Supplementary Material for Poison is Not Traceless: Black-Box Detection of Poisoning Attacks

No Author Given

No Institute Given

To ensure reproducibility, we provide additional details for the paper in this section. We include the full list of complexity measures, details on FALFA (including its time complexity), the hardware and software configurations for our experiments as well as detailed dataset descriptions and additional results.

Full List of Complexity Measures

The full set of measures is listed in Table 1 We also include the *Standard Deviations* (SDs) when possible, which are not listed in the table. These measures include F1, N2, N3, N4, T1, and Hubs. As a result, there are 28 measures in total.

Details of FALFA

We obtain the multiplier λ by generalizing all the combinations between y_i and y'_i in a binary classification task, as shown in Table 2.

Time Complexity of FALFA

FALFA is more computationally efficient than ALFA and PoisSVM by a substantial margin. Linear programming is an exponential-time algorithm, the time complexity is around $O(n^{2.5})$. Xiao *et al.*'s ALFA creates a copy of $\mathcal{Y}_{\text{tr}}$ in the linear programming step, so n is essentially doubled. Paudice *et al.*'s ALFA on NN is slower than Xiao *et al.*'s, since it traverses all combinations of $\mathcal{Y}_{\text{tr}}$ instead of using linear programming. FALFA uses linear programming but without doubling $\mathcal{Y}_{\text{tr}}$, resulting in an approximately $2^{2.5} \approx 5.6$ times faster than ALFA on each iteration. Our test shows that FALFA converges at 2 iterations on average, but ALFA takes more than 5 iterations to converge. In the worst-case scenario, FALFA poisons the CMC dataset in 22.4 ± 8.6 secs, while ALFA requires 405.8 ± 348.4 secs, and PoisSVM took over 2 hours. We observe the minimal difference on Breastcancer, where FALFA completes the task at 5.3 ± 1.9 secs, and it takes ALFA 7.4 ± 5.6 secs.

Table 1: List of measures in C-Measures. If possible, Standard Deviations (SDs) of measures are also included, but are not listed.

Category	Acronym	Description
Feature-based	F1	Maximum Fisher's discriminant ratio
	F1v	Directional-vector maximum Fisher's discriminant ratio
	F2	Volume of overlapping region
	F3	Maximum individual feature efficiency
	F4	Collective feature efficiency
Linearity	L1	Sum of the error distance by linear programming
	L2	Error rate of the linear SVM classifier
	L3	Non-linearity of the linear SVM classifier
Neighborhood	N1	Fraction of borderline points
	N2	Ratio of intra/extra class nearest-neighbors distance
	N3	Error rate of nearest-neighbors classifier
	N4	Non-linearity of nearest-neighbors classifier
	T1	Fraction of hyperspheres covering data
	LSC	Local Set average Cardinality
Network	Density	Average density of the network
	ClsCoef	Clustering Coefficient
	Hubs	Hub score – Number of connections each node has
Dimensionality	T2	Average number of features per dimension
	T3	Average number of PCA dimensions per points
	T4	Ratio of the PCA dimension to the original dimension
Class Imbalance	C1	Entropy of classes proportions
	C2	Imbalance ratio

Hardware and Software Configurations

All experiments are conducted on a workstation with the following configurations:

Table 2: All combinations for $|y_i' - y_i|$. By introducing a multiplier λ, $\lambda \cdot (y_i' - y_i)$ is equivalent to $|y_i' - y_i|$.

y_i	y_i'	$\|y_i' - y_i\|$	λ
0	0	0	1
0	1	1	1
1	0	-1	-1
1	1	0	-1

- CPU: AMD Ryzen 9 5900 24 threads @ 4.4GHz
- GPU: Nvidia GeForce RTX 3090 24GB
- Memory: 64GB
- Operating System: Ubuntu 20.04.3 LTS
- Software: Python 3.8.10, PyTorch 1.10.1+cu113, scikit-learn 1.0.2

The baseline data poisoning attacks are obtained from *Adversarial Robustness Toolbox* (ART) 1.9.1 [1] and *Secure and Explainable Machine Learning in Python* (SecML) 0.15 [2].

The execution time mentioned in the paper is evaluated using the environment above.

Datasets

Real-World Datasets. All datasets are obtained from the UCI Machine Learning Repository [3]. We apply standardization on all datasets during the preprocessing.

For multi-class classification tasks, we convert the dataset into binary based on the following:

- **Abalone:** If the ‘Rings’ attribute is less than 10, we assign the example to the negative class; Else, assign to the positive class. We exclude the categorical attribute – ‘Sex’ and the output label – ‘Rings’ from the inputs.
- **CMC:** has 3 output classes: 1. No-use, 2. Long-term, and 3. Short-term. If the class is ‘No-use‘, assign it to the negative class; Else, to the positive class.
- **Texture:** It has 10 output classes. We use a subset which contains examples labeled as ‘3’ and ‘9’. If the class is ‘3’, assign it to the negative class; Else, to the positive class.
- **Yeast:** It has 10 output classes. We select ‘’0 and ‘7’, the top two classes sorted by sample size. If the class is ‘0’, assign it to the negative class; Else, to the positive class.

[1] Source: https://github.com/Trusted-AI/adversarial-robustness-toolbox

[2] Source: https://github.com/pralab/secml

[3] Source: https://archive.ics.uci.edu/ml

Synthetic Datasets. Table 3 shows the parameters we used to generate synthetic datasets.

Table 3: Hyper-parameters for generating synthetic data

Parameter	Range
Sample size	$\{1000, 1001, \ldots, 2000\}$
# of informative features	$\{4, 5, \ldots, 30\}$
# of redundant features	$\{0, 1, \ldots, 5\}$
Class ratio	$[0.4, 0.6]$

Additional Results

Classifiers' Performance. Table 4 shows the performance of classifiers trained on poison-free data.

Table 4: Summary of the training set size (n), number of features (m), Positive Label Rate (PLR), average accuracy (%) for training and test sets across all classifiers, and difficulty (Easy/Normal/Hard).

Dataset	n	m	PLR	Train Acc.	Test Acc.	Diff.
Abalone	1600	7	0.50	79.9 ± 0.7	76.5 ± 0.5	N
Australian	552	14	0.45	91.5 ± 3.1	81.9 ± 2.1	N
Banknote	1097	4	0.44	100.0 ± 0.0	100.0 ± 0.0	E
Breastcancer	455	30	0.63	99.3 ± 0.2	95.0 ± 2.5	E
CMC	1178	9	0.77	79.9 ± 2.8	77.5 ± 0.6	N
HTRU2	1600	8	0.50	94.8 ± 0.5	92.6 ± 0.9	E
Phoneme	1600	5	0.50	89.7 ± 6.3	85.6 ± 1.3	N
Ringnorm	1600	20	0.50	99.4 ± 0.4	97.8 ± 1.1	E
Texture	800	40	0.50	100.0 ± 0.0	99.8 ± 0.5	E
Yeast	713	8	0.48	73.5 ± 4.7	65.8 ± 1.6	H

C-Measures on clean and poisoned data. When no poisoning attack is present, the C-Measures strongly correlate to the classifier's test accuracy as can be seen in Fig. 2a. When the dataset has been poisoned, the C-Measures react to it. Despite a performance drop on the training accuracy of $1.0 \pm 5.6\%$, we observe a correlation drop across all measures when measuring on poisoned data, as shown in Fig. 2b. This matches the test accuracy drop, which indicates the data becomes more complex, despite only minor changes in the training accuracy.

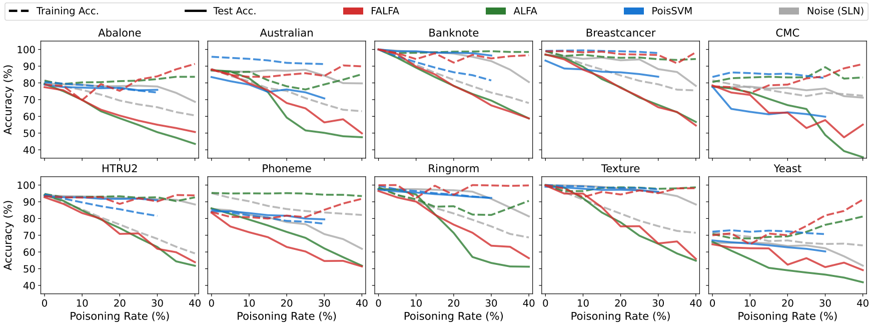

Fig. 1: Train and test accuracy at various poisoning rates when classifiers under SLN, PoisSVM, and FALFA attacks.

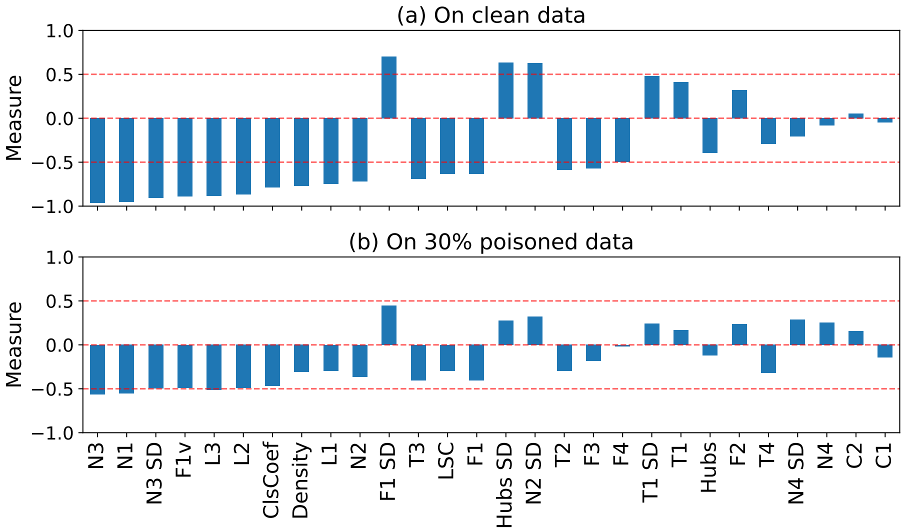

Fig. 2: (a): Correlation of each measure in C-Measures to the test accuracy on synthetic datasets without poisoning attacks. (b): Same correlation but measured on the datasets containing 30% poisoning examples.

Performance Loss. Fig. 1 shows the performance loss on all real datasets.

Performance Loss at a Low Poisoning Rate. Here, we present the performance loss at a low poisoning rate (10%) in Table 5. This is the test accuracy difference before and after the attack. PoisSVM has no meaningful impact (< 2%) on the classifiers' performance in 7 out of 10 datasets. Meanwhile, SLN leads to minor performance improvement on CMC and Yeast.

Table 5: Performance loss (%) after attacked by a poisoning attack with 10% poisoning rate.

Dataset	SLN	PoisSVM	ALFA	Falfa
Abalone	0.8 ± 0.7	1.8 ± 0.8	9.5 ± 1.9	7.7 ± 1.7
Australian	0.7 ± 0.5	4.5 ± 3.9	4.9 ± 4.0	8.3 ± 3.8
Banknote	1.4 ± 2.3	1.1 ± 1.1	10.9 ± 2.5	10.3 ± 2.9
Breastcancer	2.5 ± 0.7	5.3 ± 4.6	7.2 ± 2.0	9.1 ± 2.7
CMC	-0.2 ± 0.7	15.1 ± 4.7	3.5 ± 3.0	5.7 ± 3.3
HTRU2	0.7 ± 0.3	0.7 ± 1.3	9.2 ± 3.1	9.4 ± 2.4
Phoneme	3.5 ± 2.9	0.9 ± 2.1	6.8 ± 0.7	11.6 ± 2.1
Ringnorm	0.1 ± 0.3	1.7 ± 0.5	3.2 ± 2.5	6.4 ± 2.9
Texture	0.5 ± 1.1	1.2 ± 0.8	7.9 ± 4.6	4.9 ± 3.9
Yeast	-0.2 ± 1.6	1.9 ± 3.8	10.4 ± 4.9	2.3 ± 4.6

Receiver Operating Characteristic (ROC) Curves. Fig. 3 shows the ROC curves for both real and synthetic data.

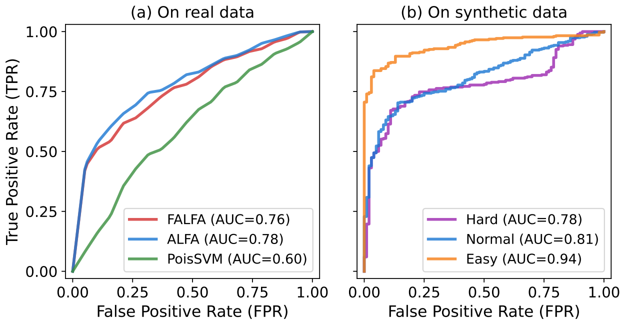

Fig. 3: ROC curve for Diva's prediction on whether the training set is poisoned. (a): Unseen real datasets poisoned by FALFA, ALFA, and PoisSVM. ALFA and PoisSVM are unknown attacks to Diva. (b): Synthetic datasets poisoned by FALFA, and grouped by the datasets' difficulty.