Systematic Poisoning Attacks on and Defenses for Machine Learning in Healthcare

Mehran Mozaffari-Kermani, *Member, IEEE*, Susmita Sur-Kolay, *Senior Member, IEEE*, Anand Raghunathan, *Fellow, IEEE*, and Niraj K. Jha, *Fellow, IEEE*

***Abstract*—Machine learning is being used in a wide range of application domains to discover patterns in large datasets. Increasingly, the results of machine learning drive critical decisions in applications related to healthcare and biomedicine. Such health-related applications are often sensitive, and thus, any security breach would be catastrophic. Naturally, the integrity of the results computed by machine learning is of great importance. Recent research has shown that some machine-learning algorithms can be compromised by augmenting their training datasets with malicious data, leading to a new class of attacks called poisoning attacks. Hindrance of a diagnosis may have life-threatening consequences and could cause distrust. On the other hand, not only may a false diagnosis prompt users to distrust the machine-learning algorithm and even abandon the entire system but also such a false positive classification may cause patient distress. In this paper, we present a systematic, algorithm-independent approach for mounting poisoning attacks across a wide range of machine-learning algorithms and healthcare datasets. The proposed attack procedure generates input data, which, when added to the training set, can either cause the results of machine learning to have targeted errors (e.g., increase the likelihood of classification into a specific class), or simply introduce arbitrary errors (incorrect classification). These attacks may be applied to both fixed and evolving datasets. They can be applied even when only statistics of the training dataset are available or, in some cases, even without access to the training dataset, although at a lower efficacy. We establish the effectiveness of the proposed attacks using a suite of six machine-learning algorithms and five healthcare datasets. Finally, we present countermeasures against the proposed generic attacks that are based on tracking and detecting deviations in various accuracy metrics, and benchmark their effectiveness.**

***Index Terms*—Healthcare, machine learning, poisoning attacks, security.**

I. Introduction

MACHINE learning is ubiquitously used to extract information patterns from datasets in a wide range of applications. Increasingly, machine-learning algorithms are being used in critical applications where they drive decisions with large personal, organizational, or societal impact. These applications include healthcare [1], network intrusion detection systems[2], spam and fraud detection, phishing detection [3], political decision making [4], adversarial advertisement detection [5], and financial engineering [6].

Among the aforementioned applications, the sensitivity of those related to healthcare calls for efficient and reliable protection against potential malicious attacks. It is important to investigate whether machine-learning algorithms used for healthcare applications are vulnerable to security and privacy threats. Many applications, such as medical machine learning, often require analysis to be performed on datasets without compromising the privacy of people or entities who provided the data. Thus, privacy-preserving machine learning and data mining have been the subject of considerable research [7]–[9]. The robustness of machine-learning algorithms to noise in the training data has also been investigated [10], [11] to evaluate its effects on the decision-making process.

More recent efforts have considered the possibility that vulnerabilities in machine-learning algorithms may be exploited by attackers to influence the algorithm's results [12]–[16]. It is now well known that classification algorithms need to take into account these adversarial intent, i.e., adversarial classification and, in general, machine learning, to preserve their effectiveness [17]–[20]. These include analyzing the vulnerabilities of algorithms and developing design approaches for their security in adversarial environments.

Two main categories of security attacks on machine learning have been considered in the literature: exploratory and causative [16], [18], [21]. Exploratory attacks exploit existing vulnerabilities without altering the training process. On the other hand, causative attacks alter the training process, typically by modifying the training dataset. Poisoning attacks [22] are a class of causative attacks in which carefully-crafted malicious instances are added to the training dataset, leaving the rest of the dataset intact.

In healthcare applications, poisoning attacks are highly relevant because although manipulation of existing data in the training dataset may be difficult or impossible for attackers, addition of new data might be relatively easy. For instance, hinderance of a hypothyroid diagnosis may have life-threatening consequences due to delayed treatment. This may reduce trust in the machine-learning algorithm. On the other hand, a false positive classification may cause unnecessary concern. If poisoning attacks are detected, the user or owner of the dataset may take appropriate action, such as disregarding the results

Manuscript received March 11, 2014; revised June 14, 2014; accepted July 26, 2014. Date of publication July 30, 2015; date of current version November 3, 2015. This work was supported in part by the National Science Foundation under Grant CNS-1219570. Recommended for publication by Associate Editor D. A Clifton.

M. Mozaffari-Kermani is with the Department of Electrical and Microelectronic Engineering, Rochester Institute of Technology, Rochester, NY 14623 USA (e-mail: m.mozaffari@rit.edu).

S. Sur-Kolay is with the Advanced Computing and Microelectronics Unit, Indian Statistical Institute, Kolkata 700108, India (e-mail: ssk@isical.ac.in).

A. Raghunathan is with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: raghunathan@purdue.edu).

N. K. Jha is with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: jha@princeton.edu).

Digital Object Identifier 10.1109/JBHI.2014.2344095

of machine learning or attempting to cleanse the dataset of the malicious data. From an attacker's perspective, it is therefore desirable to mount poisoning attacks such that they are difficult to detect. On the other hand, if such attacks are successful (which is definitely a possibility, given the need to access healthcare data anywhere/anytime and especially through cloud-based computing), the resulting erroneous conclusions may lead to serious adverse impact on people, institutions, and healthcare services.

In this paper, we present a systematic machine-learning algorithm-independent attack mechanism in the context of healthcare. To evaluate the proposed scheme, we have experimented with six different machine-learning algorithms. Moreover, we show that the proposed attack is successful even without prior knowledge of the machine-learning algorithm details (though with slight degradation in its effectiveness), e.g., the type of the discriminant function of the classifier or its parameters, such as feature weights in the case of linear algorithms. Furthermore, we elaborate upon the effectiveness of the proposed attack in the context of real patterns [16]. Although the proposed attack is of the poisoning type and acts based on the addition of malicious instances to the training patterns, it does not necessarily require knowledge of the exact attribute values in the training dataset; knowledge of their statistics is sufficient for mounting the attack. Moreover, we show that one may obtain a surrogate dataset to mount the attacks, eliminating the need for access to the training dataset.

A. Our Contributions

In this paper, we focus on poisoning attacks on machine-learning algorithms and defenses, also referred to henceforth as countermeasures, for algorithms for healthcare applications, and make the following contributions.

1) We propose a systematic approach for mounting poisoning attacks on machine learning, which is independent of the underlying machine-learning algorithm. We target both fixed datasets (in which users do not add authentic data during the attack) and evolving datasets (in which users can add authentic data). We evaluate the scheme for six machine-learning algorithms and five different healthcare datasets from various contexts. We establish that poisoning attacks can be successfully mounted even if the attacker does not know the type of algorithm used or the training dataset. We also elaborate upon the extension of our attack from datasets to real patterns.
2) Finally, we present countermeasures against the presented attacks and benchmark their effectiveness in the context of the considered machine-learning algorithms and datasets. These countermeasures are based on monitoring deviations in accuracy metrics of the training dataset and the number of instances added to it. The effectiveness of these countermeasures suggests that there is a need for devising attacks that are capable of circumventing them.

The remainder of this paper is organized as follows. In Section II, we present the relevant previous work. In Section III, we discuss preliminary concepts. In Section IV, we present and evaluate our systematic approach for mounting attacks. We describe countermeasures against these attacks in this section as well. Finally, we conclude in Section V.

II. Previous Work

Since we target causative (specifically, poisoning) attacks and countermeasures against them in this paper, we present the relevant previous work next.

Taxonomies for attacks against machine-learning systems and countermeasures against them are presented in [18] and [21]. Many of these attacks, along with countermeasures against them, were demonstrated against SpamBayes, a statistical spam filter. Experiments have been done for causative availability attacks which causes the filter to mislabel all legitimate e-mails as spam. Some countermeasures are based on methods for eliminating the newly added data instances that have a substantial negative impact on classification accuracy and also devising procedures to limit the impact of adversarial data.

A number of poisoning attacks on specific machine-learning algorithms have been proposed in [22]–[24]. The attacks presented in [23] and [24] work in the feature space. Experiments are done for an intrusion detection scenario (data from a sample of real HTTP traffic from a web server). The attack is successful with the need to only overwrite up to 35% of the initial data points. Moreover, an attacker needs to control 5–15% of traffic to successfully stage a poisoning attack. However, the work proposed in [22] only depends on the gradients of dot products of points in the input space. It investigates poisoning attacks against one particular machine-learning algorithm, namely support vector machines (SVMs), based on increasing the classifier's test error. The attacks are experimentally evaluated using a classical handwritten digit recognition dataset (a two-class subproblem consisting of discrimination between two distinct digits). In the exemplary experimental runs reported in [22], a single attack data point caused the classification error to rise from the initial error rates of 2–5% to 15–20%. Machine-learning methods that account for data manipulation by adversaries (robust classifiers) have also been investigated [25], [26].

Adversarial machine learning has gained attention in the literature [17]–[20] as an emerging field for analyzing the vulnerabilities of machine-learning algorithms in adversarial environments, developing methods for benchmarking the classifier security, and presenting countermeasures to counteract or alleviate these security concerns. A framework for evaluating classifier security in the design phase, in order to provide practical guidelines and tools for pattern recognition system designers, has been discussed in [16].

III. Preliminaries

In this section, we present preliminary concepts related to machine-learning algorithms, datasets, attack models, and notations.

A. Machine-Learning Algorithms and Datasets

In this paper, we experiment with six different machine-learning algorithms. We use these algorithms to perform various classification tasks, by constructing models based on a training dataset and using the models to classify a test dataset.

The first is a tree-based algorithm, i.e., BFTree (best-first decision tree) [27], [28]. BFTree uses a tree constructed from binary splits on attributes. The "best" node is the node that maximally reduces impurity among all nodes available for splitting. An increase of the training set size results in an increase of tree size and complexity in this algorithm. The second algorithm is Ridor (ripple-down rule learner) [29], [30], which is a rule-based algorithm that consists of a data structure and knowledge acquisition scenarios, where experts' knowledge is stored in the data structure and the knowledge is coded as a set of rules. Specifically, it generates a default rule first and then the exceptions for the default rule with the least (weighted) error rate. Then, it generates the "best" exceptions for each exception and iterates until pure. The exceptions are a set of rules that predict classes other than the default. The third algorithm is NBTree, which is a decision tree with naive Bayes classifiers at the leaves [31]. This algorithm is suitable for the learning scenarios in which many attributes are likely to be relevant for a classification task, yet the attributes are not necessarily conditionally independent given the label. The fourth algorithm is IB1 (nearest-neighbor classifier) [32], which uses normalized Euclidean distance to find the training instance closest to the given test instance. IB1 has been chosen as the simplest form of instance-based learning algorithms, yet, the presented attacks and countermeasures are, generally, suitable for other variants as well. The fifth algorithm is Multilayer Perceptron (MLP), which is based on a feedforward artificial neural network that is trained using backpropagation [33]. MLP became useful with the introduction of the backpropagation training algorithm. Its counterpart, SVM, is known to improve the generalization performance for binary classification tasks, which forms the base of the sixth algorithm: sequential minimal optimization (SMO) [34] for training an SVM [35], [36]. Note that SVM is a more recent algorithm compared to MLP and has been widely applied in biological and other sciences. The specialized SMO algorithm breaks the problem down into 2-D subproblems that may be solved analytically, eliminating the need for a numerical optimization algorithm. A summary on the aforementioned details is presented in Table I.

For our experiments, we have used two tree-based (BFTree and NBTree), one rule-based (Ridor), two functions (SMO and MLP), and one lazy (IB1) classifier. The choice of NBTree as the second tree-based classifier is based on the fact that it is the only tree-based classifier that performs naive Bayes classification at the leaves.

We evaluate the aforementioned algorithms on five different medical datasets from UCI's machine-learning repository. We use the Thyroid Disease dataset [37], Breast Cancer dataset [38], Acute Inflammations dataset [39], Echocardiogram dataset [40], and Molecular Biology (Splice-junction Gene Sequences) dataset [41]. Table II presents the details of these datasets.

TABLE I
MACHINE-LEARNING ALGORITHMS

Name	Details
BFTree (Best-first decision tree)	Tree-based with binary splits on attributes
Ridor (Ripple-down rule learner)	Rule-based through knowledge acquisition
NBTree (Naive Bayes decision tree)	Decision tree with naive Bayes classifiers
IB1 (Nearest-neighbor classifier)	Normalized Euclidean distance-based
MLP (MultilayerPerceptron)	Feedforward artificial neural network-based
SMO (Sequential minimal optimization)	Support-vector machine-based

TABLE II
DETAILS OF DATASETS WITH NUMBER OF ATTRIBUTES IN EACH TYPE IN PARENTHESES

Name	#Inst., #Attr.	Attr. types
Thyroid Disease	7104, 21	Numeric
Breast Cancer	699, 10	Nominal
Acute Inflammations	120, 6	Numeric (1), Nominal (5)
Echocardiogram	132, 12	Numeric (10), Nominal (2)
Molecular Biology	3190, 61	Nominal

Columns *#Inst.* and *#Attr.* refer to the number of data instances in the dataset and the number of attributes for each instance, respectively. Column *Attr. types* specifies the attribute types, where *Nominal* refers to attributes that can only take a limited number of distinct values, e.g., exon/intron boundaries or intron/exon boundaries in the Molecular Biology (Splice-junction Gene Sequences) dataset, and *Numeric* refers to attributes for which the values are integers or real numbers.

B. Attack Model

In our attack model, we assume that the attackers have knowledge of the training dataset and use this knowledge to construct malicious data. In practice, this knowledge can be obtained either because the dataset is publicly available or because the attackers have employed various means, such as eavesdropping on network traffic or compromising a system where the dataset is stored, in case security measures, such as the ones presented in [42], are compromised. However, the success of the proposed attacks is only dependent on the knowledge of the statistics of the training dataset, as discussed in Section IV.

In scenarios where gaining access to the training datasets is difficult, we present an alternative approach in which attackers construct a proxy training dataset drawn from the same distribution as the original dataset [13]. This is possible since our proposed attacks are based on the statistics of the training dataset (and not the exact values of attributes within the dataset). By presenting artificial test instances as inputs to the targeted machine-learning application and observing its responses, one

TABLE III
NOTATIONS

Notation	Definition
N	number of instances
M	number of attributes
$\chi_i, 1 \leq i \leq N$	ith instance
$\chi_i^j, 1 \leq i \leq N, 1 \leq j \leq M$	jth attribute value of ith instance
$\chi^j, 1 \leq j \leq M$	jth attribute
$\Upsilon_i, 1 \leq i \leq N$	ith class label

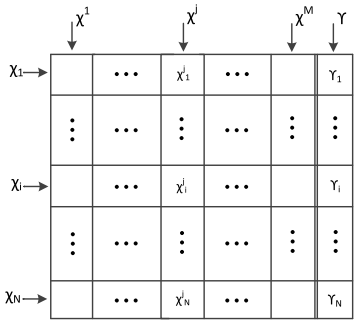

Fig. 1. Instances, attributes, and class labels.

can construct a "proxy" dataset that can be used to mount the attack.

In Section IV, we show that such an attack would be successful with only a minor degradation in the success rate compared to the case where the training dataset is directly accessible. Moreover, in many cases, launching poisoning attacks may be much easier than launching general causative attacks in which modifications to current instances are required.

We also assume that attackers have access to significant computing resources. For example, they can repeatedly modify the training dataset and evaluate the effectiveness of the modifications by constructing models and testing them on a validation dataset. The attacker can construct a golden validation dataset from the original training dataset to which access is assumed, e.g., a subset of the original training dataset before launching the attacks.

Unlike general causative attacks in which attackers are assumed to be capable of arbitrarily manipulating datasets by adding, changing, or removing data, our attack model considers poisoning attacks in which attackers can only *add* malicious data. Finally, in our attack model, we assume that attackers can add new malicious training data in a manner that generally may not raise suspicion (rather than arbitrarily generated instances that can be flagged by simple tests such as range checks on the attribute values).

C. Notations

The notations used henceforth are summarized in Table III. These include notations for data instances, class labels, and attributes. Fig. 1 depicts a training dataset that is composed of instances, which, in turn, consist of attributes and class labels.

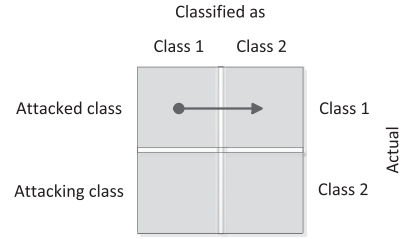

Fig. 2. Typical attacks on two-way classification problems.

IV. SYSTEMATIC MACHINE-LEARNING ATTACKS AND COUNTERMEASURES

In this section, we present and evaluate our proposed schemes for attacking machine-learning algorithms applied to medical datasets. Then, we discuss and evaluate countermeasures against the attacks.

A. Attack Objectives

In healthcare, attackers may have varying motivations for poisoning training datasets, ranging from generally degrading the accuracy of the algorithm to biasing the results in a specific, targeted manner.

As an example of targeted attacks, let us consider the Thyroid Disease dataset, in which data instances are associated with two classes: normal and hypothyroid. Targeted attacks might compromise the effectiveness of the machine-learning algorithm either to prevent a hypothyroid diagnosis or to falsely lead to a hypothyroid diagnosis. Let us first consider the former case. The prevention of such a diagnosis, when maliciously done, would, at least, cause distrust and annoyance for the patients and clinicians once diagnosis is correctly performed. Moreover, it could lead to (catastrophic) health issues if the diagnosis is mainly based on the results of the algorithm. As for the latter case, false alarms would induce distrust and force users to abandon the system. In short, these cases could either lead to an adverse health impact or a loss of trust in the system.

Let us consider the scenario in which a malicious attacker intends to prevent a hypothyroid diagnosis. In this scenario, we denote the hypothyroid class as the *attacked class* and the benign class as the *attacking class*. As shown in Fig. 2, the attacker adds malicious instances to the training dataset such that instances belonging to the attacked class (Class 1) are predicted and classified as belonging to the attacking class (Class 2). If the attacker wishes to cause false hypothyroid diagnoses, the attacking and attacked classes are switched.

The attack schemes that we propose can be used for targeted or nontargeted attacks. However, we describe the procedure in the context of targeted attacks, since attacks that simply aim to increase classification error can be viewed as a special case of targeted attacks where all classes are attacked classes.

Next, we discuss the proposed attack schemes that systematically generate inputs that are highly effective in poisoning a given medical training dataset.

B. Attack Scheme

Unlike prior work that shows how to attack specific machine-learning algorithms, our objective is to propose a generic and algorithm-independent attack scheme. In other words, the proposed attacks can be applied to a wide range of machine-learning algorithms and medical datasets. In fact, the attacker does not even need to know the type of machine-learning algorithm used to apply the proposed attack scheme. Furthermore, highly algorithm-specific attacks may be thwarted by simply changing the machine-learning algorithm used. However, knowledge of the machine-learning algorithm being used increases the efficacy of the attacks, as discussed later. In addition to our original attack scheme, we consider and benchmark through experiments four variants of the proposed scheme, i.e., attacking without access to the training dataset, attacking unknown machine-learning algorithms, adapting the attacks to real patterns, and attacking n-way classification.

The proposed attack scheme is described in Algorithm 1. Let the original dataset be denoted as $D \in (\chi, \Upsilon)$ with N instances, where χ and Υ represent an instance's attributes and class label, respectively. Algorithm 1 adds N' malicious instances to the original dataset to create a manipulated dataset $D' \in (\chi, \Upsilon)$ with $N + N'$ instances. To add a malicious instance, I pseudorandom candidates are generated (using Algorithm 2). Intuitively, Algorithm 2 generates candidates whose attribute values match the statistics of the attacked class, but whose labels are set to the attacking class (recall that the objective is to bias the model away from the attacked class and toward the attacking class). For each candidate, the algorithm builds a model on the same machine-learning algorithm and evaluates its classification accuracy on the validation set. The candidate that results in the highest degradation in classification accuracy is selected and added to the dataset. I is a constant set by the attacker and trades off efficacy of the malicious instances *versus* the computational effort expended to generate them.

Algorithm 2 is called from Algorithm 1 to generate malicious instance candidates. Its inputs are χ^j, $1 \leq j \leq M$, which represents the jth attribute set (see Fig. 1), and Υ_i, $1 \leq i \leq N$, which represents the ith class. Its output is the malicious instance candidate consisting of attribute values $(\alpha_j, 1 \leq j \leq M)$ and a class label.

For each attribute χ^j, the algorithm analyzes the training dataset to compute statistics that relate the possible values of χ^j to the class labels. We divide the range of χ^j into g bins (where g is a specified constant; the attacks can be tailored by using more bins based on the resources available to the attacker and the attack objectives). For each bin, we identify instances from the training dataset whose jth attributes assume values that lie in the bin. In Algorithm 2, $\chi^j(k)$, $1 \leq k \leq g$, are the g subsets of χ^j. We then compute the distribution of the instances in each bin across the attacked and attacking classes. For an n-way classification, let us denote $\eta_{k,j}$ and $\eta'_{k,j}$ as the number of entries in $\chi^j(k)$ corresponding to the attacked and attacking class, respectively. These statistics ($\eta_{k,j}$ and $\eta'_{k,j}$) are used for computing probabilities P_k, which are, in turn, used to generate a weighted-pseudorandom value for α_j through weighted pseudorandom functions. To choose each specific attribute value α, the weighted function S within Weka 3 machine-learning workbench [43] uses the attribute probabilities $P_k = \frac{W_k}{\sum_{1 \leq i \leq g} W_i}$, $k = 1, \ldots, g$. This function pseudorandomly chooses α, biased through attribute probabilities thus obtained. The label of the malicious instance candidate is set to the attacking class.

Algorithm 1 Algorithm-independent attacks.

Input: Dataset $D \in (\chi, \Upsilon)$ with N instances, validation dataset V, number of iterations I.
Output: Maliciously manipulated dataset $D' \in (\chi, \Upsilon)$ with $N + N'$ instances, where N' is the number of added malicious instances.
Begin
  Assign $D' \leftarrow D$
  for $k = 1$ to N' do
    //Select kth malicious instance
    for $i = 1$ to I do
      Use Algorithm 2 to generate malicious instance candidate i
      Add the candidate to D' to create a temporary training set $D_T \in (\chi, \Upsilon)$ with $N + k$ instances
      Build the model using D_T and record its classification accuracy on the validation set V as A_i
    endfor
    Select instance $\hat{i}$ such that $A_{\hat{i}} = \min(A_i), 1 \leq i \leq I$
    Add instance $\hat{i}$ to D'
  endfor
End
Return: $D' \in (\chi, \Upsilon)$.

Algorithm 2 Deriving a malicious instance candidate.

Input: χ^j, $1 \leq j \leq M$ and Υ_i, $1 \leq i \leq N$, g bins (a specified constant).
Output: $\chi_{N+1}, \Upsilon_{N+1}$ (malicious instance candidate).
Denote: $\eta_{k,j}$ and $\eta'_{k,j}$ as the number of entries in $\chi^j(k)$ corresponding to the attacked class and the attacking class, respectively.
Begin
  for $j = 1$ to M do
    for $k = 1$ to g do
      Calculate $\eta_{k,j}$ and $\eta'_{k,j}$
      Assign $W_k \leftarrow \frac{\eta_{k,j}}{\eta'_{k,j}}$
    endfor
    Compute attribute probabilities ($P_k = \frac{W_k}{\sum_{1 \leq i \leq g} W_i}$), $k = 1$ to g
    Weighted function S selects attribute value α_j pseudorandomly based on attribute probabilities
  endfor
End
Return: Malicious instance candidate is $\chi_{N+1} = \{\alpha_j, 1 \leq j \leq M\}$, $\Upsilon_{N+1} =$ Attacking class.

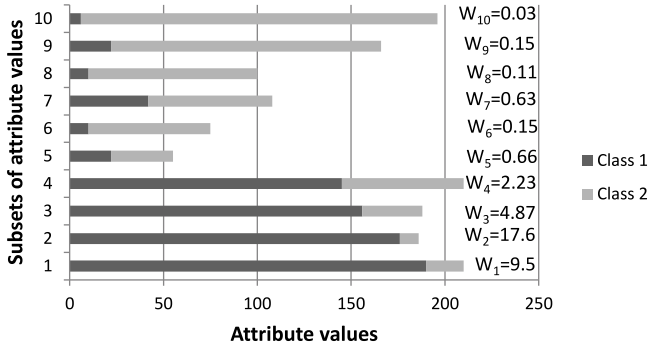

Fig. 3. Malicious attribute value derivation example ($g = 10$, $\Upsilon \in \{$Class 1, Class 2$\}$).

TABLE IV
ATTACKED AND ATTACKING CLASSES OF DATASETS FOR THE EXPERIMENTS

Dataset	Attacked class	Attacking class
Thyroid Disease	Normal	Hypothyroid
Breast Cancer	Benign	Malignant
Acute Inflammations	Negative	Positive
Echocardiogram	Dead (after 1 year)	Alive
Molecular Biology	intron/exon	exon/intron

Let us consider an illustrative example of a two-way classification problem ($\Upsilon \in \{$Class 1, Class 2$\}$) where each data instance has one attribute. We assume $g = 10$, i.e., we divide the range of the attribute into ten bins and assign the instances in the training dataset to these bins. Fig. 3 shows the distribution of instances in each bin (each bar represents a bin) across the two classes (the subbars represent the two classes). Classes 1 and 2 are the attacked and attacking classes, respectively. Using Algorithm 2, weights $W_k, 1 \leq k \leq 10$, are computed by looking at the ratio of the number of instances in bin k that belong to Class 1 to the number of instances in bin k that belong to Class 2. We show four values for W_k, $k \in \{1, 2, 9, 10\}$, in the figure. The malicious instance candidate is created by generating a weighted pseudorandom value for the attribute. $\frac{W_k}{\sum_{1 \leq j \leq 10} W_j}$ is used as the probability that the malicious instance candidate will have an attribute value in bin k. The label of the malicious instance candidate is set to Class 2.

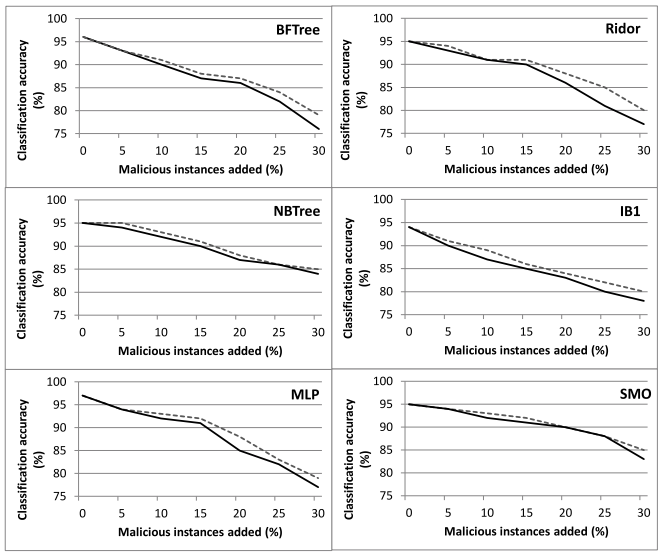

Fig. 4. Results of attacks on the Thyroid Disease dataset for the fixed (solid line) and evolving (dashed line) cases.

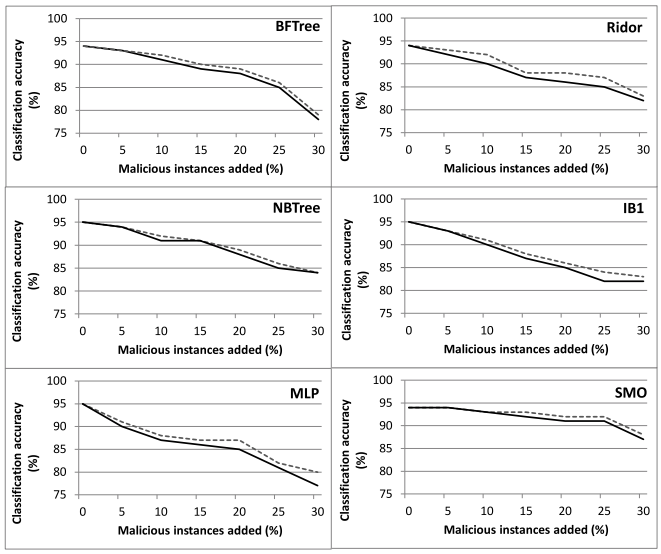

Fig. 5. Results of attacks on the Breast Cancer dataset for the fixed (solid line) and evolving (dashed line) cases.

C. Experimental Evaluation

We next present the results of applying the proposed attack procedure to the six machine-learning algorithms and five medical datasets discussed in Section III. We implemented the proposed attack scheme using the Weka 3 machine-learning workbench [43]. The training and validation datasets are extracted by splitting the original dataset.

Table IV shows the attacked and attacking classes for the chosen datasets. For the experiments reported in this paper, we simply chose the class with more data instances as the attacked class. However, note that this is not a limitation; we verified that the attacks are successful regardless of how the attacking and attacked classes are chosen. In Table IV, the datasets are based on two-way classifications. However, our attacks are applicable to n-way classification problems as well.

The results of our attacks for the five datasets are shown in Figs. 4–8. In each of these figures, the classification accuracy degradations corresponding to the classification results for the attacked class, after adding malicious data to the training dataset, are shown for the six machine-learning algorithms. The feasibility of adding malicious instances to the datasets depends on the specific applications and contexts. Moreover, the presented results for the attacks (in which up to 30% malicious instances are added) are only meant to be illustrative and provide general guidelines. In other words, as seen from the tables throughout the paper, attacks with lower number of malicious instances

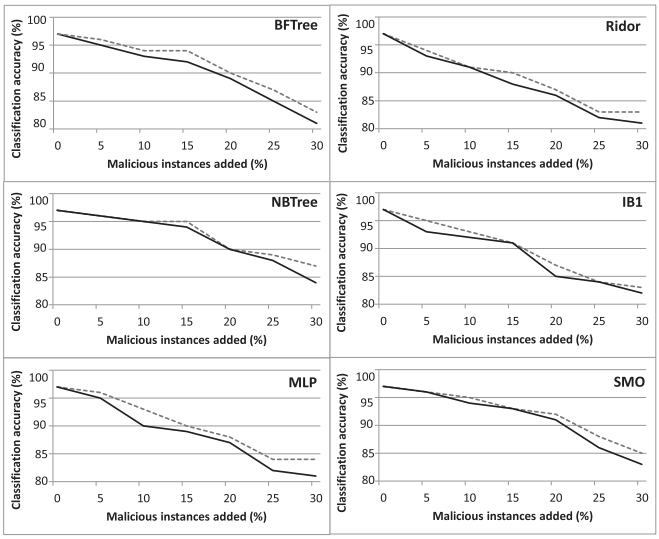

Fig. 6. Results of attacks on the Acute Inflammations dataset for the fixed (solid line) and evolving (dashed line) cases.

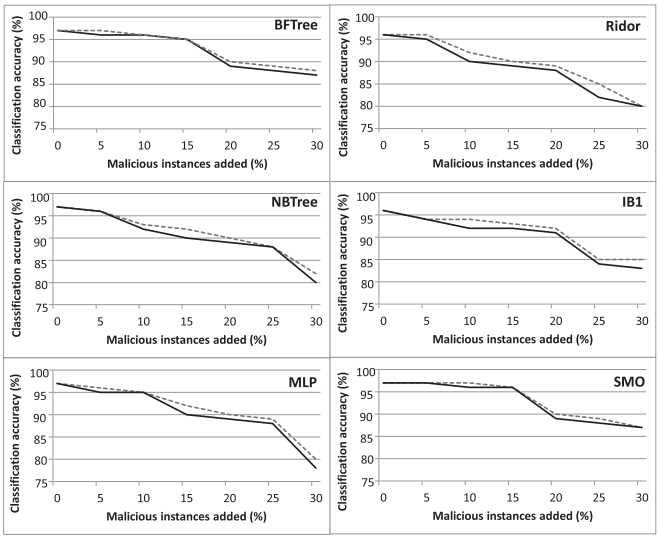

Fig. 7. Results of attacks on the Echocardiogram dataset for the fixed (solid line) and evolving (dashed line) cases.

could be considered "successful," depending on the goal of the attacker. Based on the attack objectives, restrictions on how many malicious entries can be added, and the overhead that can be tolerated, the attacker can choose an appropriate number of added malicious instances to mount the attacks.

The two curves in each graph represent two different assumptions about the nature of the training dataset. In the first one (solid lines in the graphs), the training dataset is considered to be fixed for the duration of the attack. In the second one (dashed lines in the graphs), the dataset is considered to be evolving (i.e., unknown to the attacker, other users can add authentic data to the training set during the attack). In the latter case, we assume that the number of added authentic instances is the same as the number of added malicious instances. Note

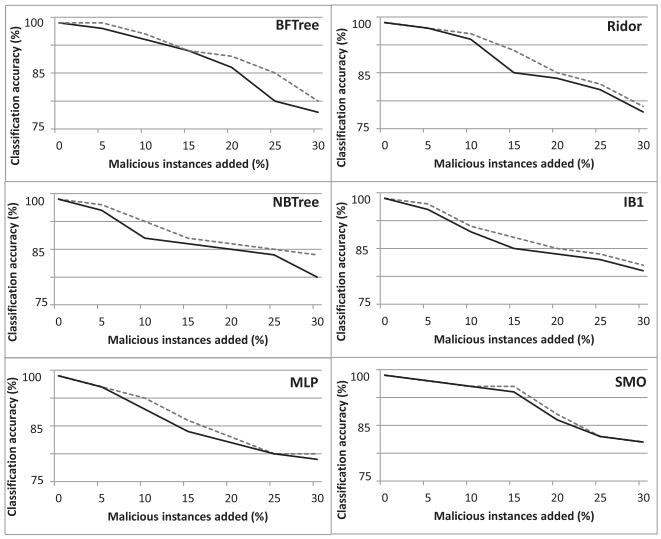

Fig. 8. Results of attacks on the Molecular Biology dataset for the fixed (solid line) and evolving (dashed line) cases.

that the evolving dataset assumption makes the problem tougher for the attacker, since the statistics of the newly added authentic instances are unknown to him, and the attack has to be based on the original training dataset. The figures show that the attacks are quite successful across all datasets and algorithms. As the number of added malicious instances increases, the misclassification increases, as indicated by the classification accuracy percentage decreasing (note that the classification accuracies for the datasets and machine-learning algorithms were originally very high). The attack remains effective even for evolving datasets, causing the misclassification to be only slightly lower than in the case of fixed datasets.

For the results described previously, we set the number of iterations I to 50 in Algorithm 1. To check that this was adequate, we also used $I = 100$ in a few cases (thus utilizing twice the CPU time). However, the misclassifications (which reflect the efficacy of the attacks) increased by less than 1% (compared to the 50 iterations case), when 30% malicious data were added to the training dataset. Since this is not that significant, we conclude that $I = 50$ appears to be sufficient for achieving the attack objectives.

We would like to emphasize that the proposed approach is successful without raising suspicion when the attacks are mounted. One may simply suggest flipping the class labels of the instances in the training dataset to create malicious instances. However, not only is such a scheme not efficient when the number of instances in the training dataset is high, but it might also raise suspicion. The precision of the proposed scheme (by changing the number of iterations at the cost of more CPU time) gives the attacker flexibility to mount attacks based on the resources available. Moreover, the analysis of statistics in the proposed approach before devising the attacks makes it applicable to large datasets. Indeed, when we deal with evolving datasets, the problem gets tougher since the statistics of the newly added

TABLE V
COMPARISON OF THE EFFECTIVENESS OF OUR ATTACKS ON THE MACHINE-LEARNING ALGORITHMS CONSIDERED

Attack	Thyroid Disease (most to least vulnerable)						Breast Cancer (most to least vulnerable)					
15% added	**IB1** (9%)	BFTree (7%)	Ridor (5%)	NBTree (4%)	MLP (3%)	**SMO** (3%)	**MLP** (14%)	IB1 (11%)	BFTree (8%)	Ridor (4%)	NBTree (3%)	**SMO** (3%)
30% added	**MLP** (20%)	Ridor (18%)	BFTree (18%)	IB1 (16%)	NBTree (13%)	**SMO** (12%)	**MLP** (26%)	BFTree (23%)	Ridor (22%)	IB1 (18%)	NBTree (16%)	**SMO** (16%)
	Acute Inflammations (most to least vulnerable)						Echocardiogram (most to least vulnerable)					
15% added	**Ridor** (9%)	BFTree (9%)	IB1 (8%)	NBTree (8%)	MLP (6%)	**SMO** (6%)	**Ridor** (8%)	NBTree (8%)	IB1 (7%)	MLP (6%)	BFTree (3%)	**SMO** (3%)
30% added	**Ridor** (21%)	BFTree (18%)	IB1 (18%)	MLP (14%)	NBTree (12%)	**SMO** (12%)	**NBTree** (20%)	IB1 (18%)	MLP (16%)	Ridor (16%)	BFTree (11%)	**SMO** (11%)
	Molecular Biology (most to least vulnerable)											
15% added	**IB1** (9%)	BFTree (9%)	Ridor (7%)	NBTree (6%)	MLP (6%)	**SMO** (5%)						
30% added	**BFTree** (18%)	IB1 (17%)	MLP (15%)	NBTree (15%)	Ridor (12%)	**SMO** (12%)						

Note: Changes in the misclassification percentage compared to the original dataset, i.e., the effectiveness of attacks, are shown in parentheses.

authentic instances are unknown. However, the presented attack mechanism is still successful in such cases.

Table V presents a comparison of the six machine-learning algorithms on the five datasets for the case of evolving datasets, to gauge the vulnerability of these algorithms to poisoning attacks. It looks at the vulnerability half-way during the attack (when half of the malicious instances are added, i.e., 15% of the original dataset), and at the end (when all malicious instances are added, i.e., 30% of the original dataset). The entries in this table indicate the reductions in the attacked label percentage that are achieved. The entries are sorted from the most vulnerable algorithm (left) to the least vulnerable algorithm (right). The results indicate that, at the end of the attacks, SMO is found to be the most robust.

In order to further evaluate the efficacy of the proposed attack strategies, we performed experiments on the Breast Cancer dataset for the six machine-learning algorithms through addition of pseudorandom malicious instances. These instances are generated by pseudorandomly choosing the attribute values without using Algorithm 2 for deriving the malicious instance candidates, i.e., equal weights are given to the subsets with weights $W \geq 1$ within the bins and the other bins are ignored. The experimental results show that after adding 30% malicious instances, a significant reduction in the success of attacks is seen compared to the proposed attack approach. Specifically, for MLP, BFTree, Ridor, IB1, NBTree, and SMO, the effectiveness for the proposed (pseudorandom) attack is, respectively: 26% (16%), 23% (15%), 22% (13%), 18% (12%), 16% (9%), and 16% (10%). We note that this type of attack in not completely random as the cases, for which $W < 1$, are ignored. The difference in the effectiveness of the proposed attack and a random attack would be even more than the above difference with respect to a pseudorandom attack.

IB1 machine learning uses normalized Euclidean distance to find the training instance closest to the given test instance, and predicts the same class as this training instance. If multiple instances have the same (smallest) distance to the test instance, the first one found is used. We also performed experiments using IBk (k-nearest neighbor classifier in which parameter k can be optimized to improve classifier performance) for the Breast Cancer dataset for $k = 2$, for which slightly better performance is observed. The results show that the proposed attacks are successful with a success rate close to that of IB1 (around 18% change in the attacked label percentage). We would like to emphasize that although the algorithm performance is an important factor, benchmarking the success of attacks (countermeasures) with respect to the original performance of the machine-learning algorithms is our primary focus.

We also experimented with a regression-based logistic classifier, SimpleLogistic [44], for building linear logistic regression models for the five datasets. After mounting the attacks (adding 30% malicious instances), the decrease in the attacked label percentages for Thyroid Disease, Breast Cancer, Acute Inflammations, Echocardiogram, and Molecular Biology datasets are 17%, 20%, 18%, 17%, and 16%, respectively. These success rates put the SimpleLogistic classifier roughly in the middle of the other classifiers in terms of vulnerability to attacks.

In some applications, when there is incomplete or incoherent knowledge, e.g., noisy data, classification based on similarity of objects (termed similarity-based classification) may be used [45]. In such approaches, a similarity criterion is used to quantify the similarities among objects by deriving a numerical measure that is eventually utilized for inference. Accordingly, for n-way classification, similarity-based classification returns n classes with which n similarity degrees are associated.

The proposed algorithm-independent attacks can be adapted to such classification approaches as well. Here, the aforementioned similarity criterion needs to be converted to a malicious one so that the incoming new objects have higher similarity degrees for a particular attacking class. In other words, the attack will be directed toward decreasing the similarity degree(s) of the attacked class(es) and increasing that of the attacking class. Depending on the method used for deriving the similarity criterion, Algorithms 1 and 2 can be used to mount the attacks. For instance, if any statistical measure (for the objects getting compared) is used as the similarity criterion, one may slightly modify Algorithm 1 to benchmark this measure (as the effectiveness factor) to cause a deviation in the similarity degree of the attacked class. We would like to emphasize that instead of attacking the training datasets or their surrogates, the similarity

criterion is maliciously targeted to reach the attacker's objectives in this case.

1) Attacking Without Access to the Training Dataset: Our experiments so far demonstrate that the proposed attack scheme can be applied to machine-learning algorithms whose training datasets are known to the attackers. We now consider the case in which prior knowledge of the statistics of the training datasets is not assumed for the attackers. The attack in this case is based on getting feedback from the machine-learning algorithm to construct an "artificial" dataset.

For performing experiments in this case, an "artificial" dataset is created by giving test inputs to the machine-learning algorithm. We note that based on the classification accuracy of the machine-learning algorithm used, this newly created dataset may contain a few incorrect class labels. This might cause minor deviations (based on the misclassification ratio) in the choice of the malicious instance candidate (see Algorithm 2). Our experiments indicate that after adding malicious data to this "artificial" dataset, the attack is still quite successful, with the results being very close to the results obtained through the original approach. Specifically, for the machine-learning algorithms BFTree (one of the most vulnerable ones in Table V) and SMO (the least vulnerable one in Table V), only 1–2% and 2–3% change are observed in the attacking class misclassification, respectively, for the five datasets.

We would like to emphasize that this approach is less effective when the classification accuracy of the machine-learning algorithms is less; nonetheless, due to inaccurate results, typically, these types of algorithms are not used for sensitive applications. The presented approach can be applied to other datasets and machine-learning algorithms as well. Moreover, the attacker does not necessarily need to know the type of the machine-learning algorithm used.

2) Attacking Unknown Machine-Learning Algorithms: Our experiments suggest that the proposed attack scheme can be applied to a wide range of machine-learning algorithms. However, in our discussions so far, the machine-learning algorithm used was assumed to be known to the attacker (recall that Algorithm 1 performs training and validation to evaluate the efficacy of the malicious instances). Algorithm 1 can be modified to the scenario where the attacker does not know the machine-learning algorithm used. We eliminate the inner loop in Algorithm 1, which generates I malicious instance candidates and evaluates them through training and validation to pick the most suitable one. Instead, we simply pick each malicious instance candidate generated by Algorithm 2 and add it to the dataset. The results of our experiments show that the attacks still succeed, although they are not as effective as when the attacker knows the algorithm being used. For instance, for the Breast Cancer dataset and for different machine-learning algorithms, our experiments show that the effectiveness of the attacks is 8–10% higher when the machine-learning algorithm is known.

In summary, knowing the machine-learning algorithm used and performing retraining and validation to select each malicious instance does result in higher attack efficacy. However, hiding this information does not eliminate the ability of the attacker to mount poisoning attacks.

3) Adapting the Attacks to Real Patterns: The proposed attacks are effective on datasets whose features (attributes) are derived through a feature extraction process (which is a reduction step) from real patterns. However, applying these attacks, in practice, would require that inputs be generated as real patterns, and not simply in the feature space. We next discuss issues involved in adapting the proposed attacks to real patterns.

It is possible to create real attack patterns from the malicious datasets derived through our proposed attacks. We note that knowledge of the feature extraction scheme is required for this backward projection. For example, consider the Breast Cancer dataset whose attributes are the tissue physical/chemical characteristics, e.g., clump thickness or uniformity of cell size, in the patterns from the reports of clinical cases. Once the proposed attacks result in malicious training datasets that are represented in the attribute space, one could generate real patterns, i.e., synthetic clinical reports that exhibit the maliciously chosen tissue characteristics. However, these patterns are not necessarily unique, i.e., there might be more than one malicious pattern corresponding to a malicious attribute vector. Moreover, one can attack patterns directly by maliciously targeting parts of the clinical reports corresponding to the target attributes.

Let us consider and elaborate upon two scenarios for the aforementioned dataset, i.e., Breast Cancer. In the first scenario, assuming that the feature extraction process is known (tissue physical/chemical characteristics), removing or adding values for tissue characteristics in order to create malicious patterns is equivalent to changing the feature vectors. In fact, one can construct attack patterns exhibiting the same feature values of a malicious instance used in the attacks; thus, simulating the attacks can be performed by directly modifying the feature vectors instead of the patterns [16]. In the second scenario, consider dealing with more than one feature extraction method for the patterns used for different applications to create different feature sets. In this case, the attacker needs to target the union of features of these methods for the attacks, e.g., a number of tissue characteristics sets for different methods. This, in turn, would increase the number of changes needed by the attackers in the real patterns to cover the feature sets (typically, the intersection of these feature sets is large). In summary, we believe that the feature vectors generated by the proposed attacks are typically extensible to real patterns; nevertheless, solving the general problem of generating real patterns is outside the scope of our study and is an interesting future research topic.

4) Attacking n-Way Classification: So far, we have presented results for two-way classifications. However, our attack scheme can be generalized to n-way classification in a straightforward manner. For example, let us consider a four-way classification problem based on the Hypothyroid dataset from UCI [46] in which class labels are negative, compensated-hypothyroid, primary-hypothyroid, and secondary-hypothyroid. In this case, an attacker can mount the proposed attack using the structure shown in Fig. 9, in which the objective of the attack is misclassification of the three attacked class labels (Classes 1, 3, and 4) toward the attacking compensated-hypothyroid label (Class 2). Table VI shows the results of the mounted attacks for the NBTree machine-learning algorithm. After adding around 20%

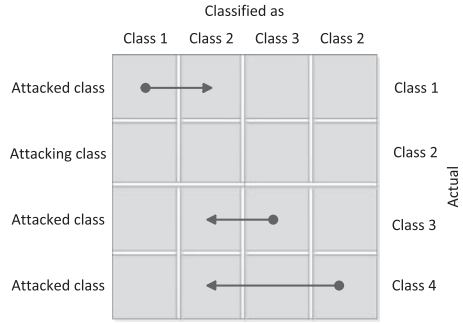

Fig. 9. Attack structure for a four-way classification problem, e.g., on the Hypothyroid dataset.

TABLE VI
ATTACK RESULTS FOR THE HYPOTHYROID DATASET AFTER ADDING 20% MALICIOUS INSTANCES

Attack Class	1	2 (attacking)	3	4
Before	92%	**4%**	2%	2%
After	80%	**18%**	1%	1%

malicious instances to the training dataset based on Algorithms 1 and 2, classification results on the validation set show a significant bias toward the attacking class label (from 4% to 18%).

D. Countermeasures Against Poisoning Attacks

In this section, we present a countermeasure to the proposed poisoning attacks. The proposed method is based on periodically constructing a model using the training dataset, evaluating its accuracy on the validation dataset, and raising an alarm in case of any suspicious change in the accuracy metrics. The effectiveness of the presented countermeasure suggests devising attacks that are resistant to such methods, capable of circumventing these countermeasures. While several metrics are used to evaluate the accuracy of classification, we consider the following two in our study.

1) *Correctly classified instances (CCI):* This statistic indicates the fraction of instances that are classified correctly in an n-way classification (the higher the CCI value, the better the classification).
2) *Kappa statistic [47]:* This statistic measures relative improvement over random predictors, i.e., it is a measure of how good the classifier is compared to a random classifier (the higher the Kappa statistic, the better the classification).

The proposed countermeasure is described in Algorithm 3. For any chosen accuracy metric, the user first computes a "golden" value by building a model when the dataset is in a trusted state and evaluating the model on a fixed validation set (note that the validation set does not evolve as new data are added to the training dataset). The user then sets a threshold value for the accuracy metric that is lower than the golden value. The model is periodically regenerated from the training dataset and evaluated on the validation set. An alarm is raised if the accuracy falls below the threshold value. The choice of the threshold value trades off the sensitivity of the countermeasure to attacks with the likelihood of false alarms due to natural variations in the benign data added to the dataset. In addition to absolute values of accuracy metrics, users can also track the rates of change of these metrics as new instances are added to the training dataset.

Algorithm 3 Presented countermeasure.

1:**Input:** Evolving Dataset $D \in (\chi, \Upsilon)$ with $N + X$ instances, validation dataset V.
2:**Output:** Attack flag AF.
3:**Begin**
4: Set a threshold for the accuracy metrics, e.g., CCI or Kappa statistic
5: Obtain accuracy metrics for V after building the model through D
6: While the accuracy metrics have not reached the set threshold, compute the accuracy metrics
7: If any of the thresholds is reached, set AF=1
8:**End**
9:**Return:** AF.

The changes in accuracy metrics for the attacks on the six machine-learning algorithms and the five evolving datasets presented earlier in this section are given in Table VII. The values tabulated correspond to the half-way and the end of the attacks (i.e., 15% and 30% increase in the dataset, respectively). The lowest and the second lowest changes shown for each metric are depicted by "++" and "+," respectively. Based on these values, we would draw the same conclusion that we drew earlier from Table V, i.e., SMO is the most difficult to attack (note that the degradation in the values of these metrics is the least; among the six algorithms, it maintains the highest accuracy under the attacks).

For setting a threshold value for accuracy metrics, e.g., CCI, a conservative rule of thumb could be to allow a 10% deviation from the original value after mounting the attacks. The reason for such a choice is that in our experiments, classification accuracies were found to be higher than 90% (inaccuracies of less than 10%), and thus, a 10% deviation would be close to twice the inaccuracies in many cases, reducing the chance of encountering false alarms. However, this choice is highly dependent on the context and datasets. Hence, we have not specified a fixed threshold for the countermeasure experiments. Based on the results shown in Table VII, with such a conservative threshold, all the attacks are detected after adding 30% malicious instances. However, except for two cases for the Breast Cancer dataset, none are detected after adding 15% malicious instances. The threshold could be tailored dynamically to trade off countermeasure sensitivity to attacks with the likelihood of false alarms.

Table VII shows changes in the two chosen accuracy metrics, i.e., CCI and Kappa, when the proposed attacks are mounted. One can set thresholds for these accuracy metrics to change the sensitivity of the countermeasures. On the other hand, Table V shows success results for the proposed attack, where success

TABLE VII
CHANGE IN ACCURACY METRICS UNDER POISONING ATTACKS

Attack		Thyroid Disease							Breast Cancer					
	Metric	SMO	NBTree	BFTree	MLP	Ridor	IB1	*Metric*	SMO	NBTree	BFTree	MLP	Ridor	IB1
15% added	CCI	3%++	4%+	7%	3%++	5%	9%	CCI	3%++	3%++	8%+	14%	4%+	11%
	Kappa	8%++	10%+	18%	8%++	13%	24%	Kappa	8%++	8%++	20%	35%	10%+	26%
30% added	CCI	12%++	13%+	18%	20%	18%	16%	CCI	12%++	13%++	18%	20%	18%	16%
	Kappa	32%++	34%+	47%	52%	47%	42%	Kappa	30%++	32%+	45%	50%	45%	40%
		Acute Inflammations							Echocardiogram					
15% added	CCI	6%++	8%+	9%	6%++	9%	8%+	CCI	3%++	8%	3%++	6%+	8%	7%
	Kappa	15%++	20%+	22%	15%++	22%	20%+	Kappa	8%++	20%	8%++	15%+	20%	17%
30% added	CCI	12%++	12%++	18%	14%+	21%	18%	CCI	11%++	20%	11%++	16%+	16%+	18%
	Kappa	30%++	30%++	45%	35%+	52%	45%	Kappa	26%++	50%	26%++	40%+	40%+	45%
		Molecular Biology												
15% added	CCI	5%++	6%+	9%	6%+	7%	9%							
	Kappa	13%++	15%+	22%	15%+	18%	22%							
30% added	CCI	12%++	15%+	18%	15%+	12%++	17%							
	Kappa	30%++	36%+	45%	36%+	30%++	43%							

Note: The lowest and the second to lowest changes in each of the statistics are depicted by "++" and "+," respectively.

is defined as maliciously increasing the classification ratio of the instances in the attacking class to the ones in the attacked class. This is why the CCI accuracy metric in Table VII is higher for the cases in which the attack is less successful in Table V. However, we emphasize that in other attack scenarios, e.g., misclassification of the instances in the two classes without necessarily increasing the classification ratio of the attacking class to the attacked class, results in Table VII can be used to raise an alarm even in the absence of a direct correlation to the classification ratio, which the results in Table V are based on. Thus, the results in these two tables complement each other.

We note that the differences in robustness or vulnerability of the machine-learning algorithms to our proposed attacks are analogous to the dissimilar effects of noise on different machine-learning algorithms [10], [11]. Finally, the presented countermeasures could be used to counteract nonmalicious changes in the classification accuracy as well. Moreover, if such changes are intended, over time, the golden validation dataset needs to be updated to reflect the changes in the training dataset.

V. CONCLUSION

In this paper, we proposed systematic attack schemes for mounting poisoning attacks against machine-learning algorithms used for medical datasets, and suggested countermeasures against them. A key feature of the proposed attack schemes is that they can be applied to a wide range of machine-learning algorithms, even when the machine-learning algorithm is unknown. We evaluated the effectiveness of the attacks against six machine-learning algorithms and five datasets [Thyroid Disease, Breast Cancer, Acute Inflammations, Echocardiogram, and Molecular Biology (Splice-junction Gene Sequences)], and ranked the algorithms based on their ability to withstand the attacks. We then presented countermeasures against these attacks and evaluated their effectiveness. Finally, we identified the machine-learning algorithms that are easiest to defend. We hope that our results will spur further research efforts on understanding and countering poisoning attacks on machine learning.

REFERENCES

[1] M. Brameier and W. Banzhaf, "A comparison of linear genetic programming and neural networks in medical data mining," *IEEE Trans. Evol. Comput.*, vol. 5, no. 1, pp. 17–26, Feb. 2001.

[2] W. Lee, S. J. Stolfo, and K. W. Mok, "Adaptive intrusion detection: A data mining approach," *Artif. Intell. Rev.*, vol. 14, no. 6, pp. 533–567, 2000.

[3] C. Whittaker, B. Ryner, and M. Nazif, "Large-scale automatic classification of phishing pages," in *Proc. Symp. Netw. Distrib. Syst. Security*, 2010, pp. 1–14.

[4] M. Conover, B. Goncalves, J. Ratkiewicz, A. Flammini, and F. Menczer, "Predicting the political alignment of twitter users," in *Proc. IEEE Int. Conf. Privacy, Security, Risk Trust*, Oct. 2011, pp. 192–199.

[5] D. Sculley, M. E. Otey, M. Pohl, B. Spitznagel, J. Hainsworth, and Y. Zhou, "Detecting adversarial advertisements in the wild," in *Proc. ACM Int. Conf. Knowl. Discovery Data Mining*, 2011, pp. 274–282.

[6] E. Kirkos, C. Spathis, and Y. Manolopoulos, "Data mining techniques for the detection of fraudulent financial statements," *Expert Syst. Appl.*, vol. 32, no. 4, pp. 995–1003, May 2007.

[7] R. Agrawal and R. Srikant, "Privacy-preserving data mining," *SIGMOD Rec.*, vol. 29, no. 2, pp. 439–450, May 2000.

[8] Y. Li, M. Chen, Q. Li, and W. Zhang, "Enabling multilevel trust in privacy preserving data mining," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 9, pp. 1598–1612, Sep. 2012.

[9] M. Kantarcioglu and C. Clifton, "Privacy-preserving distributed mining of association rules on horizontally partitioned data," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 9, pp. 1026–1037, Sep. 2004.

[10] N. Cesa-Bianchi, S. Shalev-Shwartz, and O. Shamir, "Online learning of noisy data," *IEEE Trans. Inf. Theory*, vol. 57, no. 12, pp. 7907–7931, Dec. 2011.

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

[12] B. Nelson, B. Biggio, and P. Laskov, "Understanding the risk factors of learning in adversarial environments," in *Proc. ACM Workshop Security Artif. Intell.*, 2011, pp. 87–92.

[13] B. Nelson, M. Barreno, F. J. Chi, A. D. Joseph, B. I. P. Rubinstein, U. Saini, C. Sutton, J. D. Tygar, and K. Xia, "Exploiting machine learning to subvert your spam filter," in *Proc. Usenix Workshop Large-Scale Exploits Emergent Threats*, 2008, pp. 7:1–7:9.

[14] K. M. C. Tan, J. McHugh, and K. S. Killourhy, "Hiding intrusions: From the abnormal to the normal and beyond," in *Proc. Int. Workshop Inf. Hiding*, 2003, pp. 1–17.

[15] B. I. Rubinstein, B. Nelson, L. Huang, A. D. Joseph, S.-H. Lau, S. Rao, N. Taft, and J. D. Tygar, "Stealthy poisoning attacks on PCA-based anomaly detectors," *SIGMETRICS Perform. Eval. Rev.*, vol. 37, no. 2, pp. 73–74, Oct. 2009.

[16] B. Biggio, G. Fumera, and F. Roli, "Security evaluation of pattern classifiers under attack," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 4, pp. 984–996, Apr. 2014.

[17] N. Dalvi, P. Domingos, Mausam, S. Sanghai, and D. Verma, "Adversarial classification," in *Proc. ACM Int. Conf. Knowl. Discovery Data Mining*, 2004, pp. 99–108.
[18] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar, "Can machine learning be secure?" in *Proc. ACM Symp. Inf., Comput. Commun. Security*, 2006, pp. 16–25.
[19] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. D. Tygar, "Adversarial machine learning," in *Proc. ACM Workshop Security Artif. Intell.*, 2011, pp. 43–58.
[20] M. Brückner, C. Kanzow, and T. Scheffer, "Static prediction games for adversarial learning problems," *J. Mach. Learn. Res.*, vol. 13, pp. 2617–2654, 2012.
[21] M. Barreno, B. Nelson, A. D. Joseph, and J. D. Tygar, "The security of machine learning," *Mach. Learn.*, vol. 81, no. 2, pp. 121–148, Nov. 2010.
[22] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," in *Proc. Int. Conf. Mach. Learn.*, 2012, pp. 1807–1814.
[23] B. Rubinstein, B. Nelson, L. Huang, A. Joseph, S.-H. Lau, S. Rao, N. Taft, and J. D. Tygar, "ANTIDOTE: Understanding and defending against poisoning of anomaly detectors," in *Proc. ACM SIGCOMM Conf. Internet Meas.*, 2009, pp. 1–14.
[24] M. Kloft and P. Laskov, "Online anomaly detection under adversarial impact," *J. Mach. Learn. Res.*, vol. 9, pp. 405–412, 2010.
[25] B. Biggio, I. Corona, G. Fumera, G. Giacinto, and F. Roli, "Bagging classifiers for fighting poisoning attacks in adversarial classification tasks," in *Proc. Int. Conf. Multiple Classifier Syst.*, 2011, pp. 350–359.
[26] A. Globerson and S. Roweis, "Nightmare at test time: Robust learning by feature deletion," in *Proc. Int. Conf. Mach. Learn.*, 2006, pp. 353–360.
[27] H. Shi, "Best-first decision tree learning," M.Sc. thesis, Dept. Comput. Sci., Univ. Waikato, Hamilton, New Zealand, 2007.
[28] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting," *Ann. Statist.*, vol. 28, no. 2, pp. 337–407, 2000.
[29] R. Dazeley, P. Warner, S. Johnson, and P. Vamplew, "The Ballarat incremental knowledge engine," in *Proc. Int. Conf. Knowl. Manag. Acquisition Smart Syst. Serv.*, 2010, pp. 195–207.
[30] D. Richards, "Two decades of ripple down rules research," *Knowl. Eng. Rev.*, vol. 24, no. 2, pp. 159–184, 2009.
[31] R. Kohavi, "Scaling up the accuracy of naive-Bayes classifiers: A decision-tree hybrid," in *Proc. Int. Conf. Knowl. Discovery Data Mining*, 1996, pp. 202–207.
[32] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Mach. Learn.*, vol. 6, no. 1, pp. 37–66, Jan. 1991.
[33] R. Collobert and S. Bengio, "Links between perceptrons, MLPs and SVMs," in *Proc. Int. Conf. Mach. Learn.*, 2004, pp. 23–31.
[34] J. C. Platt, *Advances in Kernel Methods*. Cambridge, MA, USA: MIT Press, 1999, pp. 185–208.
[35] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy, "Improvements to Platt's SMO algorithm for SVM classifier design," *Neural Comput.*, vol. 13, no. 3, pp. 637–649, Mar. 2001.
[36] T.-F. Wu, C.-J. Lin, and R. C. Weng, "Probability estimates for multi-class classification by pairwise coupling," *J. Mach. Learn. Res.*, vol. 5, pp. 975–1005, 2004.
[37] Thyroid disease data set. [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Thyroid+Disease, 2014.
[38] Breast cancer data set. [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+ Wisconsin+ (Original), 2014.
[39] Acute inflammations data set. [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Acute+Inflammations, 2014.
[40] Echocardiogram data set. [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Echocardiogram, 2014.
[41] Molecular biology (promoter gene sequences) data set. [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Molecular+Biology+ (Splice-junction+Gene+ Sequences), 2014.
[42] R. Azarderskhsh and A. Reyhani-Masoleh, "Secure clustering and symmetric key establishment in heterogeneous wireless sensor networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2011, pp. 16:1–16:12, Jan. 2011.
[43] Weka 3: Data mining software in Java. [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/index.html, 2014.
[44] N. Landwehr, M. Hall, and E. Frank, "Logistic model trees," *Mach. Learn.*, vol. 59, no. 1, 2, pp. 161–205, 2005.
[45] G. Bisson, "Why and how to define a similarity measure for object based representation systems," in *Proc. Towards Very Large Knowl. Bases*, 1995, pp. 236–246.
[46] Hypothyroid data set. [Online]. Available: http://repository.seasr.org/Datasets/UCI/arff/hypothyroid.arff, 2014.
[47] J. Carletta, "Assessing agreement on classification tasks: The Kappa statistic," *Comput. Linguist.*, vol. 22, no. 2, pp. 249–254, 1996.

Mehran Mozaffari-Kermani (M'11) received the B.Sc. degree in electrical and computer engineering from the University of Tehran, Tehran, Iran, in 2005, and the M.E.Sc. and Ph.D. degrees from the Department of Electrical and Computer Engineering, University of Western Ontario, London, ON, Canada, in 2007 and 2011, respectively.

He joined Advanced Micro Devices as a Senior ASIC/Layout Designer, integrating sophisticated security/cryptographic capabilities into a single accelerated processing unit. He received the prestigious Natural Sciences and Engineering Research Council of Canada (NSERC) Postdoctoral Research Fellowship. In 2012, he joined the Electrical Engineering Department, Princeton University, Princeton, NJ, USA, as an NSERC Postdoctoral Research Fellow. He is currently with the Department of Electrical and Microelectronic Engineering, Rochester Institute of Technology, Rochester, NY, USA. His current research interests include emerging security/privacy measures for deeply embedded systems, cryptographic hardware systems, fault diagnosis and tolerance in cryptographic hardware, VLSI reliability, and low-power secure and efficient FPGA and ASIC designs.

He is the Guest Editor of the IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING for the special issue of Emerging Security Trends for Deeply-Embedded Computing Systems (2014–2015). He is currently serving as the Technical Committee Member for a number of security/reliability conferences including DFT, FDTC, RFIDsec, LightSEC, and WAIFI. His research in 2014–2015 is funded through the Texas Instruments Faculty Award (Douglas Harvey).

Susmita Sur-Kolay (SM'05) received the B.Tech. degree in electronics and electrical communication engineering from IIT, Kharagpur, India, and the Ph.D. degree in computer science and engineering from Jadavpur University, Kolkata, India.

She was in the Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA, USA, from 1980 to 1984. She was a Postdoctoral Fellow in the University of Nebraska-Lincoln, Nebraska-Lincoln, NE, USA, in 1992, a Reader in Jadavpur University from 1993 to 1999, a Visiting Faculty Member with Intel Corporation, Santa Clara, CA, USA, in 2002, and a Visiting Researcher in Princeton University, Princeton, NJ, USA, in 2012. She is a Professor in the Advanced Computing and Microelectronics Unit, Indian Statistical Institute, Kolkata. She has coedited two books, authored a book chapter in the Handbook of Algorithms for VLSI Physical Design Automation, and coauthored about 100 technical articles. Her current research interests include electronic design automation, hardware security, quantum computing, and graph algorithms.

Dr. Sur-Kolay was a Distinguished Visitor of the IEEE Computer Society, India. She has been an Associate Editor of the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION SYSTEMS and is currently an Associate Editor of the *ACM Transactions on Embedded Computing Systems*. She has served on the technical program committees of several leading conferences, and as the Program Chair of the 2005 International Conference on VLSI Design, the 2007 International Symposium on VLSI Design and Test, and the 2011 IEEE Computer Society Annual Symposium on VLSI. Among other awards, she received the President of India Gold Medal from IIT, Kharagpur.

Anand Raghunathan (F'12) received the B.Tech. degree in electrical and electronics engineering from the Indian Institute of Technology, Madras, India, and received the M.A. degree from Princeton University, Princeton, NJ, USA, where he also received the Ph.D. degree in electrical engineering in 1997.

He is a Professor in the School of Electrical and Computer Engineering at Purdue University, West Lafayette, IN, USA, where he leads the Integrated Systems Laboratory. His research explores domain-specific architecture, system-on-chip design, embedded systems, and heterogeneous parallel computing. Previously, he was a Senior Research Staff Member at NEC Laboratories America and held the Gopalakrishnan Visiting Chair in the Department of Computer Science and Engineering, Indian Institute of Technology, Madras, India.

Dr. Raghunathan has coauthored a book *High-Level Power Analysis and Optimization*, eight book chapters, 21 U.S. patents, and more than 200 refereed journal and conference papers. His publications have been recognized with eight best paper awards and four best paper nominations. He received the Patent of the Year Award (recognizing the invention with the highest impact), and two Technology Commercialization Awards from NEC. He was chosen by MIT's Technology Review among the TR35 (top 35 innovators under 35 years, across various disciplines of science and technology) in 2006, for his work on "making mobile secure." He has served on the technical program and organizing committees of several leading conferences and workshops. He has chaired the ACM/IEEE International Symposium on Low Power Electronics and Design, the ACM/IEEE International Conference on Compilers, Architecture, and Synthesis for Embedded Systems, the IEEE VLSI Test Symposium, and the IEEE International Conference on VLSI Design. He has served as Associate Editor of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, IEEE TRANSACTIONS ON VLSI SYSTEMS, *ACM Transactions on Design Automation of Electronic Systems*, IEEE TRANSACTIONS ON MOBILE COMPUTING, *ACM Transactions on Embedded Computing Systems*, *IEEE Design & Test of Computers*, and *the Journal of Low Power Electronics*. He received the IEEE Meritorious Service Award in 2001 and the Outstanding Service Award in 2004. He is a Golden Core Member of the IEEE Computer Society.

Niraj K. Jha (S'85–M'85–SM'93–F'98) received the B.Tech. degree in electronics and electrical communication engineering from the Indian Institute of Technology (IIT), Kharagpur, India, in 1981, the M.S. degree in electrical engineering from S.U.N.Y. at Stony Brook, NY, USA, in 1982, and the Ph.D. degree in electrical engineering from the University of Illinois at Urbana-Champaign, Champaign, IL, USA, in 1985.

He is a Professor of electrical engineering at Princeton University, Princeton, NJ, USA. His research interests include FinFETs, low-power hardware/software design, computer-aided design of integrated circuits and systems, digital system testing, quantum computing, and secure computing. He has given several keynote speeches in the area of nanoelectronic design and test.

Dr. Jha is a Fellow of the ACM. He received the Distinguished Alumnus Award from IIT Kharagpur in 2014. He has served as the Editor-in-Chief of IEEE TRANSACTIONS ON VLSI SYSTEMS and an Associate Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I AND II, IEEE TRANSACTIONS ON VLSI SYSTEMS, IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN, and *Journal of Electronic Testing: Theory and Applications*. He is currently serving as an Associate Editor of IEEE TRANSACTIONS ON COMPUTERS, *Journal of Low Power Electronics and Journal of Nanotechnology*. He has also served as the Program Chairman of the 1992 Workshop on Fault-Tolerant Parallel and Distributed Systems, the 2004 International Conference on Embedded and Ubiquitous Computing, and the 2010 International Conference on VLSI Design. He has served as the Director of the Center for Embedded System-on-a-chip Design funded by New Jersey Commission on Science and Technology. He received the AT&T Foundation Award and NEC Preceptorship Award for research excellence, NCR Award for teaching excellence, and Princeton University Graduate Mentoring Award. He has coauthored or coedited five books titled *Testing and Reliable Design of CMOS Circuits* (Kluwer, 1990), *High-Level Power Analysis and Optimization* (Norwell, MA, USA: Kluwer, 1998), *Testing of Digital Systems* (Cambridge, U.K.: Cambridge Univ. Press, 2003), *Switching and Finite Automata Theory* (3rd ed. Cambridge, U.K.: Cambridge Univ. Press, 2009), and *Nanoelectronic Circuit Design* (New York, NY, USA: Springer, 2010). He has also authored 15 book chapters. He has authored or coauthored more than 400 technical papers. He has coauthored 14 papers, which have won various awards. These include the Best Paper Award at ICCD'93, FTCS'97, ICVLSID'98, DAC'99, PDCS'02, ICVLSID'03, CODES'06, ICCD'09, and CLOUD'10. A paper of his was selected for "The Best of ICCAD: A collection of the best IEEE International Conference on Computer-Aided Design papers of the past 20 years," two papers by IEEE Micro Magazine as one of the top picks from the 2005 and 2007 Computer Architecture conferences, and two others as being among the most influential papers of the last ten years at IEEE Design Automation and Test in Europe Conference. He has coauthored another six papers that have been nominated for best paper awards. He has received 14 U.S. patents. He has served on the program committees of more than 150 conferences and workshops.