David J. Miller
Zhen Xiang
George Kesidis

Adversarial Learning and Secure AI

CAMBRIDGE
UNIVERSITY PRESS & ASSESSMENT

Chapter 13

Error-Generic Data Poisoning Defense

Outline

1. Introduction to Error-Generic Data Poisoning (DP)
2. Some Proposed Defenses
   - KNN-D
   - GS-D
   - BIC-C-D
   - BIC-MM-TSC
3. Experiments on the two-class special case
4. Discussion

Error-Generic Data Poisoning (DP)

- Error-generic data poisoning attacks generally seek to reduce accuracy.
- A simple attack mechanism is to insert samples with the wrong labels into the training dataset, i.e., a label-flipping attack.
- Here, there is no backdoor poisoning.
- For classification, targeted models include those based on a support vector machine (SVM, see the Appendix), a Bayesian network, or a DNN.

Error Generic Data Poisoning

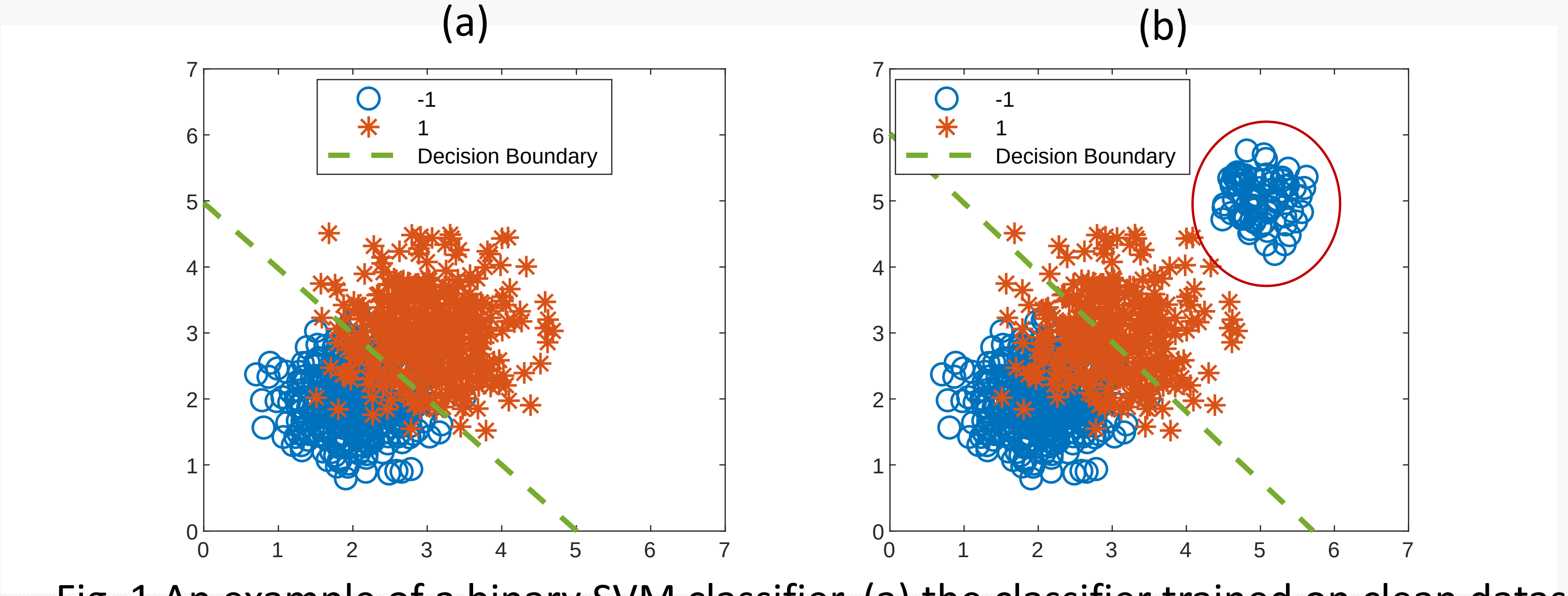


Fig. 1 An example of a binary SVM classifier. (a) the classifier trained on clean datasets, each of which has 400 data points. (b) the classifier trained on poisoned dataset, where we inject 50 red-like points into the blue set and label them as blue.

*K*NN-Defense

- Tailored to label-flipping attacks, the poisoned samples are expected to be outliers relative to untainted samples with the same labels.
- Thus, *K*NN-D relabels a sample based on the plurality label of its *K* nearest neighbors to enforce label homogeneity.
- However, this defense will fail when the number of poisoned samples is sufficiently large such that some of the neighbors of an attack sample are also attack samples.
- This defense also relies on the availability of a clean validation set to tune the sensitive hyperparameter *K*.

GS-Defense

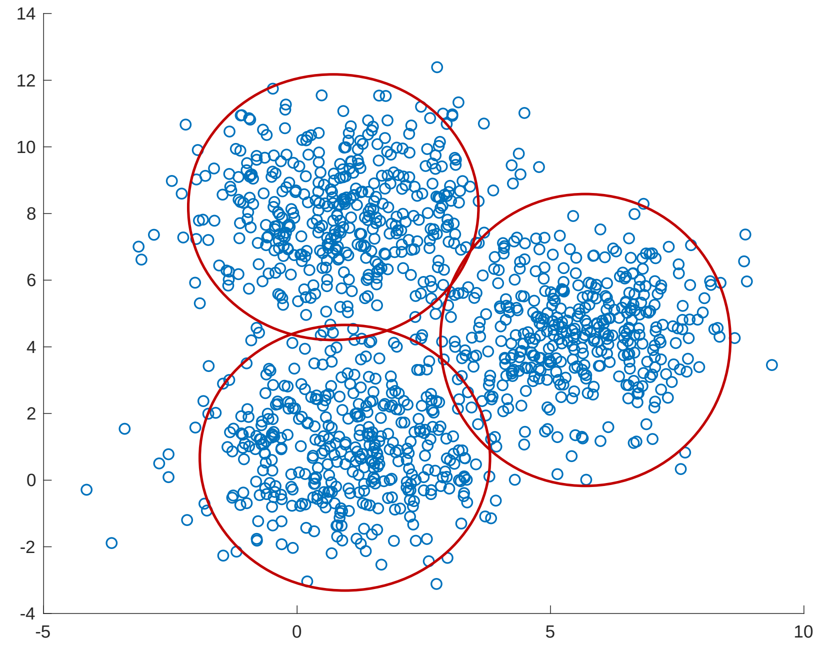
- Hypothesize that the norms of sample gradients of the loss function are larger for poisoned samples compared to clean samples.
- GS-D mitigates the effects of DP by Gradient Shaping (GS), i.e., constraining the magnitude and orientation of poisoned gradients.
- E.g., bound the gradient's l_2 norm by hyperparameter l_2_norm_clip and then add noise of a magnitude controlled by the hyperparameter noise_multiplier.

BIC-C-Defense for Two-Class Case

- Assumes that the attacker only poisons one of the two classes, with this class known to the defender.
- Thus, the defender can always model the clean class and use it as a reference to help identify poisoned samples in the corrupted class, which is especially helpful for label-flipping attacks.
- In practice, of course, the defender's assumption of which class the attacker poisons may be wrong, and the attacker may poison both classes.
- BIC-C-D is a preliminary version of the BIC-MM-TSC defense.

BIC-MM-TSC Defense: Introduction

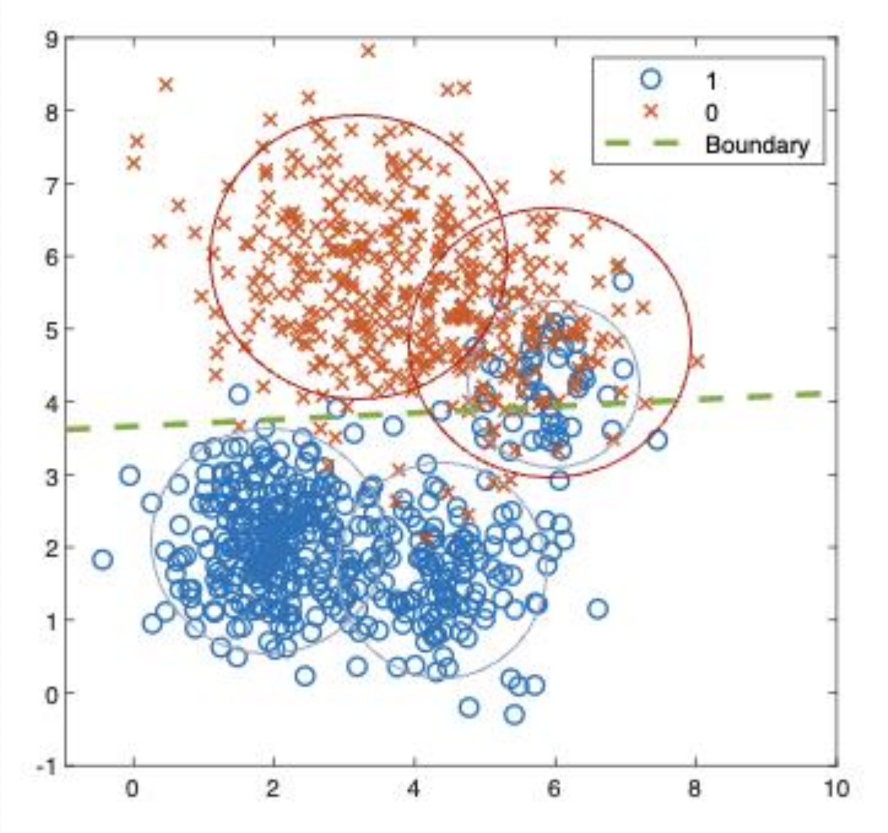
- An **unsupervised** defense against DP is based on outlier detection, a.k.a. data cleansing or sanitization, recall Chapter 5.
- An aim is to avoid performance-sensitive hyperparameters.
- Clean data may follow a (multi-modal) mixture model (MM) distribution.
- Poisoned samples may concentrate and isolate to a small subset of the mixture components.



BIC-MM-TSC: Overview

- The poisoned samples (outliers) are conjectured to form disjoint subpopulations from the clean (untainted) ones.

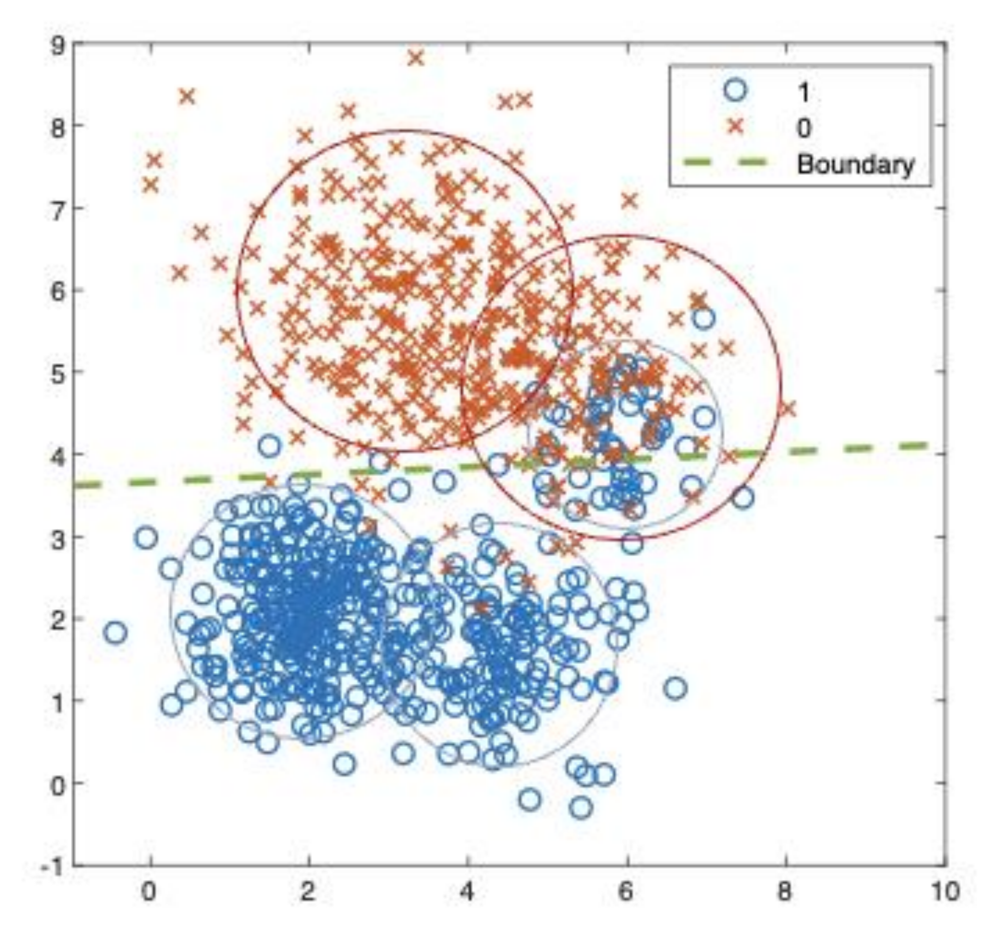
⇒ Apply mixture model to both classes to concentrate and isolate poisoned samples into several components.



Defense Methodology - Overview

- Re-distribute poisoned samples to other components to increase the data likelihood.
- Removing poisoned components may also decrease model complexity.

⇒Minimize total Bayesian Information Criterion (BIC) cost of both classes to identify poisoned components and samples. (BIC measures the ***tradeoff*** between the model complexity and the model's ability to explain training samples.)



Defense Methodology -- Overview

Defense steps:

1. Apply mixture model to concentrate and isolate poisoned samples into several components.
2. Minimize Bayesian Information Criterion (BIC) : A component is deemed poisoned if removing (or revising) it and re-distributing its samples reduces the BIC cost.
3. Samples which are re-distributed to the other class are suspicious and are removed pre-training.

Background – Mixture Model

Mixture model is a probabilistic model for representing the presence of subpopulations within an overall population.

For an individual sample x, labeled to class c, its density (likelihood) is

$$\mathrm{P}[\boldsymbol{x}|\Theta_{\boldsymbol{c}}] = \sum_{j=1}^{M^c} \alpha_j^c P[\boldsymbol{x}|\theta_j^c]$$

- M^c is model order;
- α_j^c is mass of component j, which satisfies $0 \leq \alpha_j^c \leq 1$ and $\sum_{j=1}^{M^c} \alpha_j^c = 1$;
- θ_j^c is the set of parameters specifying the joint probability mass function (PMF) or probability density function (PDF) for component j;
- Note that $\Theta_{\boldsymbol{c}} = \{\theta_j^c\} \cup \{\alpha_j^c\}$ and $\Theta = \bigcup_c \Theta_{\boldsymbol{c}}$

Recall BIC

- Recall the **Bayesian Information Criterion (BIC)** of the maximum likelihood estimation framework for model selection, see Chapter 3.
- BIC cost:

$$\mathrm{BIC} = |\Theta|k + L(\mathcal{D}|\Theta)$$

- where $|\Theta|$ is the number of parameters specifying a density function model for the dataset $\mathcal{D}$, $\mathrm{k} = \frac{1}{2}\log|\mathcal{D}|$ is the cost (penalty) for describing an individual model parameter, and $L(\mathcal{D}|\Theta)$ is the log-likelihood of the dataset $\mathcal{D}$.

BIC objective: Notation for 2-class case

- $c \in \{0,1\}$, index of a class; $y_i \in \{0,1\}$, label of sample $\boldsymbol{x}_i$; $T = |\mathcal{D}_{Train}|$
- $r_j^c = \begin{cases} 1, & component\ j\ in\ class\ c\ is\ poisoned \\ 0, & else \end{cases}$
- $q_j^c = \begin{cases} 1, component\ j\ in\ class\ c\ needs\ to\ be\ revised \\ 0, component\ j\ in\ class\ c\ needs\ to\ be\ removed \end{cases}$, q_j^c is configured only when r_j^c=1.
- $(t_i, j_i) = \underset{t=\{0,1\}, j=\{1,\ldots,M^t\}}{\arg\max} \mathrm{P}[\boldsymbol{x}_i | \theta_j^t]$, the class t_i and component under this class j_i that best-explain sample $\boldsymbol{x}_i$
- $S = \{(c,j) | c = 0,1, j = 1, \ldots, M^c\}$ be the set of components across all classes
- Complete data log-likelihood for the data from component j in class c is $L_j^c = \sum_{x \in X_j^c} \log P[x|\theta_j^c]$, where $x \in X_j^c$ if and only if, for x labeled to class c, $\mathrm{P}\left[x \middle| \theta_j^c\right] > \mathrm{P}\left[x \middle| \theta_{j'}^c\right] \forall j' \neq j, j' = 1, \ldots, M^c$

BIC objective (cont)

- The complete data BIC cost function to be minimized is

$$\mathrm{BIC}(\Theta) = \sum_{c\in\{0,1\}} \sum_{j=1}^{M^c} \left(\left(1 - r_j^c\left(1 - q_j^c\right)\right) k\left|\theta_j^c\right| + 1 + \delta\left(r_j^c, 1\right)\right)$$
$$- \sum_{c\in\{0,1\}} \sum_{j=1}^{M^c} ((1 - r_j^c) L_j^c(\theta_j^c) + r_j^c \sum_{x\in X_j^c} \log P[x|\theta_{j_i}^{t_i}])$$

- The model parameters are $\Theta = \{\{\theta_j^c\}, \{r_j^c\}, \{q_j^c\}\}$, where r_j^c and q_j^c each require one bit to specify (hence the '1' and $\delta\left(r_j^c, 1\right)$ contributions to the model complexity term).
- By contrast, t_i and j_i are hidden data assignments, not model parameters.

BIC objective: Cases for r,q variables

Each feasible joint configuration of the variables (r_j^c, q_j^c) for component j in class c corresponds to one of the three cases:

- **Case 1**: $\boldsymbol{r_j^c = 0}$, the component is formed by clean samples, and there is no need to revise this component (i.e., change in model complexity $\Delta\Omega_{j,1}^c = 0$) or re-distribute its samples (i.e., change in complete data log-likelihood $\Delta L_{j,1}^c = 0$). The change in BIC is thus

$$\Delta BIC_j^c = \Delta\Omega_{j,1}^c + \Delta L_{j,1}^c = 0$$

BIC objective: r,q variables (cont)

- **Case 2**: $\boldsymbol{r_j^c = 1, q_j^c = 0}$, the component is poisoned, and we choose to remove it, changing the model complexity term by

$$\Delta\Omega_{j,2}^c = -|\theta_j^c|\frac{1}{2}\log T$$

Each sample $x_i \in X_j^c$ is re-assigned to component j_i of class t_i, where

$$(t_i, j_i) = \underset{(t,j')\in S\setminus\{(c,j)\}}{\arg\max} P[\boldsymbol{x}_i|\theta_{j'}^t]$$

Let $Q = \{(t_i, j_i)|\forall i, x_i \in X_j^c\}$ be the components that receive the re-assigned samples. We re-estimate the parameters of each of the components $(w, j') \in Q$ on $\hat{X}_{j'}^w = X_{j'}^w \cup \{x_i \in X_j^c | t_i = w, j_i = j'\}$ by maximum likelihood estimation (MLE) and denote it as $\theta_{j'}^{w,new}$.

The total data log-likelihood changes by

$$\Delta L_{j,2}^c = -\sum_{(w,j')\in Q}\sum_{x_i\in\hat{X}_{j'}^w}\log P\left[x_i|\theta_{j'}^{w,new}\right] + \sum_{(w,j')\in Q}\sum_{x_i\in X_{j'}^w}\log P\left[x_i|\theta_{j'}^w\right] + \sum_{x_i\in X_j^c}\log P[x_i|\theta_j^c]$$

BIC objective: r,q variables (cont)

- **Case 3**: $\boldsymbol{r_j^c = 1, q_j^c = 1}$, component j in class c is poisoned, and we choose to revise it and re-distribute its samples.

Revising a component does not change the model complexity cost ($\Delta\Omega_{j,3}^c = 0$).

The parameters θ_j^c are re-estimated by MLE on its surviving samples

$$\hat{X}_j^c = \{x_i \in X_j^c | t_i = c\} \text{ and denote it as } \theta_j^{c,new}.$$

Let $Q' = \{(w, j') \in Q | w \neq c\} \cup \{(c, j)\}$ be the components to be revised.

The total data log-likelihood changes by

$$\Delta L_{j,2}^c = -\sum_{(w,j')\in Q'} \sum_{x_i \in \hat{X}_{j'}^w} \log P\left[x_i | \theta_{j'}^{w,new}\right] + \sum_{(w,j')\in Q'} \sum_{x_i \in X_{j'}^w} \log P\left[x_i | \theta_{j'}^w\right]$$

BIC-MM-TSC: BIC Minimization

- To minimize the complete data BIC objective, for each component j in class c ∈ {0,1}, choose the configuration of the parameters $(\theta_j^c, r_j^c, q_j^c)$ that reduces BIC the most (i.e., minimizes ΔBIC_j^c).
- However, the optimal configuration for any component j depends on the configurations of others.
- It is thus intractable to define an algorithm guaranteed to find a globally optimal configuration over all components.
- Instead, at each optimization step, we separately trial-update each component's configuration, and then only permanently update for the component that yields the greatest reduction in BIC.
- This is repeated until there are no further changes in BIC. This optimization approach is non-increasing in the BIC objective and results in a locally optimal solution.

Experiments for 2-class case

- TREC05 email dataset with K=2 classes: spam and ham.
- Classifers are: SVM, Logistic Regression (LR); bi-directional one-layer LSTM with 128 hidden units (neurons)
- Attack Scenarios: poison class $c \in \{0,1\}$ with mislabelled samples from class $1 - c$.
- All hyperparameters of *K*NN-D, GC-D and BIC-C-D defenses optimistically set based on a clean dataset assumed to be available to the defender - BIC-MM-TSC requires no such hyperparameters.

Experimental Results

# Poisoned Ham,Spam	0,0	0,1k	0,2k	0,3k	0,4k	0,5k	0,6k	1k,1k	1k,2k	2k,1k	2k,2k	2k,4k	4k,2k
SVM													
Poisoned	0.95	0.89	0.85	0.82	0.79	0.77	0.75	0.83	0.79	0.78	0.75	0.71	0.71
BIC-MM-TSC	**0.97**	**0.96**	**0.95**	**0.94**	**0.94**	**0.94**	**0.93**	**0.95**	**0.93**	**0.94**	**0.91**	**0.90**	**0.87**
KNN-D	0.90	0.90	0.88	0.87	0.84	0.80	0.78	0.90	0.89	0.89	0.88	0.84	0.84
GS-D	0.96	0.94	0.92	0.90	0.81	0.70	0.63	0.91	0.88	0.87	0.86	0.82	0.77
BIC-C-D	0.96	0.94	0.91	0.85	0.69	0.60	0.57	0.92	0.91	0.91	0.83	0.64	0.72
LR													
Poisoned	0.96	0.92	0.88	0.84	0.82	0.78	0.75	0.88	0.85	0.85	0.82	0.76	0.74
BIC-MM-TSC	**0.97**	**0.97**	**0.96**	**0.95**	**0.95**	**0.94**	**0.94**	**0.95**	**0.94**	**0.95**	**0.93**	**0.91**	**0.88**
KNN-D	0.91	0.91	0.90	0.88	0.85	0.81	0.78	0.92	0.90	0.90	0.90	0.86	0.87
GS-D	0.96	0.94	0.92	0.86	0.82	0.71	0.67	0.93	0.91	0.90	0.88	0.81	0.78
BIC-C-D	0.96	0.96	0.92	0.86	0.69	0.62	0.58	0.94	0.92	0.92	0.84	0.64	0.72
LSTM													
Poisoned	0.96	0.93	0.91	0.89	0.87	0.82	0.80	0.88	0.87	0.87	0.85	0.78	0.80
BIC-MM-TSC	**0.97**	**0.97**	**0.96**	**0.96**	**0.95**	**0.95**	**0.94**	**0.96**	**0.95**	**0.96**	**0.94**	**0.92**	**0.90**
KNN-D	0.93	0.93	0.92	0.89	0.87	0.85	0.80	0.93	0.91	0.90	0.91	0.89	0.88
GS-D	0.83	0.82	0.81	0.78	0.73	0.72	0.68	0.84	0.82	0.82	0.82	0.77	0.79
BIC-C-D	0.96	0.96	0.92	0.87	0.69	0.61	0.59	0.94	0.92	0.93	0.84	0.65	0.74

Table 13.1 Test set classification accuracy of victim classifiers as a function of attack strength on poisoned and sanitized TREC05 datasets. (Poisoned Ham and Poisoned Spam samples in increments of 1k=1000.)

Experimental Results (cont)

# Poisoned Ham,Spam	0,0	0,1k	0,2k	0,3k	0,4k	0,5k	0,6k	1k,1k	1k,2k	2k,1k	2k,2k	2k,4k	4k,2k
True Positive Rates (TPRs)													
BIC-MM-TSC	-	**0.89**	**0.90**	**0.90**	**0.87**	**0.90**	**0.89**	0.86	**0.87**	0.89	0.84	0.81	0.81
KNN-D	-	0.84	0.82	0.79	0.73	0.65	0.58	**0.90**	0.85	**0.91**	**0.88**	**0.84**	**0.83**
BIC-C-D	-	0.88	0.83	0.73	0.36	0.20	0.11	0.86	0.84	0.83	0.75	0.21	0.44
False Positive Rates (FPRs)													
BIC-MM-TSC	**0.018**	**0.02**	**0.08**	**0.09**	**0.06**	**0.09**	**0.07**	**0.05**	**0.06**	**0.06**	**0.07**	**0.08**	**0.11**
KNN-D	0.07	0.08	0.09	0.11	0.14	0.18	0.21	0.09	0.11	0.10	0.11	0.13	0.15
BIC-C-D	0.05	0.07	0.08	0.09	0.32	0.36	0.39	0.06	0.07	0.06	0.21	0.30	0.27

Table 13.2 TPRs and FPRs of three defenses on the TREC05 dataset under all attack cases.

# Poisoned Ham,Spam	0,0	0,1k	0,2k	0,3k	0,4k	0,5k	0,6k	1k,1k	1k,2k	2k,1k	2k,2k	2k,4k	4k,2k
# Cmps	(21,18)	(29,16)	(22,18)	(25,17)	(19,20)	(24,20)	(24,31)	(49,27)	(25,15)	(37,29)	(48,28)	(40,29)	(36,28)
# Rev Cmps	(1,5)	(0,6)	(6,11)	(5,10)	(1,16)	(2,9)	(7,11)	(19,18)	(11,7)	(17,12)	(9,7)	(14,11)	(14,13)
# Rem Cmps	(0,1)	(5,3)	(2,6)	(1,2)	(2,4)	(3,4)	(4,11)	(7,4)	(4,2)	(4,6)	(12,5)	(10,5)	(8,11)

Table 13.3 The number of components (cmps), and the number of revised (Rev) components and removed (Rem) components by BIC-MM-TSC, for each class, under all attack cases, on the TREC05 dataset.

Discussion

- Unsupervised BIC-MM-TSC outperforms other defenses even when the latters' hyperparameters are (for them) optimistically set.
- BIC false positives are close to the decision boundary so their removal actually improves accuracy.
- Similar superior performance is demonstrated in [Li et al. '22] for experiments involving datasets with more than two classes (with BIC-C-D defense replaced by SVD-D).
- Computational complexity of: BIC-MM-TSC is similar to that required to train the DNN, GC-D and SVD-D are significantly more, *K*NN-D is negligible.
- BIC-MM-TSC can act as a precursor to the training of **any** type of classifier.
- UnivBD may also detect error-generic data poisoning, see Chap. 9.

Some additional related work: De-Pois

- De-Pois [IEEE TIFS '21] employs a GAN, trained on a clean dataset assumed to be possessed by the defender (unlike BIC-MM-TSC), to produce synthetic data on which a surrogate model is trained.
- Test samples that have different predictions from the mimic model and from the target model are deemed poisoned.
- The clean dataset is assumed sufficiently large to train an accurate GAN but somehow not large enough to train an accurate classifier model from scratch.

Some additional related work: DPA

- DPA [ICLR'21,ICML'22] uses an ensemble of classifiers, each learned from a different subset of the training dataset; where
- each model has a front-end feature representation based on predicting angular rotations of the images, which is then fine-tuned using their class labels.
- The resulting features are assumed representative of the true classes.
- DPA has important hyperparameters that need setting, e.g., the number of models, sizes of the training subsets, model size (relative to that of a single model based on the whole training dataset), and training parameters (e.g., random dropout).
- DPA may give reduced accuracy when the training dataset is unpoisoned compared to a single model learned using the whole training dataset.
- Note that the poisoning rate is preserved when randomly subsampling the training dataset re. the ensemble-models element of this defense.
- Preliminary experiments show BIC-MM-TSC gives better accuracy than DPA on a held-out test set (using DPA code provided on GitHub) with a 5-model ensemble.

References for BIC-MM-TSC

- X. Li, D.J. Miller, Z. Xiang, G. Kesidis. A BIC based Mixture Model Defense against Data Poisoning Attacks on Classifiers. http://arxiv.org/abs/2105.13530
- Shorter conference version in Proc. IEEE MLSP 2023.