Securing Federated Sensitive Topic Classification against Poisoning Attacks

Tianyue Chu[1]
Alvaro Garcia-Recuero[1]
Costas Iordanou[2]
Georgios Smaragdakis [3]
Nikolaos Laoutaris[1]

[1] IMDEA Networks Institute
[2] Cyprus University of Technology
[3] TU Delft

[Developing the Science of Networks]

Background

(a) Zone: European Union

(b) Zone: California (USA)

❑ Legislations (GDPR)

- define sensitive data : related to health, political opinions, religious beliefs, sexual orientation and racial ethnic origin
- ensure data protection and safeguard online content that contains sensitive data

Background

(a) Zone: European Union

(b) Zone: California (USA)

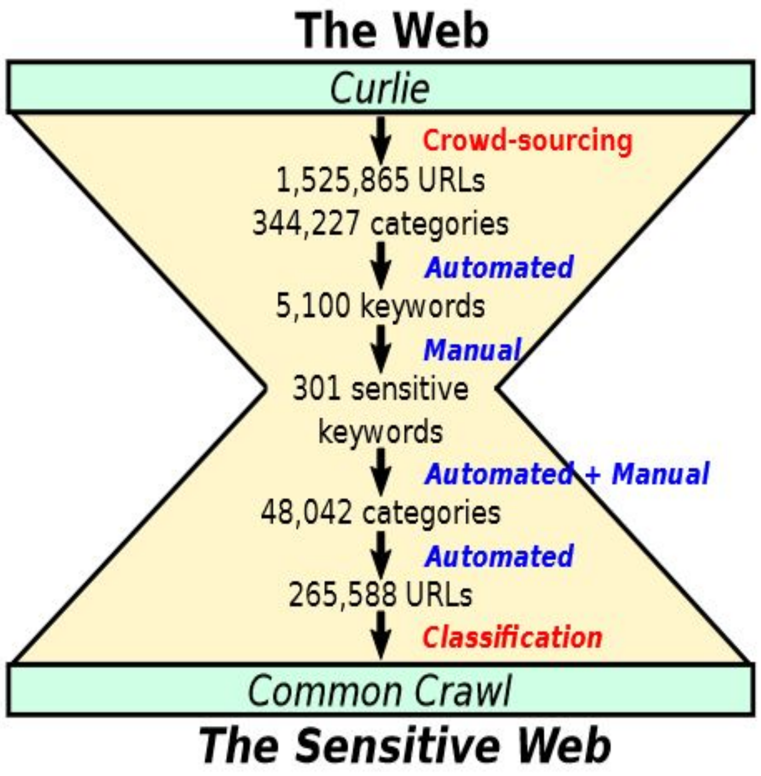
Legislations (GDPR)

- define sensitive data : related to health, political opinions, religious beliefs, sexual orientation and racial ethnic origin
- ensure data protection and safeguard online content that contains sensitive data

How to detect whether the content of a URL relates to any of the sensitive data?

Previous Work

- **A classifier for detecting GDPR sensitive data [2]**
  - Defined on GDPR sensitive categories
  - train a <u>centralized classifier</u> to detect whether the content of a URL relates to sensitive categories
  - trained using 156k sensitive URLs from Curlie with <u>over 90% accuracy</u>



[1] Matic, Srdjan, Costas Iordanou, Georgios Smaragdakis, and Nikolaos Laoutaris. "Identifying sensitive urls at web-scale." In *Proceedings of the ACM Internet Measurement Conference*, pp. 619-633. 2020.

Motivation

- Why do we need to identify sensitive URLs
  - 30% of sensitive URLs are hosted in domains that fail to use HTTPS.
  - In sensitive web pages with third-party cookies, 87% of the third-parties set at least one persistent cookie.

Motivation

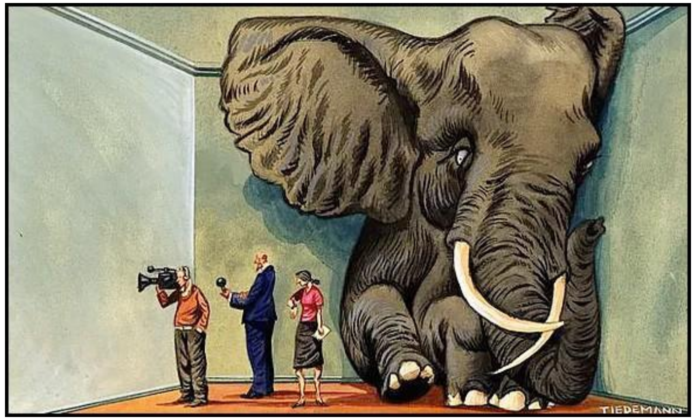
- Why do we need to identify sensitive URLs
  - 30% of sensitive URLs are hosted in domains that fail to use HTTPS.
  - In sensitive web pages with third-party cookies, 87% of the third-parties set at least one persistent cookie.

Being tracked when visiting web pages that contain sensitive content

Motivation

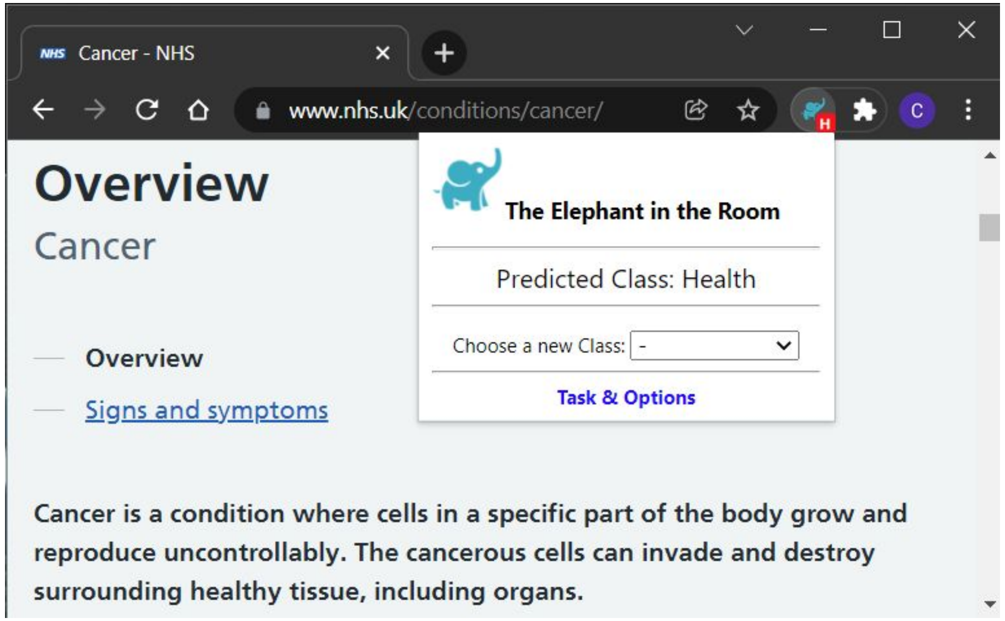
- Why do we need to identify sensitive URLs
  - 30% of sensitive URLs are hosted in domains that <u>fail to use HTTPS</u>.
  - In sensitive web pages with third-party cookies, <u>87% of the third-parties </u>set <u>at least one</u> persistent cookie.

the "Elephant in the Room" of privacy



Motivation

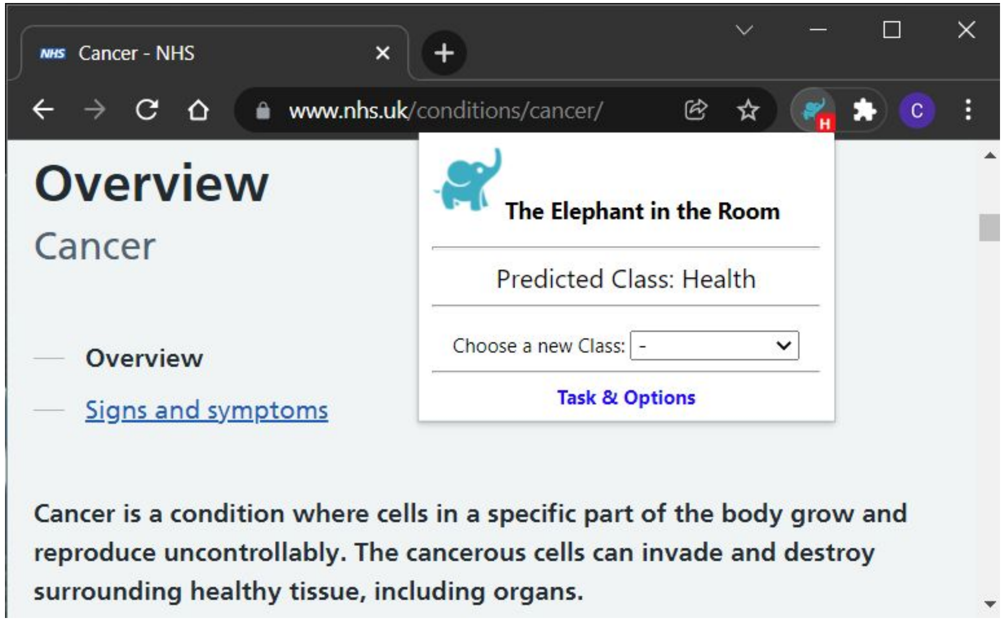
- What do we want to do?



A classifier capable of detecting URLs containing sensitive content in real-time

Motivation

- What do we want to do?

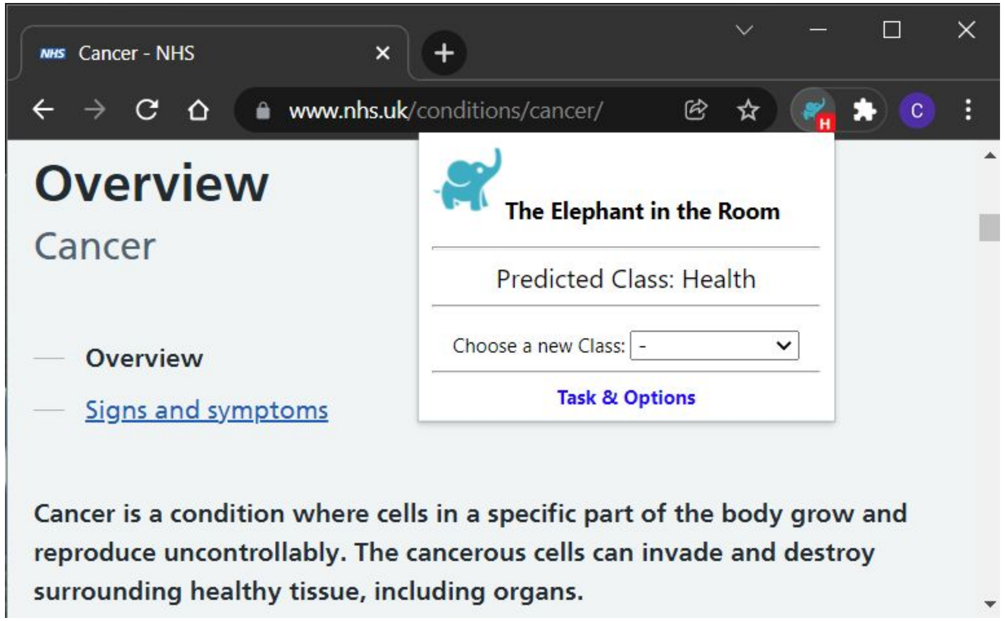


A classifier capable of detecting URLs containing sensitive content in <u>real-time</u>

when you visit a webpage, it pops up which category this webpage falls in

Motivation

- What do we want to do?

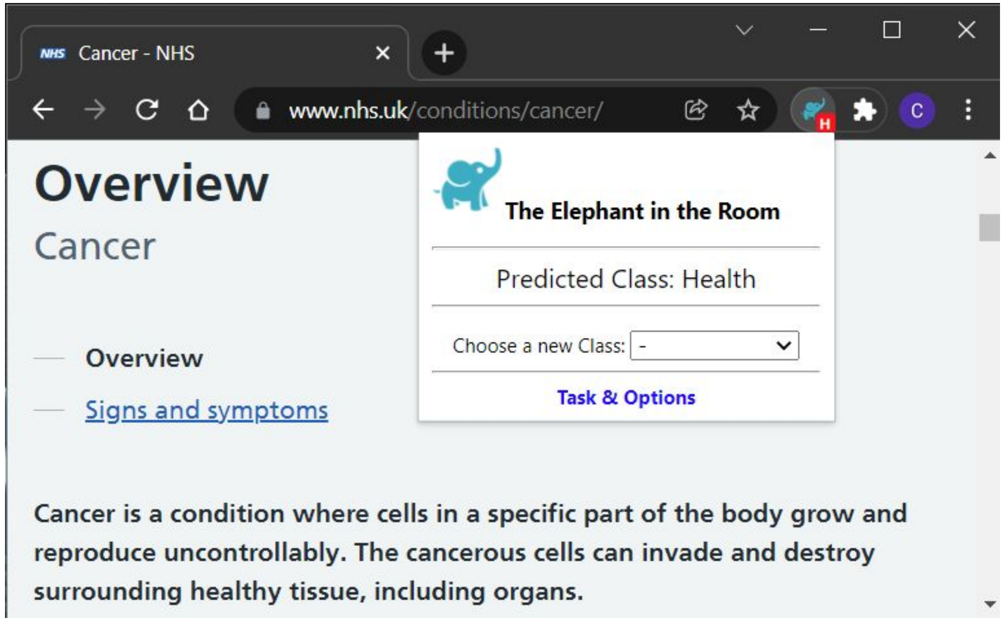


A classifier capable of detecting URLs containing sensitive content in <u>real-time</u>

when you visit a webpage, it pops up which category this webpage falls in

Motivation

- What do we want to do?

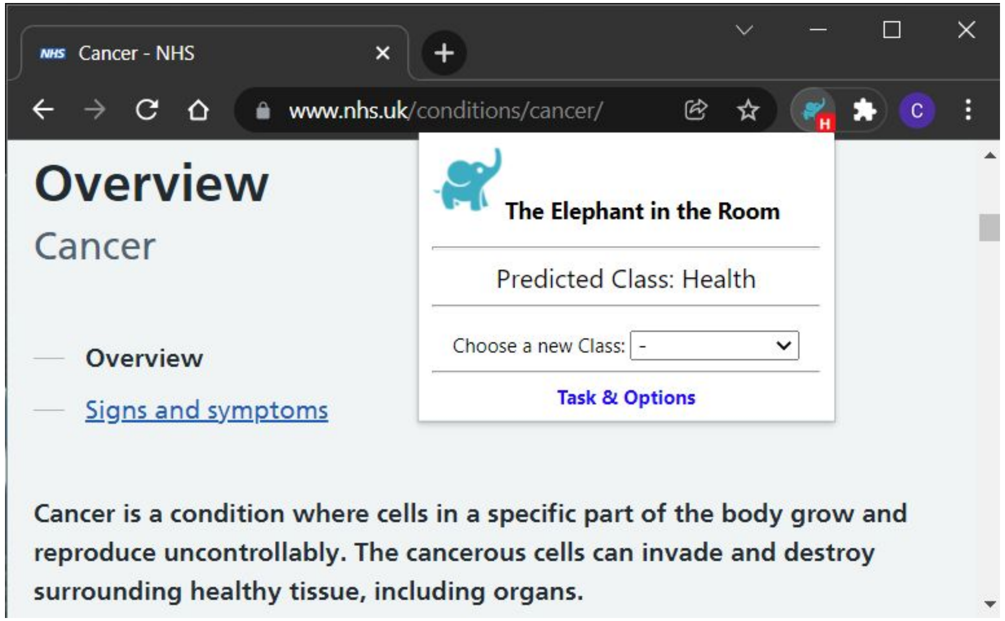


when you visit a webpage, it pops up which category this webpage falls in

A classifier capable of detecting URLs containing sensitive content in <u>real-time</u>

Motivation

- What do we want to do?



when you visit a webpage, it pops up which category this webpage falls in

A classifier capable of detecting URLs containing sensitive content in real-time

Limitations

- **Limitations of centralised classifier**
  - tied to a fixed training set: difficult to quickly cover labels related to **yet unseen** sensitive content
  - being centralized: cannot be used to drive a **privacy-preserving** **distributed** classification system

Limitations

- **Limitations of centralised classifier**
  - tied to a fixed training set: difficult to quickly cover labels related to yet unseen sensitive content

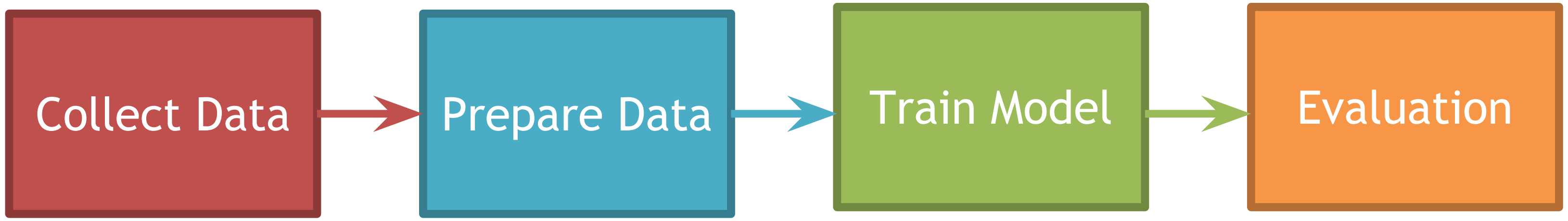
  - being centralized: cannot be used to drive a privacy-preserving distributed classification system

institute IMdea networks

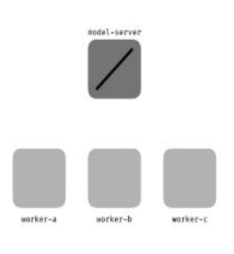
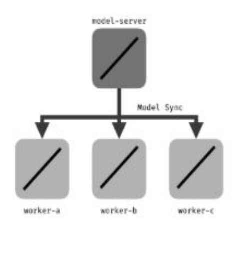
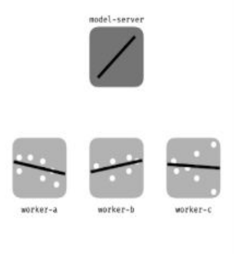
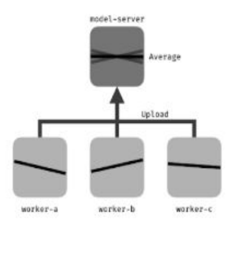
Limitations

- **Limitations of centralised classifier**
  - tied to a fixed training set: difficult to quickly cover labels related to yet unseen sensitive content

  - being centralized: cannot be used to drive a privacy-preserving distributed classification system

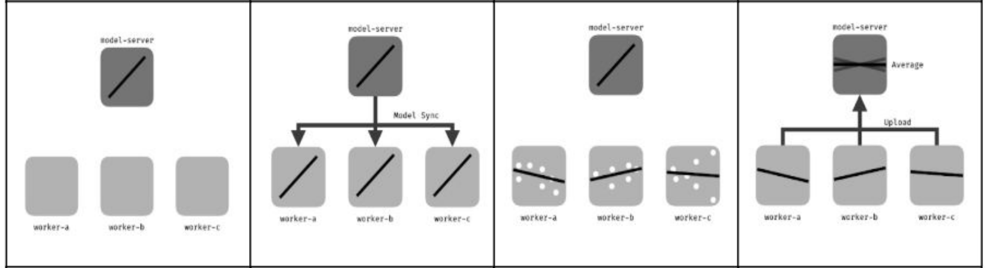


Solution: Federated learning (FL) [3]

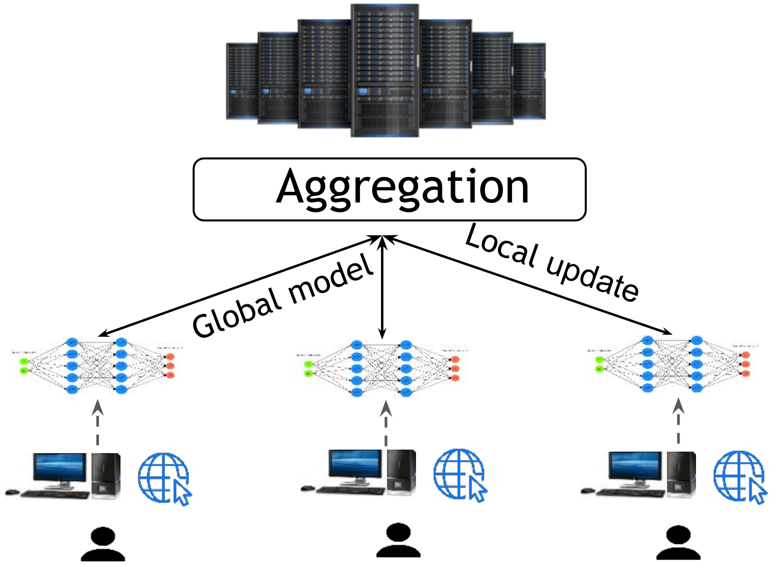
Step 1	Step 2	Step 3	Step 4
			
Central server chooses a statistical model to be trained	Central server transmits the initial model to several nodes	Nodes train the model locally with their own data	Central server pools model results and generate one global mode without accessing any data

[3] McMahan, Brendan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. "Communication-efficient learning of deep networks from decentralized data." In *Artificial intelligence and statistics*, pp. 1273-1282. PMLR, 2017.

Solution: Federated learning (FL) [3]



Step 1	Step 2	Step 3	Step 4
Central server chooses a statistical model to be trained	Central server transmits the initial model to several nodes	Nodes train the model locally with their own data	Central server pools model results and generate one global mode without accessing any data



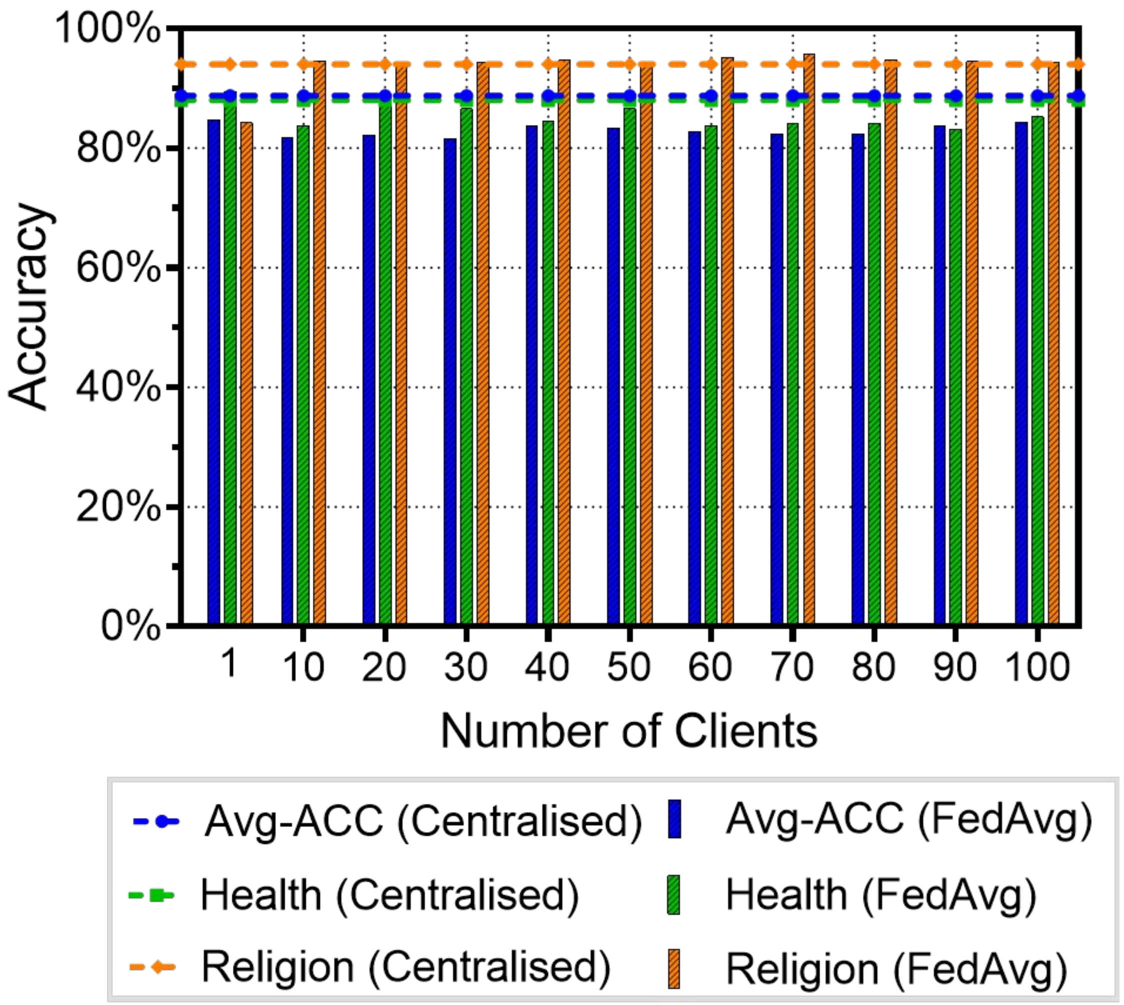
- Up-to-date: continuously learn from real-time web data gathered by users
- Being distributed with privacy: users train the classifier locally using personal data

[3] McMahan, Brendan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. "Communication-efficient learning of deep networks from decentralized data." In *Artificial intelligence and statistics*, pp. 1273-1282. PMLR, 2017.

FL Framework

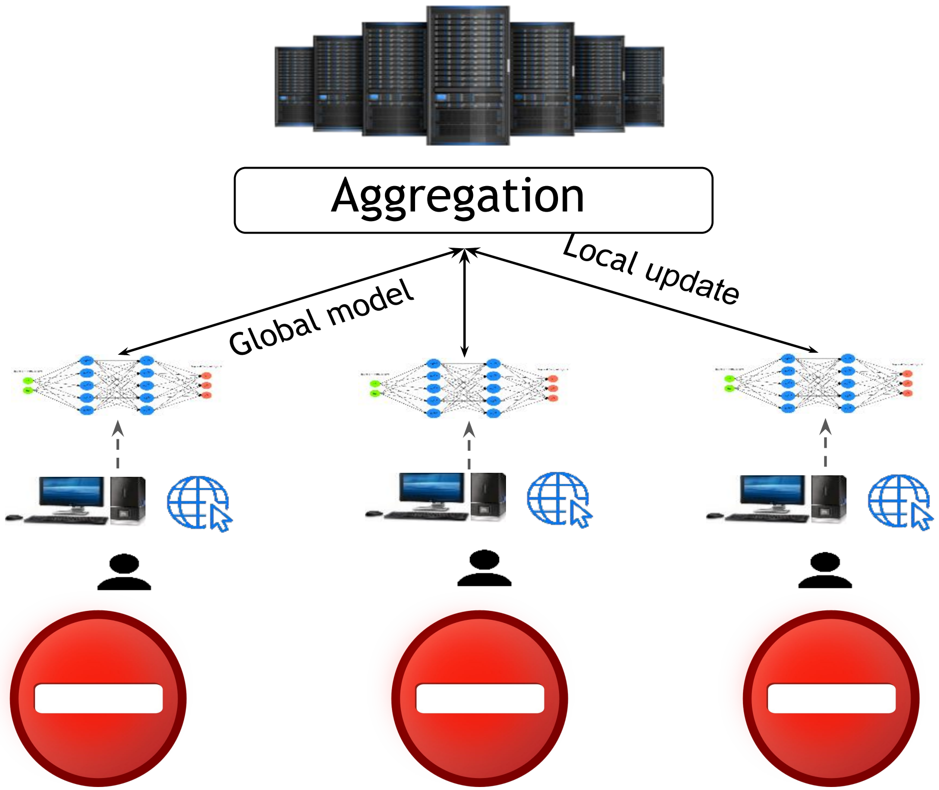
- Centralized classifier
  - Naïve-Bayes
- FL-based classifier
  - A simple neural network
  - FedAvg [3]
  - Performance

Accuracy	FL	Centralized
Health	85%	88%
Religion	93%	94%

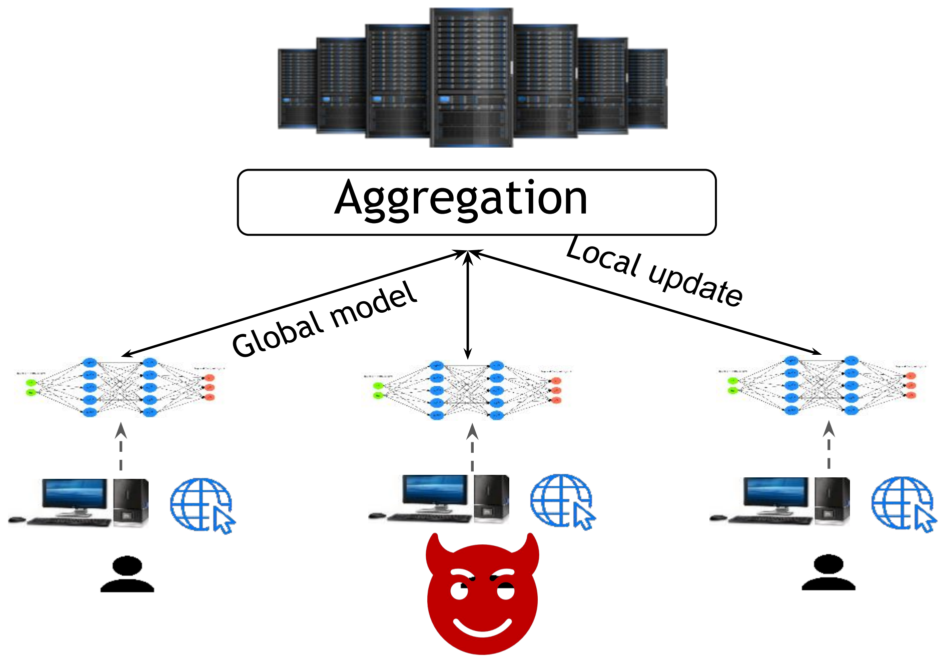


Federated learning

- **Federated Learning (FL) is vulnerable**



- **Federated Learning (FL) is vulnerable**



FL is vulnerable to so-called poisoning attacks

Byzantine-robust FL methods

- Several Byzantine-robust defense methods have been developed
- Recent studies [4], [5], [6] have shown some methods are **forgetful** by not tracking information from previous aggregation rounds

[4] Bagdasaryan, Eugene, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. "How to backdoor federated learning." In *International Conference on Artificial Intelligence and Statistics*, pp. 2938-2948. PMLR, 2020.

[5] Bhagoji, Arjun Nitin, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. "Analyzing federated learning through an adversarial lens." In *International Conference on Machine Learning*, pp. 634-643. PMLR, 2019.

[6] Karimireddy, Sai Praneeth, Lie He, and Martin Jaggi. "Learning from history for byzantine robust optimization." In *International Conference on Machine Learning*, pp. 5311-5319. PMLR, 2021.

Byzantine-robust FL methods

- Several Byzantine-robust defense methods have been developed
- Recent studies [4], [5], [6] have shown some methods are **forgetful** by not tracking information from previous aggregation rounds

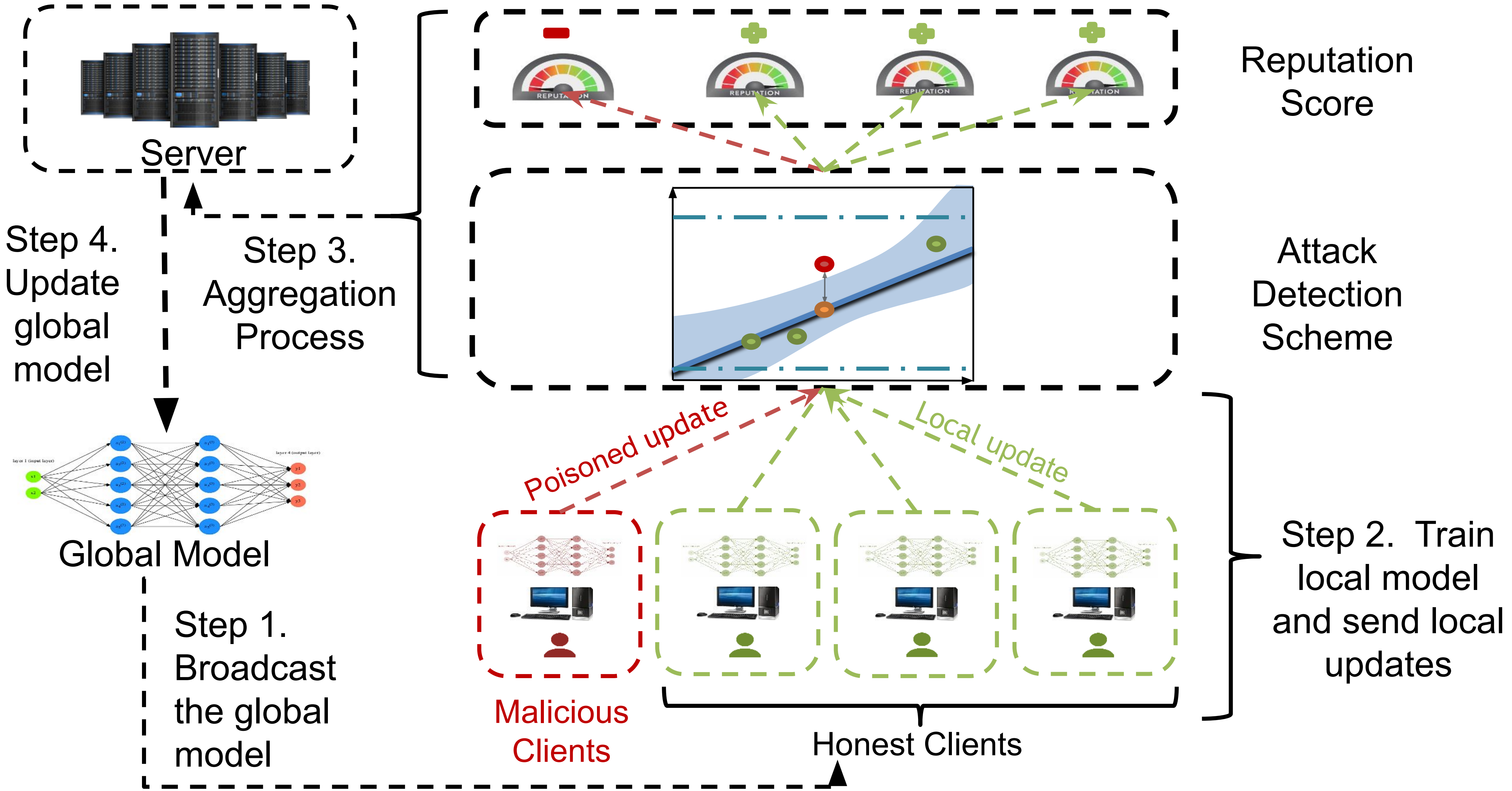
We design a robust FL aggregation method to generate **reputation scores** of clients based on their **historical behaviors**

[4] Bagdasaryan, Eugene, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. "How to backdoor federated learning." In *International Conference on Artificial Intelligence and Statistics*, pp. 2938-2948. PMLR, 2020.

[5] Bhagoji, Arjun Nitin, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. "Analyzing federated learning through an adversarial lens." In *International Conference on Machine Learning*, pp. 634-643. PMLR, 2019.

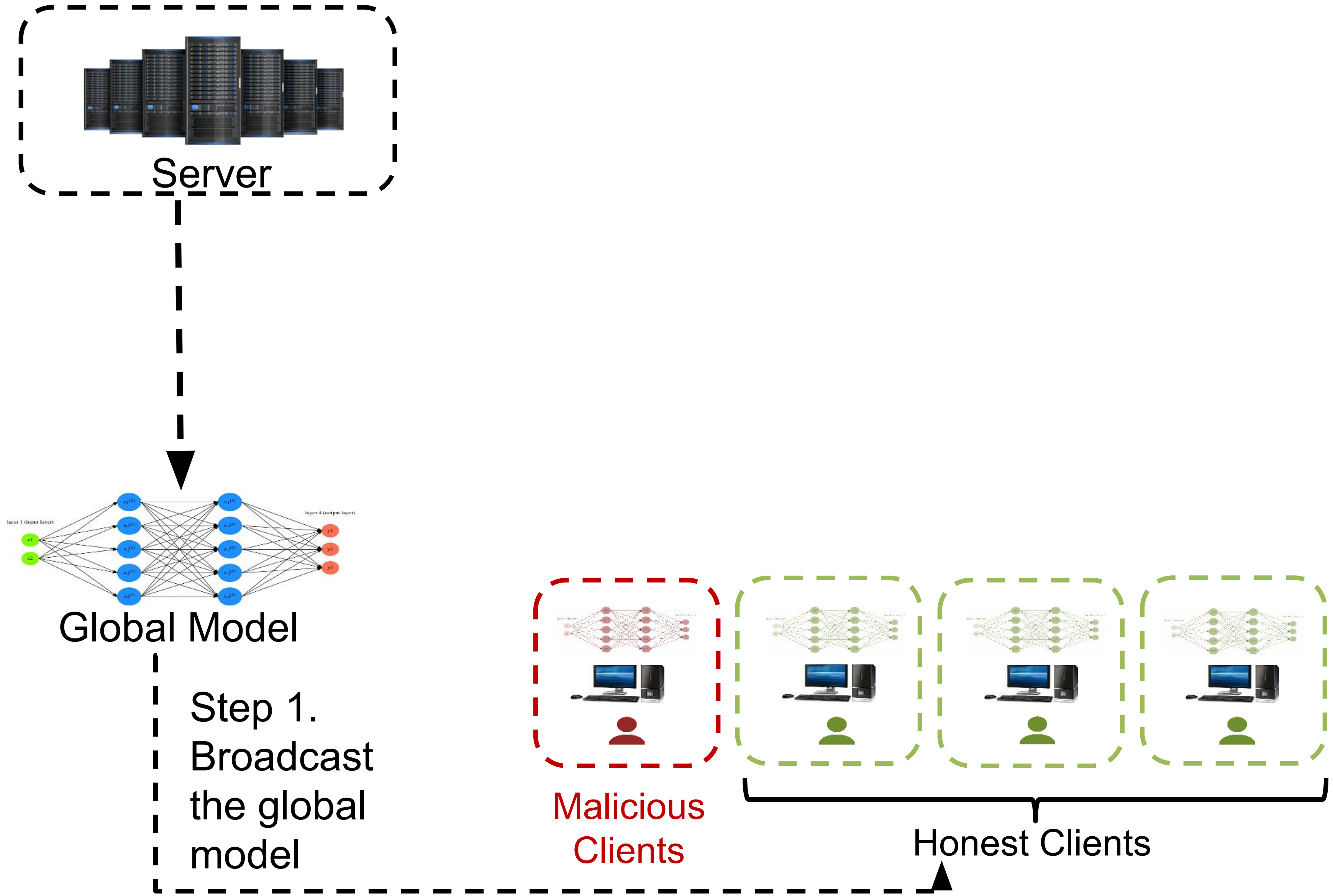
[6] Karimireddy, Sai Praneeth, Lie He, and Martin Jaggi. "Learning from history for byzantine robust optimization." In *International Conference on Machine Learning*, pp. 5311-5319. PMLR, 2021.

Algorithm

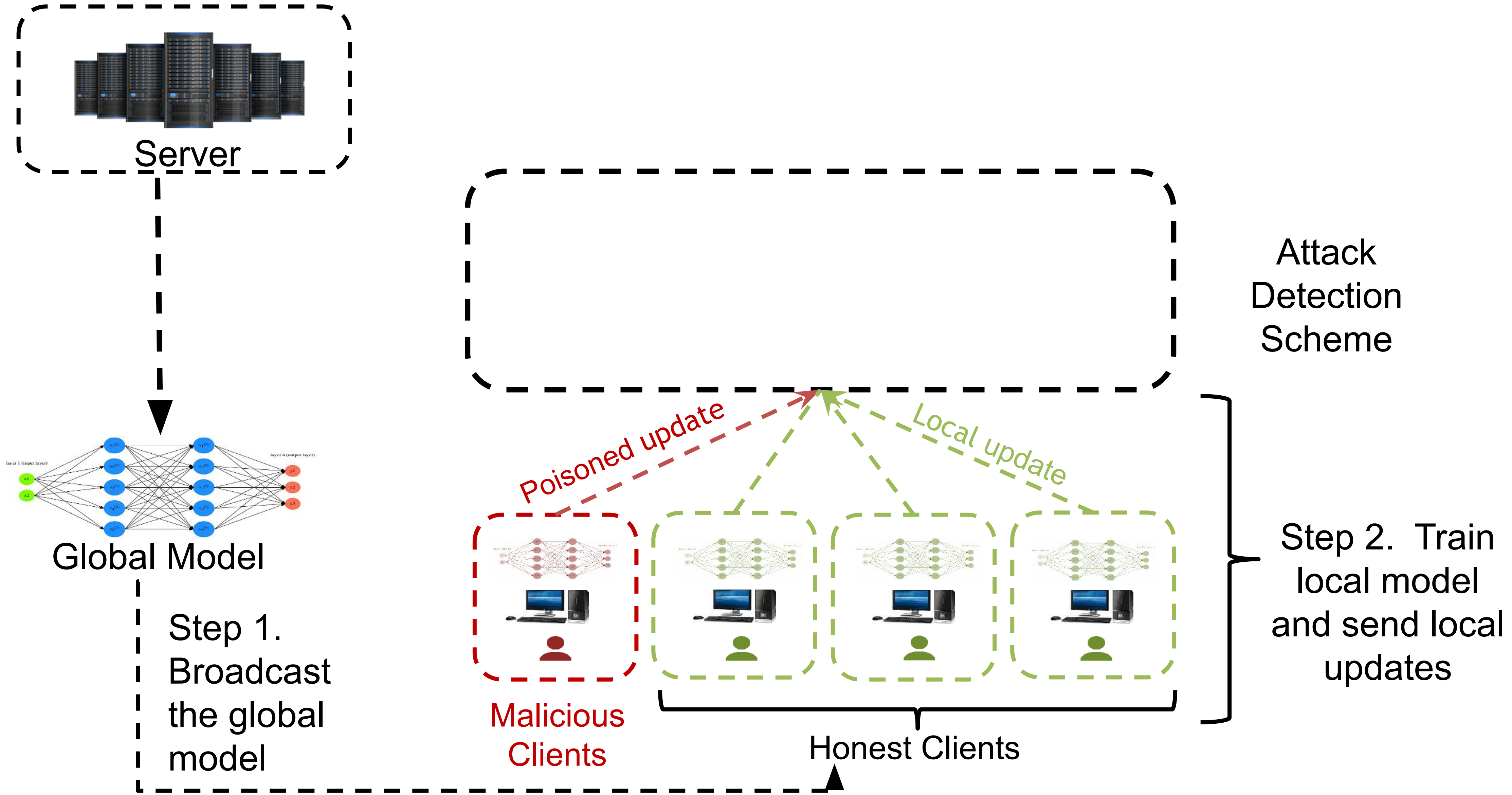


Server
Step 4. Update global model
Step 3. Aggregation Process
Reputation Score
Attack Detection Scheme
Poisoned update
Local update
Global Model
Step 1. Broadcast the global model
Malicious Clients
Honest Clients
Step 2. Train local model and send local updates

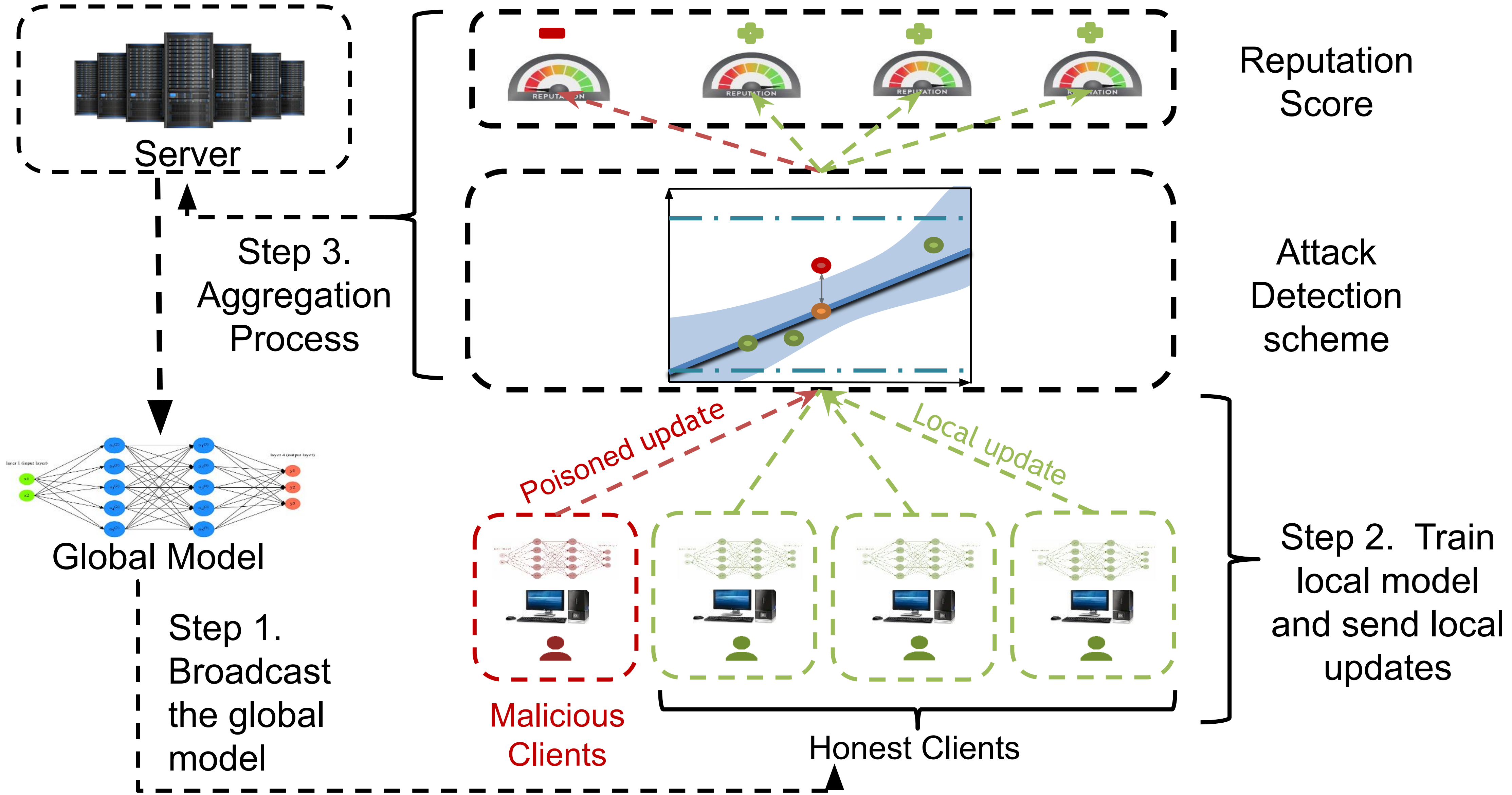
Algorithm


Server
Global Model
Step 1. Broadcast the global model
Malicious Clients
Honest Clients

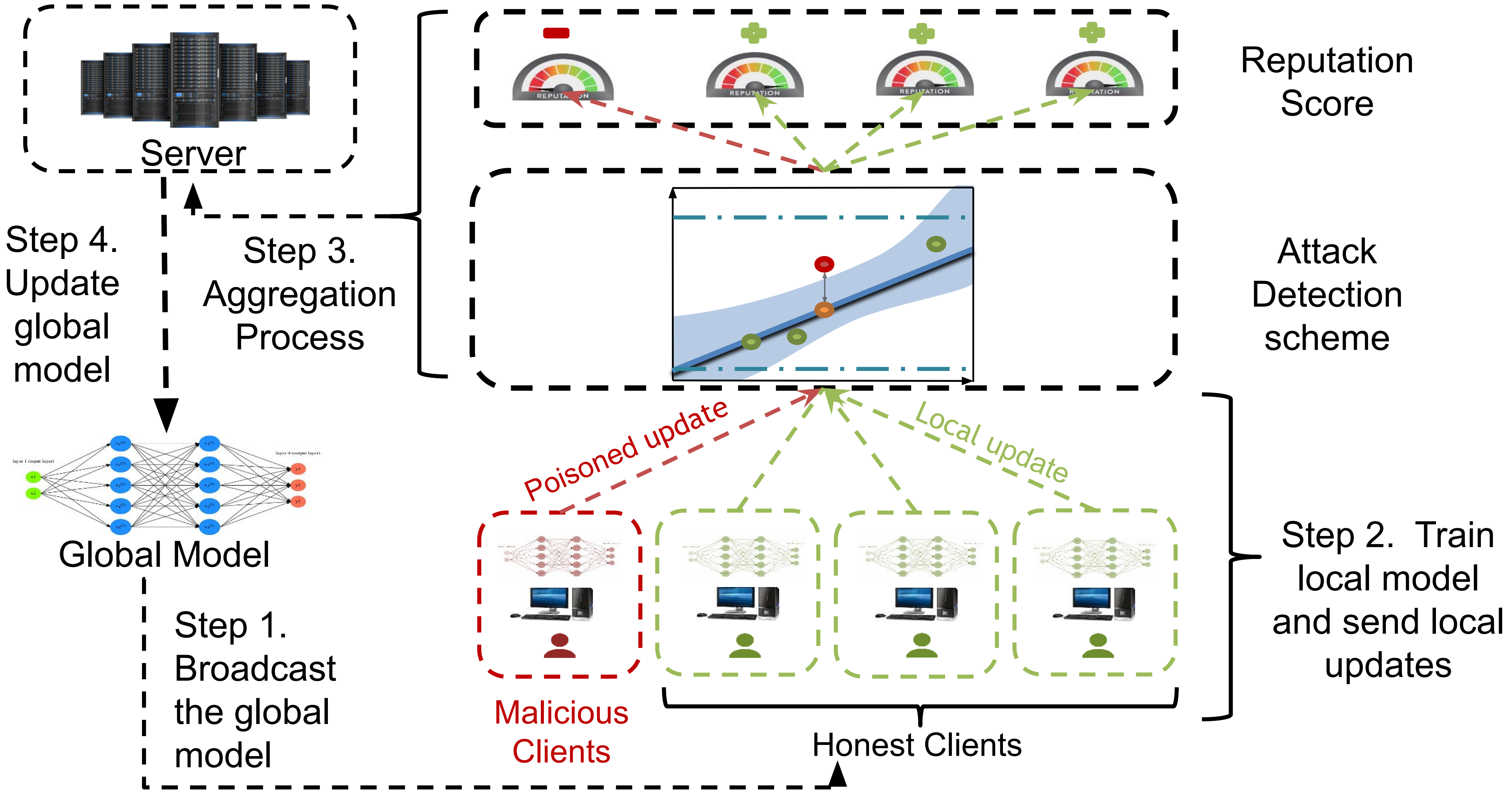
Algorithm


Server
Global Model
Step 1. Broadcast the global model
Poisoned update
Local update
Attack Detection Scheme
Step 2. Train local model and send local updates
Malicious Clients
Honest Clients

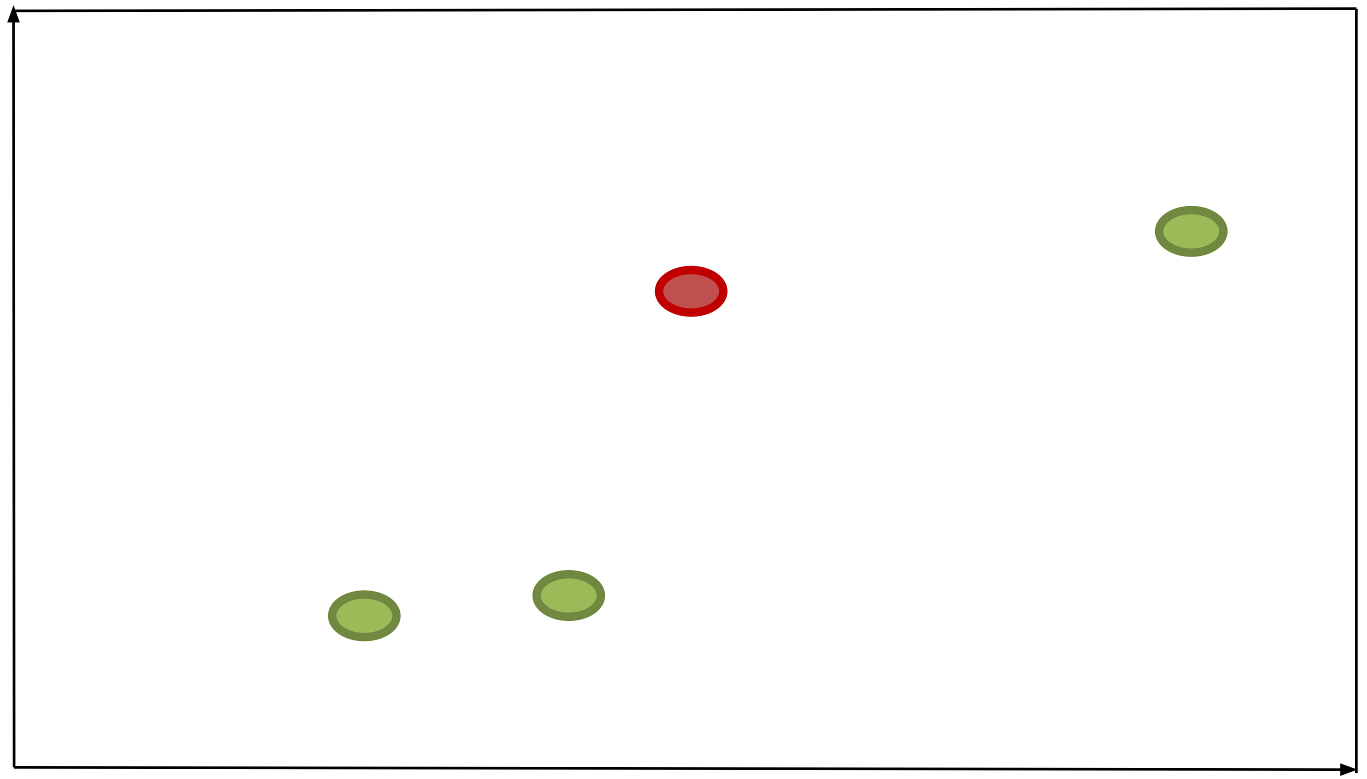
Algorithm


Server
Step 3.
Aggregation
Process
Reputation
Score
Attack
Detection
scheme
Poisoned update
Local update
Global Model
Step 1.
Broadcast
the global
model
Malicious
Clients
Honest Clients
Step 2. Train
local model
and send local
updates

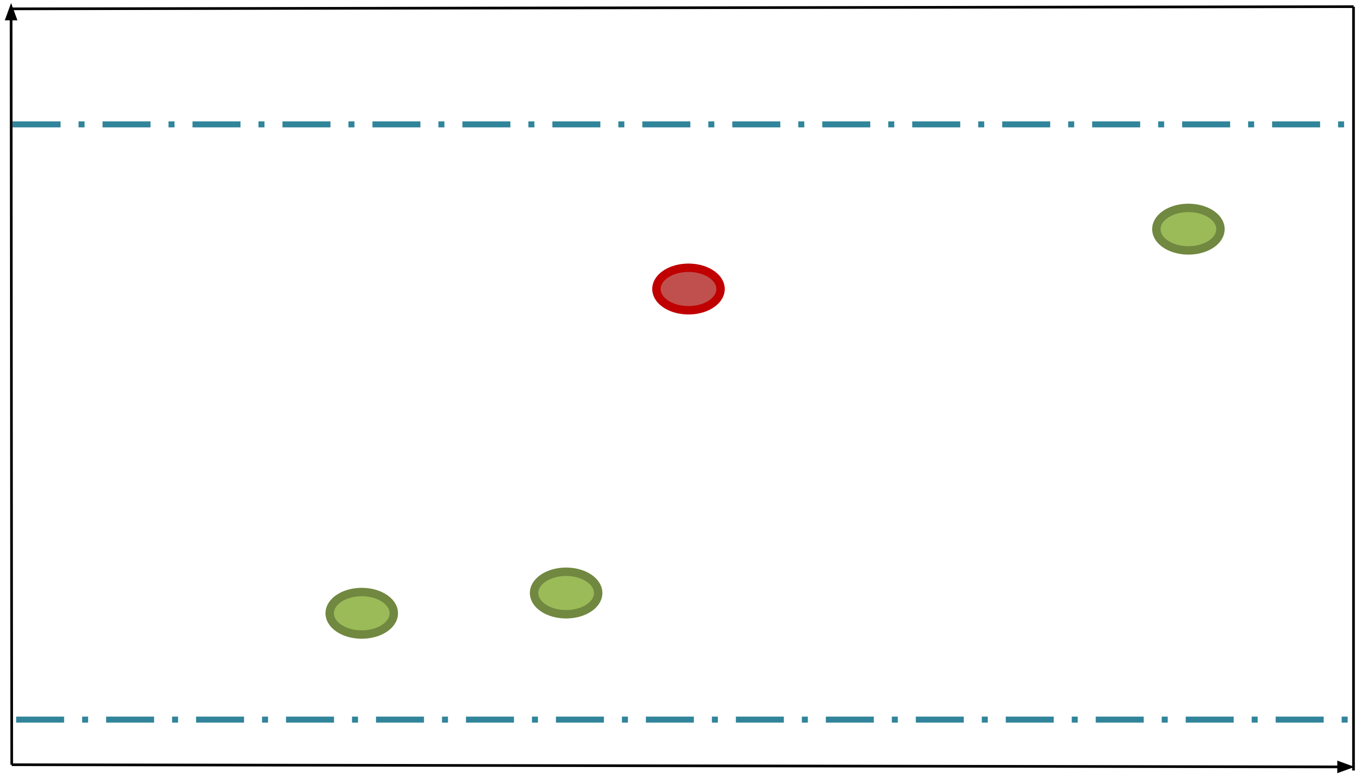
Algorithm


Server
Reputation Score
Attack Detection scheme
Step 4. Update global model
Step 3. Aggregation Process
Global Model
Poisoned update
Local update
Step 1. Broadcast the global model
Step 2. Train local model and send local updates
Malicious Clients
Honest Clients

Attack Detection Scheme



Attack Detection Scheme



Algorithm 1: Rescale(w)

Input : $\{w_{i,n}^t\} \leftarrow$ Local Model parameters in round t

Output: $\{w_{i,n}^t\}$ with range of value less than ϖ

for $n \leftarrow 1$ **to** N **do**

  // Determine the maximum range

  $Rm = \max w_{i,n}^t - \min w_{i,n}^t = w_{i,n}^{t,(Max)} - w_{i,n}^{t,(Min)}$

  while $Rm > \varpi$ **do**

    // Rescale range based on standard deviation.

    $w_{i,n}^{t,(Max)} := w_{i,n}^{t,(Max)} - \sigma(w_{i,n}^t)$;

    $w_{i,n}^{t,(Min)} := w_{i,n}^{t,(Min)} + \sigma(w_{i,n}^t)$;

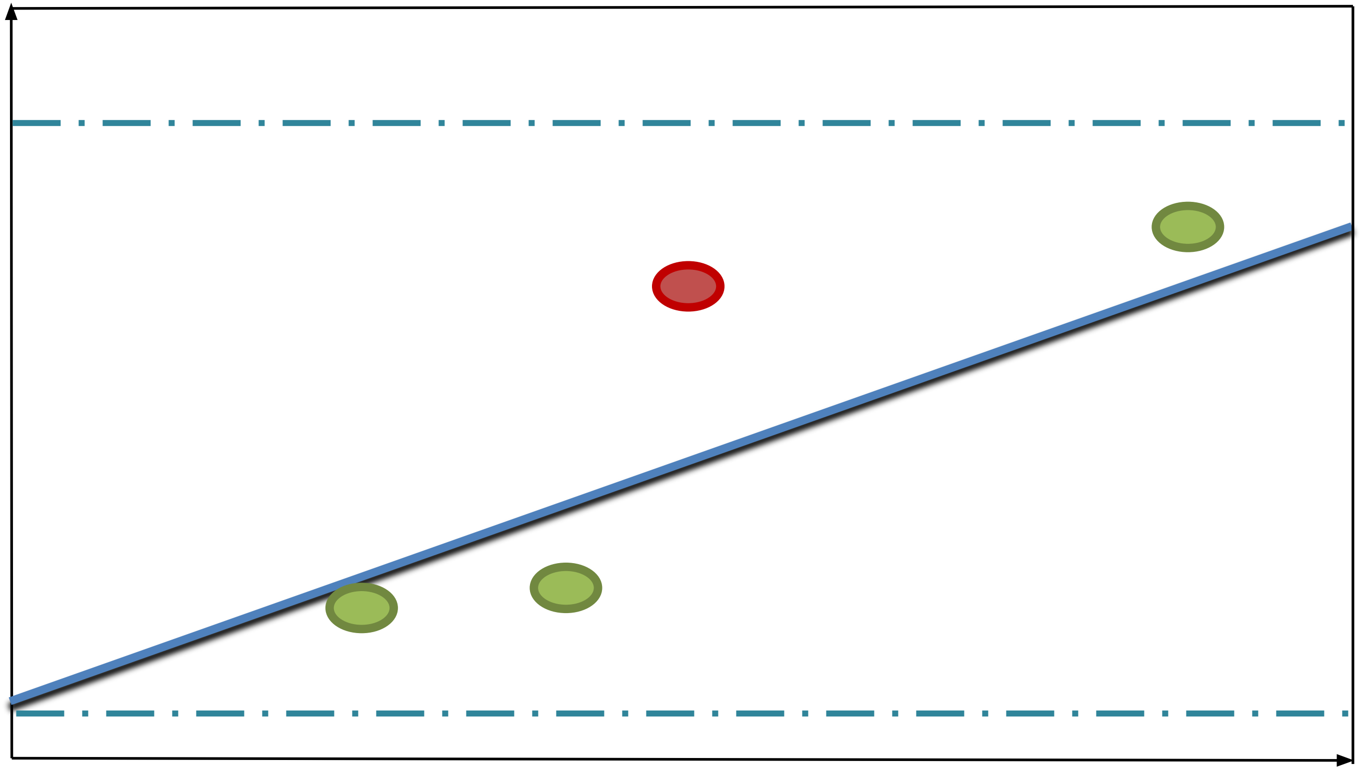
    // Updated Rm.

    $Rm = \max w_{i,n}^t - \min w_{i,n}^t = w_{i,n}^{t,(Max)} - w_{i,n}^{t,(Min)}$

  end while

end for

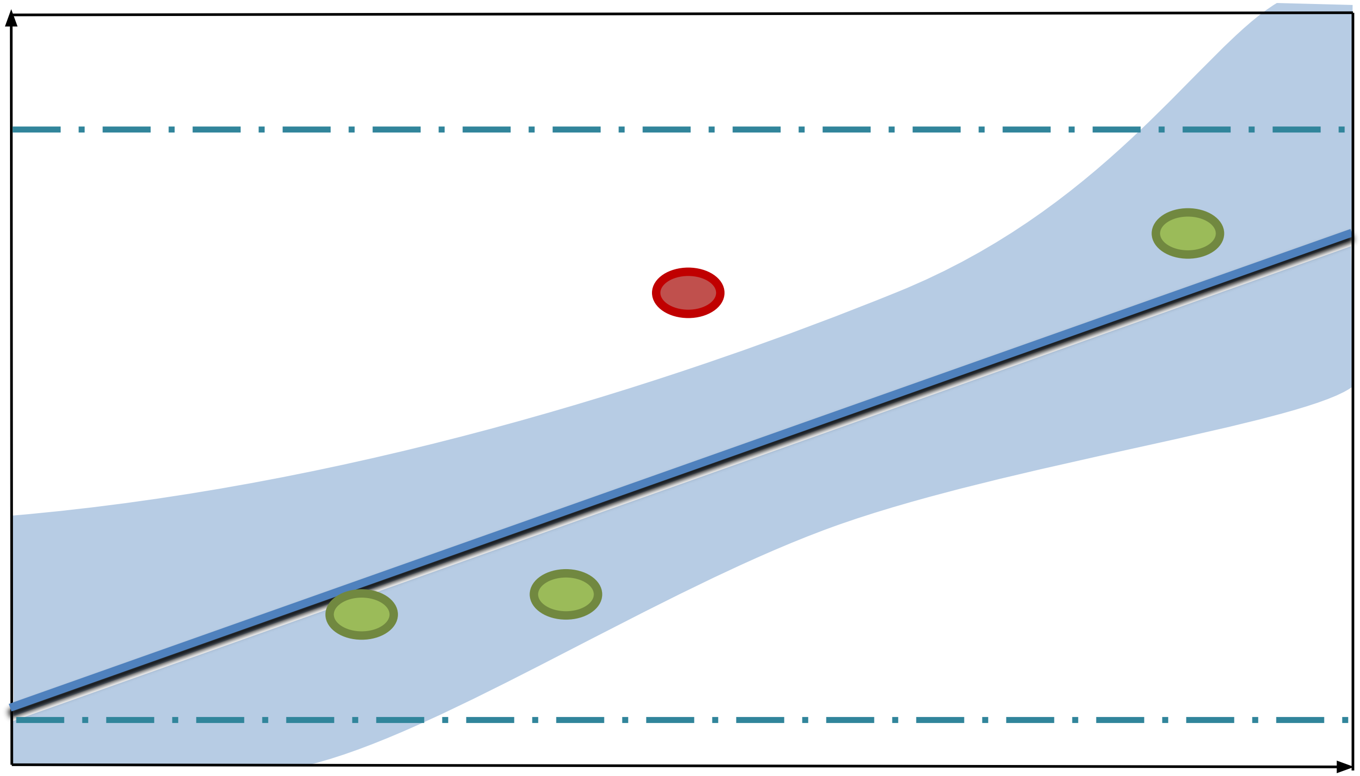
Attack Detection Scheme



$$\hat{B}_n = \operatorname*{median}_i \left\{ \operatorname*{median}_{i \neq j} \{B_n(i,j)\} \right\}$$

$$\hat{A}_n = \operatorname*{median}_i \left\{ w_{i,n} - \hat{B}_n x_{i,n} \right\}$$

Attack Detection Scheme

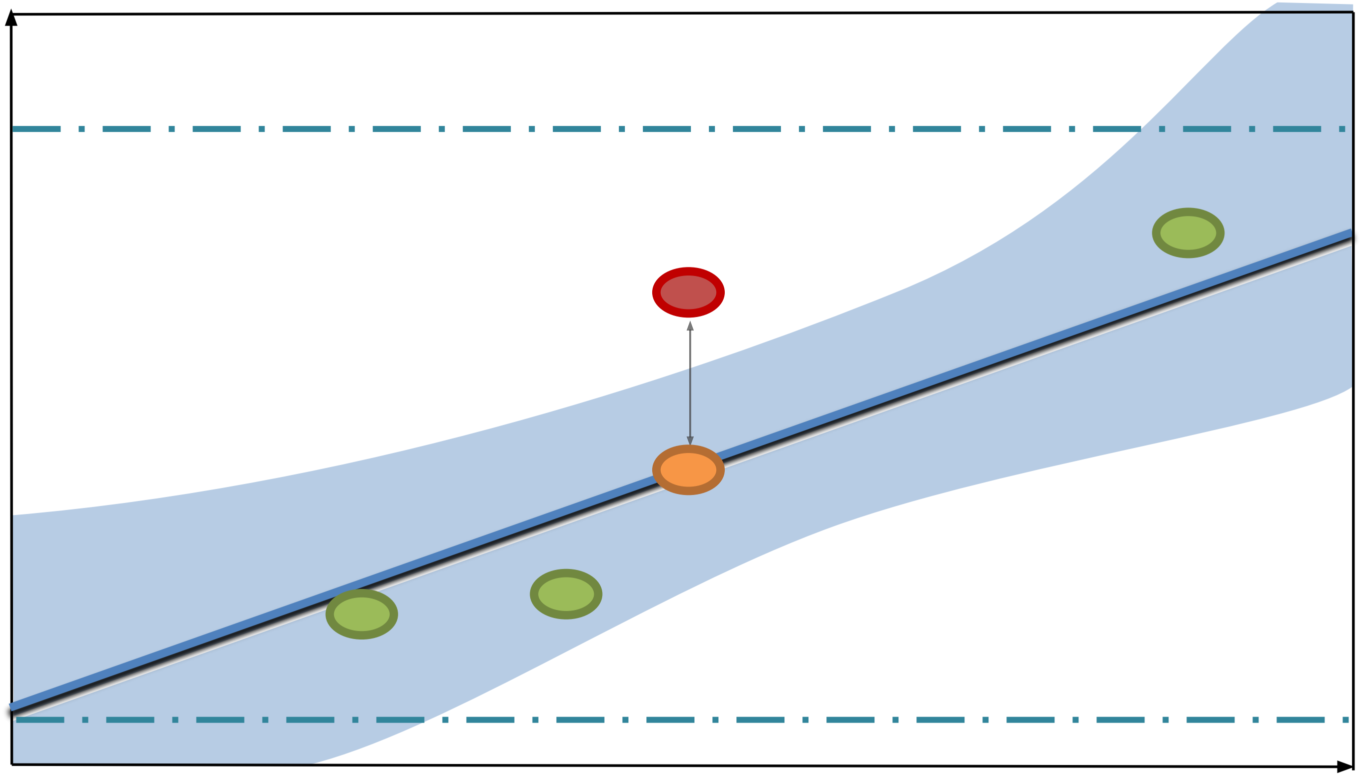


$$s_{i,n}^t = \frac{\sqrt{1 - diag(H_n^t)}}{e_{i,n}^t} \Psi\left(\frac{e_{i,n}^t}{\sqrt{1 - diag(H_n^t)}}\right)$$

where confidence interval $\Psi(x)$:

$$\Psi(x) = \max\{-\lambda\sqrt{2/M}, \min(\lambda\sqrt{2/M}, x)\}$$

Attack Detection Scheme



An update with confidence value less than threshold δ, it replaces it with the median

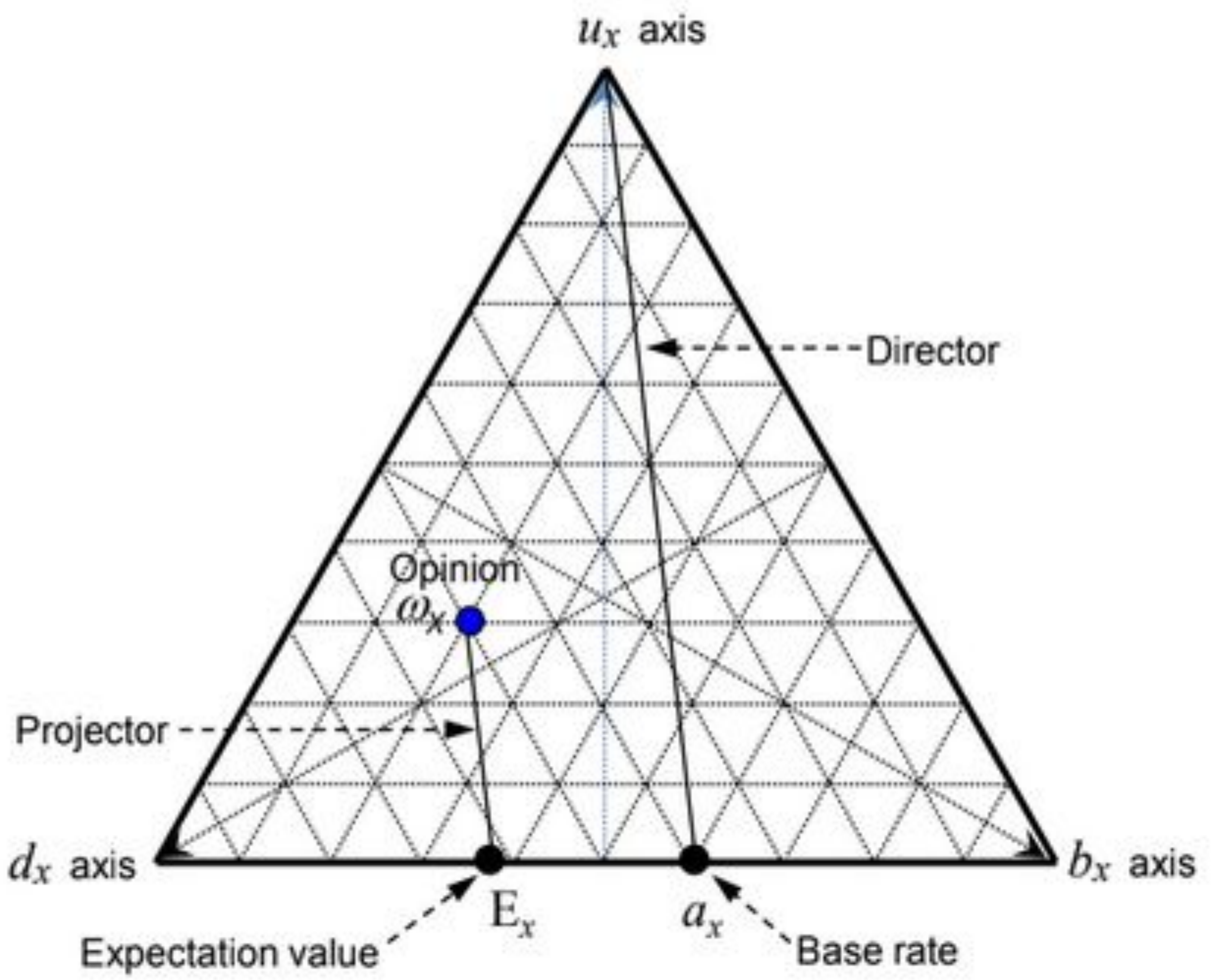
$$\mathbf{w}_{i,n}^t = \begin{cases} \mathbf{w}_{i,n}^t & \text{if } s_{i,n}^t > \delta \\ \underset{i}{\text{median}}\left\{\mathbf{w}_{i,n}^t\right\} & \text{if } s_{i,n}^t \leq \delta \end{cases}$$

$$s_{i,n}^t = \frac{\sqrt{1 - diag(H_n^t)}}{e_{i,n}^t} \Psi\left(\frac{e_{i,n}^t}{\sqrt{1 - diag(H_n^t)}}\right)$$

where confidence interval $\Psi(x)$:

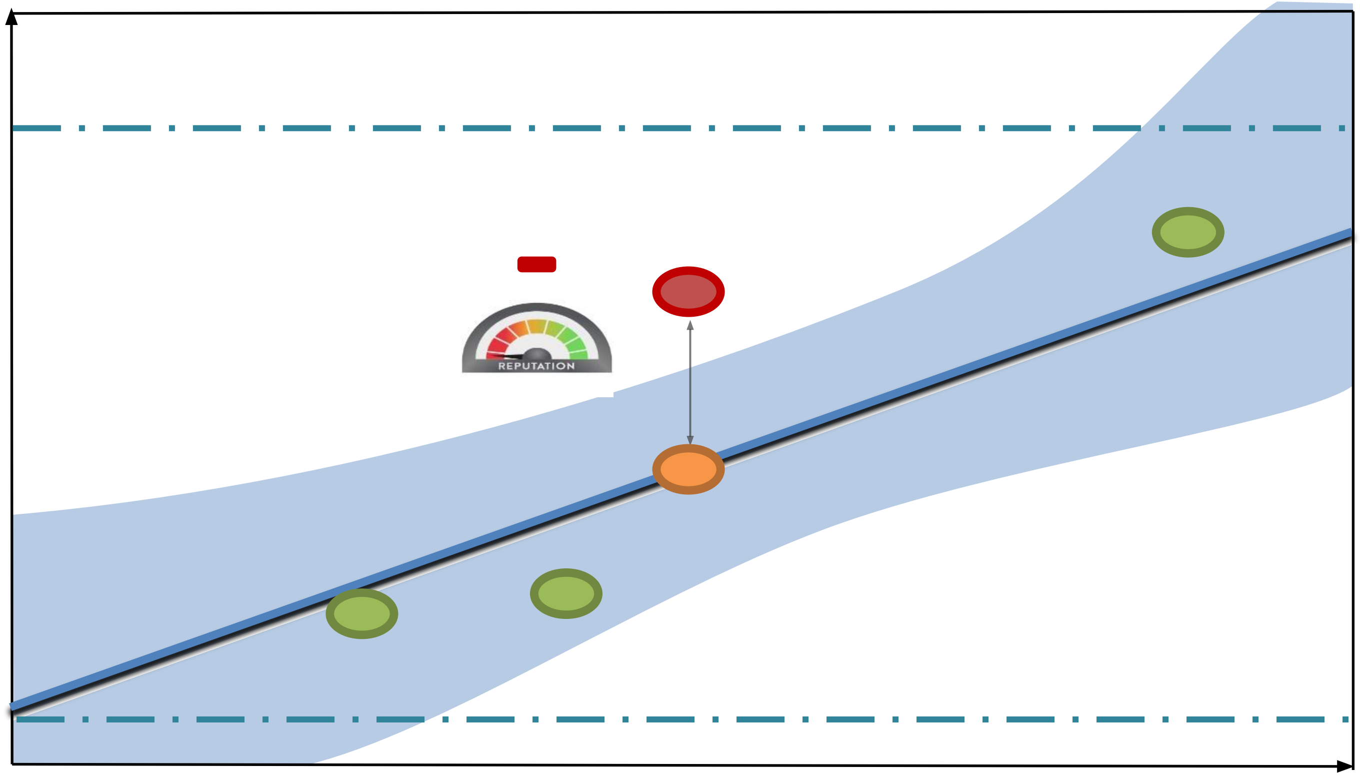
$$\Psi(x) = \max\{-\lambda\sqrt{2/M}, \min(\lambda\sqrt{2/M}, x)\}$$

Subjective logic model



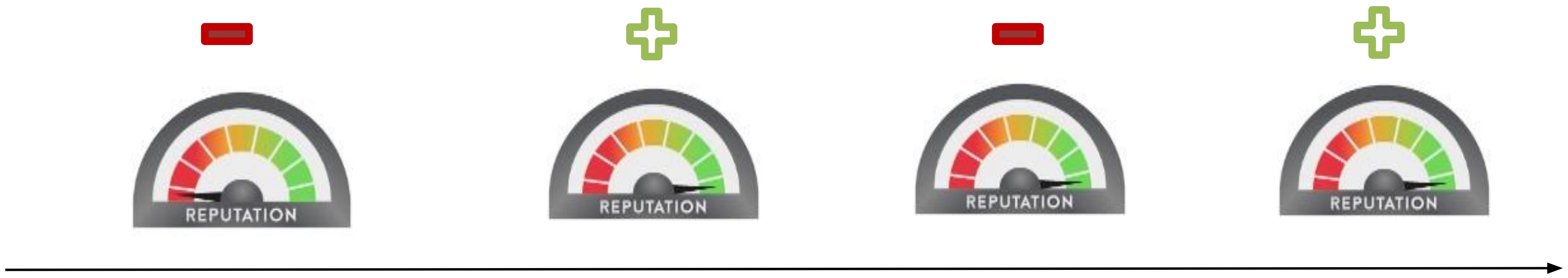
$$
\begin{cases}
b_i^t = \dfrac{\kappa P_i^t}{\kappa P_i^t + \eta N_i^t + W} \\
d_i^t = \dfrac{\eta N_i^t}{\kappa P_i^t + \eta N_i^t + W} \\
u_i^t = \dfrac{W}{\kappa P_i^t + \eta N_i^t + W}
\end{cases}
$$

Subject logic model



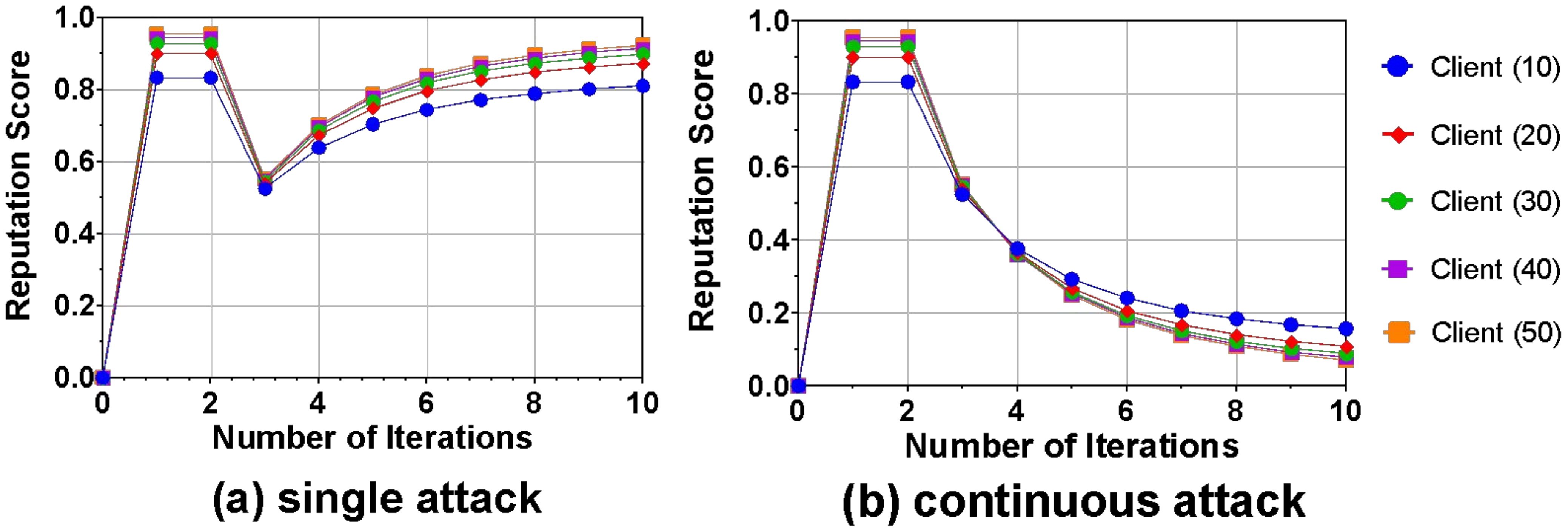
$$\begin{cases} b_i^t = \dfrac{\kappa P_i^t}{\kappa P_i^t + \eta N_i^t + W} \\ d_i^t = \dfrac{\eta N_i^t}{\kappa P_i^t + \eta N_i^t + W} \\ u_i^t = \dfrac{W}{\kappa P_i^t + \eta N_i^t + W} \end{cases}$$

Subject logic model



$$\theta_{j,t} = \exp(-c(t-j)) \longrightarrow \tilde{R}_i^t = \frac{\sum_{j=\tilde{s}}^{t} \theta_{j,t} R_i^j}{\sum_{j=\tilde{s}}^{t} \theta_{j,t}}$$

Reputation

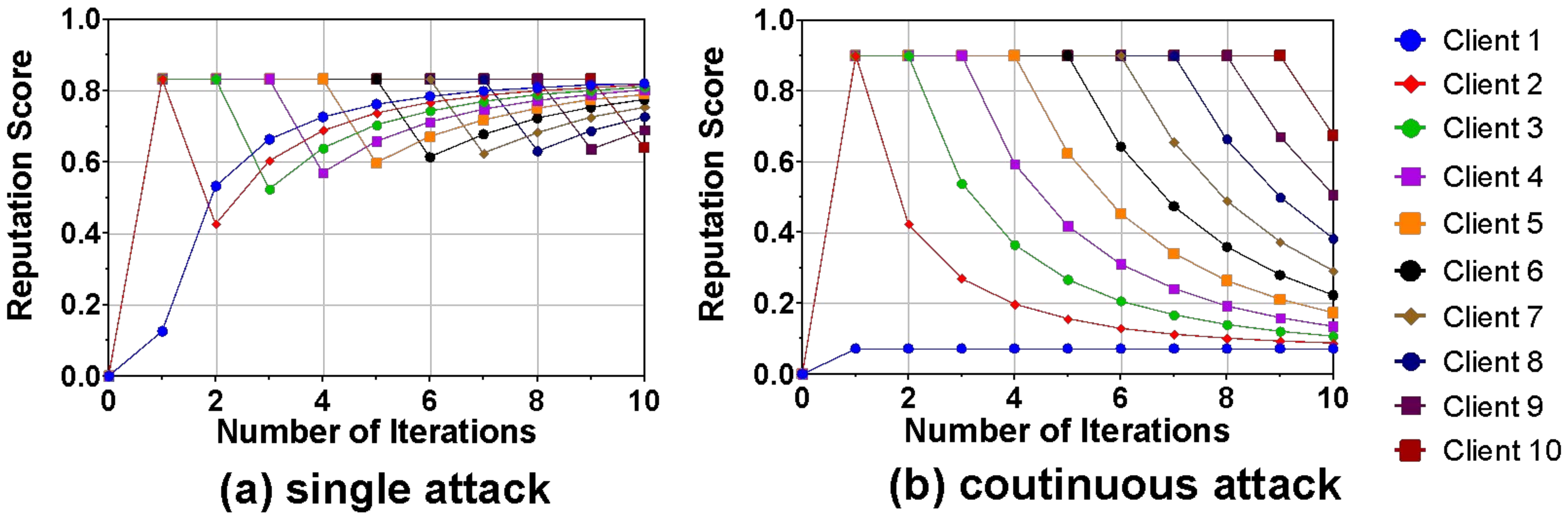


The decay of reputation score in Client (X) with X model parameters when they

(a) attack once at 3rd iteration

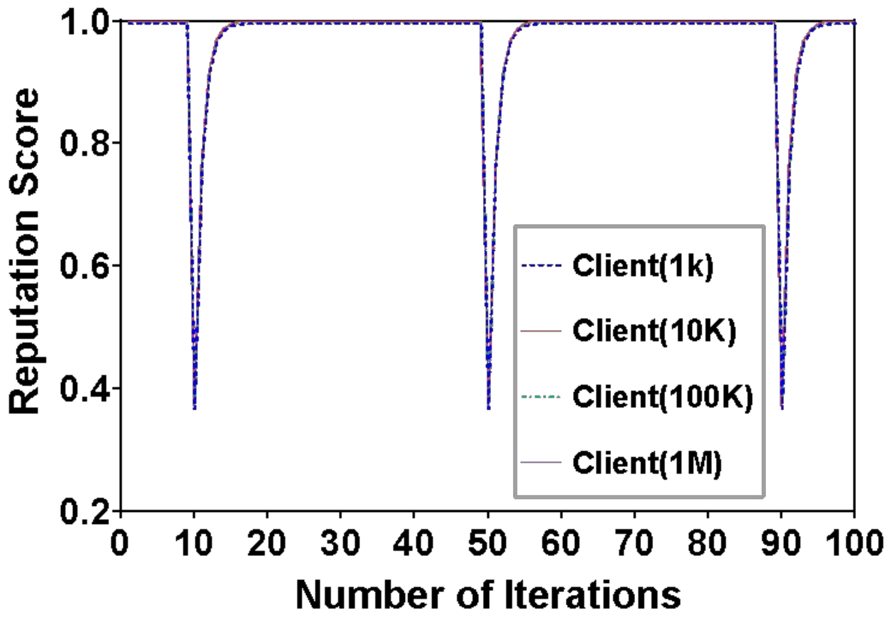
(b) attack continuously at and after the 3rd iteration

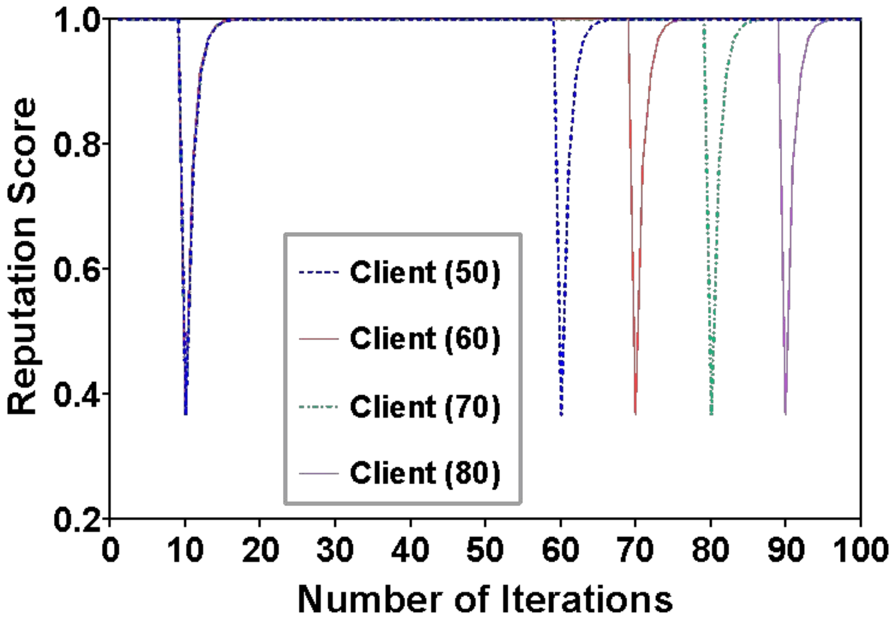
Reputation



- The decay of reputation score in Client X with same model parameters when they
  (a) attack once at X iteration and
  (b) attack continuously after starting to attack at X iteration

Reputation





- The decay of reputation score in

(a) client with X model parameters when they attack at 10, 50 and 90 iteration;
(b) client X with 1M parameters when they attack at 10 and 10 + X iteration

Theoretical Guarantees

Theorem shows the convergence is guaranteed

Theorem

Under Assumptions, $\exists \epsilon > 0$ *that:*

$$\sqrt{\frac{d \log(1 + \hat{Q}MLD)}{M(1-p)}} + C\frac{\mathcal{G}_w}{\sqrt{\hat{Q}}} + p \leq \frac{1}{2} - \epsilon \quad (1)$$

After t rounds, Our Algorithm converges with probability at least $1 - \xi \in \left[1 - \frac{4d}{(1+\hat{Q}ML\upsilon)^d}, 1\right)$ *as*

$$\|w^t - w^*\|_2 \leq (1 - Lr)^t \|w^0 - w^*\|_2 + \frac{\sqrt{N}}{L}\Delta_1 + \frac{1}{L}\Delta_2 \quad (2)$$

Theoretical Guarantees

The Corollary establishes the converge rate and error rate

Corollary

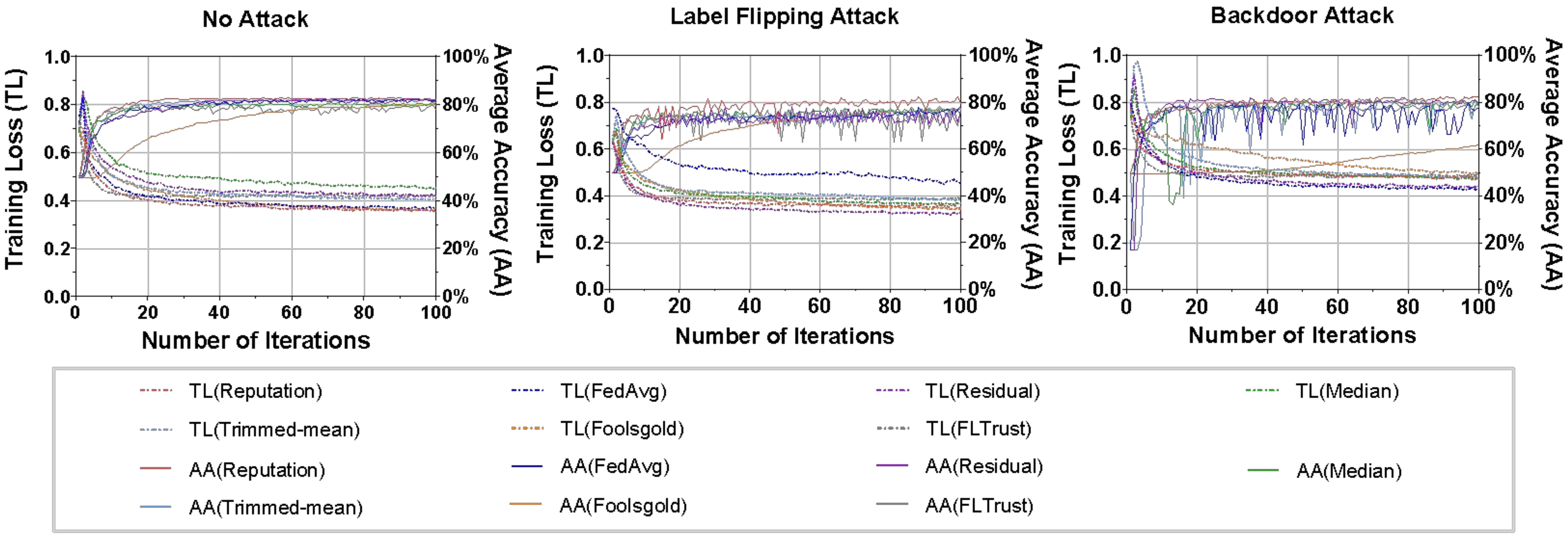
Continuing with Theorem 1, when the iterations satisfy

$t \geq \frac{1}{Lr} \log \left(\frac{L}{\sqrt{N}\Delta_1 + \Delta_2} \left\| w^0 - w^* \right\|_2 \right)$, $\exists \xi \in \left(0, \frac{4d}{(1+\hat{Q}ML\upsilon)^d}\right]$, *we have:*

$$\mathbb{P}\left(\left\| w^t - w^* \right\|_2 \leq \frac{2\sqrt{N}}{L}\Delta_1 + \frac{2}{L}\Delta_2 \right) \geq 1 - \xi$$

- The convergence is guaranteed in <u>bounded time</u>
- The <u>trade-off</u> between converge rate and error rate
- Guidance for hyper-parameters tuning

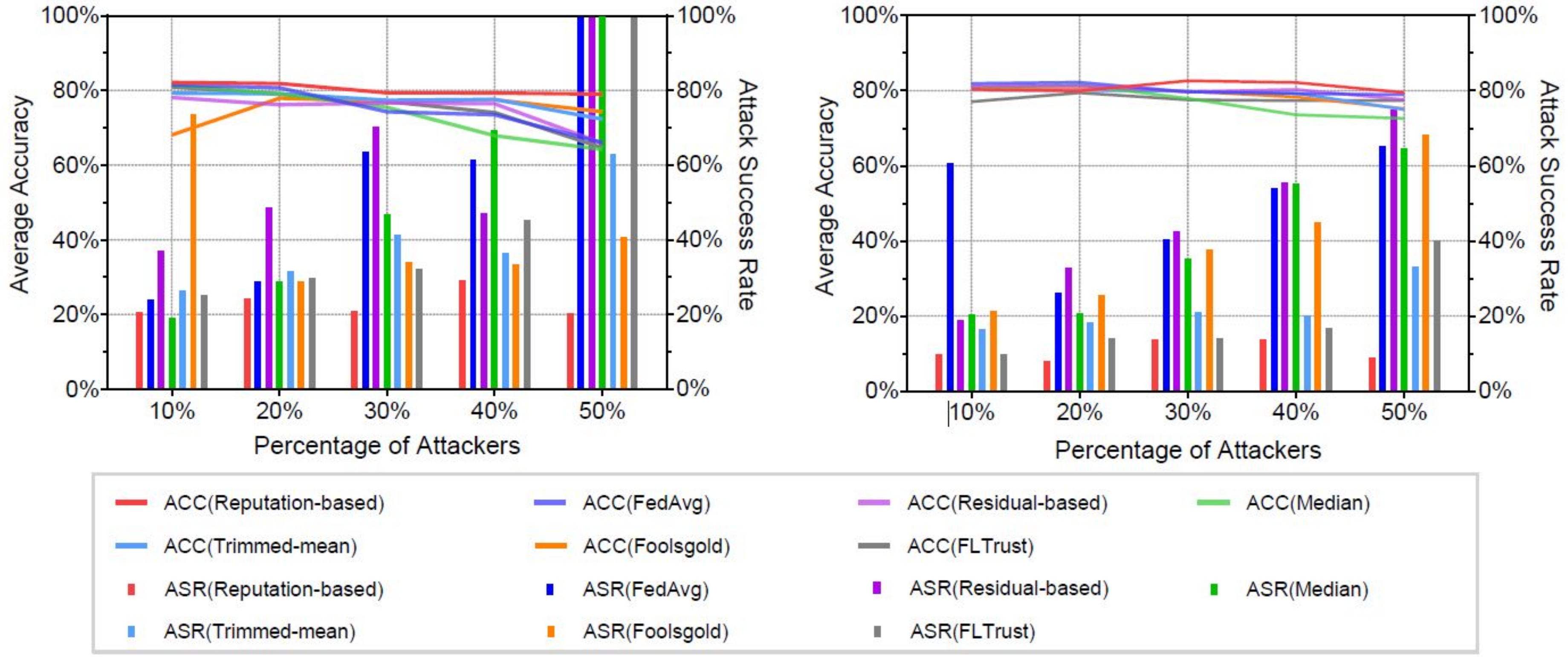
Convergence and Accuracy



– **Converges 1.6× to at least 4.2× faster** than all competing state-of-the-art methods

– Provides the **same or better accuracy** than competing methods

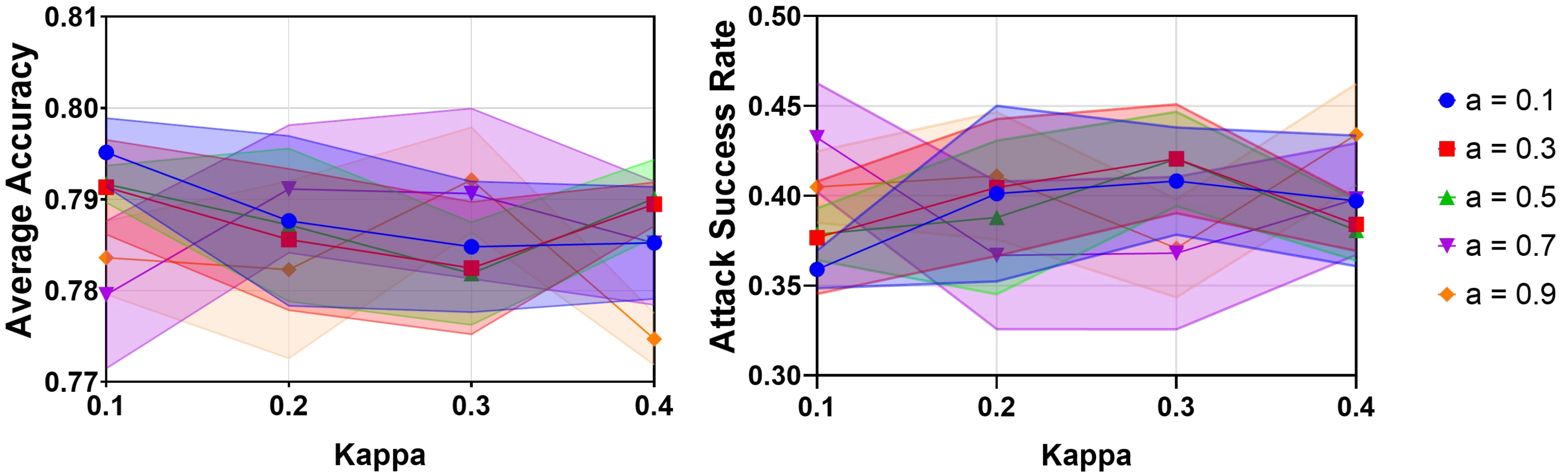
Attack Success rate



Varying percentage of attackers from 10% to 50%

- Yields the **lowest ASR** compared to all other methods,with the average ASR of them being at least 72% higher than ours

Hyper-parameters Searching

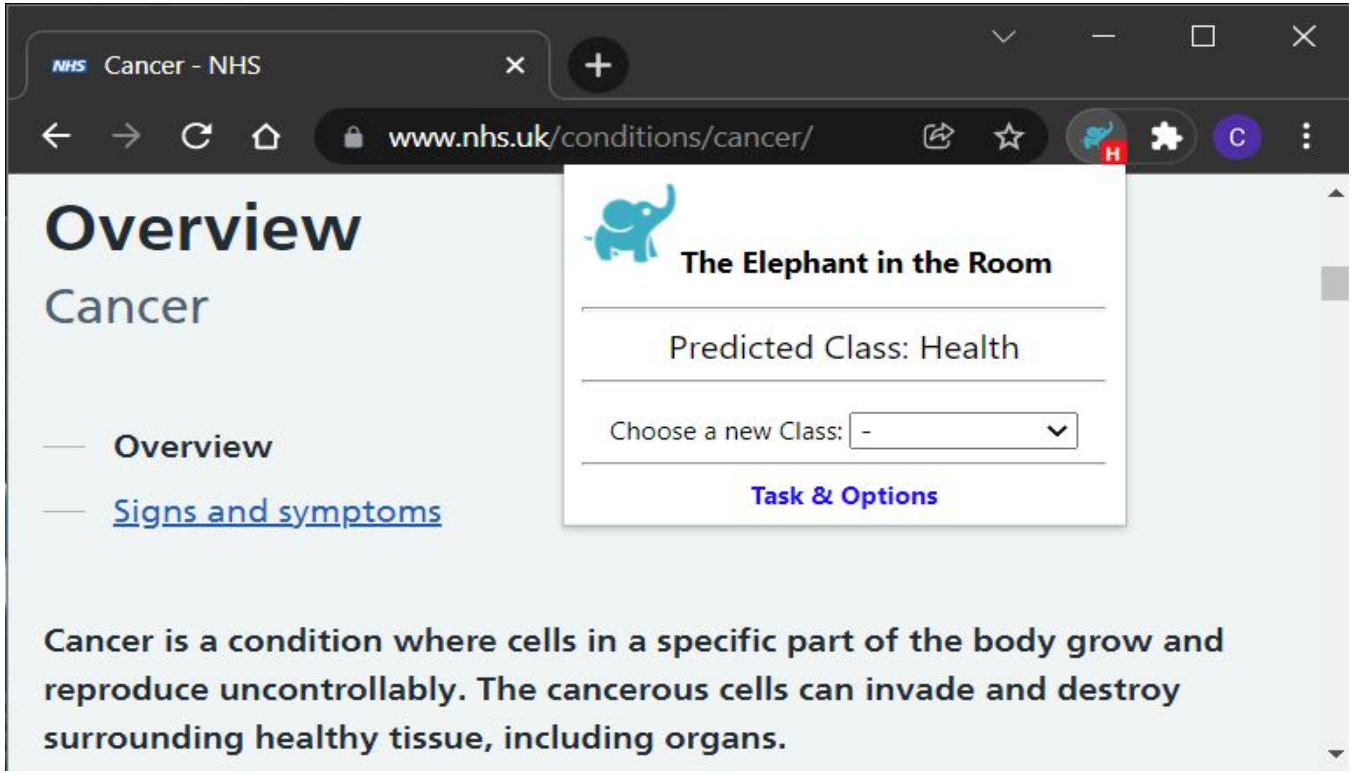


The result demonstrates that our method is

- very stable and efficient in terms of hyper-parameter selection
- and it achieves a high degree of precision
- the result is compatible with the theoretical analysis

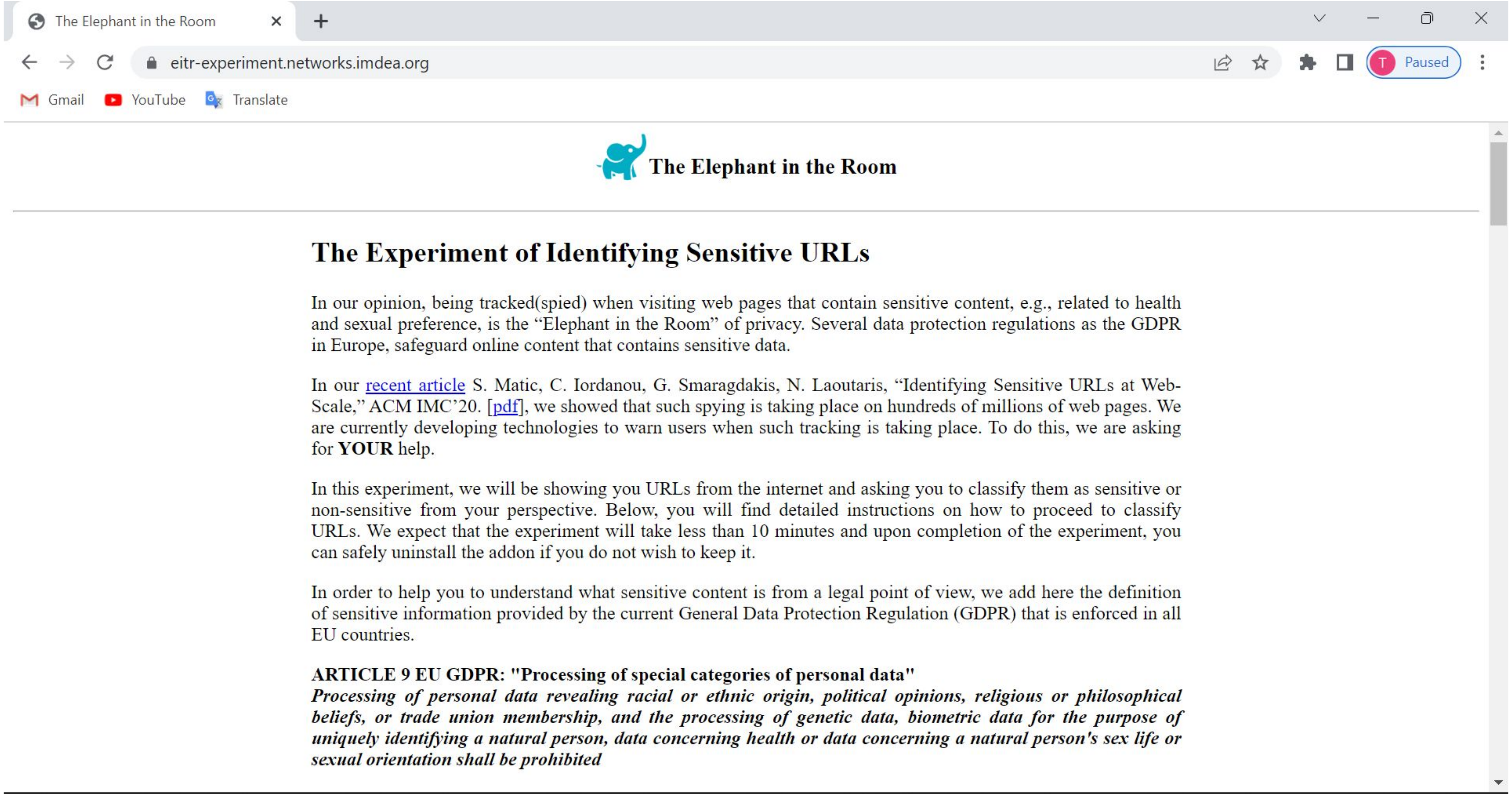
EITR System

- A research prototype to evaluate the robustness of our algorithm in a simple real-world setting with real users



https://eitr-experiment.networks.imdea.org/

Real-user Experiment



The Elephant in the Room

The Experiment of Identifying Sensitive URLs

In our opinion, being tracked(spied) when visiting web pages that contain sensitive content, e.g., related to health and sexual preference, is the “Elephant in the Room” of privacy. Several data protection regulations as the GDPR in Europe, safeguard online content that contains sensitive data.

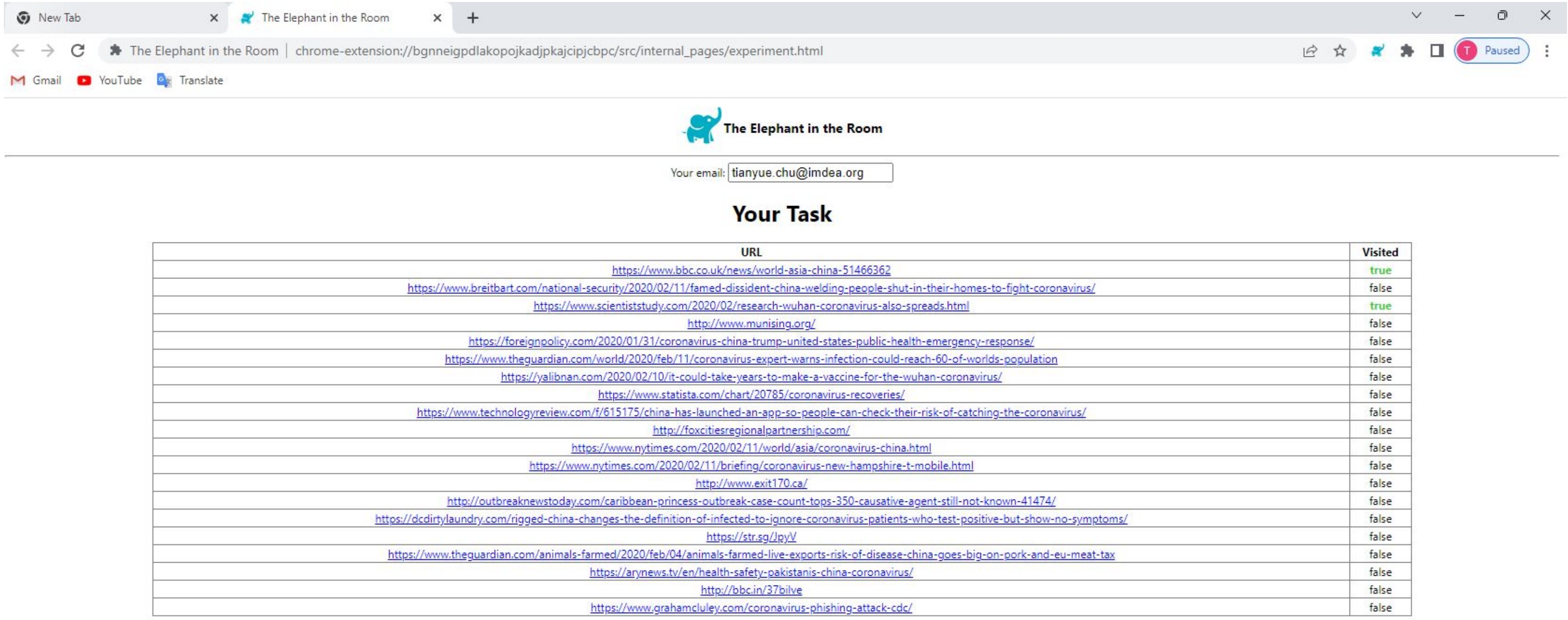
In our recent article S. Matic, C. Iordanou, G. Smaragdakis, N. Laoutaris, “Identifying Sensitive URLs at Web-Scale,” ACM IMC’20. [pdf], we showed that such spying is taking place on hundreds of millions of web pages. We are currently developing technologies to warn users when such tracking is taking place. To do this, we are asking for **YOUR** help.

In this experiment, we will be showing you URLs from the internet and asking you to classify them as sensitive or non-sensitive from your perspective. Below, you will find detailed instructions on how to proceed to classify URLs. We expect that the experiment will take less than 10 minutes and upon completion of the experiment, you can safely uninstall the addon if you do not wish to keep it.

In order to help you to understand what sensitive content is from a legal point of view, we add here the definition of sensitive information provided by the current General Data Protection Regulation (GDPR) that is enforced in all EU countries.

ARTICLE 9 EU GDPR: "Processing of special categories of personal data"
Processing of personal data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, or trade union membership, and the processing of genetic data, biometric data for the purpose of uniquely identifying a natural person, data concerning health or data concerning a natural person's sex life or sexual orientation shall be prohibited

Real-user Experiment



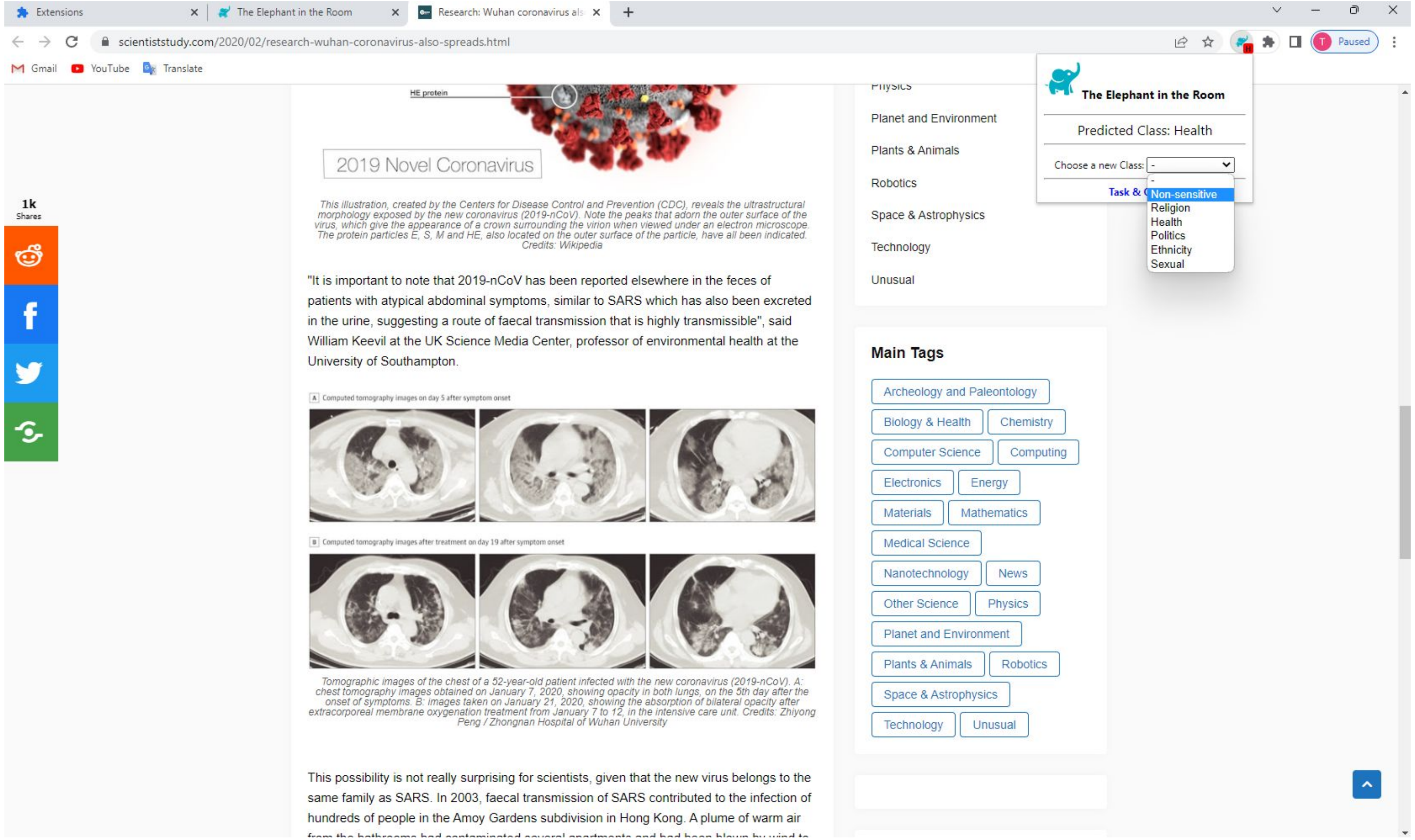
The Elephant in the Room

Your email: tianyue.chu@imdea.org

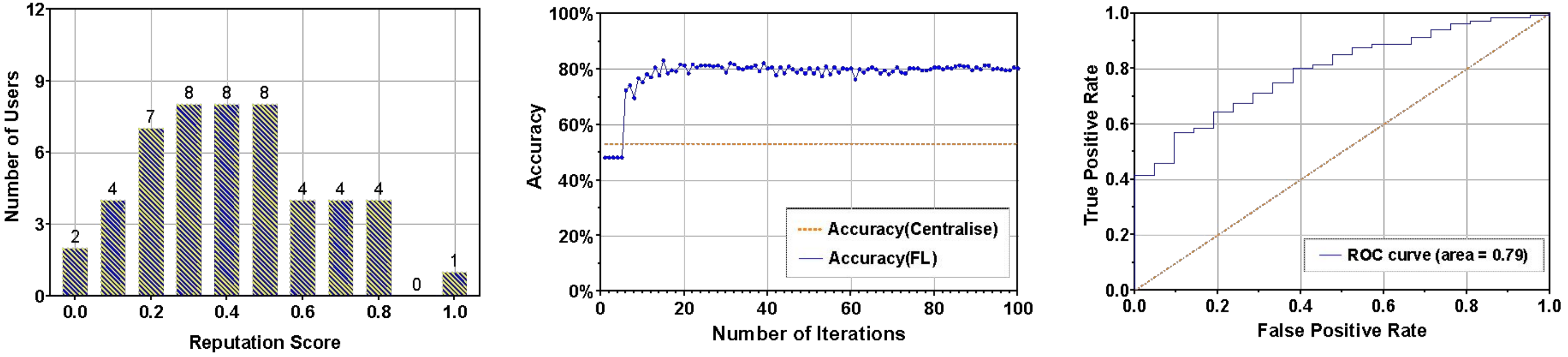
Your Task

URL	Visited
https://www.bbc.co.uk/news/world-asia-china-51466362	true
https://www.breitbart.com/national-security/2020/02/11/famed-dissident-china-welding-people-shut-in-their-homes-to-fight-coronavirus/	false
https://www.scientiststudy.com/2020/02/research-wuhan-coronavirus-also-spreads.html	true
http://www.munising.org/	false
https://foreignpolicy.com/2020/01/31/coronavirus-china-trump-united-states-public-health-emergency-response/	false
https://www.theguardian.com/world/2020/feb/11/coronavirus-expert-warns-infection-could-reach-60-of-worlds-population	false
https://yalibnan.com/2020/02/10/it-could-take-years-to-make-a-vaccine-for-the-wuhan-coronavirus/	false
https://www.statista.com/chart/20785/coronavirus-recoveries/	false
https://www.technologyreview.com/f/615175/china-has-launched-an-app-so-people-can-check-their-risk-of-catching-the-coronavirus/	false
http://foxcitiesregionalpartnership.com/	false
https://www.nytimes.com/2020/02/11/world/asia/coronavirus-china.html	false
https://www.nytimes.com/2020/02/11/briefing/coronavirus-new-hampshire-t-mobile.html	false
http://www.exit170.ca/	false
http://outbreaknewstoday.com/caribbean-princess-outbreak-case-count-tops-350-causative-agent-still-not-known-41474/	false
https://dcdirtylaundry.com/rigged-china-changes-the-definition-of-infected-to-ignore-coronavirus-patients-who-test-positive-but-show-no-symptoms/	false
https://str.sg/JpyV	false
https://www.theguardian.com/animals-farmed/2020/feb/04/animals-farmed-live-exports-risk-of-disease-china-goes-big-on-pork-and-eu-meat-tax	false
https://arynews.tv/en/health-safety-pakistanis-china-coronavirus/	false
http://bbc.in/37bilve	false
https://www.grahamcluley.com/coronavirus-phishing-attack-cdc/	false

Real-user Experiment



Validation with real users



- **Result with real users**
  - The divergence of the user's interpretation of the sensitive information
  - Our method converges as rapidly as in simulation and achieves an average accuracy of 80%.

Our FL-based solution can quickly learn to classify sites about COVID even with inconsistent input provided by real users

Conclusion

Employ FL for sensitive content classification

Implement EITR browser extension

Design a robust FL aggregation with reputation

- Theoretical analysis
- Experimental evaluation

Implementation available online:
https://github.com/FRM-Sec/FRM

Questions:
Email: tianyue.chu@imdea.org

THANK YOU!

Attack detection scheme

Algorithm 1: Rescale(w)

Input : $\{w_{i,n}^t\} \leftarrow$ Local Model parameters in round t

Output: $\{w_{i,n}^t\}$ with range of value less than ϖ

for $n \leftarrow 1$ **to** N **do**
  `// Determine the maximum range`
  $Rm = \max w_{i,n}^t - \min w_{i,n}^t = w_{i,n}^{t,(Max)} - w_{i,n}^{t,(Min)}$
  while $Rm > \varpi$ **do**
    `// Rescale range based on standard deviation.`
    $w_{i,n}^{t,(Max)} := w_{i,n}^{t,(Max)} - \sigma(w_{i,n}^t)$;
    $w_{i,n}^{t,(Min)} := w_{i,n}^{t,(Min)} + \sigma(w_{i,n}^t)$;
    `// Updated` Rm.
    $Rm = \max w_{i,n}^t - \min w_{i,n}^t = w_{i,n}^{t,(Max)} - w_{i,n}^{t,(Min)}$
  end while
end for

Algorithm 2: Aggregation Algorithm

Server :

Input : $w^0 \leftarrow$ Pretrained Model
$\kappa,\eta,a,W,c,s \leftarrow$ Reputation parameters

Output: Global model M_{global} with w^T

for *Iteration* $t \leftarrow 1$ **to** T **do**
  `// Broadcast global model to clients`
  send(w^{t-1});
  `// Wait until all updates arrive`
  receive(w^t);
  `// Rescale parameters by Algorithm 1`
  $\bar{w}^t \leftarrow$ `Rescale`(w^t);
  for $n \leftarrow 1$ **to** N **do**
    for $i \leftarrow 1$ **to** M **do**
      `// Compute parameter confidence`
      $s_{i,n}^t = Eq\ 4(\bar{w}_{i,n}^t)$;
      `// Rectify abnormal parameters`
      $w_{i,n}^t := Eq\ 5\ (s_{i,n}^t, \delta)$;
      record (P_i^t, N_i^t);
    end for
  end for
  for $i \leftarrow 1$ **to** M **do**
    `// Calculate reputation score`
    $\tilde{R}_i^t = Eq\ 12(P_i^t,\ N_i^t,\ \kappa,\ \eta,\ a,\ W,\ c,\ s)$;
  end for
  `// Normalisation`
  $\bar{R}^t \leftarrow$ `Norm`$(\tilde{R}^t)$;
  for $n \leftarrow 1$ **to** N **do**
    `// Update the parameters`
    $w_n^t := \sum_{i=1}^{M} \frac{R_i^t}{\sum_{i=1}^{M} R_i^t} w_{i,n}^t$;
  end for
  `// Obtain parameters for global model`
  $w^t := [w_1^t, \cdots, w_n^t]$;
end for

Client :

for *Client* $i \leftarrow 1$ **to** M **do in parallel**
  receive(w^{t-1});
  `// Train local model`
  $w_i^t \leftarrow w_i^{t-1} - r\frac{\partial \ell_i(w_i^{t-1})}{\partial w}$;
  send(w_i^t);
end forpar

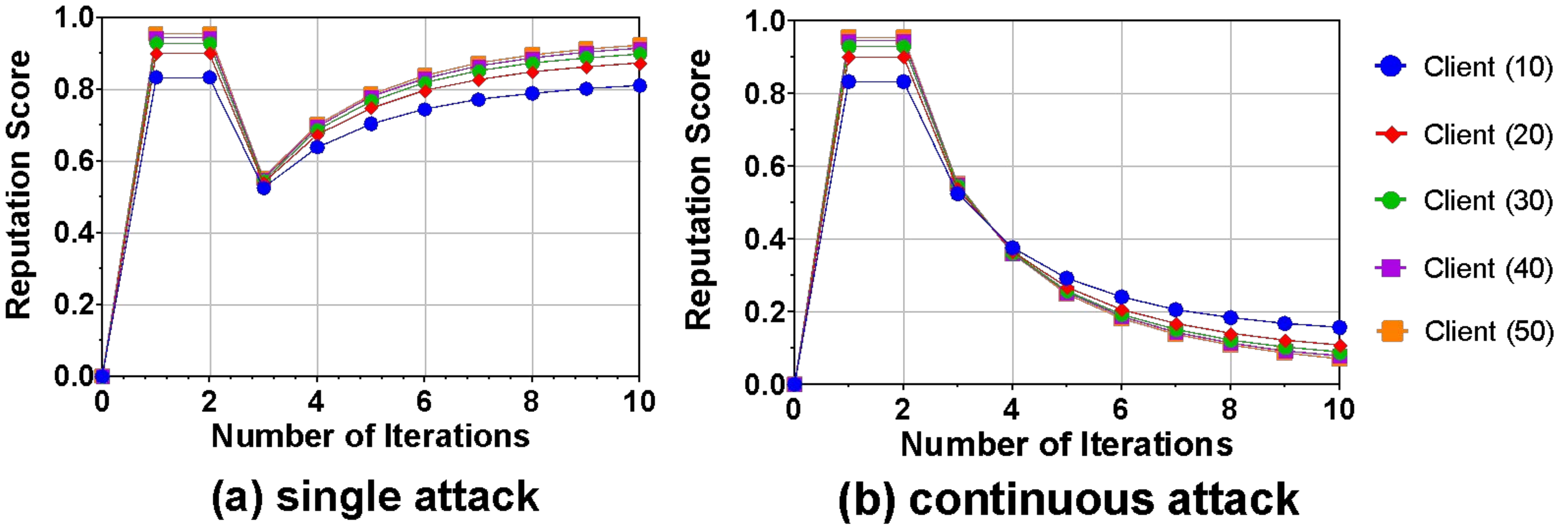
Repeated Median

IRLS scheme

Reputation:

Time decay function

Reputation



The decay of reputation score in Client (X) with X model parameters when they

(a) attack once at 3rd iteration

(b) attack continuously at and after the 3rd iteration

Centralized vs Distributed

Pros: centralised training performs better for some tasks, but for the task of detecting sensitive URLs to protect personal online privacy, a distributed classifier is a preferable solution.

Cons:

- Privacy: In a centralized manner, even with semi-supervised learning, manual labelling of certain training data is still required. However, because users are labelling sensitive data, their privacy will be harmed if the server has access to their labelling data for centralised training. Therefore, FL is a natural privacy-preserving method for conducting distributed collaborative model training across clients that do not disclose their local data.
- Efficiency: for the centralized training, the server has to update the dataset for learn new content, maybe by paying people to do it. but for FL, it can voluntarily and continuously learn from real-time web data gathered by users, and will represent the user's interpretation of sensitive content.

The number of attackers

we acknowledge that constraining the number of attackers is not realistic, but

- as with other works [7][15][19][20][21] on this topic, we follow the same assumption that the percentage of attackers is lower than the percentage of benign users in the system for our evaluation.
- In Sec. 3C Theoretical Guarantees, we also demonstrate how the number of attackers influences the performance of our algorithm.
- Additionally, we did not restrict the number of attackers in the real-user experiment, the results show that our method still performs well.

Attack Strategies

In a label flipping attack, the attacker flips the labels of training samples to a targeted label and trains the model accordingly.
In our case, the attacker changes the label of “Health” to “Non-sensitive”.

In a backdoor attack, attackers inject a designed pattern into their local data and train these manipulated data with clean data, in order to develop a local model that learns to recognise such pattern.

We realise backdoor attacks inserting the top 10 frequent words with their frequencies for the “Health” category. Therein the backdoor targets are the labels “non-sensitive”. A successful backdoor attack would acquire a global model that predicts the backdoor target label for data along with specific pattern

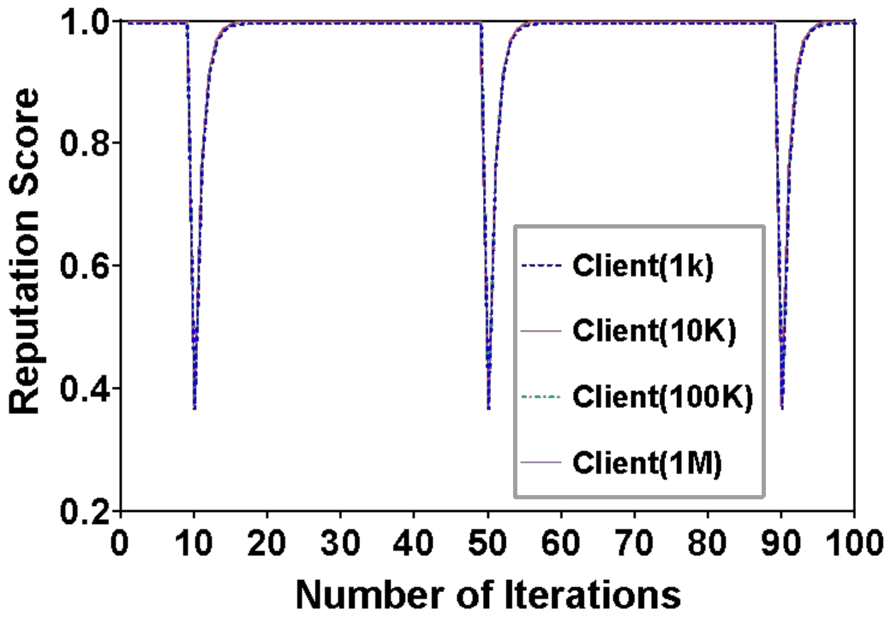
Attacks

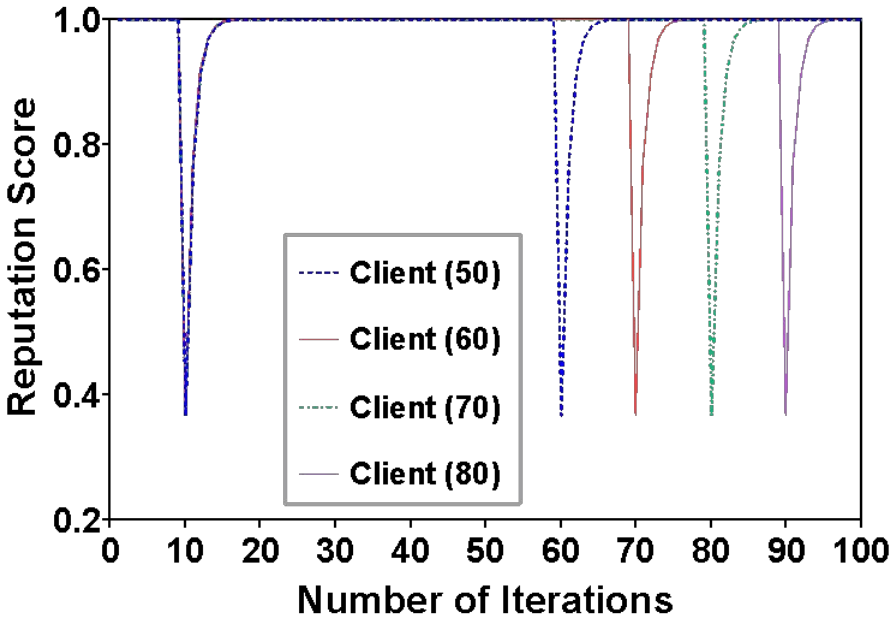
Consider two attacks:

We focus on two common attack strategies for sensitive context classification, namely, (i) label flipping attack [24] and (ii) backdoor attack [12].

Comparing to other attacks, for example model poisoning attack [29], [63], these two data poisoning attacks are more likely to be carried out by real users in the real world via our browser extension described in Section V, since polluting data is easier than manipulating model updates using the browser extension. Note that privacy attacks including membership inference attack [65] and property inference attack [64], are out of the scope of this paper,but form part of our ongoing and future work

Reputation



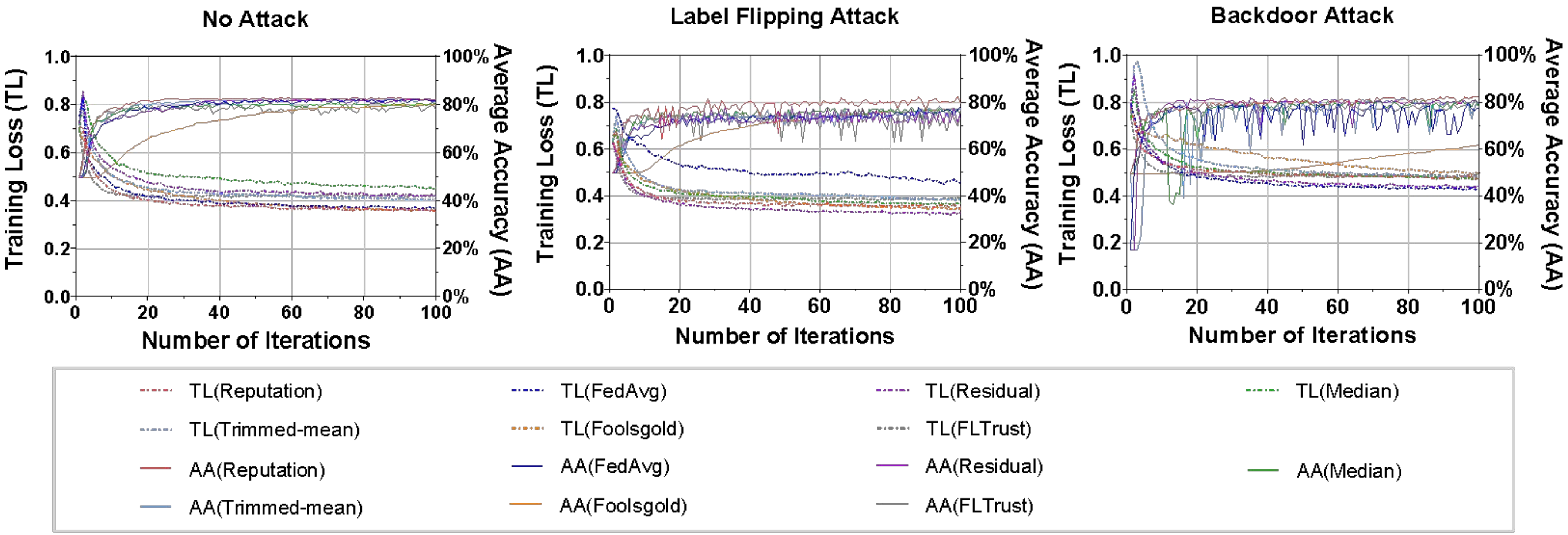


These figures show that even if attackers spread our poisoning over multiple iterations and then try to recover their reputation score by acting benignly, our detection scheme can still identify them.
This is because our attack detection and reputation schemes work in sequence. The attack detection scheme detects malicious updates without considering any reputation scores and rectifies them to mitigate damage.
Then, the reputation scheme modifies the reputation scores based on the detection results. Also, attackers that employ a higher number of model parameters suffer a slightly higher reduction of reputation, which is consistent with Corollary

Experimental Evaluation

- **Objectives**
  - Compare our method with other SOA
  - Use a text based real-world dataset of sensitive categories
  - Show our experimental result are consistent with theoretical analysis
  - Under different scenarios: no attack, under attack

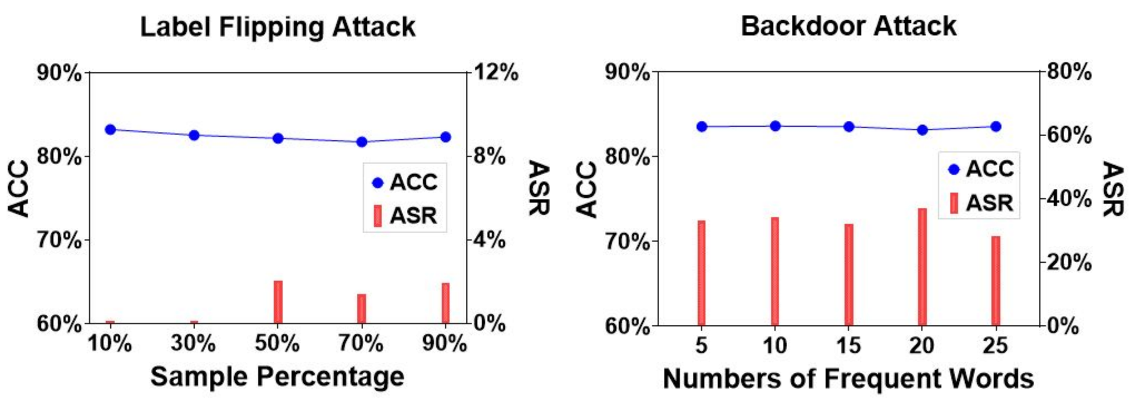
Convergence and Accuracy

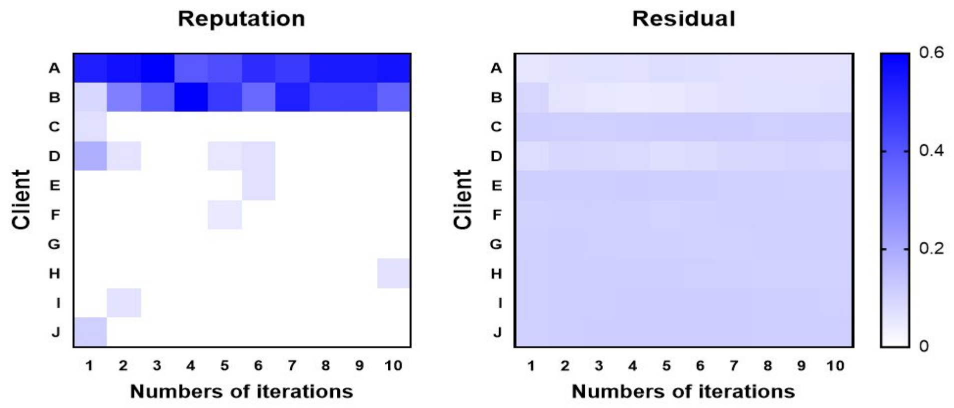


why converge faster:

This is due to the fact that in our algorithm we give higher weights to the clients with high-quality updates, as illustrated in Figure~\ref{fig:comparision}[nextt slides], causing the model to converge rapidly and retain consistent accuracy. In addition, even under the two different attacks, our method:

Other Evaluations



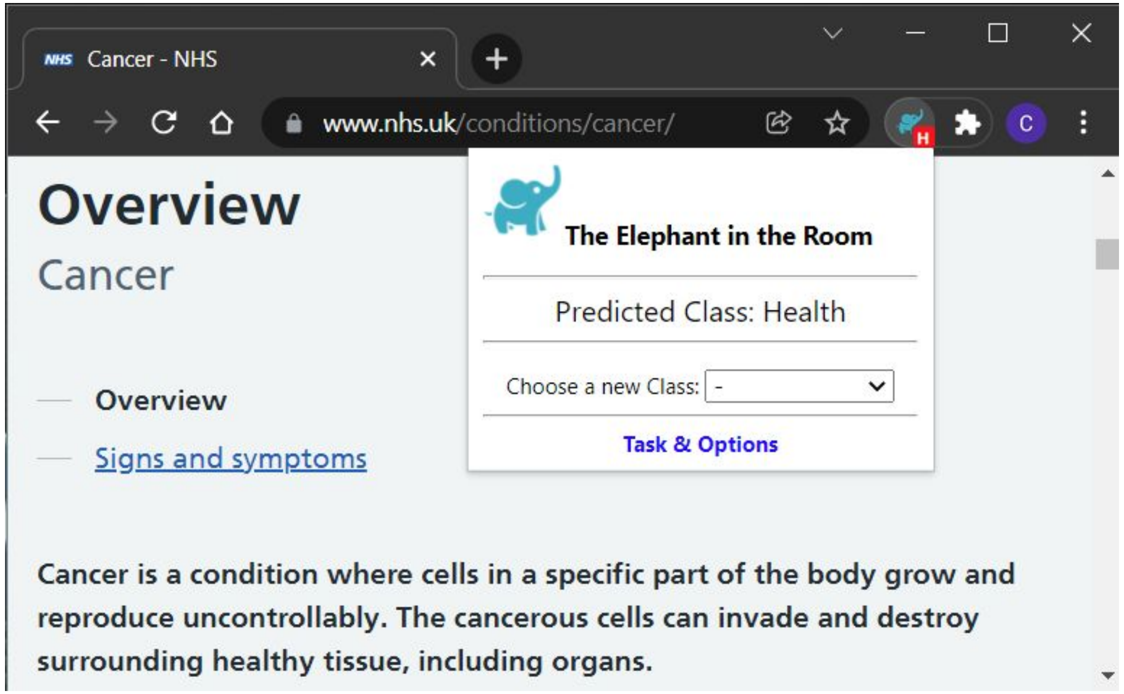


- The percentage of the poisoned sample is increased, it leads to the decrease of the accuracy of the model and to a slight increase of ASR

- our reputation method the aggregation weights of malicious clients, which are their reputations, are rectified near to 0, outperform the residual-based method

EITR System

- A research prototype to evaluate the robustness of our algorithm in a simple real-world setting with real users



The back-end server is responsible to distribute the initial classification model and the consequently updated model(s) to the clients, and receive new annotations from the different clients of the system.

Fake account

We agree that in the real world, attackers can create a large number of new accounts and launch a single attack to damage the reputation mechanism. One solution to multiple account creation is to attach participation via certificates to the real-world identities of users. The method of authenticating and binding identities will be implemented in future work.

References

1. Commission, E. Data protection in the EU, The General Data Protection Regulation (GDPR); Regulation (EU) 2016/679. (2018), https://ec.europa.eu/%0Ainfo/law/law-topic/data-protection
2. Matic, Srdjan, Costas Iordanou, Georgios Smaragdakis, and Nikolaos Laoutaris. "Identifying sensitive urls at web-scale." In *Proceedings of the ACM Internet Measurement Conference*, pp. 619-633. 2020.
3. McMahan, Brendan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. "Communication-efficient learning of deep networks from decentralized data." In *Artificial intelligence and statistics*, pp. 1273-1282. PMLR, 2017.

institute imdea networks