Iterative Classification for Sanitizing Large-Scale Datasets

Bo Li, Yevgeniy Vorobeychik, and Muqun Li
Electrical Engineering and Computer Science
Vanderbilt University
Nashville, TN
{bo.li.2,yevgeniy.vorobeychik,muqun.li}@vanderbilt.edu

Bradley Malin
Biomedical Informatics
Vanderbilt University
Nashville, TN
b.malin@vanderbilt.edu

***Abstract*—Cheap ubiquitous computing enables the collection of massive amounts of personal data in a wide variety of domains. Many organizations aim to share such data while obscuring features that could disclose identities or other sensitive information. Much of the data now collected exhibits weak structure (e.g., natural language text) and machine learning approaches have been developed to identify and remove sensitive entities in such data. Learning-based approaches are never perfect and relying upon them to sanitize data can leak sensitive information as a consequence. However, a small amount of risk is permissible in practice, and, thus, our goal is to balance the value of data published and the risk of an adversary discovering leaked sensitive information. We model data sanitization as a game between 1) a publisher who chooses a set of classifiers to apply to data and publishes only instances predicted to be non-sensitive and 2) an attacker who combines machine learning and manual inspection to uncover leaked sensitive entities (e.g., personal names). We introduce an iterative greedy algorithm for the publisher that provably executes no more than a linear number of iterations, and ensures a low utility for a resource-limited adversary. Moreover, using several real world natural language corpora, we illustrate that our greedy algorithm leaves virtually no automatically identifiable sensitive instances for a state-of-the-art learning algorithm, while sharing over 93% of the original data, and completes after at most 5 iterations.**

I. Introduction

Vast quantities of personal data are now collected in a wide variety of domains [1]. It is anticipated that such data can enable significant improvements in the quality of services provided to individuals and facilitate new discoveries for society. At the same time, the data collected is often sensitive, and regulations, such as the Privacy Rule of the Health Insurance Portability and Accountability Act of 1996 (when disclosing medical records) [2], Federal Rules of Civil Procedure (when disclosing court records) [3], and the European Data Protection Directive [4] often recommend the removal of identifying information. To accomplish such goals, the past several decades have brought forth the development of numerous data protection models [5]. These models invoke various principles, such as hiding individuals in a crowd (e.g., k-anonymity [6]) or perturbing values to ensure that little can be inferred about an individual even with arbitrary side information (e.g., ϵ-differential privacy [7]). All of these approaches are predicated on the assumption that the publisher of the data knows where the identifiers are from the outset. More specifically, they assume the data has an explicit representation, such as a relational form [8], where the data has at most a small set of values per feature [9], [10].

However, it is increasingly the case that the data we generate lacks a formal relational (or explicitly structured) representation. A clear example of this phenomenon is the substantial quantity of natural language text which is created in many different arenas, ranging from field reports of intelligence agencies [11] to clinical notes in medical records [12] to microblogging over social media platforms [13]. To protect such data, there has been a significant amount of research into natural language processing (NLP) techniques to detect (and subsequently redact or substitute) identifiers [14], [15]. The most scalable versions of such techniques are rooted in machine learning methods [16], in which the publisher of the data annotates instances of identifiers in the text, and the machine attempts to learn a classifier (e.g., a grammar) to predict where such identifiers reside in a much larger corpus. Unfortunately, no learned classifier is perfect, and some sensitive information will invariably leak through to the data recipient. This is clearly a problem if, for instance, the information leaked corresponds to direct identifiers (e.g., personal name) or quasi-identifiers (e.g., ZIP codes or dates of birth) which may be exploited in re-identification attacks, such as the re-identification of Thelma Arnold in the search logs disclosed by AOL [17] or the Social Security Numbers in Jeb Bush's emails [18].

Rather than attempt to detect and redact every sensitive piece of information, our goal is to guarantee that even if sensitive entities remain in the published data, the adversary cannot easily find them. Fundamental to our approach is the acceptance of non-zero privacy risk, which we view as unavoidable. This is consistent with most privacy regulation, such as HIPAA, which allows expert determination that privacy risk is sufficiently small [2], and the EU Data Protection Directive, which "does not require anonymisation to be completely risk-free" [19]. Our starting point is a threat model in which an attacker uses published data to first train a classifier to predict sensitive entities (based on a labeled subset of the data), prioritizes inspection based on the predicted positives, and inspects (and verifies the true sensitivity status of) B of these in a prioritized order, where B is the budget available to inspect (or read) instances. An illustration of such a setting is depicted in Figure 1. In this threat model, we consider an idealized adversary with several elements of omniscience. First, we assume that the adversary can always correctly assess the

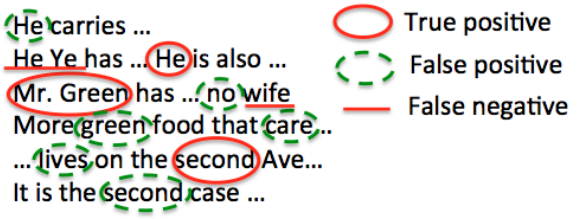

Fig. 1. An example of sensitive and non-sensitive instances that need to be distinguished via manual inspection.

true sensitivity for any manually inspected instance. Second, we assume that the adversary computes an optimal classifier (that is, a classifier with maximum accuracy within a given hypothesis class).

We use this threat model to construct a game between a *publisher*, who 1) applies a collection of classifiers to an original data set, 2) prunes all the positives predicted by any classifier, and 3) publishes the remainder, and an *adversary* acting according to our threat model. We show that, in any locally optimal publishing strategy, when the risk associated with exploited sensitive entities is high, an adversary cannot learn a classifier with a high true positive count. When these conditions hold, we say that sensitive data is *hiding in plain sight*—even though it may be leaked, it is difficult for a motivated adversary to discover. Moreover, we exhibit a greedy publishing strategy which is guaranteed to converge to a local optimum (and consequently guarantees the above two properties) in a linear (in the size of the data) number of iterations. Our experiments on four distinct data sets demonstrate the power of our approach, showing that 1) the number of residual true positives is always quite small, and 2) demonstrating that most ($> 93\%$) of the original data is nevertheless published.

II. Model

TABLE I. Table of Notations

n	$\triangleq$	number of total instances
$\mathcal{H}$	$\triangleq$	hypothesis class of the publisher
H	$\triangleq$	the subset of classifiers chosen by the publisher
S	$\triangleq$	sensitive instances
N	$\triangleq$	non-sensitive instances
$TP(h, P)$	$\triangleq$	number of *true positives* by h on P
$TN(h, P)$	$\triangleq$	number of *true negatives* by h on P
$FP(h, P)$	$\triangleq$	number of *false positives* by h on P
$FN(h, P)$	$\triangleq$	number of *false negatives* by h on P
α	$\triangleq$	percent of identifiers in data
h_A	$\triangleq$	the attacker's classifier
$T(H)$	$\triangleq$	loss function of data publisher for H

Table I summarizes all the notation used in this paper. Imagine that a publisher's dataset consists of a set of n entities (or words), $X = \{x_1, \ldots, x_n\}$, of which he will publish a subset $P \subseteq X$. The publisher may have an additional data set for training a classifier to predict whether an entity x is sensitive. We let α denote the fraction of the original n entities that are sensitive. A learning algorithm is designed to select a hypothesis that best supports the data. Here we consider the hypothesis to be a function f mapping from the data space $\mathcal{D}$ to the response space $\mathcal{E}$; i.e., $f : \mathcal{D} \to \mathcal{E}$. Of course there are many such hypotheses. We assume f belongs to a family of all possible hypotheses $\mathcal{H}$, and assume this hypothesis class $\mathcal{H}$ is known to the public, including the publisher and attackers. This is a natural assumption, considering that state-of-the-art machine learning algorithms are well known and typically have multiple high-quality open source implementations. Specifically, the response space $\mathcal{E} = \{0, 1\}$ within our problem indicates whether the entity x is sensitive (S, $f(x) = 1$) or non-sensitive (N, $f(x) = 0$), and $\mathcal{H}$ represents a set of binary classifiers.

We use h to denote a classifier chosen from the hypothesis class $\mathcal{H}$. For a classifier h and a data set Y, we introduce the following notation:

- $FP(h, Y) = |\cup_{x \in Y} \{x \in N | h(x) = 1\}|$: the number of false positive instances of h on Y,
- $TP(h, Y) = |\cup_{x \in Y} \{x \in S | h(x) = 1\}|$: the number of true positive instances of h on Y,
- $FN(h, Y) = |\cup_{x \in Y} \{x \in S | h(x) = 0\}|$: the number of false negative instances of h on Y, and
- $TN(h, Y) = |\cup_{x \in Y} \{x \in N | h(x) = 0\}|$: the number of true negative instances of h on Y.

Clearly, if $|Y| = m$, $FP(h, Y) + TP(h, Y) + FN(h, Y) + TN(h, Y) = m \ \forall h \in \mathcal{H}$. Finally, we define $FP(h, \emptyset) = FN(h, \emptyset) = TP(h, \emptyset) = TN(h, \emptyset) \equiv 0$.

Threat Model

Suppose that an adversary obtains the published data $P \subseteq X$. We assume that an adversary has a fixed inspection budget, B, which can be thought of as manual inspection of actual instances to verify whether or not they are sensitive (and, consequently, have value to the adversary). If a sensitive instance is found, we assume that it is exploited by an adversary, who gains L for every such instance, which is identical to the publisher's loss. Thus, when the attacker selects a set $I \subseteq P$ of instances for inspection, such that $|I| \leq B$, his utility is

$$U_A(I) = L|\{\text{sensitive instances} \in I\}| = L \sum_{x \in I} S(x),$$

where $S(x) = 1$ iff x is sensitive. A central aspect of the threat model is the specific way that the attacker chooses the set I of instances to inspect. A simple baseline is to choose I uniformly at random from P. We use U_A to denote the utility that the attacker obtains when using this simple baseline. Presumably, however, the attacker can do better by using a more sophisticated strategy. In particular, we suppose that a *sophisticated* attacker proceeds as follows:

1) Choose a classifier

$$h_A(P) \in \arg\min_{h \in \mathcal{H}} \frac{FP(h, P) + FN(h, P)}{|P|}.$$

In other words, the attacker chooses an optimal classifier from $\mathcal{H}$ in terms of accuracy. From the publisher's perspective, this is a very pessimistic limit of an attacker who uses a subset of P for training a standard classification algorithm, such as an SVM.

2) Prioritize instances in P by ranking all $x \in P$ with $h^*(x) = 1$ first, followed by those with $h^*(x) = 0$. Within each class, the order is arbitrary.
3) Choose I in this ranked order until it contains B instances. In other words, first the attacker will choose the predicted positives, followed by predicted negatives (if there is any budget remaining).

We simply refer to h_A where P is clear from context. We let U_A^* denote the attacker's utility when using this more sophisticated learning-based strategy. A technical caveat is that depending on the quality of the classifier, U_A^* is not necessarily higher than U_A; below, we provide a sufficient condition for $U_A^* \geq U_A$.

As an illustration, let us return to Figure 1 which presents an example of the behavior of an attacker given a published dataset containing sensitive and non-sensitive instances. Assume the circled words are classified as positives by h_A. Therefore, the attacker would inspect these words and their surrounding context first. However, in this setting, some of the words inspected are not sensitive instances (i.e., false positives; shown in dashed ovals). For example, the first dashed "He" is a pronoun, while the solid circled "He" is actually the name of a person. Therefore, if the attacker has sufficient budget to inspect all of the circled instances, he would gain 3 units of utility (i.e., true positives, shown in solid ovals), and waste 3 units of budget (again, in dashed ovals).

Data Publisher Model

To develop some intuition for our publisher model, let us first consider the typical approach for sanitizing data (we assume that the defender is able to learn an optimal classifier):

1) Learn a classifier
$$\bar{h} \in \arg\min_{h \in \mathcal{H}} \frac{FP(h, X) + FN(h, X)}{|X|}.$$
Let $X_1 = \{x \in X | \bar{h}(x) = 1\}$ (i.e., X_1 is the set of predicted positives).
2) Publish the data set $P = X \setminus X_1$.

Essentially all of the approaches in the literature assume this, or a similar, form. To apply our threat model above, we consider two possibilities: a) the attacker's classifier h_A can successfully identify residual sensitive instances, or b) the attacker's classifier cannot detect residual positives. If we are in situation (b), the publisher can view the sanitization as a success. Situation (a), on the other hand, is clearly problematic, but it also suggests a natural solution: the publisher can apply h_A to residual data, remove the sensitive instances, and only then publish the data. Indeed, this is where the symmetry between the publisher and attacker, taking advantage of the common knowledge of $\mathcal{H}$, is pivotal. Specifically, *the publisher can simulate anything that the attacker would do*.

Moreover, there is no reason to stop at this point. In fact, the publisher should continue as long as the simulated classifier that would be used by the attacker is sufficiently good. This observation also offers the key intuition for our results. Whenever the publisher chooses to stop, the attacker's ability to identify sensitive instances must inherently be relatively weak. Of course, this will depend on the relative loss to the publisher from correctly identified sensitive entities and the value of publishing data.

Using the developed intuition, we model the publisher as selecting a finite set of classifiers $H \subseteq \mathcal{H}$, where $H = \{h_1, h_2, ..., h_D\}$. Figure 2 shows the process of applying a set of classifiers before publishing the data. After applying each classifier h_i, the positive instances are replaced with the fake tokens, such as "**[NAME]**" replacing an individual's name.

Let $X_1(H) = \cup_{h \in H}\{x \in X | h(x) = 1\}$, that is, the set of all positives predicted by the classifiers in H, and let $P(H) = X \setminus X_1(H)$; we use P with no argument where H is clear from context. The publisher's approach is:

1) Choose a collection of classifiers H (we address this choice below).
2) Publish the data set $P(H) = X \setminus X_1(H)$.

Let $FN(H)$ be the number of false negatives of H in X, which we define as all residual sensitive instances in P, and let $FP(H)$ be the number of false positives in X, that is, all predictive positives by any $h \in H$ which are, in fact, not sensitive. It is immediate that for any H, $FN(H) \leq \alpha n$ (the number of false negatives is at most the total number of sensitive entities in the original data) and $TN(H) \leq (1 - \alpha)n$ (the number of true negatives is at most the total number of non-sensitive entities). If we allow the attacker to have an infinite budget, then every false negative will be exploited, resulting in the total loss of $L \cdot FN(H)$. In addition, each false positive costs the publisher a fixed amount C, which we can interpret as the value of not publishing the data. Thus, we define the (worst-case) total loss to the publisher from using a set of classifiers H as

$$T(H) = L \cdot FN(H) + C \cdot FP(H),$$

where $FN(H) = |\cap_{h \in H}\{x \in S | h(x) = 0\}|$ and $FP(H) = |\cup_{h \in H}\{x \in N | h(x) = 1\}|$, where S, N represent the sensitive and non-sensitive instances, respectively. $TN(H)$ and $TP(H)$ can be defined similarly.

III. A Greedy Algorithm for Automated Data Sanitization

Given a formal model, we can now present our iterative algorithm for automated data sanitization, which we term *GreedySanitize*. Our algorithm (shown as Algorithm 1) is simple to implement and involves iterating over the following steps: 1) compute a classifier on training data, 2) remove all predicted positives from the training data, and 3) add this classifier to the collection. The algorithm continues until a specified stopping condition is satisfied, at which point we publish only the predicted negatives, as above. It is important to emphasize that *GreedySanitize* is *qualitatively different* from typical ensemble learning schemes in several ways. First, it is a crucial feature of the algorithm that a classifier is re-trained in every iteration on data which includes only predicted negatives from all prior iterations; this is entirely unlike the mechanics of any ensemble learning algorithm we are aware of.[1] Second, our

[1]Typical ensemble learning algorithms will either focus on mistakes made in prior iterations (boosting is an example of this), take no note of performance by other members of the ensemble (e.g., bagging), or use a fixed set of classifiers as inputs into a meta-classifier [20].

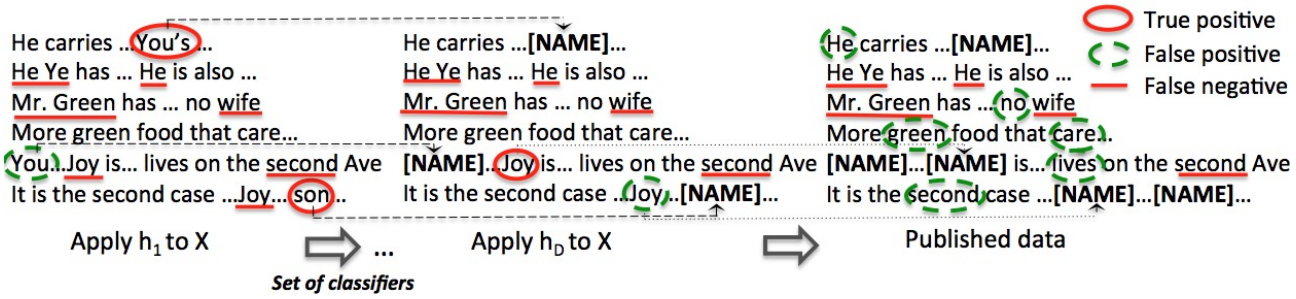

Fig. 2. The process for applying a set of classifiers H to data X.

Algorithm 1 GreedySanitize(X_t), X_t : training data.

$H \leftarrow \{\}$, $k \leftarrow 0$, $h_0 \leftarrow \emptyset$, $D_0 \leftarrow X_t$,
repeat
  $H \leftarrow H \cup h_k$
  $k = k + 1$
  $h_k \leftarrow$LearnClassifier(D_{k-1})
  $D_k \leftarrow$RemovePredictedPositives(D_{k-1}, h_k)
until $T(H \cup h_k) - T(H) \geq 0$
return H

algorithm removes the union of all predicted positives, whereas ensemble learning typically applies a weighted voting scheme to predict positives; our algorithm, therefore, is fundamentally more conservative when it comes to sensitive entities in the data. Third, the stopping condition is uniquely tailored to the algorithm, and is critical in enabling us to provide provable guarantees about privacy-related performance of the algorithm.

Given the iterative nature of the algorithm, it is not obvious that it is always guaranteed to terminate. The following theorem asserts that *GreedySanitize* will always terminate in a linear number of iterations.

Theorem 3.1: Algorithm 1 terminates after at most $|X_t|$ iterations.

We omit all proofs due to the space constraints.

IV. Analysis of *GreedySanitize*

Our theoretical analysis of the proposed *GreedySanitize* algorithm focuses on the following questions: what kinds of privacy guarantees does this algorithm offer? To address this question, we abstract away the details of our algorithm behind the veil of its stopping condition, which turns out to be the primary driver of our results. This also allows us to state the privacy guarantees in much more general terms.

Analysis of Locally Optimal Publishing Policies

In this section we analyze the adversary's ability to infer sensitive information from published data if the defender's choice of classifiers H to apply to original data satisfies the following *local optimality* condition.

Definition 4.1: A set of classifiers $H \subseteq \mathcal{H}$ is a *local optimum* if $T(H \cup h_A) - T(H) \geq 0$.

In plain terms, a subset is a local optimum if the adversary's optimal classifier h_A (that is, the attacker's best classifier choice to apply to the published data), when added to this subset, does not improve the publisher's utility. Under a minor regularity condition that $\mathcal{H}$ contains an identity (which can always be added), there is always a trivial local optimum of not releasing any data. Notice that the local optimality condition is exactly the stopping condition of *GreedySanitize*, which means that when the algorithm terminates, its output set of hypotheses H is guaranteed to be a local optimum.

We now state the primary result, which characterizes all locally optimal solutions H.

Theorem 4.1: $H \subseteq \mathcal{H}$ is a local optimum if and only if either $TP(h_A, P) = 0$ or $\frac{FP(h_A,P)}{TP(h_A,P)} \geq \frac{L}{C}$.

Below, we simplify notation by defining $FP_A \equiv FP(h_A, P)$, and defining FN_A TP_A, and TN_A similarly, with H becoming an implicit argument throughout. Now, observe that if $L/C > (1-\alpha)n$, the only locally optimal solutions have $TP_A = 0$, because otherwise $\frac{FP_A}{TP_A} \leq (1-\alpha)n < L/C$.

As a direct consequence of Theorem 4.1, we can bound TP_A in all locally optimal solutions.

Theorem 4.2: For any locally optimal $H \subseteq \mathcal{H}$, $TP_A \leq \frac{C}{L}(1-\alpha)n$.

The upshot of Theorem 4.2 is that when C is small relative to L, any locally optimal H will guarantee that the attacker cannot learn a classifier that correctly identifies more than a few sensitive instances.

V. Experiments

In this section, we assess the performance of *GreedySanitize* (GS) using four text data sets to protect the personal sensitive identifiers (here we only consider the individuals' names): 1) publicly accessible medical records from the I2B2 corpus [21], 2) a private electronic medical records (EMR) dataset from the Vanderbilt University Medical Center (VUMC), 3) Enron email data, and 4) newsgroup data [22]. We used three state-of-the-art learning algorithms for sensitive entity recognition: conditional random fields (CRF), which consistently ranks as the best method for identifying personal health information in electronic medical records [14], [21], support vector machine (SVM) [23], which makes use of the features of the word itself, part-of-speech (POS), morphologic information, and the history class of previous features assigned by the classifier; as well as a recently proposed ensemble method [24], which applies CRF to classify first and then uses SVM to reduce the false positives. All these play a dual-role in our experiments: they serve as a comparison baseline

to prior art, as well as the core learning algorithms in our own Algorithm 1 (GreedySanitize). In all the experiments, the attacker first runs all three of these algorithms on the training holdout from published data, and then chooses the best performing classifier. Our evaluation is based on four-fold cross-validation, with GreedySanitize running on the training data and using the incidence of true and false negatives on training data to determine when to stop The implementation and datasets are available at: https://www.dropbox.com/s/aq1je9dydhiq3m0/privacy_icdm.zip?dl=0.

Privacy Risk

When the budget of the attacker is small, our theoretical results provide an upper bound on the expected number of identified instances. While this upper bound suggests that risk becomes arbitrarily small when the associated loss is large, it is not tight. In Figure 3 we demonstrate that the number of identified instances (which is equivalent to the number of true positives for the attacker's classifier) typically becomes negligible even when L is quite small relative to C.

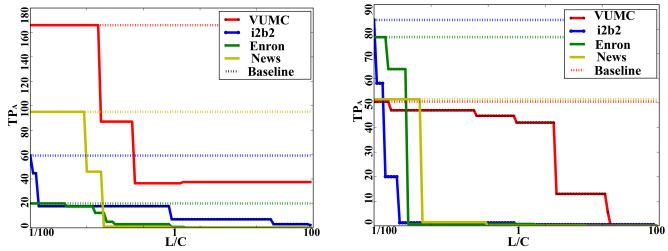

Fig. 3. The number of residual *true positive* instances TP_A (equivalently, identified instances for an attacker with a small budget) after running GreedySanitize for the i2b2, VUMC, Enron, and Newsgroup datasets. (a) GreedySanitize using CRF (dashed lines, or baseline, correspond to standard application of CRF). (b) GreedySanitize using the best classifier from {CRF, SVM, Ensemble} (dashed lines correspond to the baseline application of the best classifier from this collection).

Data Utility

Our next evaluation concerns whether we can still retain the data utility with such a high privacy preserving requirement. This is something that motivates the presented approach (as compared to simply suppressing all data), but that we did not explicitly consider in the theoretical analysis. Intuitively, the proposed *GreedySanitize* algorithm should strike a reasonable balance: it stops immediately after a local optimum is reached. Here, we evaluate the data utility of the published data using the *publish ratio*, which is defined as the proportion of the original number of entities in the published data.

Figure 4 compares GreedySanitize to cost-sensitive variants of the baseline algorithms (CRF, SVM, and Ensemble). GreedySanitize preserves most of the data utility even when L/C is high. Specifically, in both of the EMR datasets over 98% of the data is published, *even when L/C is quite high*. The performance for the other two data sets is lower, but still, over 93% of the data is ultimately published, even with large L/C ratios. In contrast, when the loss due to re-identification is moderate or high, cost-sensitive algorithms essentially suppress most of the data, resulting in very low

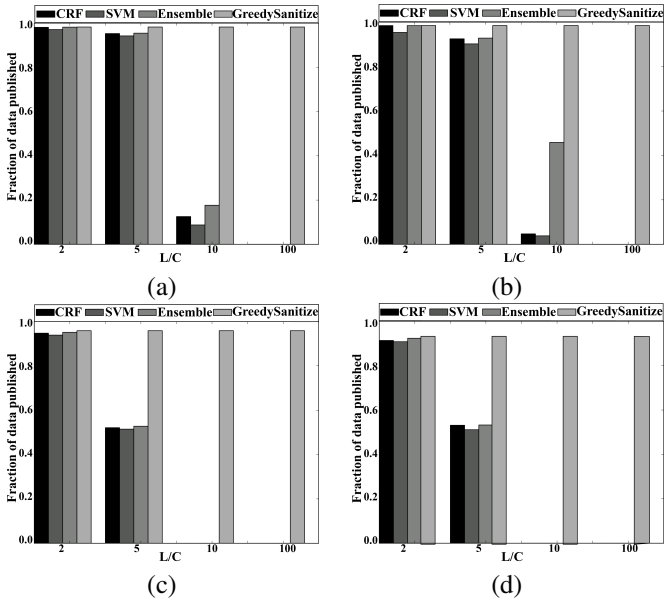

Fig. 4. Fraction of data published for different classifiers with cost sensitive learning. (a)-(d) corresponds to the i2b2, VUMC, News, and Enron datasets.

utility. GreedySanitize therefore offers a far better balance between risk and utility than the state-of-the-art alternatives.

Impact of the Size of the Hypothesis Space

One important issue in applying GreedySanitize is that perhaps the attacker will make use of a new algorithm that the publisher had not considered. We now explore this issue by considering the quality of decisions when the publisher uses a single classifier, or the best of all three that we consider, at the core of GS.

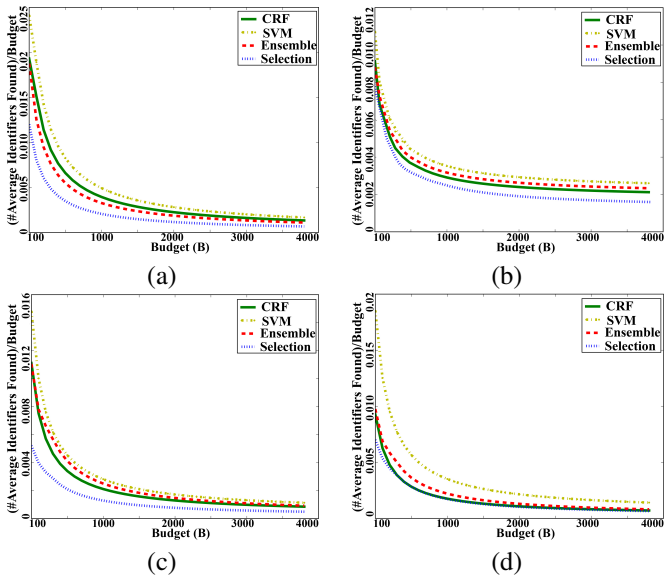

Fig. 5. The ratio of average sensitive identifiers found by the attacker and the adversarial budget, while the publisher applies different classifiers as CRF, SVM, Ensemble, and Selection which allows the publisher to choose a learner with highest accuracy from {CRF, SVM, Ensemble} for GreedySanitize (L/C=10). (a)-(d) corresponds to the i2b2, VUMC, News, and Enron datasets.

Figures 5 compare these four options (the three single-

classifier options, and the last, called "Selection", where the most accurate of these three classifiers is chosen in each iteration), evaluated when the adversary chooses the most accurate of these. The overall observation is that while increasing the space of classifiers to choose from does help, the difference is relatively small, so that significant underestimation of the attacker's strength appears unlikely to make much impact.

Number of Greedy Iterations

The final issue we consider is the number of iterations of GreedySanitize for the different data sets. We found that for all four datasets (and for the entire range of L/C that we consider) the average number of iterations is less than 5. Our theoretical upper bound is, therefore, extremely pessimistic. Indeed, for some datasets, such as the VUMC EMR dataset, the average number of iterations is just above 2 even when the loss from leaking sensitive information is quite high.

VI. Conclusion

Our ability to take full advantage of large amounts of unstructured data collected across a broad array of domains is limited by the sensitive information contained therein. We proposed a novel framework for sanitization of such data at scale, introducing, for the first time, a principled threat model, a very general class of publishing strategies, and a greedy, yet effective, data publishing algorithm. Our analysis with multiple natural language corpora demonstrates that any locally optimal solution in our general framework will suppress enough sensitive data to make the problem of re-identification (i.e., uncovering identifiers leaked through) extremely challenging even for a highly capable adversary armed with state-of-the-art machine learning tools. Our experimental evaluation confirms these results, and further shows that our algorithm is: a) substantially better than existing approaches for suppressing sensitive data, and b) retains most of the value of the data, suppressing less than 10% of information on all four data sets we considered in evaluation. In contrast, cost-sensitive variants of standard learning methods yield virtually no residual utility, suppressing most or all of the data, when the loss associated with privacy risk is even moderately high. Since our adversarial model is deliberately extremely strong, our results suggest feasibility for data sanitization at scale.

VII. Acknowledgments

This work was supported by the NIH (R01-LM011366, R01- HG006844, R01-LM009989, U01-HG006478, U01-HG006385), NSF (CCF-0424422), AFRL (FA8785-14-2-0180), ONR (N00014-15-1-2621), Sandia National Labs (contract 2191), and Symantec Labs Graduate Research Fellowship.

References

[1] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 1, pp. 97–107, 2014.

[2] U.S. Dept. of Health and Human Services, "Standards for privacy and individually identifiable health information; final rule," *Federal Register*, vol. 65, no. 250, pp. 82 462–82 829, 2000.

[3] Committe on the Judiciary House of Representatives, "Federal Rules of Civil Procedure," 2014.

[4] European Parliament and Council of the European Union, "Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data," *Official Journal L*, vol. 281, pp. 0031–0050, 1995.

[5] B. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-preserving data publishing: A survey of recent developments," *ACM Computing Surveys*, vol. 42, no. 4, p. 14, 2010.

[6] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.

[7] C. Dwork, "Differential privacy: A survey of results," in *Proc. 5th International Conference on Theory and Applications of Models of Computation*, 2008, pp. 1–19.

[8] L. Sweeney, "Achieving *k*-anonymity privacy protection using generalization and suppression," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 571–588, 2002.

[9] G. Poulis, A. Gkoulalas-Divanis, G. Loukides, S. Skiadopoulos, and C. Tryfonopoulos, "SECRETA: A system for evaluating and comparing relational and transaction anonymization algorithms," in *Proc. 17th International Conference on Extending Database Technology*, 2014, pp. 620–623.

[10] Y. He and J. F. Naughton, "Anonymization of set-valued data via top-down, local generalization," *Proc. VLDB Endowment*, vol. 2, no. 1, pp. 934–945, 2009.

[11] J. Olive, C. Christianson, and J. McCary, *Handbook of Natural Language Processing and Machine Translation: DARPA Global Autonomous Language Exploitation*. Springer Press, 2011.

[12] P. Nadkarni, L. Ohno-Machado, and W. Chapman, "Natural language processing: an introduction," *Journal of the American Medical Informatics Association*, vol. 18, no. 5, pp. 544–551, 2011.

[13] A. Ritter, Mausam, O. Etzioni, and S. Clark, "Open domain event extraction from Twitter," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2012, pp. 1104–1112.

[14] J. Aberdeen, S. Bayer, R. Yeniterzi, B. Wellner, C. Clark, D. Hanauer, B. Malin, and L. Hirschman, "The MITRE Identification Scrubber Toolkit: design, training, and assessment," *International Journal of Medical Informatics*, vol. 79, no. 12, pp. 849–859, 2010.

[15] A. Benton, S. Hill, L. Ungar, A. Chung, C. Leonard, C. Freeman, and J. H. Holmes, "A system for de-identifying medical message board text," *BMC Bioinformatics*, vol. 12 Suppl 3, p. S2, 2011.

[16] O. Ferrandez, B. R. South, S. Shen, F. J. Friedlin, M. H. Samore, and S. M. Meystre, "BoB, a best-of-breed automated text de-identification system for VHA clinical documents," *Journal of the American Medical Informatics Association*, vol. 20, no. 1, pp. 77–83, 2013.

[17] M. Barbaro, T. Zeller, and S. Hansell, "A face is exposed for aol searcher no. 4417749," *New York Times*, vol. 9, no. 2008, p. 8, 2006.

[18] R. Hackett, "Jeb Bush exposed 13,000 social security numbers: Here's where they were hiding," *Forbes*, 2015 Feb 13.

[19] I. C. Office, "Anonymisation: managing data protection risk code of practice," 2012.

[20] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. Springer, 2011.

[21] Ö. Uzuner, Y. Luo, and P. Szolovits, "Evaluating the state-of-the-art in automatic de-identification," *Journal of the American Medical Informatics Association*, vol. 14, no. 5, pp. 550–563, 2007.

[22] E. Minkov, R. C. Wang, and W. W. Cohen, "Extracting personal names from email: applying named entity recognition to informal text," in *Proc. Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 2005, pp. 443–450.

[23] S. Doan and H. Xu, "Recognizing medication related entities in hospital discharge summaries using support vector machine," in *International Conference on Computational Linguistics: Posters*. Association for Computational Linguistics, 2010, pp. 259–266.

[24] O. Ferrández, B. R. South, S. Shen, F. J. Friedlin, M. H. Samore, and S. M. Meystre, "BoB, a best-of-breed automated text de-identification system for vha clinical documents," *Journal of the American Medical Informatics Association*, vol. 20, no. 1, pp. 77–83, 2013.