed method under different label flip attacks against linear SVM with attack strength (0.05, 0.1, 0.15 and 0.2) on letter recognition

| Attack method | Attack ratio | RONI (%) | | | Our method (%) | | |
|---|---|---|---|---|---|---|---|
| | | R | P | $F_1$ | R | P | $F_1$ |
| Nearest | 0.05 | 3.60 | 40.99 | 6.62 | 19.60 | 28.04 | 23.07 |
| | 0.1 | 2.88 | 29.95 | 5.25 | 44.20 | 37.76 | 40.73 |
| | 0.15 | 4.03 | 60.30 | 7.55 | 24.53 | 79.46 | 37.49 |
| | 0.2 | 5.96 | 63.17 | 10.89 | 24.30 | 71.58 | 36.28 |
| Furthest | 0.05 | 16.16 | 49.51 | 24.37 | 100.00 | 57.83 | 73.28 |
| | 0.1 | 13.16 | 61.13 | 21.66 | 100.00 | 63.41 | 77.61 |
| | 0.15 | 13.71 | 53.02 | 21.78 | 92.80 | 72.37 | 81.32 |
| | 0.2 | 13.36 | 53.66 | 21.39 | 78.50 | 76.98 | 77.73 |
| Far-rotate | 0.05 | 14.40 | 34.30 | 20.28 | 97.20 | 66.15 | 78.72 |
| | 0.1 | 17.04 | 51.57 | 25.62 | 96.80 | 67.45 | 79.50 |
| | 0.15 | 17.15 | 74.73 | 27.89 | 80.00 | 70.06 | 74.70 |
| | 0.2 | 14.50 | 60.59 | 23.40 | 54.40 | 58.49 | 56.37 |
| MaxErr | 0.05 | 20.68 | 43.86 | 28.11 | 100.00 | 41.92 | 59.08 |
| | 0.1 | 25.36 | 52.49 | 34.20 | 79.60 | 69.14 | 74.00 |
| | 0.15 | 19.31 | 64.05 | 29.67 | 33.60 | 53.82 | 41.37 |
| | 0.2 | 21.02 | 63.63 | 31.60 | 51.70 | 53.08 | 52.38 |

**Table 9** Recall rate (R), precision rate (P) and F-score ($F_1$) of RONI and the proposed method under different label flip attacks against nonlinear SVM with attack strength (0.05, 0.1, 0.15 and 0.2) on letter recognition

| Attack method | Attack ratio | RONI (%) | | | Our method (%) | | |
|---|---|---|---|---|---|---|---|
| | | R | P | $F_1$ | R | P | $F_1$ |
| Nearest | 0.05 | 4.80 | 39.67 | 8.56 | 52.00 | 28.51 | 36.83 |
| | 0.1 | 4.12 | 45.56 | 7.56 | 67.00 | 54.31 | 59.99 |
| | 0.15 | 3.95 | 55.30 | 7.37 | 62.00 | 81.76 | 70.52 |
| | 0.2 | 4.82 | 43.75 | 8.68 | 50.00 | 88.47 | 63.89 |
| Furthest | 0.05 | 7.92 | 36.65 | 13.03 | 93.60 | 49.45 | 64.72 |
| | 0.1 | 7.28 | 48.10 | 12.65 | 81.20 | 73.17 | 76.97 |
| | 0.15 | 5.39 | 51.45 | 9.75 | 97.20 | 74.35 | 84.25 |
| | 0.2 | 4.30 | 48.05 | 7.89 | 81.70 | 70.55 | 75.72 |
| Far-rotate | 0.05 | 5.36 | 26.25 | 8.90 | 94.00 | 41.11 | 57.20 |
| | 0.1 | 4.72 | 36.43 | 8.36 | 85.60 | 70.45 | 77.29 |
| | 0.15 | 2.64 | 40.84 | 4.96 | 93.73 | 66.51 | 77.81 |
| | 0.2 | 4.46 | 50.32 | 8.19 | 85.70 | 68.54 | 76.16 |
| MaxErr | 0.05 | 7.12 | 29.53 | 11.47 | 44.40 | 52.90 | 48.28 |
| | 0.1 | 6.64 | 33.81 | 11.10 | 73.80 | 57.04 | 64.35 |
| | 0.15 | 7.60 | 47.37 | 13.10 | 68.27 | 61.34 | 64.62 |
| | 0.2 | 6.12 | 50.41 | 10.91 | 56.30 | 58.97 | 57.61 |

to a dataset is measured by considering the change of the data complexity measures of the dataset after removing the sample point and its nearest samples. Legitimate and contaminated samples are then separated by a clustering method based on different influences of removing a legitimate sample and a contaminated one.

A number of experiments were carried to evaluate the performance of the proposed data sanitization method using artificial and security-related datasets under four well known label flip attacks. Our method achieves superior performance in reducing the influence of furthest, far-rotate and maxErr attack to a learning process in comparison with RONI. Both RONI and our method cannot detect the contaminated samples by nearest attack efficiently. The reason might be that the contaminated samples of nearest attack are similar to the untainted samples located near the class boundary if there is an overlap between different classes.

Thus, the performances of RONI and no sanitization are similar. The experimental results indicate that the accuracy of the proposed method against the nearest attack is lower than RONI and no sanitization. The reason might be that the sanitization is counter-productive in the nearest attack

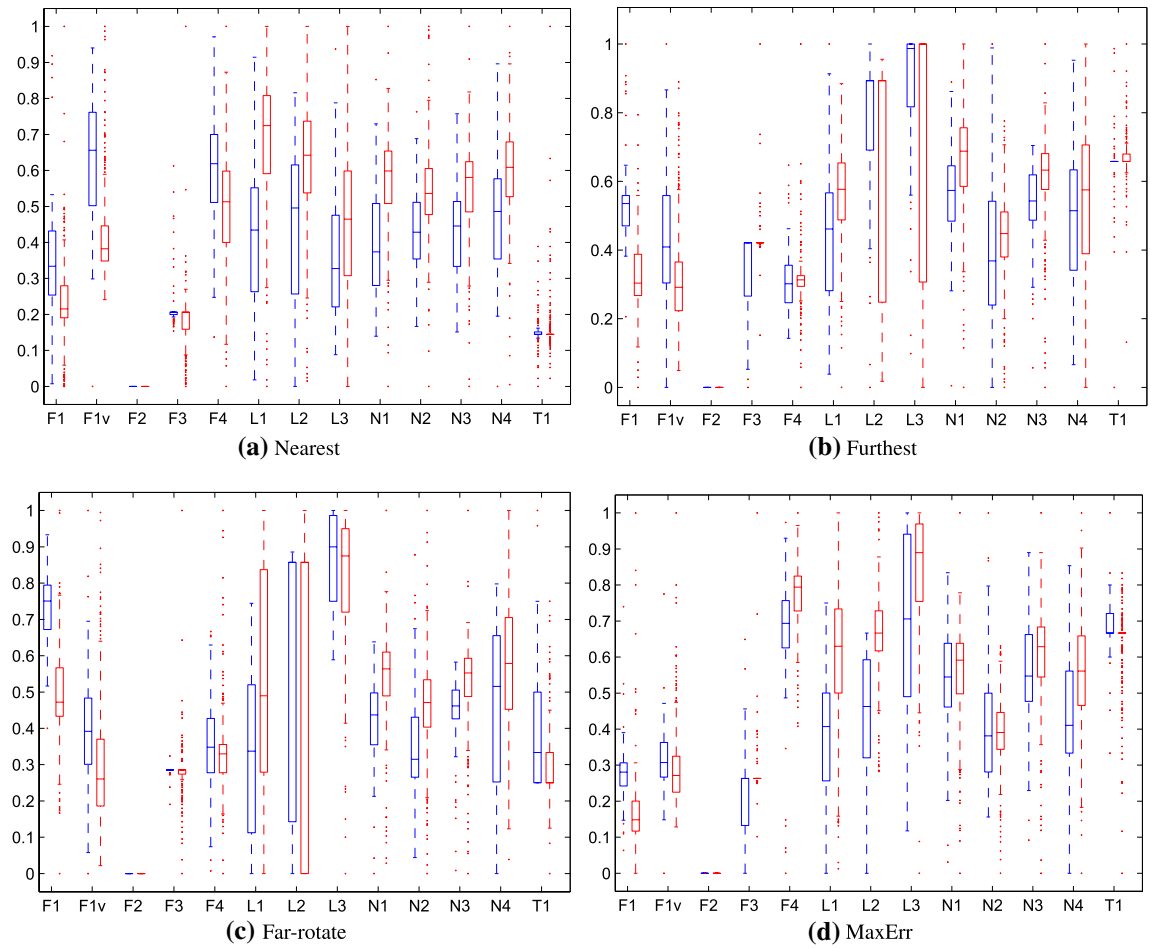



**Fig. 4** Difference of data complexity measures between removing untainted (*in red*) and contaminated (*in blue*) samples by label flip attack on Spam filtering

if the contaminated samples of one class is detected while the contaminated samples of the other class is not as shown in Fig. 3(i). Furthermore, the discriminant ability of data complexity measures used in our method is illustrated, and the parameter selection is also discussed experimentally.

One of the advantages of our method is that no untainted dataset is required. Obtaining a small untainted dataset may be difficult or even unrealistic in some applications. Moreover, different from RONI and micro-model method which rely on classification error only, a number of data complexity measures are used in our method. As data complexity measure captures different geometric characteristics of the dataset, influence of a sample can be evaluated more completely. A cluster effect is also considered. A single sample may not affect a learning process, while a number of samples around this sample considered as a whole may significantly influence a classification system. Our method does not only consider one sample each time, but also its $k$ nearest neighbors. However, our method demands for an increased computational complexity due to the data complexity calculation for each sample. This may not be a critical issue, since data sanitization is often carried out offline. Improving the efficiency of our method remains an open issue to be investigated. A possible solution to overcome this limitation is to apply less data complexity measures to our method. Feature selection methods can be applied to choose a subset of data complexity according to their contribution on contaminated sample identification. However, it may decrease the general performance on different kinds of label flip attacks.

Evaluating whether data complexity is useful to sanitize a contaminated dataset by feature noise attack is another interesting extension. Feature noise attack, which misleads a learning process by manipulating feature values, is another kind of poisoning attack. This attack usually aims to contaminate a particular set of features, and may not cause a significant change on the decision boundary. It increases the difficulty of the detection of contaminated samples. Our experiment indicates that the performance of our method relies on the influence degree of contaminated

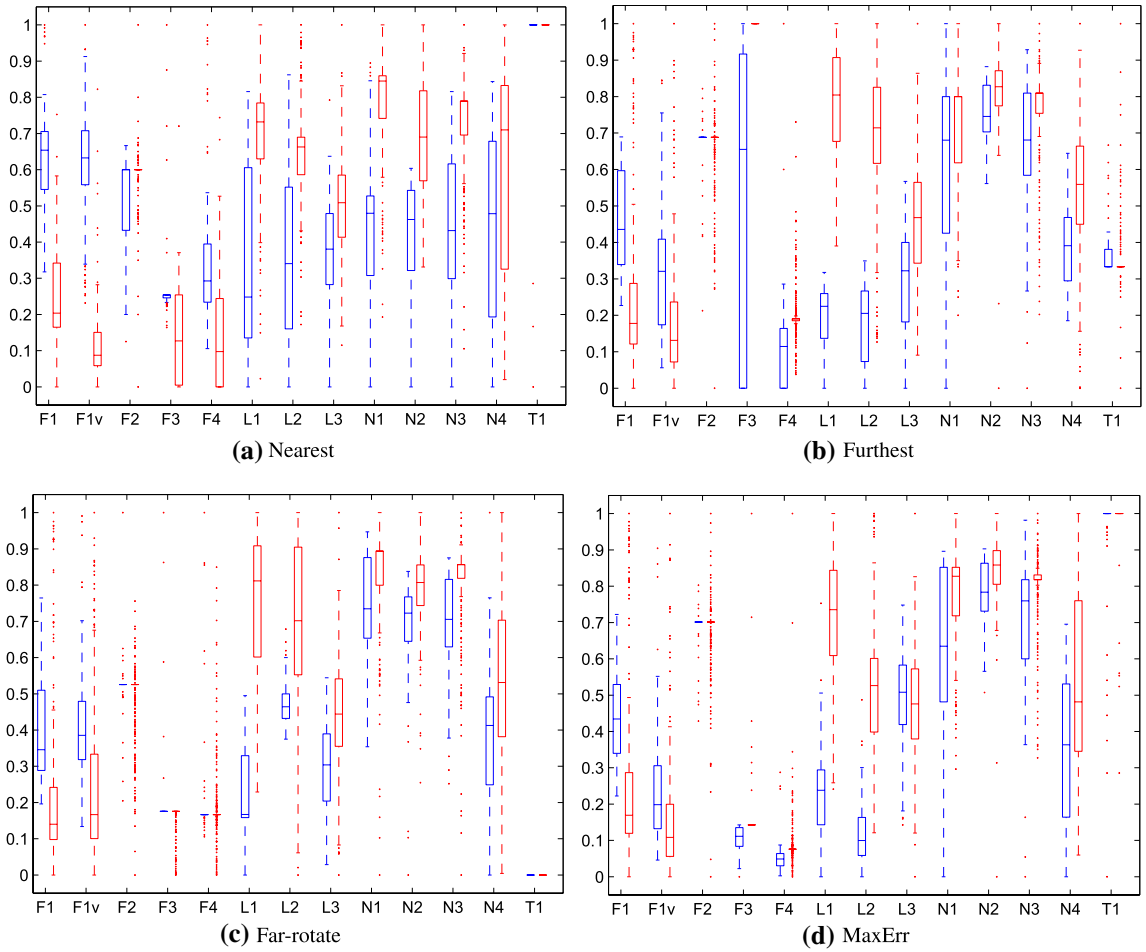



**Fig. 5** Difference of data complexity measures between removing untainted (*in red*) and contaminated (*in blue*) samples by label flip attack on letter recognition

samples to geometric characteristics of a dataset. Specifically, the accuracy of our method on nearest attack which has less influence on the distribution of a dataset is the lowest among other attacks. Rather than directly applying our proposed method to feature noise attack, it may be better to capture the sensitivity of the data complexity of a dataset by moving a sample. If the values of the data complexity change significantly with a small movement on a sample, the sample may be suspected to be contaminated.

**Acknowledgements** This work is supported by the Fundamental Research Funds for the Central Universities (2015ZZ092).

## References

1. Alfeld S, Zhu X, Barford P (2016) Data poisoning attacks against autoregressive models. In: Proceedings of the thirtieth AAAI conference on artificial intelligence, AAAI'16, pp 1452–1458
2. Amos B, Turner H, White J (2013) Applying machine learning classifiers to dynamic android malware detection at scale. In: Wireless communications and mobile computing conference (IWCMC), IEEE, pp 1666–1671
3. Barreno M, Nelson B, Sears R, Joseph AD, Tygar JD (2006) Can machine learning be secure? In: Proceedings of the 2006 ACM symposium on information, computer and communications security, ACM, pp 16–25
4. Barreno M, Nelson B, Joseph AD, Tygar J (2010) The security of machine learning. Mach Learn 81(2):121–148
5. Bernadó-Mansilla E, Ho TK (2005) Domain of competence of xcs classifier system in complexity measurement space. IEEE Trans Evol Comput 9(1):82–104
6. Biggio B, Fumera G, Roli F (2010) Multiple classifier systems for robust classifier design in adversarial environments. Int J Mach Learn Cybernet 1(1–4):27–41
7. Biggio B, Corona I, Fumera G, Giacinto G, Roli F (2011a) Bagging classifiers for fighting poisoning attacks in adversarial classification tasks. In: Multiple classifier systems. Springer, Berlin, pp 350–359
8. Biggio B, Fumera G, Roli F (2011b) Design of robust classifiers for adversarial environments. In: IEEE international conference on systems, man, and cybernetics (SMC), IEEE, pp 977–982
9. Biggio B, Nelson B, Laskov P (2011c) Support vector machines under adversarial label noise. In: ACML, pp 97–112

10. Biggio B, Nelson B, Laskov P (2012) Poisoning attacks against support vector machines. In: 29th intl conf. on machine learning (ICML), pp 1807–1814
11. Biggio B, Fumera G, Roli F (2014) Security evaluation of pattern classifiers under attack. IEEE Trans Knowl Data Eng 26(4):984–996
12. Brückner M, Kanzow C, Scheffer T (2012) Static prediction games for adversarial learning problems. J Mach Learn Res 13(1):2617–2654
13. Chan PPK, Yang C, Yeung DS, Ng WWY (2015) Spam filtering for short messages in adversarial environment. Neurocomputing 155(C):167–176
14. Corona I, Giacinto G, Roli F (2013) Adversarial attacks against intrusion detection systems: taxonomy, solutions and open issues. Inf Sci 239:201–225
15. Cretu GF, Stavrou A, Locasto ME, Stolfo SJ, Keromytis AD (2008) Casting out demons: sanitizing training data for anomaly sensors. In: IEEE symposium on security and privacy, IEEE, pp 81–95
16. Dalvi N, Domingos P, Sanghai S, Verma D, et al (2004) Adversarial classification. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp 99–108
17. Fefilatyev S, Shreve M, Kramer K, Hall L, Goldgof D, Kasturi R, Daly K, Remsen A, Bunke H (2012) Label-noise reduction with support vector machines. In: 21st international conference on pattern recognition (ICPR), IEEE, pp 3504–3508
18. Georgala K, Kosmopoulos A, Paliouras G (2014) Spam filtering: an active learning approach using incremental clustering. In: Proceedings of the 4th international conference on web intelligence, mining and semantics (WIMS14), ACM, pp 1–12
19. Globerson A, Roweis S (2006) Nightmare at test time: robust learning by feature deletion. In: Proceedings of the 23rd international conference on Machine learning, ACM, pp 353–360
20. He ZM, Chan PPK, Yeung DS, Pedrycz W, Ng WWY (2015) Quantification of side-channel information leaks based on data complexity measures for web browsing. Int J Mach Learn Cybernet 6(4):607–619
21. Ho TK, Basu M (2002) Complexity measures of supervised classification problems. IEEE Trans Pattern Anal Mach Intell 24(3):289–300
22. Huang L, Joseph AD, Nelson B, Rubinstein BI, Tygar J (2011) Adversarial machine learning. In: Proceedings of the 4th ACM workshop on security and artificial intelligence, ACM, pp 43–58
23. Jorgensen Z, Zhou Y, Inge M (2008) A multiple instance learning strategy for combating good word attacks on spam filters. J Mach Learn Res 9:1115–1146
24. Kong JS, Rezaei B, Sarshar N, Roychowdhury VP (2006) Collaborative spam filtering using e-mail networks. Computer 39(8):67–73
25. Lee H, Ng AY (2005) Spam deobfuscation using a hidden markov model. In: CEAS
26. Li B, Wang Y, Singh A, Vorobeychik Y (2016) Data poisoning attacks on factorization-based collaborative filtering. In: Advances in neural information processing systems, pp 1885–1893
27. Lichman M (2013) UCI machine learning repository. http://archive.ics.uci.edu/ml
28. Lowd D, Meek C (2005) Adversarial learning. In: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, ACM, pp 641–647
29. Luengo J, Herrera F (2012) Shared domains of competence of approximate learning models using measures of separability of classes. Inf Sci 185(1):43–65
30. Michie D, Spiegelhalter DJ, Taylor CC, Campbell J (eds) (1994) Machine learning, neural and statistical classification. Ellis Horwood, Upper Saddle River
31. Nelson B, Barreno M, Chi FJ, Joseph AD, Rubinstein BI, Saini U, Sutton CA, Tygar JD, Xia K (2008) Exploiting machine learning to subvert your spam filter. LEET 8:1–9
32. Nelson B, Barreno M, Chi FJ, Joseph AD, Rubinstein BI, Saini U, Sutton C, Tygar J, Xia K (2009) Misleading learners: co-opting your spam filter. In: Machine learning in cyber trust. Springer, Berlin, pp 17–51
33. Rubinstein BI, Nelson B, Huang L, Joseph AD, Lau Sh, Rao S, Taft N, Tygar J (2009) Antidote: understanding and defending against poisoning of anomaly detectors. In: Proceedings of the 9th ACM SIGCOMM conference on internet measurement conference, ACM, pp 1–14
34. SáEz JA, Luengo J, Herrera F (2013) Predicting noise filtering efficacy with data complexity measures for nearest neighbor classification. Pattern Recognit 46(1):355–364
35. Sahs J, Khan L (2012) A machine learning approach to android malware detection. In: Intelligence and security informatics conference (EISIC), IEEE, pp 141–147
36. Saini U (2008) Machine learning in the presence of an adversary: attacking and defending the spambayes spam filter. Tech. rep, DTIC Document
37. Satpute K, Agrawal S, Agrawal J, Sharma S (2013) A survey on anomaly detection in network intrusion detection system using particle swarm optimization based machine learning techniques. In: Proceedings of the international conference on frontiers of intelligent computing: theory and applications (FICTA), pp 441–452
38. Servedio RA (2003) Smooth boosting and learning with malicious noise. J Mach Learn Res 4:633–648
39. Singh S (2003) Multiresolution estimates of classification complexity. IEEE Trans Pattern Anal Mach Intell 12:1534–1539
40. Smith FW (1968) Pattern classifier design by linear programming. IEEE Trans Comput 100(4):367–372
41. Suthaharan S (2014) Big data classification: problems and challenges in network intrusion prediction with machine learning. ACM SIGMETRICS Perform Eval Rev 41(4):70–73
42. Wittel GL, Wu SF (2004) On attacking statistical spam filters. In: Conference on email and anti-spam
43. Xiao H, Xiao H, Eckert C (2012) Adversarial label flips attack on support vector machines. In: ECAI, pp 870–875
44. Xiao H, Biggio B, Brown G, Fumera G, Eckert C, Roli F (2015a) Is feature selection secure against training data poisoning? In: Proceedings of the 32nd international conference on machine learning (ICML'15), pp 1689–1698
45. Xiao H, Biggio B, Nelson B, Xiao H, Eckert C, Roli F (2015b) Support vector machines under adversarial label contamination. Neurocomputing 160:53–62
46. Zhang F, Chan P, Biggio B, Yeung D, Roli F (2016) Adversarial feature selection against evasion attacks. IEEE Trans Cybernet 46:766–777
47. Zhou B, Yao Y, Luo J (2014) Cost-sensitive three-way email spam filtering. J Intell Inf Syst 42(1):19–45