Mitigating Poisoning Attacks on Machine Learning Models: A Data Provenance Based Approach

Reviewed by : Aditya Jain & Manish Reddy

Outline

Introduction to causative attacks

The paper considers a causative attack model a.k.a Poisoning (Barreno et al., 2010), which can be thought of as a game between two players: ***the defender (who seeks to learn a model Θ)****, and the* ***adversary (who wants to reduce the performance of the model)****.*

The scenario is particularly challenging in **online learning** where the model is **periodically retrained** to learn new behavior from dataset shifts.

Eg. Microsoft's AI chatbot Tay, which learned to be racist and offensive from twitter users

Novelty of the approach - Data Provenance

Provenance: “the beginning of something's existence; something's origin”

Provenance Data: Meta-data specifying the origin of the input data. Eg. for a tweet, the provenance data includes the twitter account, time of tweet etc.

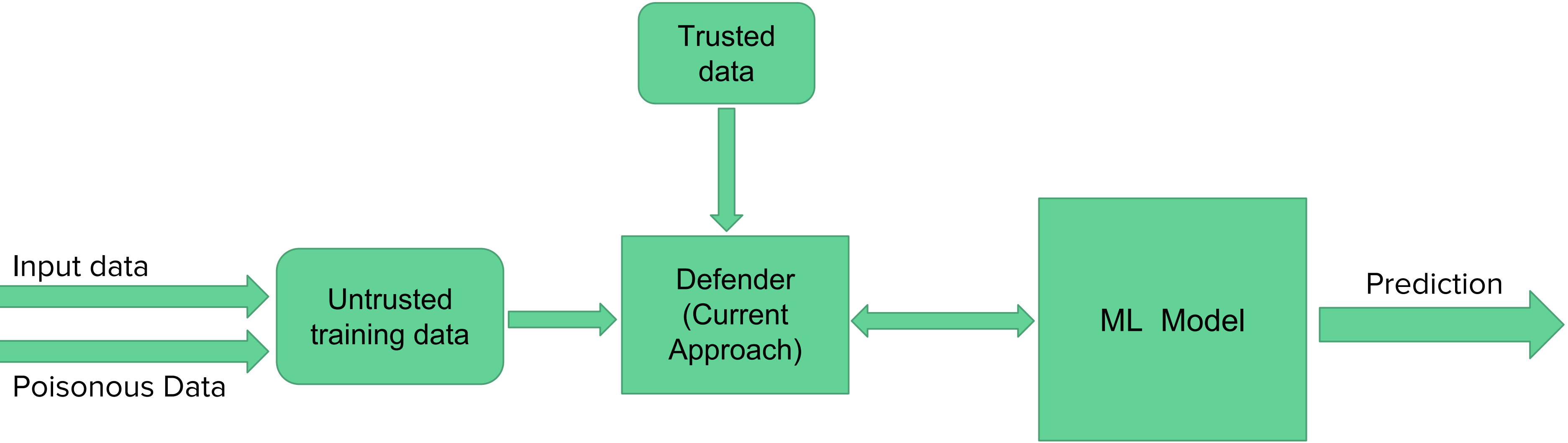
Provenance Signature: Value of a feature of a provenance data. Eg. the feature could be the *twitter account id* with provenance signature as *actual id(xyz@gmail.com)*

Data Segment of Signature *i* : All input data points sharing the same provenance signature. Eg. All tweets from the same twitter account.

Underlying Assumptions

- The sole aim of the Adversary is to try and reduce model accuracy
- Provenance data is available for each data point and the adversary cannot tamper with it
- Adversary has access to the dataset and can generate similar looking datasets
- Adversary can only modify data points sharing a certain provenance signature
- Input data sharing a *provenance* signature have highly correlated likelihood of being poisoned

Overview



Based on whether trusted data is available or not, there are two variants of the current Provenance based approach

Algorithms

CASE 1

Partially Trusted Data Stream

CASE 2

Fully Untrusted Data Stream

Algorithm 1 - Partially Trusted Data

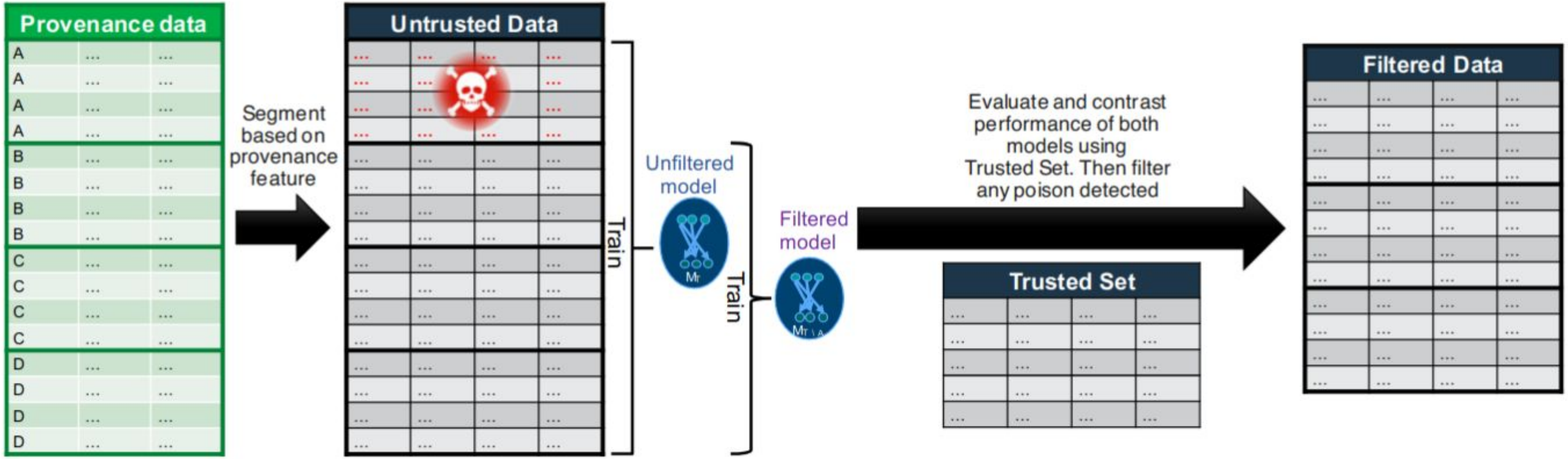
- Assumption: Poisoned devices can be clustered based on provenance data.

- Idea: If we can cluster them, then a model trained with a poisoned cluster included in the dataset will perform worse than a model trained without.

- We can use the trusted part of the dataset to measure ‘performance’

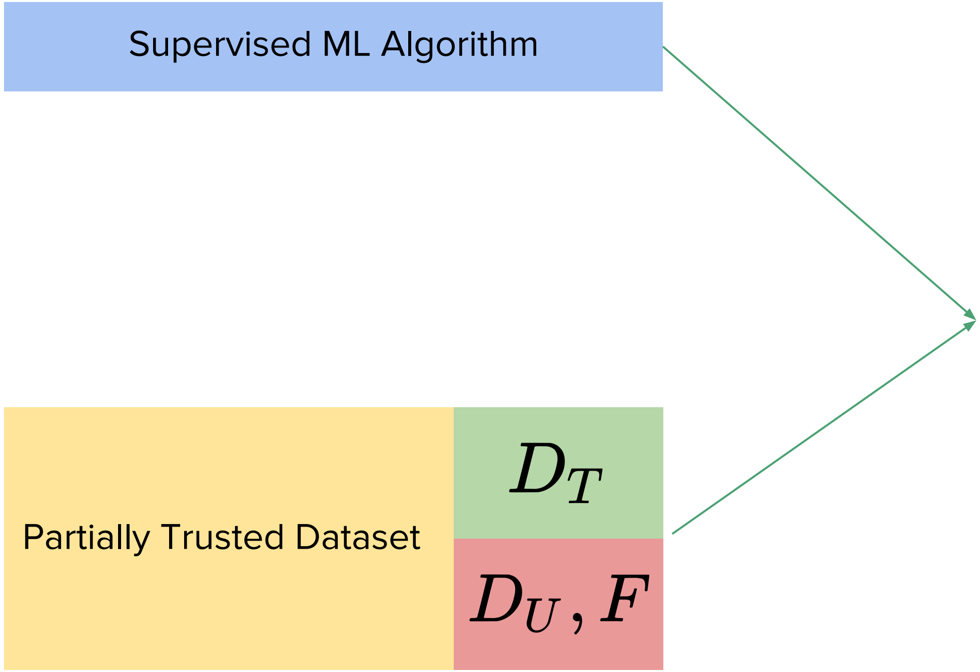
- Approach: Cross-Validation like!

Algorithm 1: Partially Untrusted Data



- Very Similar to Cross-Validation!

Algorithm 1: Details



Algorithm 1 findPoisonDataPartiallyTrusted(D, D_T, $\mathcal{F}$)

Input: D := all data points, D_T := trusted data points (trusted set), $\mathcal{F}$:= Provenance feature to be used for segmentation

Output: Set of data points that are suspected of being poisonous.

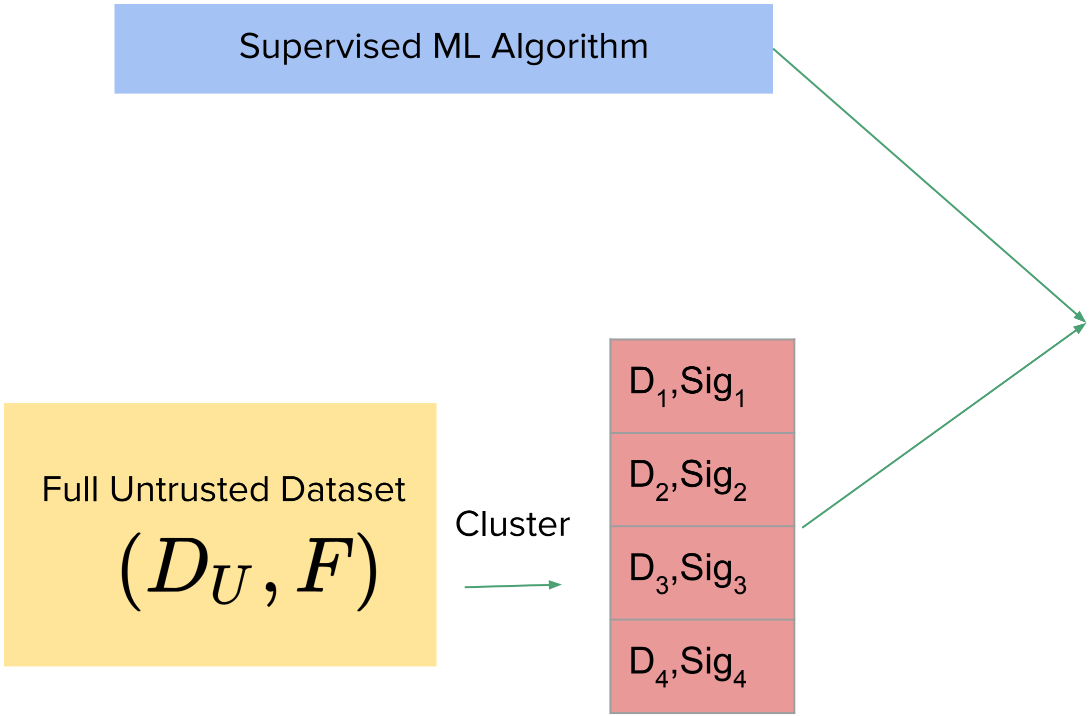
```
D_poisoned ← ∅
D_U ← D \ D_T {Untrusted data}
F ← segmentByProvenanceFeature(D_U, 𝓕)
for all ⟨D_i, Sig_i⟩ ∈ F do
    Model_filtered ← trainModel(D_U \ D_i)
    Model_unfiltered ← trainModel(D_U)
    if performance(Model_unfiltered, D_T) < performance(Model_filtered, D_T) ) then
        D_poisoned ← D_poisoned ∪ ⟨D_i, Sig_i⟩ {Flag as suspicious}
        D_U ← D_U \ D_i {Remove from training set}
    end if
end for
return D_poisoned
```

Algorithm 2: Fully Untrusted Data

- Problem: No Trusted Set for measuring performance.

- Idea: Measure on the entire dataset, and *pray* that it works as a good proxy.

Algorithm 2: Details



Algorithm 2 findPoisonDataFullyUntrusted($D_U, \mathcal{F}$)

Input: D_U := all data points (all are untrusted), $\mathcal{F}$:= Provenance feature to be used for segmentation

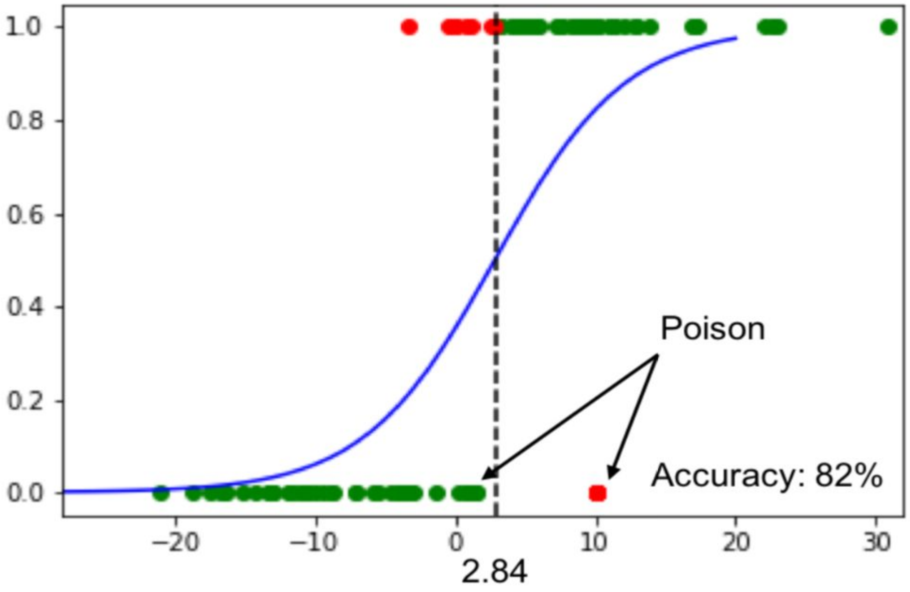
Output: Set of data points that are suspected of being poisonous.

$D_{poisoned} \leftarrow \emptyset$
$F \leftarrow$ segmentByProvenanceFeature($D_U, \mathcal{F}$)
$F_{train} \leftarrow \emptyset$, $F_{eval} \leftarrow \emptyset$
for all $\langle D_i, Sig_i \rangle \in F$ **do**
  Randomly assign half of the data in D_i to F_{train} and half to F_{eval}
end for
for all $\langle D_i, Sig_i \rangle \in F_{train}$ **do**
  $Model_{filtered} \leftarrow$ trainModel($D_{train} \setminus D_i$)
  $Model_{unfiltered} \leftarrow$ trainModel(D_{train})
  $\langle D_{eval_i}, Sig_i \rangle \leftarrow$ getSegment(F_{eval}, Sig_i)
  $D_{filteredEval} \leftarrow D_{eval} \setminus D_{eval_i}$
  if performance($Model_{unfiltered}, D_{filteredEval}$) $<$ performance($Model_{filtered}, D_{filteredEval}$) **then**
    $D_{poisoned} \leftarrow D_{poisoned} \cup \langle D_i, Sig_i \rangle$ {Flag as suspecious}
    $D_{train} \leftarrow D_{train} \setminus D_i$ {Remove from training set}
    $D_{eval} \leftarrow D_{eval} \setminus D_{filteredEval}$ {Remove from validation set}
  end if
end for
return $D_{poisoned}$

Toy Example: Logistic Regression

Legitimate data: 200 pts of normal(0,10) ,   y: $P(y_i = 1 \mid x_i)$: $1/(1+\exp(-x_i))$

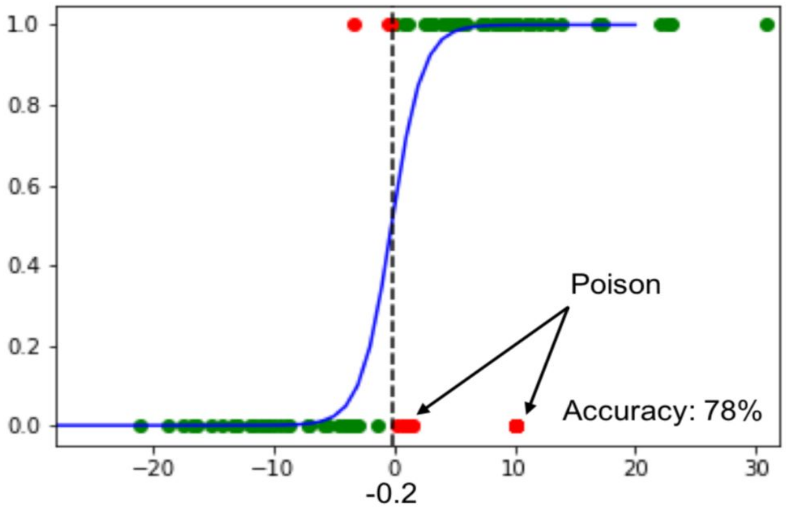
Poisoned Data : x = 10, y=0 (20 pts) & x=1, y=0 (20 pts)

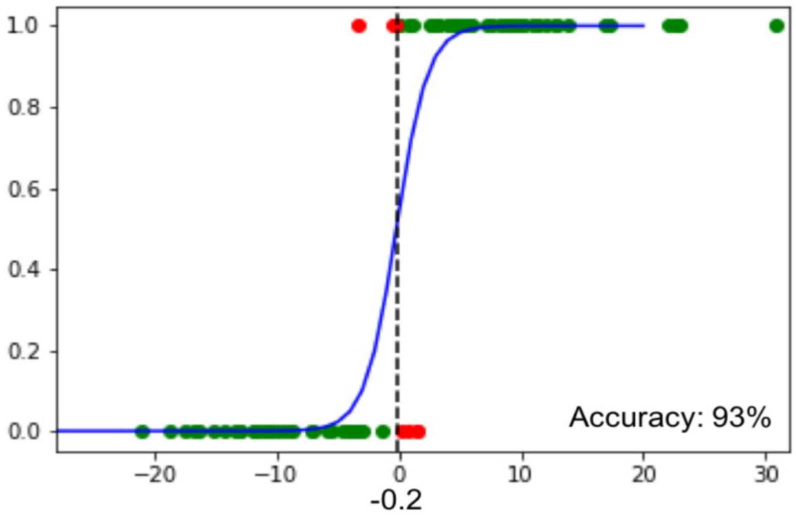


Adversary can poison evaluation process

Retraining without poisoned dataset shifts the mean to 0.2 but reduces accuracy and thus will not be flagged as poisoned

Removing poisoned dataset from both the training and evaluation dataset, mean is 0.2 and accuracy increases. Thus poisoned data will be flagged



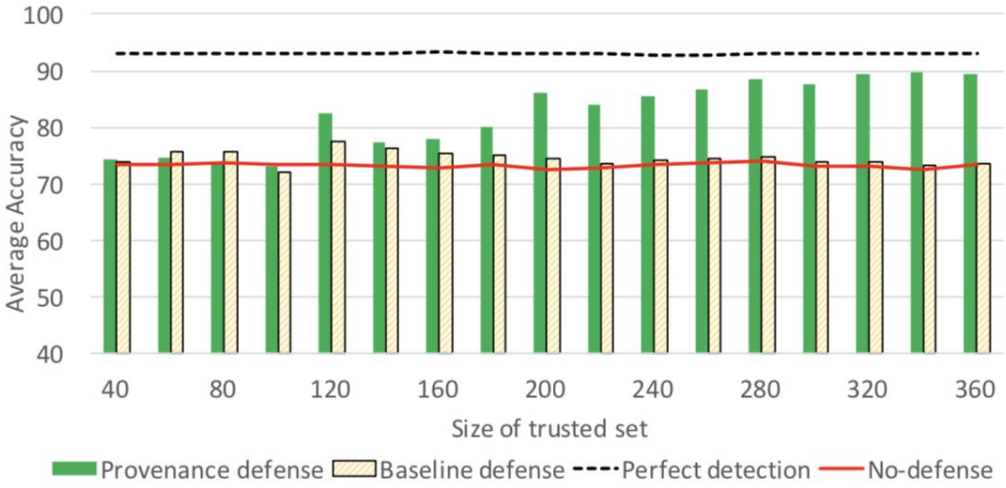


Experiments

Experiment 1: Setting

- Poisoned data - Synthetic data with two features and two classes
    - Attack factor: 0 no attack, 1 aggressive attack (0.5 here)
    - Separation: difference between the poisoned data and actual data (small separation here)
- Compromised devices give compromised data points while honest devices give honest data points.
- Different Comparisons :
    - **Perfect model** : trained on legitimate data points
    - **No defense model**: trained on all data points
    - **Baseline defense**: trained on data points after filtering through RONI
    - **Provenance model**: trained on data points after filtering through current approach
- **Baseline**: Calibrated Reject on Negative Impact (RONI)
    - Similar to the current approach using individual points instead of segments

Effect of Increasing Trusted set size



	Poisoned Data	Non Poisoned Data
Trusted data	0	Varying
Untrusted data	200	800

Increasing the size of trusted data, increases the accuracy of the identifying step for poisoned signatures

performance(model$_{unfiltered}$, trusted_dataset) < performance(model$_{filtered}$, trusted_dataset)
and thus increases accuracy

After a point, the increase plateaus

20% drop in accuracy without any poison defense

Effect of number of compromised devices

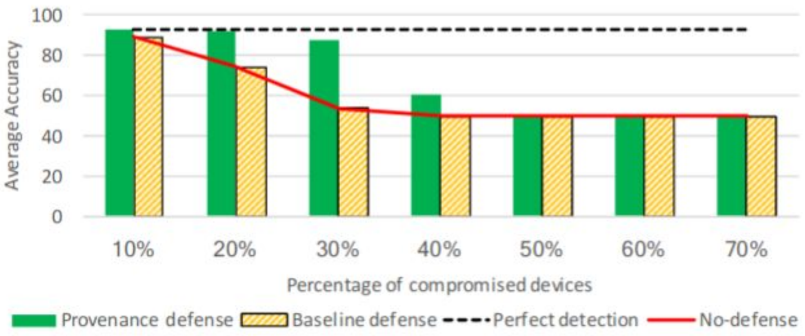


Figure 4: Effect of increasing the percentage of compromised devices on the average accuracy achieved under poison I.

	Poisoned Data	Non Poisoned Data
Trusted data	0	Fixed
Untrusted data	Increase by p%	Decreased by p%

Increased poisoning leads to decreased accuracy in the model, thresholded to be the lowest at random chance.

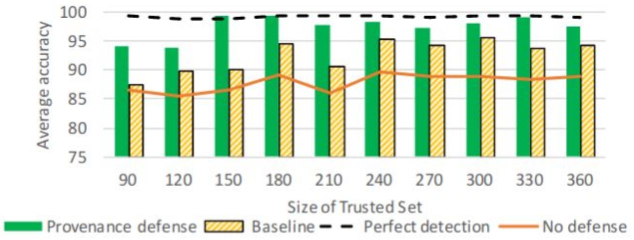
Effect of number of datapoints contributed per device:

		Average Accuracy				
Data points per device	%Devices compromised	Perfect detection	No-defense	Provenance defense	Baseline defense	Average Improvement
10	10%	87.32	68.44	80.18	73.47	8%
	20%	90.47	50.14	75.36	50.58	33%
	30%	88.84	50.00	66.47	50.00	25%
	40%	85.34	50.00	67.23	50.00	26%
	50%	84.61	50.00	67.01	50.00	25%
	60%	78.85	50.00	57.09	50.00	12%
	70%	76.90	50.00	50.00	50.00	0%
50	10%	93.06	85.79	83.43	89.04	-7%
	20%	92.98	62.09	72.84	65.91	10%
	30%	92.64	50.15	73.02	50.62	31%
	40%	92.70	50.00	73.84	50.00	32%
	50%	92.47	50.00	83.25	50.00	40%
	60%	92.38	50.00	72.79	50.00	31%
	70%	91.36	50.00	56.29	50.00	11%
70	10%	92.87	87.82	87.99	90.09	-2%
	20%	92.97	67.56	79.18	72.76	8%
	30%	92.97	51.01	72.84	52.17	28%
	40%	92.85	50.00	76.03	50.02	34%
	50%	92.63	50.00	71.97	50.00	31%
	60%	92.45	50.00	68.98	50.00	28%
	70%	92.56	50.00	59.77	50.00	16%

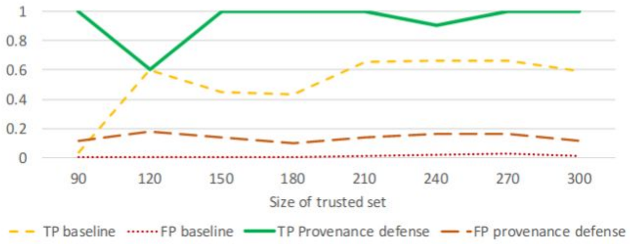
Experiment Setup 2

- Dataset: MNIST
- Classifier Model: SVM
- Poisoning method: Gradient Ascent

- Similar trends to previous experimental setup.



(a) Average accuracy



(b) Average true positive rate (i.e. recall) and false positive rate (i.e. fall-out)

Figure 6: Effect of increasing the size of the trusted set for poison II

Conclusions

- Many assumptions - simple methods.
- Specific experiments - limited conclusions.
- Probably the best paper to Critique.
- Critique Team: Go have a field day!