International Journal of Machine Learning and Cybernetics (2021) 12:103–116
https://doi.org/10.1007/s13042-020-01159-7

ORIGINAL ARTICLE

Causative label flip attack detection with data complexity measures

Patrick P. K. Chan[1] · Zhimin He[2] · Xian Hu[3] · Eric C. C. Tsang[4] · Daniel S. Yeung[5] · Wing W. Y. Ng[1]

Received: 13 November 2019 / Accepted: 13 June 2020 / Published online: 3 July 2020

Abstract
A causative attack which manipulates training samples to mislead learning is a common attack scenario. Current countermeasures reduce the influence of the attack to a classifier with the loss of generalization ability. Therefore, the collected samples should be analyzed carefully. Most countermeasures of current causative attack focus on data sanitization and robust classifier design. To our best knowledge, there is no work to determinate whether a given dataset is contaminated by a causative attack. In this study, we formulate a causative attack detection as a 2-class classification problem in which a sample represents a dataset quantified by data complexity measures, which describe the geometrical characteristics of data. As geometrical natures of a dataset are changed by a causative attack, we believe data complexity measures provide useful information for causative attack detection. Furthermore, a two-step secure classification model is proposed to demonstrate how the proposed causative attack detection improves the robustness of learning. Either a robust or traditional learning method is used according to the existence of causative attack. Experimental results illustrate that data complexity measures separate untainted datasets from attacked ones clearly, and confirm the promising performance of the proposed methods in terms of accuracy and robustness. The results consistently suggest that data complexity measures provide the crucial information to detect causative attack, and are useful to increase the robustness of learning.

Keywords Adversarial learning · Causative attack detection · Label flip attack · Data complexity

1 Introduction

Pattern classification techniques have been widely applied to many security applications like spam filtering [49, 53], malware detection [57], biometric authentication [31] and intrusion detection system [38, 58] to distinguish the legitimate from the malicious samples. Different from conventional classification problems, the data in security problems may be manipulated by an adaptive and intelligent adversary to mislead the decision of a classification system [5, 13, 19, 68]. Unfortunately, the assumption of samples being stationary so that the training and test data follow the same distribution made in the conventional pattern classification researches is violated in an adversarial environment due to the manipulation by an adversary [26]. The performance of the conventional classification methods may be downgraded significantly. Current researches show that many state-of-the-art classifiers including support vector machine, deep neural networks and ensemble classifiers are vulnerable to adversarial attack [14, 37, 46, 59].

A causative attack, also named a poisoning attack, is one of common adversarial attack scenarios in which the training samples are manipulated by injecting well-crafted label [12, 27, 63] or feature noise [40, 45] in order to mislead a learning process. For example, in a worm detection problem, an adversary will contact a node to perform some suspicious actions, *e.g.,* scanning the network to attract the attention

✉ Zhimin He
zhmihe@gmail.com

Patrick P. K. Chan
patrickchan@ieee.org

Xian Hu
huxian.scut@gmail.com

Wing W. Y. Ng
wingng@ieee.org

[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] Tencent, Shenzhen, China

[4] Faculty of Information Technology, Macau University of Science and Technology, Macau, China

[5] Hong Kong, Hong Kong

from the automatic signature generation (ASG) system [24]. Then, any traffic sent from the suspicious node, including the legitimate traffic, will be labeled as a malicious pattern in the ASG system. As a result, the dataset for worm detection is manipulated by the adversary. In addition, when the size of the dataset is huge, for instance, in natural language processing applications, labels of the samples may come from unreliable sources, *e.g.,* Amazon's Mechanical Turk [18, 62]). A label flip attack can be conducted by any adversarial labeler.

Data sanitization and robust learning are common countermeasures against causative attack. Data sanitization is a pre-processing technique which removes contaminated samples from a training data set before training a classifier [20, 30], while robust learning focuses enhances the robustness of a particular learning algorithm, for example, Support Vector Machines [12, 28], Multiple Classifier Systems [10], Deep Autoencoder [42] and Deep Neural Network [66]. The countermeasures usually sacrifice the generalization ability of a classifier to gain the security [36, 44]. For example, accuracy of SVM with robust learning can be 25% less than the one with standard training [12]. However, in real applications, we do not know whether a collected dataset is contaminated or not. The cleanness of the dataset should be verified before using the countermeasure methods.

Although the causative attack detection is an essential problem of the defence against causative attacks [4], it has not been thoroughly investigated. A naive causative attack detection which requires some untainted samples is reported in [4]. However, its assumption may not be practical since it is difficult to determine whether a sample is untainted or not. Some techniques identify the patterns which are different from the prototypical data. For example, anomaly detection [21] finds out the samples different from the samples in the majority, while concept drift detection [29] produces an alert when the newly arrived samples in the data stream are different from the original ones. These techniques focus on the difference between newly arrived samples and existing ones in real time. However, in causative attack, contaminated samples are mixed with untainted samples in a collected dataset, *i.e.,* there is no prior knowledge on untainted data.

Causative attack modifies a dataset according to a particular algorithm in order to increase the generalization error of a classifier. Geometrical natures of the dataset are changed due to manipulation of the samples. This observation leads us to propose a causative attack detection method based on the geometrical characteristics quantified by the data complexity measures [35]. In this study, we firstly investigate whether and which current data complexity measures are able to capture the difference between untainted and tainted datasets efficiently in Sect. 5. By learning the different geometrical characteristics of untainted and tainted datasets, a dataset containing a causative attack can be detected. In our model, the causative attack detection is formulated as a 2-class classification problem in which a dataset is represented in the feature space formed by the data complexity measures in Sect. 6.1. Furthermore, a two-step secure classification model against causative label flip attack is proposed (Sect. 6.2) to verify that the proposed causative attack detection is able to improve the robustness of learning. For an application, a dataset is firstly examined by our causative attack detection. A traditional learning method, which only maximizes the generalization ability, is applied if a dataset is classified as a untainted class. Otherwise, a robust learning which considers generalization ability and also robustness is used.

In the experiments (Sect. 7), we consider five well-known causative label flip attacks including random label flip, nearest-first label flip, furthest-first label flip, Far-rotate label flip and the advanced adversarial label flip attack [12, 63, 65]. Eleven proposed data complexity measures have been used in our model to quantify the geometric characteristic of a dataset. Ensembles of Decision Trees (EDTs) is used to classify a dataset being attacked or not due to EDTs' satisfying performance in a 2-class problem and independent to monotonic scaling of features [60]. The performance of the proposed two-step secure system is verified under the advanced adversarial label flip attack in the perfect knowledge scenario in which a defender has full information of the attack, *i.e.,* the type and the strength. EDTs firstly determines whether a dataset is contaminated. If so, the robust SVM is applied; otherwise, the traditional SVM is used. Datasets of the five security-related applications are used. The result shows the satisfying performance of the proposed two-step secure system in comparison with the traditional and robust SVMs. Besides, we also conduct the experiments to analyze the discriminating ability of different data complexity measures for causative attack detection. Finally, the conclusion and future work are discussed in Sect. 8.

2 Notation

In this study, geometrical characteristic of a data set $\mathscr{D}$ is quantified by a number of data complexity measures and regarded as a sample $\mathbf{x}^D \in \mathscr{X}$, where $\mathscr{X}$ is the input space composed by data complexity measures. A class label $y^D \in \mathscr{Y}_b = \{-1, +1\}$ is assigned to each $\mathbf{x}^D$. If $\mathbf{x}^D$ contains the causative attack, $y^D = +1$; otherwise, $y^D = -1$. Let $D = \{(\mathbf{x}_i^D, y_i^D)\}_{i=1}^n$ be the training set with n samples drawn independently and identically distributed from a joint distribution $P_{\mathscr{X}\times\mathscr{Y}}$. D_0^{Tr} and D_0^{Ts} denote the untainted training and test dataset. $D_{m,j}^{Tr}$ and $D_{m,j}^{Ts}$ denote the contaminated training and test datasets separately according to the attack m with j% attack strength.

3 Adversarial learning and causative attack

Pattern classification technique has been applied to many security applications in which an adversary aims to downgrade the performance of the system by manipulating the data. It becomes the arms race between the attacker and the defender countering each other's strategies [9, 15, 50]. The attacker explores the vulnerabilities of security systems and devises novel attacks [5, 36], while the defender enhances the security of systems by carefully selecting the inputs (*e.g.,* data sanitization [25, 30, 45]) and implementing robust learning algorithms [12, 28, 51, 55]. As the black-box adversarial attack requires no knowledge on the details of the target system such as its parameters [7, 47], it is more practical than the white-box attack.

A taxonomy of adversarial attacks against machine learning is described in [4, 5]. It considers adversarial attack from different aspects, including attack influence, security violation, and attack specificity. Attack can be divided into causative and exploratory attack based on its influence on the data. Causative attack misleads a training process by manipulating training samples [39, 64, 65] while test samples are directly modified to cause misclassification in an exploratory attack [14, 67]. According to security violation, the attack can be divided into availability, integrity and privacy attack. Availability attack decreases the overall classification accuracy in order to cause a denial of service, while integrity attack only targets on malicious samples. Privacy attack explores sensitive information from a security system. Targeted and Indiscriminate attacks are defined based on attack specificity. Targeted attack focuses on one or a specific set of samples while indiscriminate attack involves a broad range of samples. According to the taxonomy, the adversarial attack considered in this paper is causative-availability-indiscriminate attack.

Many causative label flip attacks are proposed in recent years [12, 63, 65]. These methods assume that the training set has been exposed to by an adversary. They aim to decrease classification accuracy of a classifier by injecting well-crafted label noise to training samples to mislead learning. Their attack strength is limited by the percentage of training samples being attacked. Several simple label flip attacks including random label flip (*rand*), nearest-first label flip (*near*) and furthest-first label flip (*far*), are mentioned in [12, 63]. In *rand* attack, samples in an untainted dataset are randomly chosen and their labels are flipped to the opposite class. It can be regarded as random noise to labels. *Near* attack reverses labels of samples closest to a decision hyperplane trained on an untainted dataset. This attack complicates the decision boundary. Differently, samples furthest away from a decision hyperplane are attacked in *far* attack.

An advanced adversarial label flip attack (*far-rotate*) is proposed for Support Vector Machines (SVMs) in [12]. Samples locating furthest to both hyperplanes obtained by using an untainted training set and generated randomly are attacked. Different from *far* attack, *far-rotate* attack considers an artificial hyperplane in order to cause the rotation of the original hyperplane easily. The detail process is described in Algorithm 1.

Algorithm 1 Adversarial label flip of *far-rotate*

Input: the untainted dataset $\mathscr{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$; the number of label flips L; the number of iterations R; the penalty parameter C of the error term in SVM; the trade-off parameters λ_1 and λ_2.

Output: the tainted dataset $\mathscr{D}'$.

1: train an SVM on the untainted dataset $\mathscr{D}$ and get α and b;
2: calculate $s_i = y_i(\sum_{j=1}^n y_j \alpha_j K(\mathbf{x}_j, \mathbf{x}_i) + b)$ for each sample in $\mathscr{D}$ and make a normalization $s_i = s_i / \max(s_1, ..., s_n)$;
3: **for** $k = 1, ..., R$ **do**
4: train a random SVM and get its α^{rnd} and b^{rnd}
5: calculate $q_i = y_i(\sum_{j=1}^n y_j \alpha_j^{\text{rnd}} K(\mathbf{x}_j, \mathbf{x}_i) + b^{\text{rnd}})$ for each sample in $\mathscr{D}$ and make a normalization $q_i = q_i / \max(q_1, ..., q_n)$;
6: calculate the weight for each sample in $\mathscr{D}$: $v_i = \alpha_i / C - \lambda_1 s_i - \lambda_2 q_i$
7: flip the label of L samples which have the L lowest weights and generate a new tainted dataset $\mathscr{D}'_k$;
8: train an SVM on $\mathscr{D}'_k$ and estimate the error rate (e_k) on the untainted dataset $\mathscr{D}$;
9: **end for**
10: **return** $\mathscr{D}'_k$ which yields the maximum e_k;

Another advanced adversarial label flip attack (*maxErr*) [63, 65] aims to maximize the classification error of the SVM trained on tainted data ($\mathscr{D}' = \{(\mathbf{x}_i, y_i')\}_{i=1}^n$) and tested on untainted data ($\mathscr{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$) . To simplify the algorithm, the expanded representation $U = \{(\mathbf{x}_i, y_i)\}_{i=1}^{2n}$ of $\mathscr{D}$ is constructed by duplicating each sample in D with a flipped label where $(\mathbf{x}_i, y_i) \in \mathscr{D}$ when $i = 1, \ldots, n$, and $\mathbf{x}_i = \mathbf{x}_{i-n}, y_i = -y_{i-n}$ when $i = n+1, \ldots, 2n$. The near-optimal problem is formulated as follows:

$$\begin{aligned} \min_{\mathbf{q},\mathbf{w},\boldsymbol{\xi},b} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{2n} q_i(\xi_i - \xi_i^{\mathscr{D}}) \\ \text{s.t.} \; & y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, 2n, \\ & \sum_{i=n+1}^{2n} q_i \leq L, \\ & q_i + q_{i+n} = 1, \quad i = 1, \ldots, n, \\ & q_i \in \{0, 1\}, \quad i = 1, \ldots, 2n, \end{aligned} \tag{1}$$

where $\mathbf{w}$, $\boldsymbol{\xi}$, b and C are the weight vector, slack variable, bias and penalty parameter of the SVM trained on the tainted dataset respectively. $\boldsymbol{\xi}_i^{\mathscr{D}} = \max(0, 1 - y_i f_{\mathscr{D}}(\mathbf{x}_i))$ denotes the hinge loss of sample $(\mathbf{x}_i, y_i)$ resulting from the classifier trained with untainted data ($f_{\mathscr{D}}$). $\mathbf{q}$ is a binary vector controlling the labels of which samples are flipped. $q_i + q_{i+n} = 1$

makes sure that only one label can be chosen for the sample $\mathbf{x}_i$. L represents the number of label flips. Eq. (1) can be solved by optimizing a linear programming and a quadratic programming iteratively. The vector $\mathbf{q}$ is obtained to determine which samples are attacked. Causative attacks are also designed against online Learning where training samples arrive in a streaming manner [61].

4 Data complexity

Data complexity measures [35] quantify the difficulty of a classification problem by measuring the geometrical characteristics of data. They have been applied to many applications with satisfying performances. For example, they can indicate whether a type of classifiers is suitable for a classification problem before the training process [54], measure the domains of competence of an individual [6, 41] and a combined classifier [17, 34], decrease the information leaks during web browsing [33], sanitize the samples contaminated by causative attack [20] and predict whether a noise filter should be used [52].

Twelve data complexity measures [35] are briefly introduced in this section. Only eleven of them are investigated in this paper as $T2$ is not suitable for attack detection. It is because $T2$ calculates the ratio of the number of samples and features which remain the same after the attack. Data complexity measures are categorized into three types [35]: *overlap of individual feature values*, *separability of classes*, and *geometry, topology and density of manifolds*.

4.1 Measures of overlap of feature values

Discriminative power of features defined as the overlaps among different classes is described. A classification problem is difficult if there exists large overlap between different classes.

Fisher's discriminant ratio ($F1$) represents the separability by the most discriminating feature. The ratio of the square of the difference between the centers of two classes to the summation of the variance of each class is calculated for each feature. $F1$ is defined as the maximum value among all features. A large value of $F1$ indicates that at least one feature can well separate different classes, and is defined as:

$$F1 = max_{i=1}^{d} f_d \tag{2}$$

where d is the number of input features, and f_d is the discrimant ratio of each feature.

Volume of overlapping region ($F2$) calculates the overlapping region based on the product of the ratio between the width of overlapping region and the range of values spanned by both classes for each feature. $F2$ is defined as:

$$F2 = \prod_{j=1}^{d} \frac{\min\left(\max(x_j^{(+)}), \max(x_j^{(-)})\right) - \max\left(\min((x_j^{(+)})), \min(x_j^{(-)})\right)}{\max\left(\max(x_j^{(+)}), \max(x_j^{(-)})\right) - \min\left(\min(x_j^{(+)}), \min(x_j^{(-)})\right)} \tag{3}$$

where $x_j^{(+)}$ and $x_j^{(-)}$ are the jth feature of class + and − respectively, and d denotes the dimensionality of the feature space. A dataset with a large value of $F2$ indicates that the overlapping area of classes is large, *i.e.,* a more difficult classification problem.

Maximum (individual) feature efficiency ($F3$) is the largest fraction of points lying outside the overlapping region for each individual feature. Large value of $F3$ indicates that at least one feature can discriminate most of the samples.

4.2 Measures of separability of classes

Measures of separability of classes represent the difficulty of a classification problem by linear separability ($L1$ and $L2$) and class boundary ($N1$, $N2$ and $N3$) of data.

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 [56]. 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 the construction of a minimum spanning tree (MST) connecting all the samples regardless of class label. It returns the ratio of the number nodes connecting different classes to the total number of samples in a dataset. This measure estimates the length of the class boundary. 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 their classes to the total distance between all samples and their nearest samples of any other classes. A large value of $N2$ indicates that the sample is close to the sample from other class and far from the sample of its own class, which indicates a 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. $N3$ describes how many samples are closer to the sample from other class than the one from the same class. A large value of $N3$ indicates the samples from different classes are highly interleaved.

4.3 Measures of geometry, topology, and density of manifolds

This kind of measures assumes 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) is the error rate of a linear classifier trained on the original dataset and tested on an artificial dataset. The artificial test set is generated by linear interpolation with random coefficients between two samples randomly drawn from the same class. *L*3 is large when the classes are interleaved.

Nonlinearity of 1NN classifier (N4) is the error rate of nearest-neighbor classifier on a artificial test set generated by the same linear interpolation adopted in *L*3. A large value of *N*4 indicates that the data set is highly interleaved.

Fraction of points with associated adherence subsets retained (T1) describes 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. *T*1 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 *T*1 represents that the samples of different classes are close to each other, *i.e.,* the shape of the decision boundary is complicated.

Average number of points per dimension (T2) is the ratio of the number of samples to the number of features. This measure is not adopted in this paper since neither the number of samples nor the number of features is changed in label flip attack.

5 Illustration on the change of geometrical characteristics by causative attacks via a toy example

The training samples are manipulated to mislead the learning of a classifier in a causative attack. This causes an untainted and tainted training set having different geometrical characteristics [8, 11], which can be captured by data complexity measures. We use a simple toy example to illustrate how the geometrical characteristics change due to a causative attack, and also how the data complexity measures capture the change.

An artificial two-dimensional dataset containing 200 samples for each of two classes is generated. The samples of the classes follow a Gaussian distribution with different means ($\mu_+ = \{0.4, 0.5\}$, $\mu_- = \{0.8, 0.5\}$) and the same standard deviation ($\sigma_+ = \sigma_- = 0.1$). The untainted dataset is shown in Fig. 1(a). The decision hyperplane of a SVM with linear kernel and $C = 1$ is illustrated by the solid line. 10% samples of the untainted dataset are attacked by the causative label flip attacks, including *rand*, *near*, *far*, *far-rotate* and *maxErr* separately, shown in Fig. 1(b-f). The random hyperplane generated by *far-rotate* attack is marked by a dotted line in (e). Tainted samples are highlighted with a black circle.

The geometrical characteristics of a tainted dataset attacked by *rand* are obviously different from those of

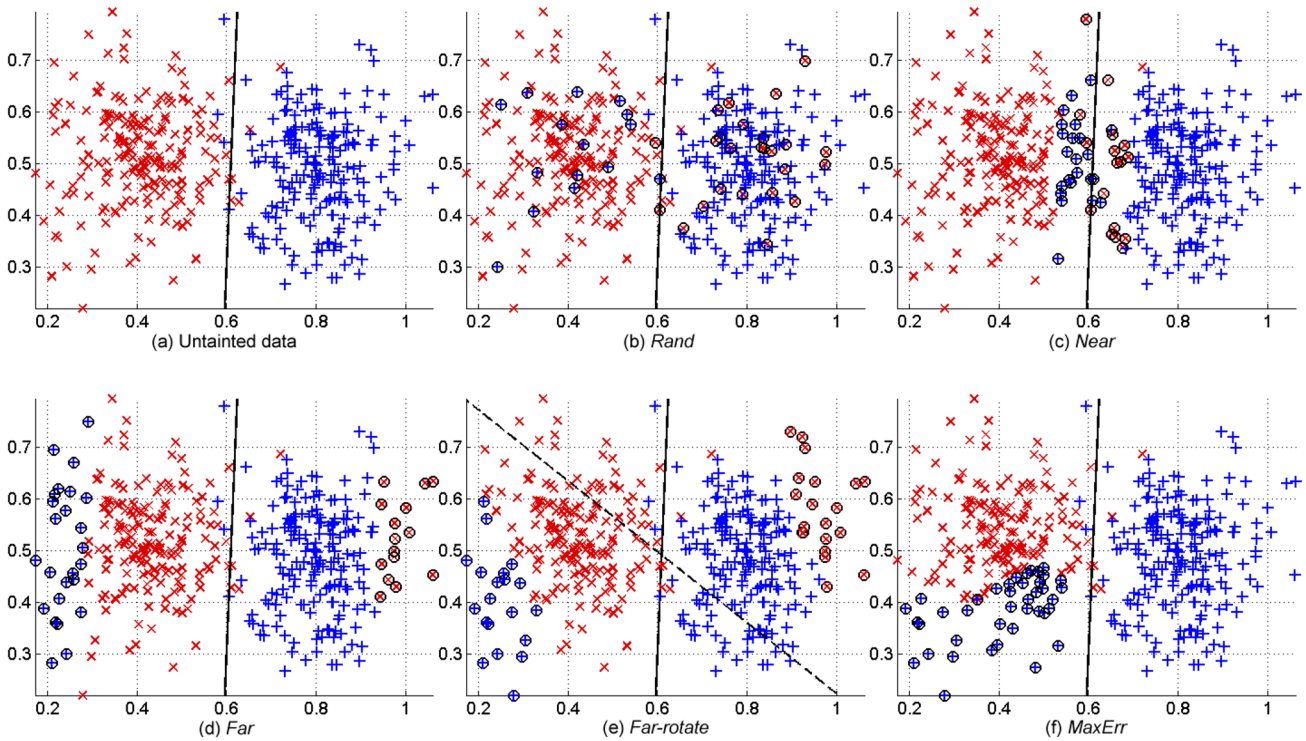

Fig. 1 The artificial datasets in absence of an attack and with different causative label flip attacks

an untainted one since there are some samples spanning over the region of the other class, shown in Fig. 1b. Thus *rand* attack should be easily recognized by using the data complexity measures based on overlapping region, *e.g.,* *F*2 and *F*3, as it significantly increases the overlap region between two classes. Fig. 1c shows that all the attack points by *near* attack are located near the decision boundary. Since this attack complicates the boundary between two classes, the data complexity measures describing the complexity of a class boundary, *e.g.,* *N*1, can detect *near* attack. The attacked samples by *far* attack are far away from the decision hyperplane as shown in Fig. 1d. The classes are interleaved after the attack. The classification error of linear interpolation usually increases dramatically. *L*3 and *N*4 should be useful for detecting this kind of attack. Similar to *far* attack, *far-rotate* attack also prefers to flip the labels of samples which are far away from the decision hyperplane. *L*3 and *N*4 should also be suitable for detecting this kind of attack. All samples in a bottom left corner are attacked in *MaxErr*. This maximizes the classification error of the SVM trained by using the tainted training data on untainted test data. Generally, *MaxErr* attack makes the class boundary more complicated. It may be captured by the complexity measures describing the geometrical characteristics of class boundary, *e.g.,* *N*1 and *T*1. This toy example shows that some specific geometrical characteristics are yielded by different causative attacks, which provide some clues to distinguish contaminated datasets from untainted ones.

The changes of the geometrical characteristics are quantified by the data complexity measures. The results are shown in Table 1. $\Delta\mathbf{x}$ is defined as Eq. (4) to measure the difference of data complexity measures between untainted dataset ($\mathbf{x}$) and its tainted dataset ($\mathbf{x}'$) using ℓ_1-norm.

$$\Delta\mathbf{x} = \frac{1}{d}\sum_{i=1}^{d}\Delta x_i = \frac{1}{d}\sum_{i=1}^{d}\frac{|x_i - x_i'|}{ub_i - lb_i} \tag{4}$$

To avoid domination, all values of a feature are normalized by the difference between its upper bound (*ub*) and lower bound (*lb*). As the upper bounds of *F*1, *L*1 and *N*2 are infinite, we set their upper bounds as 10, 1 and 1 respectively in this example to obtain meaningful values of $\Delta\mathbf{x}$. As shown in Table 1, the change of geometrical characteristics can be captured by data complexity measures efficiently, *i.e.,* $\Delta\mathbf{x}$ is above 0.18 for different causative attacks except for *near* attack (the range of $\Delta\mathbf{x}$ is from 0 to 1). As described in [35], a classification problem is simple as long as there is a discriminating feature. Thus the maximum value of Δx_i for each attack is also shown in the last row of Table 1. The maximum Δx_i for each causative attack is over 0.25, which indicates the existence of at least one data complexity measure which can successfully distinguish between untainted and tainted dataset.

This toy example shows that geometrical characteristics of a dataset are changed after a causative label flip attack. Moreover, the change on characteristics can be quantified by data complexity measures efficiently. Therefore, the untainted and tainted datasets should locate differently in the space combined by data complexity measures. As a result, causative label flip attack can be detected by using data complexity measures.

6 Proposed methods

In this study, we focus on the causative-availability-indiscriminate attack which aims to increase the generalization error of a classifier by flipping the label of the training samples. These attacks usually yield some specific geometrical

Table 1 Data complexity measures of the artificial dataset in absence of an attack and with different causative label flip attacks

Measure	Range	Untainted data	*Rand*	*Near*	*Far*	*Far-rotate*	*MaxErr*
*F*1	[0,inf)	8.66	2.25	6.15	0.94	0.99	2.68
*F*2	[0,1]	0.14	0.74	0.17	0.65	0.74	0.39
*F*3	[0,1]	0.90	0.05	0.86	0.09	0.04	0.43
*L*1	[0,inf)	0.37	0.53	0.40	0.64	0.61	0.53
*L*2	[0,1]	0.02	0.11	0.08	0.17	0.13	0.12
*L*3	[0,1]	0.01	0.09	0.04	0.15	0.15	0.11
*N*1	[0,1]	0.05	0.29	0.09	0.11	0.10	0.08
*N*2	[0,inf)	0.11	0.51	0.14	0.20	0.18	0.14
*N*3	[0,1]	0.03	0.20	0.05	0.06	0.05	0.05
*N*4	[0,1]	0.01	0.15	0.05	0.11	0.12	0.06
*T*1	[0,1]	0.46	0.91	0.55	0.67	0.67	0.60
$\Delta\mathbf{x}$	[0,1]	/	0.35	0.06	0.29	0.29	0.18
$max(\Delta x_i)$	[0,1]	/	0.85	0.25	0.81	0.86	0.60

characteristics to a dataset, discussed in Sect. 5. We capture these specific characteristics from a number of untainted and attacked datasets by different data complexity measures, aiming to determine whether an unseen training set is contaminated or not. In this section, we first propose a causative detection method (CAD) based on data complexity measures. Then we construct a simple two-step secure classification model to illustrate the importance of causative attack detection in a situation where we have no knowledge on whether the training data is contaminated or not.

6.1 Formulation for causative attack detection

The problem formulation of the causative attack detection based on data complexity measures is devised in this section. We let a sample $\mathbf{x}^D = \{x_1, x_2, \ldots, x_d\}$ be a dataset and x_i be its geometrical characteristic quantified by a data complexity measure where $i = 1, 2, \ldots, d$. $\mathscr{X}$ is the input space composed by data complexity measures and $\mathscr{X} \subset \mathbb{R}^d$, *i.e.*, $\mathbf{x}^D \in \mathscr{X}$.

As we aim to determine whether the causative attack is presented in the given dataset, the detection problem can be formulated as a 2-class problem. For the 2-class problem, a class label $y^D \in \mathscr{Y}_b = \{-1, +1\}$ is assigned to each $\mathbf{x}$. If $\mathbf{x}$ contains the causative attack, $y = +1$; otherwise, $y = -1$.

Let $D = \{(\mathbf{x}_i^D, y_i^D)\}_{i=1}^n$ be the training set with n samples drawn independently and identically distributed from a joint distribution $P_{\mathscr{X}\times\mathscr{Y}}$. The classification function trained on D is defined as $f : \mathscr{X} \mapsto \mathscr{Y}$. The objective is to increase the accuracy of f on classifying whether the causative attack is presented in a given dataset ($\mathbf{x}$).

Figure 2 shows the architecture of the proposed causative attack detection. A number of untainted datasets are collected firstly. The contaminated datasets are prepared by adding simulated attack to the collection of datasets. The features of untainted and contaminated samples are prepared by extracting data complexity measures from untainted and contaminated datasets. The label indicates whether the dataset is attacked (+1) or not (-1). Finally, a 2-class classifier is trained to detect whether a new dataset is contaminated. It should be noted that contaminated datasets are distinguished from the untainted ones according to geometrical characteristics measured by data complexity measures, but not only how complex it is. Thus, a complex dataset may not necessarily be classified as attacked by the classifier.

6.2 Two-step secure classification model

Since most countermeasures of causative attack, *i.e.,* robust learning and data sanitization, sacrifice the generalization ability to gain the security of a classifier [12], we shall well study a collected dataset before applying these countermeasures. A simple two-step secure classification model is proposed to illustrate how the proposed causative attack detection improves the robustness of learning.

The proposed model contains two steps: causative attack detection and classifier training. The proposed causative attack method as described in Sect. 6.1 is firstly applied to determine whether a given dataset is contaminated. If the dataset is contaminated, a robust learning algorithm, which increases both generalization ability and robustness of a classifier, will used; otherwise, a traditional one, which focuses on generalization ability, will be chosen. The architecture of the two-step secure classification model is shown in Fig. 3. Ensembles of decision trees (EDTs) is used as a classifier in our system since it is able to deal with the input features with different ranges. As shown in Table 1, the ranges of different complexity measures vary a lot. This model is expected to perform better than both traditional and

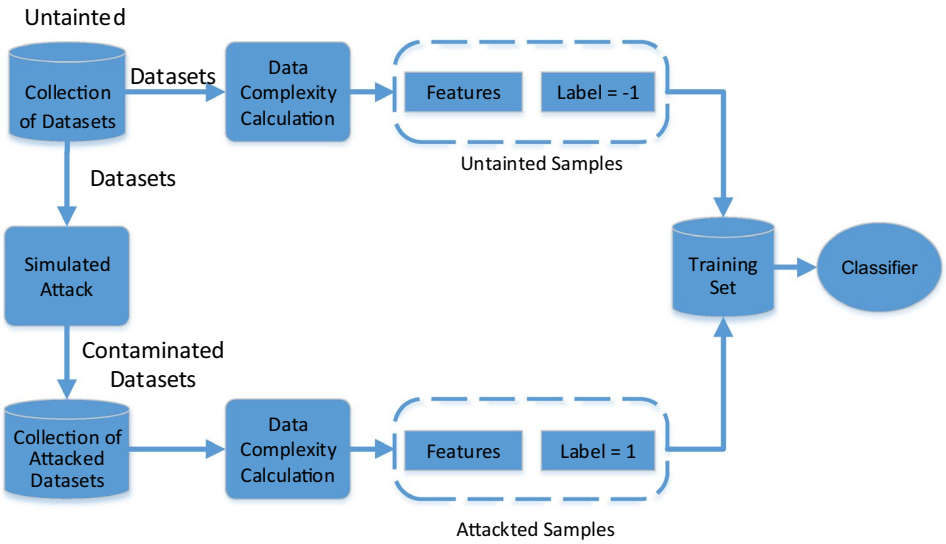

Fig. 2 Architecture of the proposed causative attack detection

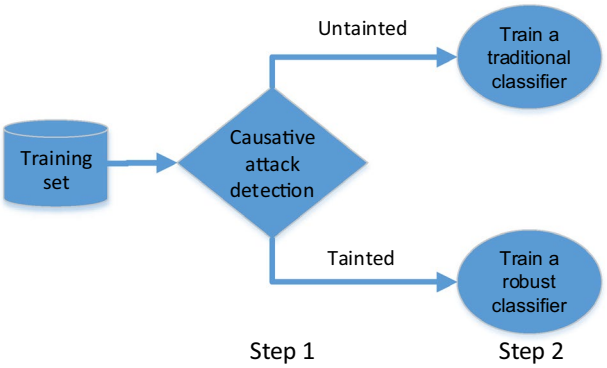

Fig. 3 Architecture of the proposed two-step secure classification model

robust learning algorithms with and without the existence of an adversary since a collected dataset is analyzed firstly.

7 Experiments

The proposed causative attack detection method using 11 data complexity measures mentioned in Sect. 4 is evaluated experimentally in this section. The five well known causative label flip attacks including *rand*, *near*, *far*, *far-rotate* and *maxErr* described in Sect. 3 are studied in the experiments. Sect. 7.1 shows the performances of detection of different causative attacks with different attack strengths. The performance of a two-step secure system composed of causative attack detection and classifier training is measured with five security-related datasets in Sect. 7.2. The distributions of the untainted datasets and the datasets with different kinds of causative attacks are illustrated visually and the discriminating ability of each data complexity measure is analyzed in Sect. 7.3.

7.1 Causative label flip attack detection based on data complexity measures

The causative attack detection based on data complexity measures is evaluated experimentally in this section. The causative attack detection is formulated as a 2-class classification problem, which determines whether a dataset is attacked.

According to the problem formulation discussed in Sect. 6.1, we would like to detect the causative attack based on the difference between geometrical characteristics of untainted and tainted datasets. Therefore, a set of datasets will be prepared for learning. 70 two- and multi-class datasets with a wide range of variety are selected from the UCI Machine Learning Repository [3] and KEEL-dataset Repository [2] (see Appendix A).

The 70 datasets are randomly split into a training (D_0^{Tr}) and a test set (D_0^{Ts}). Similar to previous works [35, 41], multi-class datasets in D^{Tr} and D^{Ts} are then divided into two-class datasets by considering each pair of classes as an individual classification problem respectively. Only a two-class problem is investigated since it is the common formulation in security problems in which only the legitimate and malicious classes are considered. As a result, $D_0^{Tr} \cup D_0^{Ts}$ contains 384 untainted two-class datasets. Tainted training and test sets ($D_{m,i}^{Tr}$ and $D_{m,i}^{Ts}$) are generated by contaminating D_0^{Tr} and D_0^{Ts} separately according to the attack m (*i.e.,* *rand*, *near*, *far*, *far-rotate* and *maxErr*) with $i\%$ attack strength, where $i = 2.5, 5, 7.5, \ldots 20$.

The causative attack detection is considered in three scenarios characterized by different levels of the defender's knowledge on the attack. In *Perfect Knowledge* scenario, we assume that the defender has perfect knowledge of the type of attack and the attack strength. A classifier is trained for each type of causative attack with each attack strength. The training set is combined by D_0^{Tr} $(y = -1)$ and $D_{m,i}^{Tr}$ $(y = +1)$. In *Partial Knowledge* scenario, the defender knows the type of attack but has no knowledge on the attack strength. A classifier is trained for each type of causative attack with the combination of different attack strengths. Thus the training set consists of the untainted samples (D_0^{Tr}) and the tainted samples attacked by a specific attack m (D_m^{Tr}). *No Knowledge* scenario, in which the defender has no knowledge of the attack type and the attack strength, is the worst case for the defender. Only one classifier is trained on a dataset containing different types of attacks with different attack strengths. As a result, the training set contains the untainted samples D_0^{Tr} $(y = -1)$ and the tainted samples with different types of attack and strength D^{Tr} $(y = +1)$.

The number of the positive samples is much larger than the negative samples in the second and the last scenarios. To avoid an imbalance problem, a subset is randomly selected from each attack set to form the positive class in order to maintain the same ratio between the positive and the negative samples. The test set, which contains D_0^{Tr} $(y = -1)$ and $D_{m,i}^{Tr}$ $(y = +1)$ for any m and i, is the same for all scenarios. As the ranges of data complexity measures vary a lot, we use ensembles of Decision Trees (EDTs) which are in general to any monotonic scaling of the features, *i.e.,* EDTs are able to handle features in their original forms. In each experiment, the test set is $D_0^{Te} \cup D_{m,i}^{Te}$. The number of decision trees is set to 100. Each experiment is repeated 30 times independently.

The experimental results are illustrated in Fig. 4. The causative attack detection using data complexity measures shows satisfying performance. The detection accuracies of causative attacks are higher than 82%, 80% and 75% for

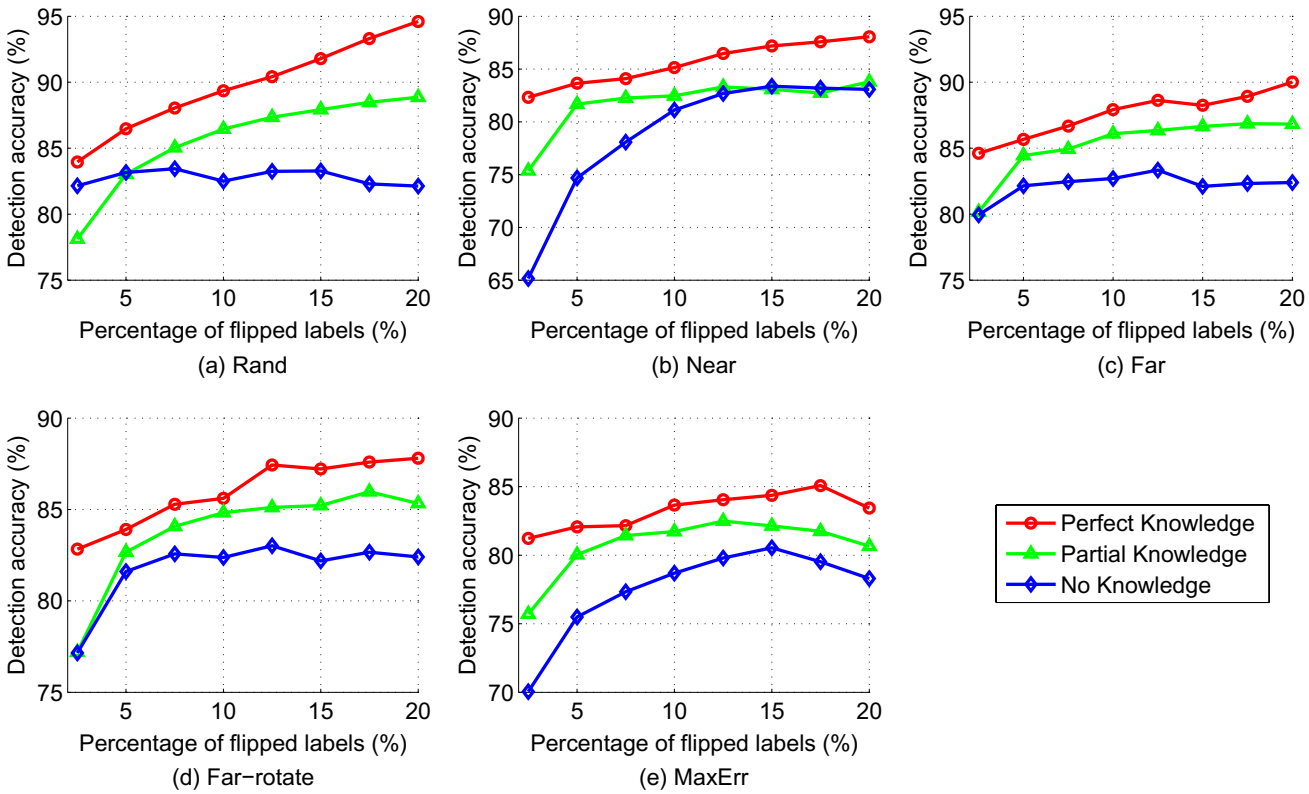

Fig. 4 Detection accuracy of different causative attacks with different attack strengths

Perfect Knowledge, *Partial Knowledge* and *No Knowledge* scenarios respectively, even though the attack strength is only 5%. When the attack strength is 15%, the detection accuracies of *Perfect Knowledge*, *Partial Knowledge* and *No Knowledge* scenarios increase to 84%, 82% and 80% respectively. Then the accuracies remain constant or fluctuate slightly in most cases when the attack strength increases from 15% to 22%.

The accuracies are the highest in *Perfect Knowledge* scenario and the lowest in *No Knowledge* scenario among all cases. This is because in *Perfect Knowledge* scenario we assume that the defender knows exactly the type of attack as well as its parameters. A classifier trained for this particular attack setting is chosen for detection. Differently, no information is obtained in *No Knowledge* scenario. Only one classifier is trained to detect all kinds of attacks. Since classifying all kinds of attack is more complicated than classifying a specific one, the accuracies of our proposed model in *Perfect Knowledge* scenario is obviously higher than the ones in *No Knowledge* scenario. However, the average accuracy of the classifier in *No Knowledge* scenario still achieves 80.61%. It confirms the data complexity measures are useful for causative label flip attack detection.

As described in [35], since the distributions of data with random noise and the real world data are obviously different in the data complexity space, the average accuracy of *rand* attack detection is 86.06%, which is the highest among all kinds of attacks. The average detection accuracies of *far* (85.02 %) and *far-rotate* (83.84%) are similar since both of these attack strategies prefer to attack the samples which are far away from the decision hyperplane. *Near* (82.10%) and *maxErr* attacks (80.48%) are difficult to be detected. The reason might be that these attacks do not significantly introduce special characteristics which seldomly appear in untainted datasets. *MaxErr* attack is the most difficult to be detected (80.48%). The reason might be that *maxErr* does not significantly introduce special characteristics to a dataset, *i.e.,* the values of data complexity measures are similar, as shown in Sect. 7.3.

7.2 A two-step secure system

This section aims to demonstrate the essentiality of the proposed two-step secure classification method. In this experiment, we assume that the defender has *Perfect Knowledge* defined in last section, *i.e.,* the attack type and strength have been known. *MaxErr*, which is one of the most representative label flip attack, is used. Its attack strength is 10% which is a common setting in a causative attack [12, 63]. Five security-related datasets, including Steghide, MB1, F5,

PDF and Spam, are used. Steghide, MB1 and F5 are datasets of steganalysis [32], while PDF and Spam are datasets of malware detection and spam filtering. More details of these datasets are given in [16].

For the first step of the proposed model, the training set of the proposed causative attack detection is prepared as follows. Data complexity measures of the 384 untainted datasets described in Sect. 7.1, and also 384 contaminated datasets generated by introducing *MaxErr* attack with 10% attack strength to these untainted datasets are extracted to generate a training set. EDTs with 100 trees are trained to classify whether a dataset is contaminated. It should be noted that five security-related datasets are not used in the first step.

A traditional SVM [22] and a robust SVM with kernel matrix correction proposed in [12] are used in untainted and contaminated situation in the second step of our model. For each of the five security-related datasets, 50% of the data are randomly chosen for training while the rest is used as a test set. We generate contaminated training set with the same attack setting in the first step. For the untainted and contaminated training set, we firstly detect whether it is attacked by the EDTs trained in the first step. If the output of EDTs is +1 (*i.e.,* attacked), we will use a robust SVM, otherwise a traditional SVM will be used. Parameters of the SVM are determinated by an 5-fold cross-validation to maximize the classification accuracy. The parameter μ in the robust SVM [12] is set to 0.1. Each experiment is carried out 10 times independently.

Experimental results are described in Table 2. The robust SVM has worse performance than the traditional one when the training set is untained, especially for Steghide and MB1, in which the robust SVM is 5% less accurate. This confirms that robust learning should only be used when a dataset is contaminated by the causative attack. On the other hand, when the training set is contaminated, although performances of all methods drop, the accuracy of the robust SVM is much higher than the traditional SVM. It explains why causative attack detection should be carried out before choosing the learning method.

We also show the accuracy of the causative detection (see the last column in Table 2) in Step 1. Performance of our proposed 2-step classification model is closely related to the accuracy of its causative attack detection as it determinates which classifier to be used. 100% accuracy of CAD means that we can correctly recognize whether the training set is contaminated all the times.. When the causative attack detection is 100% accurate, the proposed model achieves the same accuracy as the traditional SVM without attack and the robust SVM with attack. The accuracy of the 2-step classification model drops with the decrease of accuracy of causative attack detection. For example, the accuracy of CAD is only 70% in Steghide with attack, the 2-step classification model is less accurate than the robust SVM. Our method achieves the highest average accuracy in all datasets. It suggests that our method perform better than the traditional SVM and the robust SVM in general, *i.e.,* the proposed causative attack detection is able to improve the robustness of learning.

Table 2 Classification accuracies (%) of traditional SVM, robust SVM, the proposed 2-step classification model (Proposed Model), and also the proposed causative attack detection method (CAD) on different security datasets with and without causative attack

Dataset	Training set	SVM	Robust SVM	Proposed model	CAD
Steghide	Untainted	**97.94**	92.72	**97.94**	100.00
	tainted	86.80	**90.64**	89.40	70.00
	Average	92.37	91.68	**93.67**	85.00
MB1	Untainted	**95.92**	89.74	**95.92**	100.00
	tainted	86.12	**89.36**	**89.36**	100.00
	Average	91.02	89.55	**92.64**	100.00
F5	Untainted	**92.22**	90.78	**92.22**	100.00
	tainted	83.12	**91.20**	91.08	90.00
	Average	87.67	90.99	**91.65**	95.00
PDF	Untainted	**98.88**	98.87	**98.88**	100.00
	tainted	87.9	**93.45**	**93.45**	100.00
	Average	93.39	96.16	**96.17**	100.00
Spam	Untainted	**99.36**	99.26	**99.36**	100.00
	tainted	88.42	**94.50**	**94.50**	100.00
	Average	93.89	96.88	**96.93**	100.00

Bold values indicate the highest ones among SVM, robust SVM and proposed model

7.3 Discriminating ability of the data complexity measure on causative label flip attack detection

The distributions of untainted datasets and the datasets with different causative attacks are illustrated. Each of 384 datasets in D_0 described in APPENDIX A is attacked according to *rand*, *near*, *far*, *far-rotate* and *maxErr* to form the sets of tainted datasets, denoted by D_{rand}, D_{near}, D_{far}, D_{maxErr} and $D_{far-rotate}$ respectively. The attack strength is set to 10%, which is a common setting in a study of the causative attack [12, 63]. 11 data complexity measures are calculated for each dataset. The probability density function (PDF) estimated based on a normal kernel function of each set on each data complexity individually is plotted in Fig. 5.

Larger values of complexity measures, except for $F1$ and $F3$, indicate that the classification problem is more difficult. As shown in Fig. 5, the distributions of attacked sets are located on the right of the untainted set and also their means are larger than the ones of the untainted set, except for $F1$ and $F3$. This indicates that causative attacks increase the difficulty of a classification problem in general.

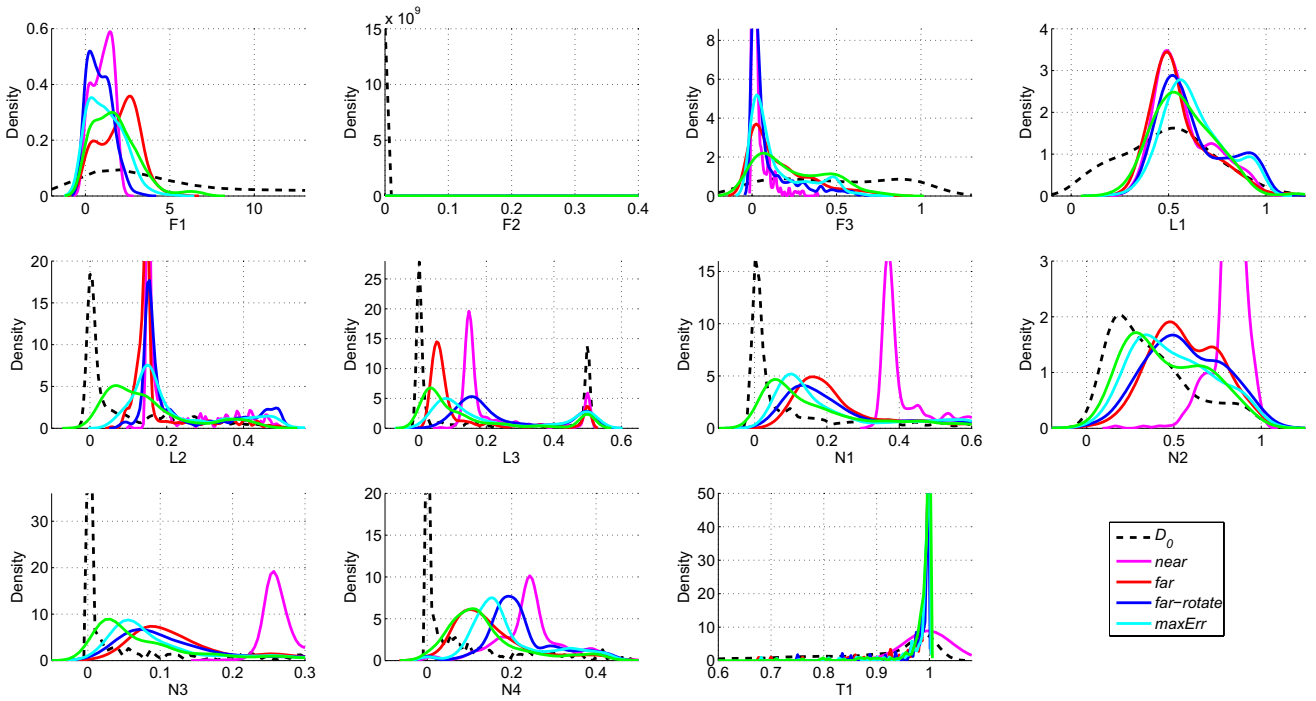

Fig. 5 Probability density functions of untainted datasets and the datasets with *rand*, *near*, *far*, *far-rotate* and *maxErr* attacks on different complexity measures

The overlap of distributions of untainted sets (denoted by the black dotted lines) and tainted sets (denoted by the colored solid lines) is small in some data complexity measures, *e.g.,* $L2$, $L3$, $N1$ and $N4$. This means that a dataset in the absence of an attack can be distinguished from tainted datasets efficiently using those data complexity measures. By contrast, the distributions of untainted and tainted sets overlap significantly using $L1$ and $N2$, *i.e.,* these two measures are less useful to detect causative attacks individually. Although no single data complexity measure can perfectly distinguish the untainted from the tainted sets, the detection accuracy can be increased by combining all data complexity measures. This confirms the underlying rationality of the proposed causative attack detection based on data complexity measures.

We follow the the definition of discriminating ability, which is quantified by using the Fisher's discriminant ratio, in previous studies [23, 35, 43] in order to analyze the relevance of a data complexity measure to causative attack detection. Fisher's discriminant ratio calculates the ratio between the square of the difference between the centers of two classes and the summation of the variance of each class. Larger value of Fisher's discriminant ratio indicates larger discriminating ability. Fisher's discriminant ratio of a set combined with D_0 and one of the attacked sets (*i.e.,* D_{rand}, D_{near}, D_{far}, D_{maxErr} and $D_{far-rotate}$) for a data complexity measure is shown in Table 3. The rank of data complexity measures according to the discriminating ability is shown in the parentheses (rank 1 indicates the highest discriminating ability).

Table 3 The discriminating ability of different data complexity (DC) measures for causative attack detection

DC	*Rand*	*Near*	*Far*	*Far-rotate*	*MaxErr*
F1	0.099 (11)	0.049 (9)	0.109 (11)	0.095 (11)	0.038 (9)
F2	3.718 (5)	0.307 (7)	1.112 (3)	0.628 (4)	0.059 (6)
F3	1.589 (6)	0.454 (4)	0.915 (4)	0.726 (3)	0.161 (4)
L1	0.101 (10)	0.003 (11)	0.199 (10)	0.194 (10)	0.028 (10)
L2	0.855 (7)	0.188 (8)	1.189 (2)	1.039 (2)	0.053 (7)
L3	0.404 (9)	0.018 (10)	0.684 (5)	0.408 (5)	0.004 (11)
N1	5.442 (2)	1.000 (2)	0.631 (6)	0.355 (7)	0.208 (2)
N2	5.079 (3)	0.458 (3)	0.491 (8)	0.271 (9)	0.106 (5)
N3	6.596 (1)	1.013 (1)	0.509 (7)	0.279 (8)	0.195 (3)
N4	3.917 (4)	0.383 (5)	2.573 (1)	1.591 (1)	0.050 (8)
T1	0.411 (8)	0.370 (6)	0.382 (9)	0.373 (6)	0.315 (1)
Mean	2.564	0.386	0.799	0.542	0.111

A dataset with *rand* attack is easy to be distinguished from an untainted one in general as the average Fisher's discriminant ratios for all data complexity measures are high. Data complexity measures with a ratio larger than 1 are $N3$, $N1$, $N2$, $N4$, $F2$ and $F3$ in a descending order. As the attacked samples are selected randomly, they are similar to

the random noise. $N1$, $N2$, $N3$ and $N4$ describe the separability of classes based on the nearest neighbor measure which is noise sensitive [1, 48]. $F2$ and $F3$ are sensitive to noise samples as they can significantly change the overlapping region.

$N3$ and $N1$ have the highest discriminating ability to separate a dataset attacked by *near* attack from the untainted one. $N1$ describes the fraction of points with the nearest samples not in the same class, while $N3$ measures the size of the gap between two classes which is measured by the Euclidean distance between the closest samples from two classes. Since the labels of samples closest to a decision boundary are flipped, the width of the decision boundary will decrease. It explains why *near* attack can be captured by $N1$ and $N3$ efficiently.

$N4$, $L2$, $F2$, $F3$ and $L3$ are discriminating for *far* attack. *Far* increases the error rate of a classifier on the dataset generated by linear interpolation significantly, which are measured by $N4$ and $L3$. $L2$ is also good for detecting *far* attack since it causes the classes interleaving which increases the error of a linear classifier. Flipping the labels of samples which are the furthest to the decision hyperplane significantly increases the size of the overlapping region of different classes. Thus $F2$ and $F3$, which calculate the overlapping region of tainted datasets attacked by *far* are obviously different from those of untainted ones. As discussed before, the behavior of *far-rotate* is similar to *far*. The ranks of *far* and *far-rotate* are similar, so $N4$, $L2$, $F3$, $F2$ and $L3$ also have good performances in discriminating *far-rotate* attack.

The ratios of data complexity measures are relatively smaller for classifying the *maxErr* attack. Three largest values are 0.315 ($T1$), 0.208 ($N1$) and 0.195 ($N3$). As the complexity of the class boundary increases after the *maxErr* attack, $N1$, $T1$ and $N3$ describing the geometrical characteristic of class boundary are most discriminating.

The experimental results show that distributions of data complexity measures for the untainted and the contaminated datasets are different. This confirms that the influence of the label flip attack on a dataset can be captured by data complexity measures. The data complexity measure provides useful information for label flip attack detection.

8 Conclusion and future studies

Causative attack detection is an essential issue in adversarial learning as most countermeasures usually increase the security by sacrificing the generalization ability. To the best of our knowledge, the issue of causative attack detection has not been deeply investigated. Although some existing methods may be related, only using the information provided by the samples collected from a classification problem is not enough to detect whether the dataset contains a causative attack.

Inspired by the observation that characteristics of a dataset are changed by a causative attack, we investigate the geometrical nature of untainted and tainted datasets by using the data complexity measure. The five well-known label flip attacks including random label flip, nearest-first label flip, furthest-first label flip, advanced adversarial label flips are studied. By formulating a dataset as a sample in an input space composed by 11 data complexity measures, distributions of untainted and tainted datasets with different attacks are separated significantly. Moreover, the causative attack detection is formulated as a 2-class classification problem. Ensemble of decision trees is applied as a classifier. It achieves satisfying performance in three scenarios characterized by different levels of the defender's knowledge on the attack. A two-step secure classification model is also implemented to illustrate the importance and necessity of the causative attack detection before training a classifier.

One possible extension of this work is related to the detection of causative attacks with feature noise. The situation is quite different from the label flip attack, since the geometrical nature of a dataset is changed slightly by the attack with feature noise, the detection may be more difficult. Considering defense-awareness causative attack is another meaningful study. Difficulty of the detection is expected to increase since the contaminated datasets will be much more similar to the untainted ones when a defense mechanism is considered in the attack. More features or an advanced classifier may be considered. However, most existing causative label flips attacks only focus on the attack performance but not the concealment. No defense-awareness causative label flips attack can be found in the literature. Thus, we would leave this to the future work.

Acknowledgements This work is supported by Natural Science Foundation of Guangdong Province, China (No. 2018A030313203), the Fundamental Research Funds for the Central Universities SCUT (No. 2018ZD32), the National Natural Science Foundation of China (No. 61802061) and the Project of Department of Education of Guangdong Province (No. 2017KQNCX216).

Dataset used in the experiments

Total 70 2-class and multi-class datasets with a wide range of variety are selected from UCI Machine Learning Repository [3] and KEEL-dataset Repository [2] are as follows: ala, australian, banana shape, banana subset, card, circular, diabetes, fourclass, german numer, hepatitis, highleyman, ionosphere, krvskp, liver-disorders, pima, promoters, sonar, splice, tic-tac-toe, two spirals, vote, balance, dna, gene, horse, iris, post-op, wine, hypo, lymph, vehicle, anneal, page-blocks, autos, dermatology, glass, satimage, segment, zoo, ecoli, pendigits, poker, car, mushroom, spambase, bands, yeast, cleveland, contraceptive,

wisconsin, dermatology, ecoli, wdbc, haberman, hayes-roth, hepatitis, housevotes, titanic, mammographic, marketing, monk-2, newthyroid, texture, optdigits, page-blocks, penbased, phoneme, tae, ring, and saheart.

References

1. Aha DW, Kibler D (1989) Noise-tolerant instance-based learning algorithms. In: Proceedings of the 11th international joint conference on artificial intelligence—Volume 1, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, IJCAI'89, pp 794–799
2. Alcalá-Fdez J, Fernández A, Luengo J, Derrac J, García S, Sánchez L, Herrera F (2011) Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. J Multiple-Valued Logic Soft Comput 17(2–3):255–287
3. Bache K, Lichman M (2013) UCI machine learning repository. http://archive.ics.uci.edu/ml. Accessed 6 Oct 2018
4. 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, ASIACCS '06, pp 16–25
5. Barreno M, Nelson B, Joseph AD, Tygar JD (2010) The security of machine learning. Mach Learning 81(2):121–148
6. Bernado-Mansilla E, Ho TK (2005) Domain of competence of xcs classifier system in complexity measurement space. IEEE Trans Evolut Comput 9(1):82–104
7. Bhagoji AN, He W, Li B, Song D (2018) Practical black-box attacks on deep neural networks using efficient query mechanisms. In: European conference on computer vision, Springer, pp 158–174
8. Biggio B (2010) Adversarial pattern classification. PhD thesis, University of Cagliari, Cagliari (Italy)
9. Biggio B, Roli F (2018) Wild patterns: ten years after the rise of adversarial machine learning. Pattern Recognition 84:317–331
10. Biggio B, Fumera G, Roli F (2010) Multiple classifier systems for robust classifier design in adversarial environments. Int J Mach Learning Cybernet 1(1–4):27–41
11. Biggio B, Corona I, Fumera G, Giacinto G, Roli F (2011a) Bagging classifiers for fighting poisoning attacks in adversarial classification tasks. In: International workshop on multiple classifier systems. Springer, Berlin, pp 350–359
12. Biggio B, Nelson B, Laskov P (2011b) Support vector machines under adversarial label noise. In: Journal of machine learning research—proc. 3rd Asian conference on machine learning (ACML 2011), Taoyuan, Taiwan, vol 20, pp 97–112
13. Biggio B, Nelson B, Laskov P (2012) Poisoning attacks against support vector machines. In: 29th Int'l Conf. on Machine Learning (ICML), Omnipress
14. Biggio B, Corona I, Maiorca D, Nelson B, Srndic N, Laskov P, Giacinto G, Roli F (2013) Evasion attacks against machine learning at test time. In: European conference on machine learning and principles and practice of knowledge discovery in databases (ECML PKDD), Springer-Verlag Berlin Heidelberg, vol 8190, pp 387–402
15. Biggio B, Fumera G, Roli F (2014) Security evaluation of pattern classifiers under attack. IEEE Trans Knowl Data Eng 26:984–996
16. Biggio B, Corona I, He ZM, Chan PPK, Giacinto G, Yeung DS, Roli F (2015) One-and-a-half-class multiple classifier systems for secure learning against evasion attacks at test time. Int'l Workshop Multiple Classifier Syst (MCS) 9132:168–180
17. Britto AS, Sabourin R, Oliveira LE (2014) Dynamic selection of classifiersa comprehensive review. Pattern Recognition 47(11):3665–3680
18. Callison-Burch C, Dredze M (2010) Creating speech and language data with amazon's mechanical turk. In: Proceedings of the NAACL HLT 2010 workshop on creating speech and language data with Amazon's Mechanical Turk, Association for Computational Linguistics, pp 1–12
19. Carlini N, Wagner D (2017) Towards evaluating the robustness of neural networks. In: 2017 IEEE symposium on security and privacy (SP), IEEE, pp 39–57
20. Chan PP, He ZM, Li H, Hsu CC (2018) Data sanitization against adversarial label contamination based on data complexity. Int J Mach Learn Cybernet 9(6):1039–1052
21. Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: a survey. ACM Comput Surveys (CSUR) 41(3):15
22. Chang CC, Lin CJ (2011) Libsvm: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST) 2(3):27
23. Cheng H, Yan X, Han J, Hsu CW (2007) Discriminative frequent pattern analysis for effective classification. In: IEEE 23rd international conference on data engineering, IEEE, pp 716–725
24. Chung SP, Mok AK (2006) Allergy attack against automatic signature generation. In: Proceedings of the 9th international conference on recent advances in intrusion detection, Springer-Verlag, RAID'06, pp 61–80
25. Cretu GF, Stavrou A, Locasto ME, Stolfo SJ, Keromytis AD (2008) Casting out demons: sanitizing training data for anomaly sensors. In: Security and privacy, 2008. SP 2008. IEEE symposium on, IEEE, pp 81–95
26. Dalvi N, Domingos P, Mausam, Sanghai S, Verma D (2004) Adversarial classification. In: Proceedings of the Tenth ACM SIGKDD International conference on knowledge discovery and data mining, ACM, KDD '04, pp 99–108
27. Dekel O, Shamir O (2009) Good learners for evil teachers. In: Proceedings of the 26th annual international conference on machine learning, ACM, pp 233–240
28. Demontis A, Biggio B, Fumera G, Giacinto G, Roli F (2017) Infinity-norm support vector machines against adversarial label contamination. In: ITASEC, pp 106–115
29. Dries A, Rückert U (2009) Adaptive concept drift detection. Stat Anal Data Mining 2(5–6):311–327
30. 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
31. Fierrez-Aguilar J, Ortega-Garcia J, Gonzalez-Rodriguez J, Bigun J (2005) Discriminative multimodal biometric authentication based on quality measures. Pattern Recognition 38(5):777–779
32. He ZM (2012) Cost-sensitive steganalysis with stochastic sensitvity and cost sensitive training error. Int Conf Mach Learn Cybernet 1:349–354
33. 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
34. Ho TK (2002) A data complexity analysis of comparative advantages of decision forest constructors. Pattern Anal Appl 5(2):102–112
35. Ho TK, Basu M (2002) Complexity measures of supervised classification problems. IEEE Trans Pattern Anal Mach Intell 24(3):289–300
36. 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

37. Kantchelian A, Tygar J, Joseph A (2016) Evasion and hardening of tree ensemble classifiers. In: International conference on machine learning, pp 2387–2396
38. Lakhina A, Crovella M, Diot C (2004) Diagnosing network-wide traffic anomalies. ACM SIGCOMM Comput Commun Rev ACM 34:219–230
39. 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
40. Lowd D, Meek C (2005) Adversarial learning. In: Proceedings of the eleventh ACM SIGKDD international conference on knowledge discovery in data mining, ACM, New York, NY, USA, KDD '05, pp 641–647
41. Luengo J, Herrera F (2012) Shared domains of competence of approximate learning models using measures of separability of classes. Inform Sci 185(1):43–65
42. Madani P, Vlajic N (2018) Robustness of deep autoencoder in intrusion detection under adversarial contamination. In: Proceedings of the 5th annual symposium and Bootcamp on hot topics in the science of security, ACM, p 1
43. Mao K (2002) Rbf neural network center selection based on fisher ratio class separability measure. IEEE Trans Neural Netw 13(5):1211–1217
44. Nelson B (2010) Behavior of machine learning algorithms in adversarial environments. PhD thesis, EECS Department, University of California, Berkeley
45. 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
46. Papernot N, McDaniel P, Jha S, Fredrikson M, Celik ZB, Swami A (2016) The limitations of deep learning in adversarial settings. In: IEEE European symposium on security and privacy, IEEE, pp 372–387
47. Papernot N, McDaniel P, Goodfellow I, Jha S, Celik ZB, Swami A (2017) Practical black-box attacks against machine learning. In: Proceedings of the 2017 ACM on Asia conference on computer and communications security, pp 506–519
48. Pekalska E, Paclik P, Duin RPW (2002) A generalized kernel approach to dissimilarity-based classification. J Mach Learn Res 2:175–211
49. Ramachandran A, Feamster N, Vempala S (2007) Filtering spam with behavioral blacklisting. In: Proceedings of the 14th ACM conference on computer and communications security, ACM, pp 342–351
50. Roli F, Biggio B, Fumera G (2013) Pattern recognition systems under attack. Progress in pattern recognition, image analysis, computer vision, and applications. Springer, Berlin, pp 1–8
51. 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
52. SáEz JA, Luengo J, Herrera F (2013) Predicting noise filtering efficacy with data complexity measures for nearest neighbor classification. Pattern Recognition 46(1):355–364
53. Sahami M, Dumais S, Heckerman D, Horvitz E (1998) A bayesian approach to filtering junk e-mail. In: Learning for text categorization: papers from the 1998 workshop, vol 62, pp 98–105
54. Sánchez JS, Mollineda RA, Sotoca JM (2007) An analysis of how training data complexity affects the nearest neighbor classifiers. Pattern Anal Appl 10(3):189–201
55. Servedio RA (2003) Smooth boosting and learning with malicious noise. J Mach Learn Res 4:633–648
56. Smith FW (1968) Pattern classifier design by linear programming. IEEE Trans Comput 100(4):367–372
57. Smutz C, Stavrou A (2012) Malicious pdf detection using metadata and structural features. In: Proceedings of the 28th annual computer security applications conference, ACM, pp 239–248
58. Soule A, Salamatian K, Taft N (2005) Combining filtering and statistical methods for anomaly detection. In: Proceedings of the 5th ACM SIGCOMM conference on internet measurement, USENIX Association
59. Su J, Vargas DV, Sakurai K (2019) One pixel attack for fooling deep neural networks. IEEE Transactions on Evolutionary Computation
60. Tuv E, Borisov A, Runger G, Torkkola K (2009) Feature selection with ensembles, artificial variables, and redundancy elimination. J Mach Learn Res 10(Jul):1341–1366
61. Wang Y, Chaudhuri K (2018) Data poisoning attacks against online learning. arXiv preprint arXiv:180808994
62. Whitehill J, Wu Tf, Bergsma J, Movellan JR, Ruvolo PL (2009) Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In: Advances in neural information processing systems, pp 2035–2043
63. Xiao H, Xiao H, Eckert C (2012) Adversarial label flips attack on support vector machines. 20th European Conference on artificial intelligence (ECAI). Montepellier, France, pp 870–875
64. 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
65. Xiao H, Biggio B, Nelson B, Xiao H, Eckert C, Roli F (2015b) Support vector machines under adversarial label contamination. Neurocomputing 160:53–62
66. Zhang F, Chan PP, Tang TQ (2015) L-gem based robust learning against poisoning attack. In: 2015 International conference on wavelet analysis and pattern recognition (ICWAPR), IEEE, pp 175–178
67. Zhang F, Chan P, Biggio B, Yeung D, Roli F (2016) Adversarial feature selection against evasion attacks. IEEE Trans Cybernet 46:766–777
68. Zügner D, Akbarnejad A, Günnemann S (2018) Adversarial attacks on neural networks for graph data. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, ACM, pp 2847–2856