Verifying the Robustness of KNNs against Data-Poisoning Attacks

Yannan Li, Jingbo Wang, Chao Wang

10/19/2022

USC University of Southern California

fmcad

Formal Methods in Computer-Aided Design

Limitation of Prior Works

- Verifying n-poisoning robustness of KNNs
  - *Jia et al., Certified robustness of nearest neighbors against data poisoning attacks and backdoor attacks. AAAI 2022.*
  - **Only verifies part of problem (not handle complex "parameter tuning")**
- Verifying n-poisoning robustness of decision trees
  - *Drews et al., Proving data-poisoning robustness in decision trees. PLDI 2020.*
  - **Method works for decision trees only (but not for KNNs)**

Our method is the only method for the entire KNN algorithm and is more accurate than [Jia et al.] for the prediction step

Outline

- Background
  - Data Poisoning Attacks
  - KNNs (k-nearest neighbors)
- Data Poisoning Robustness of KNNs
- Our Method
- Evaluation
- Conclusion

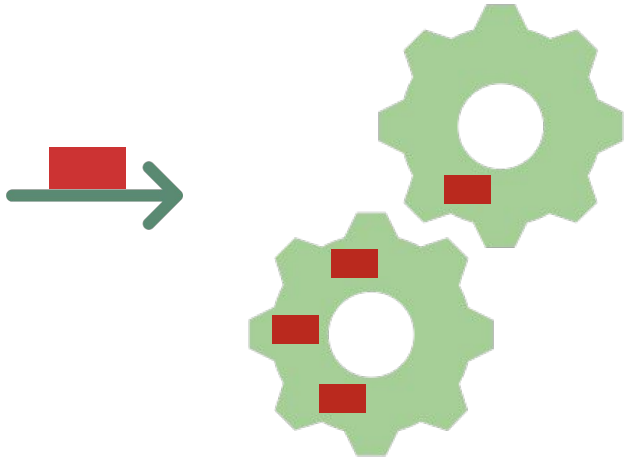
Outline

- Background
  - Data Poisoning Attacks
  - KNNs (k-nearest neighbors)
- Data Poisoning Robustness of KNNs
- Our Method
- Evaluation
- Conclusion

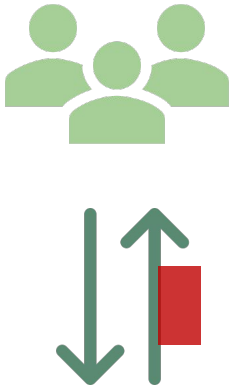
Background – *machine learning steps*

< 0.2% poisoning*

Collect Data



Train Model

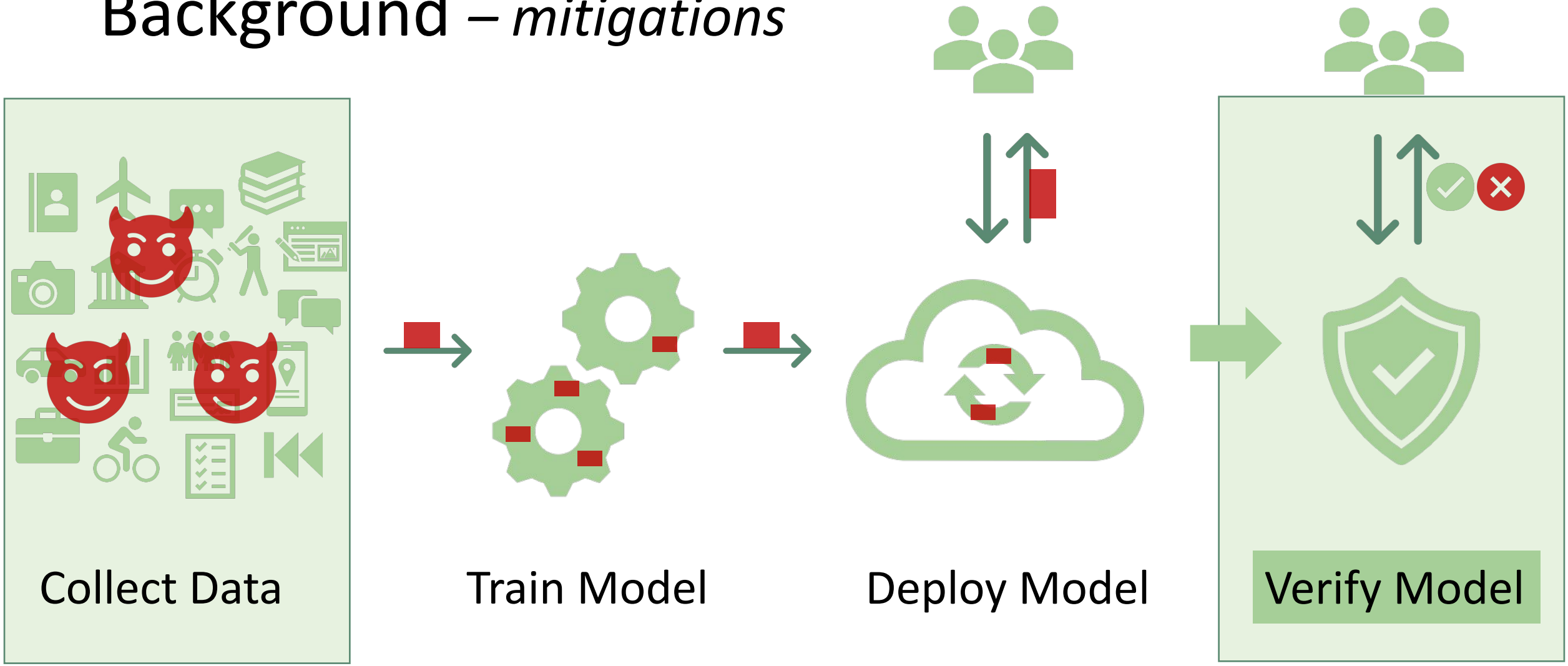


97% attack success rate*

Deploy Model

* Chen et al. attacked VGG-Face in "Targeted back-door attacks on deep learning systems using data poisoning", arXiv, 2017

Background – *mitigations*



Security Property – *n-poisoning robustness*

Training Dataset	Learned Model	Prediction Result of x
Current Dataset: **ABCD**	M	$M(x) = dog$
(n = 3) Possible Clean Dataset: ~~A~~BCD ~~AB~~CD AB~~C~~~~D~~ ~~A~~B~~C~~~~D~~ ~~A~~BCD ~~AB~~CD AB~~C~~~~D~~ ~~A~~B~~C~~~~D~~ ~~A~~BCD ~~AB~~CD AB~~C~~~~D~~ ~~A~~B~~C~~~~D~~ ~~A~~BCD ~~AB~~CD AB~~C~~~~D~~ ~~A~~B~~C~~~~D~~ 14 situations		

Combinatorial explosion!

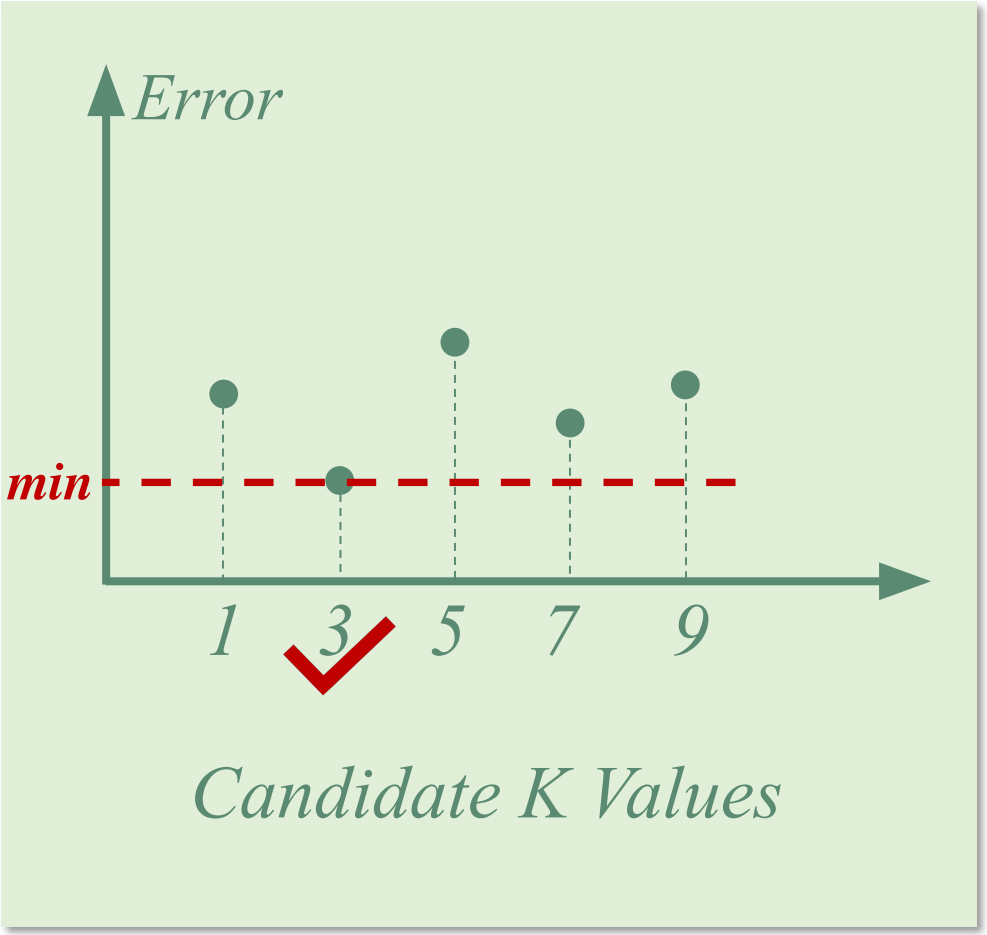
Training size =100 and n =5, almost $8*10^7$ situations!

Secure Definition: $\forall i,\ M_i(x) = M(x)$

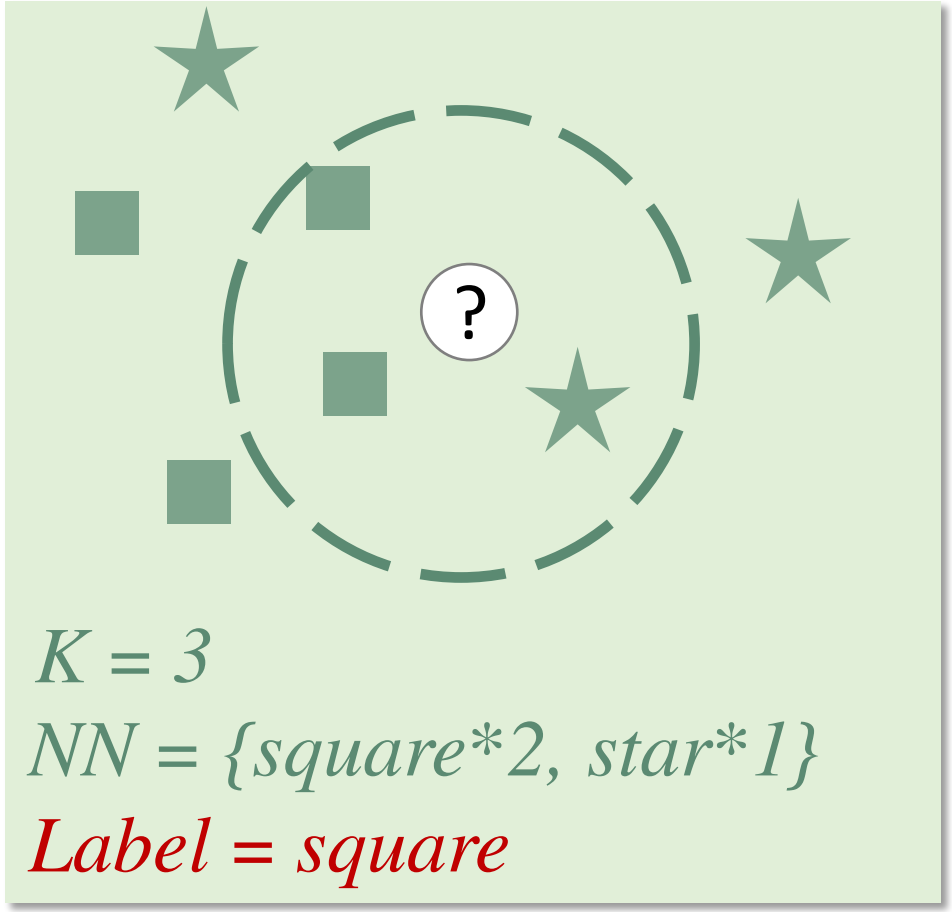
Outline

- Background
  - Data Poisoning Attacks
  - KNNs (k-nearest neighbors)
- Data Poisoning Robustness of KNNs
- Our Method
- Evaluation
- Conclusion

KNN (k-Nearest Neighbors)



Opt K



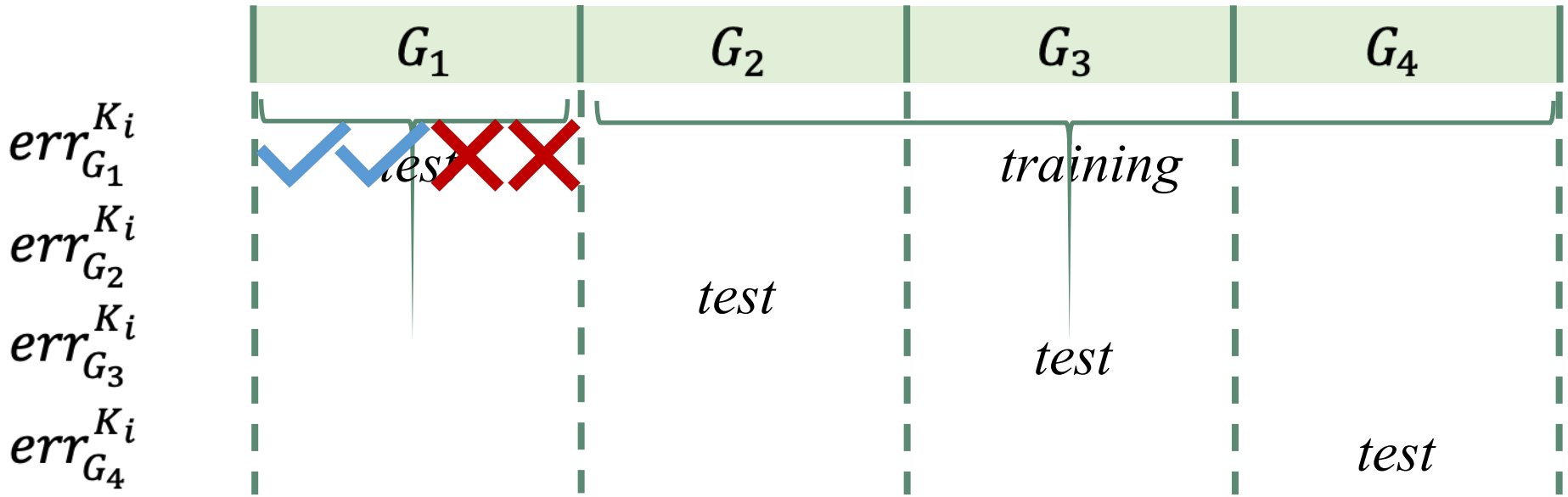
Parameter Tuning

Label Prediction

KNN parameter tuning: 4-fold cross validation

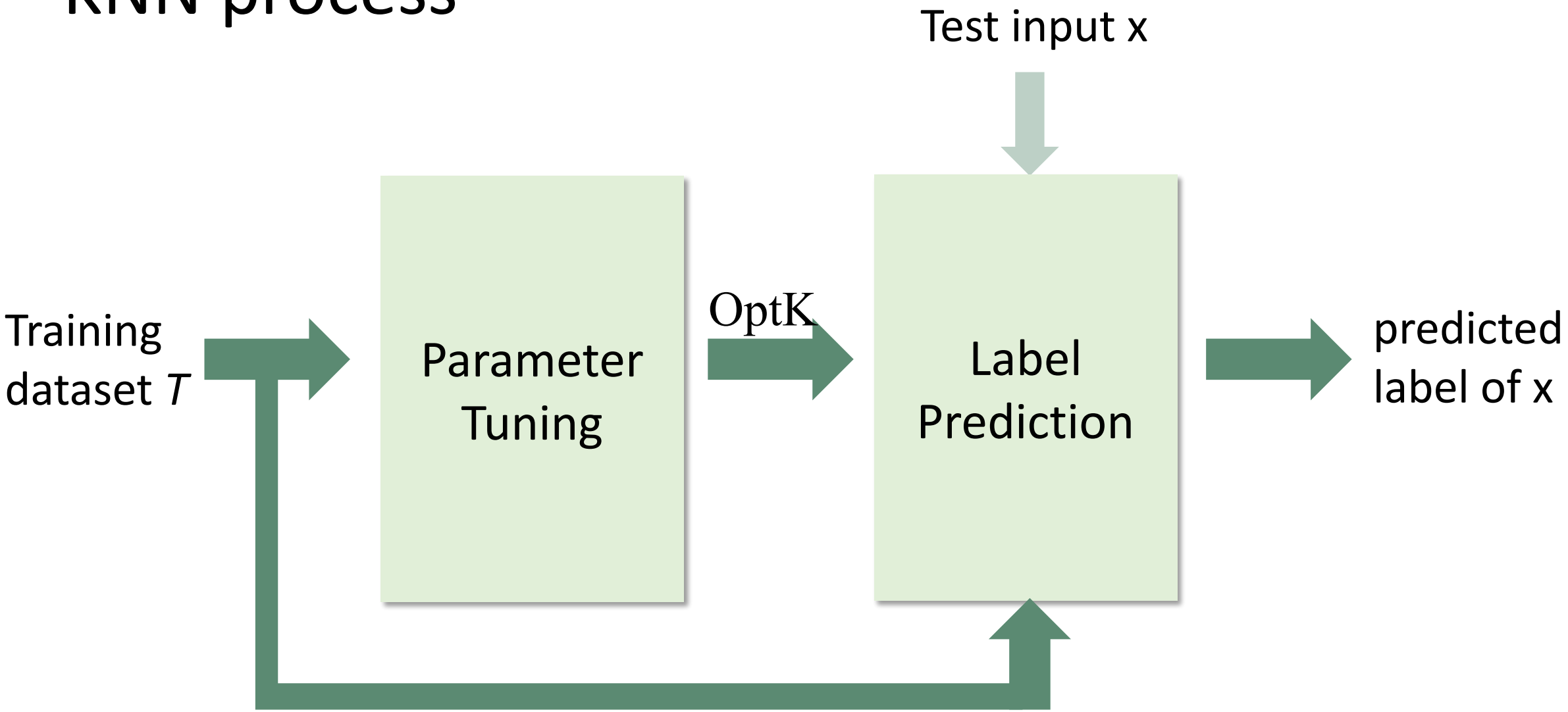
For one K_i

Training Dataset



$$err^{K_i} = \frac{1}{4}\sum_{j=1}^{4} err_{G_j}^{K_i}$$

KNN process


Test input x
Training dataset *T*
Parameter Tuning
OptK
Label Prediction
predicted label of x

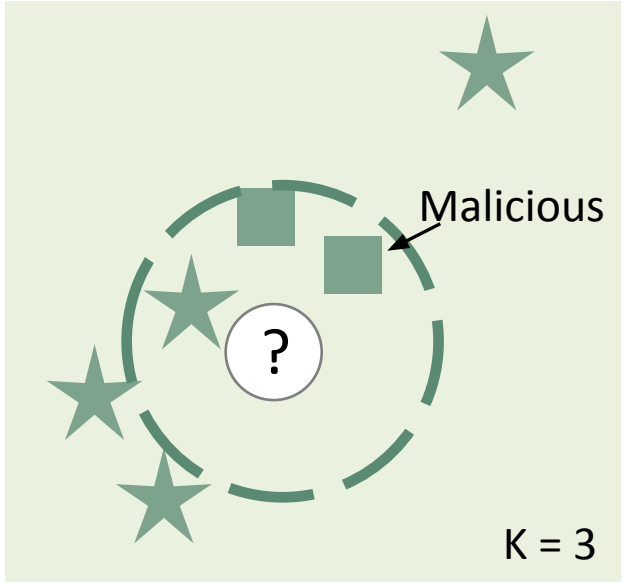
Outline

- Background
  - Data Poisoning Attacks
  - KNNs (k-nearest neighbors)
- Data Poisoning Robustness of KNNs
- Our Method
- Evaluation
- Conclusion

Poisoning Impact

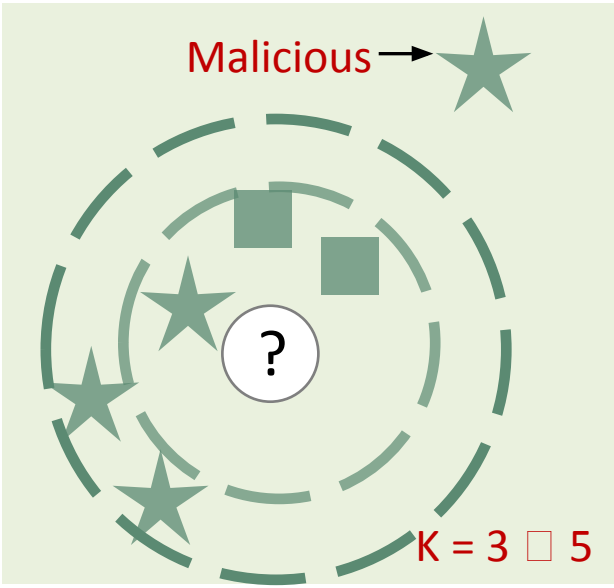
(1) Direct influence: change neighbors of test input x

- Only need to check poisoning situations near x

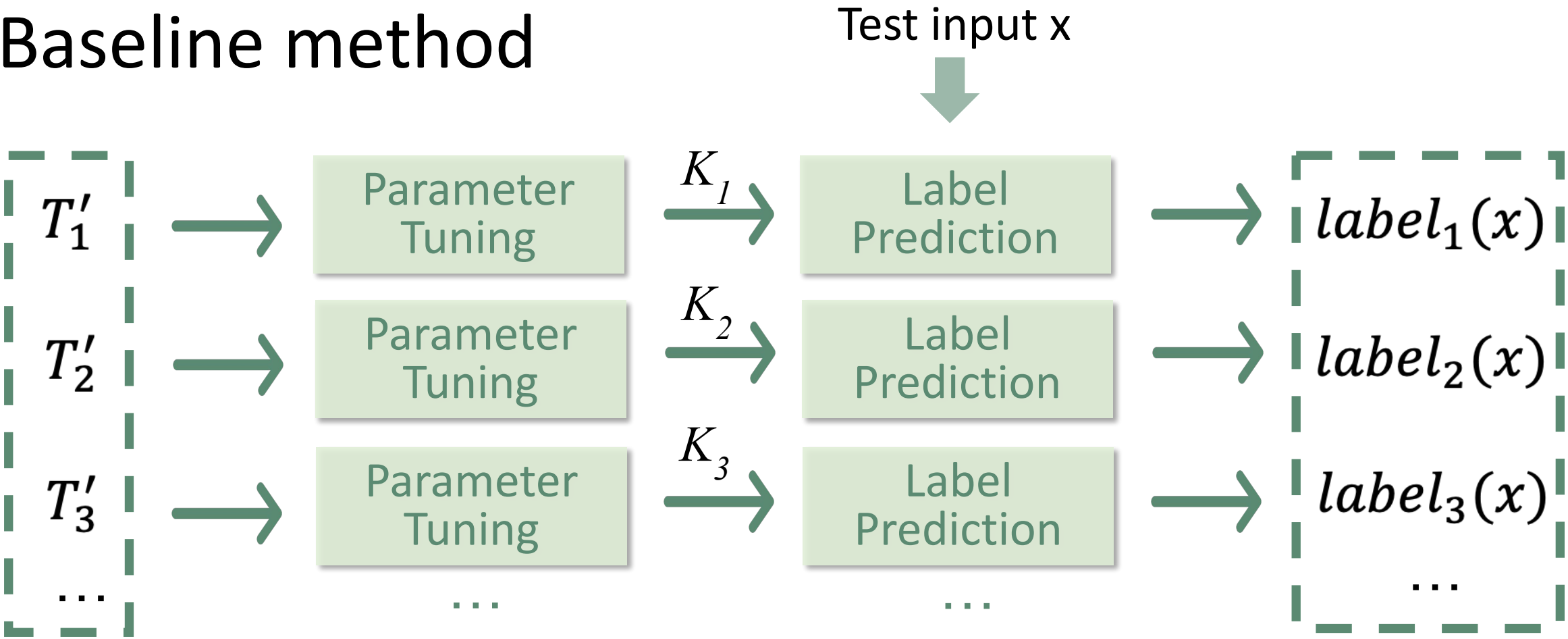


(2) ***Indirect influence***: change the optimal K

- **Need to check all the poisoning situations**



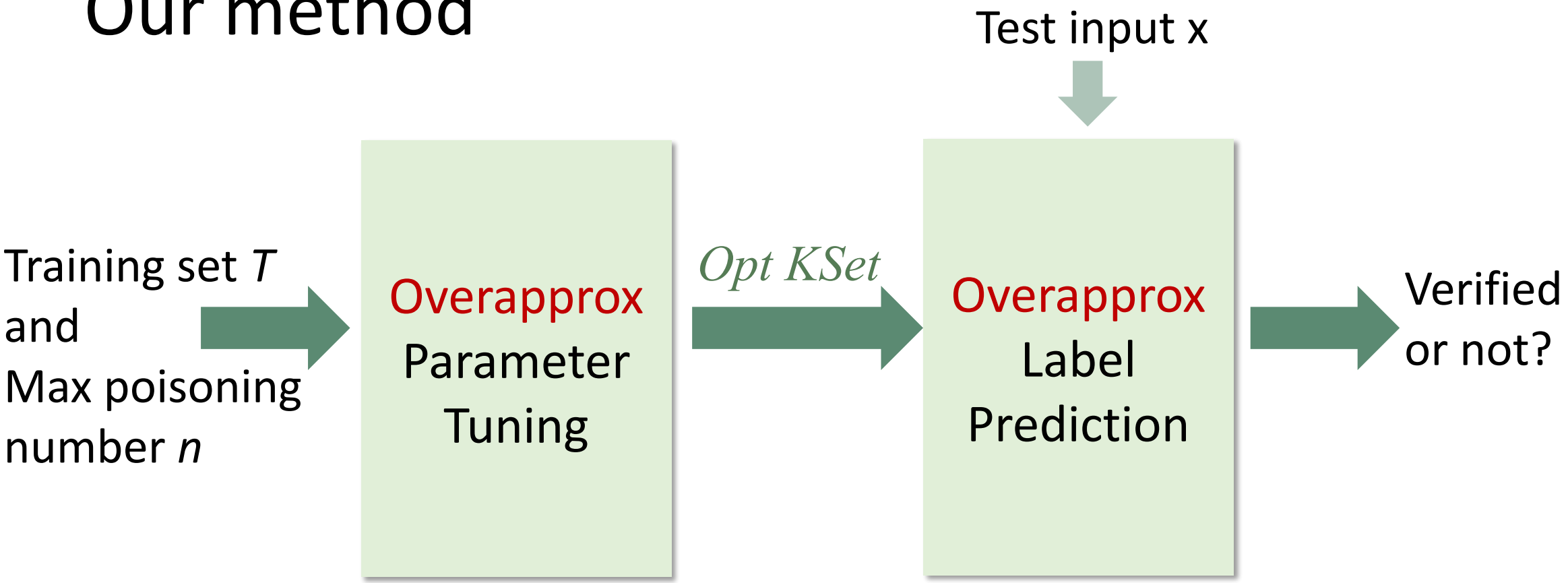
Baseline method



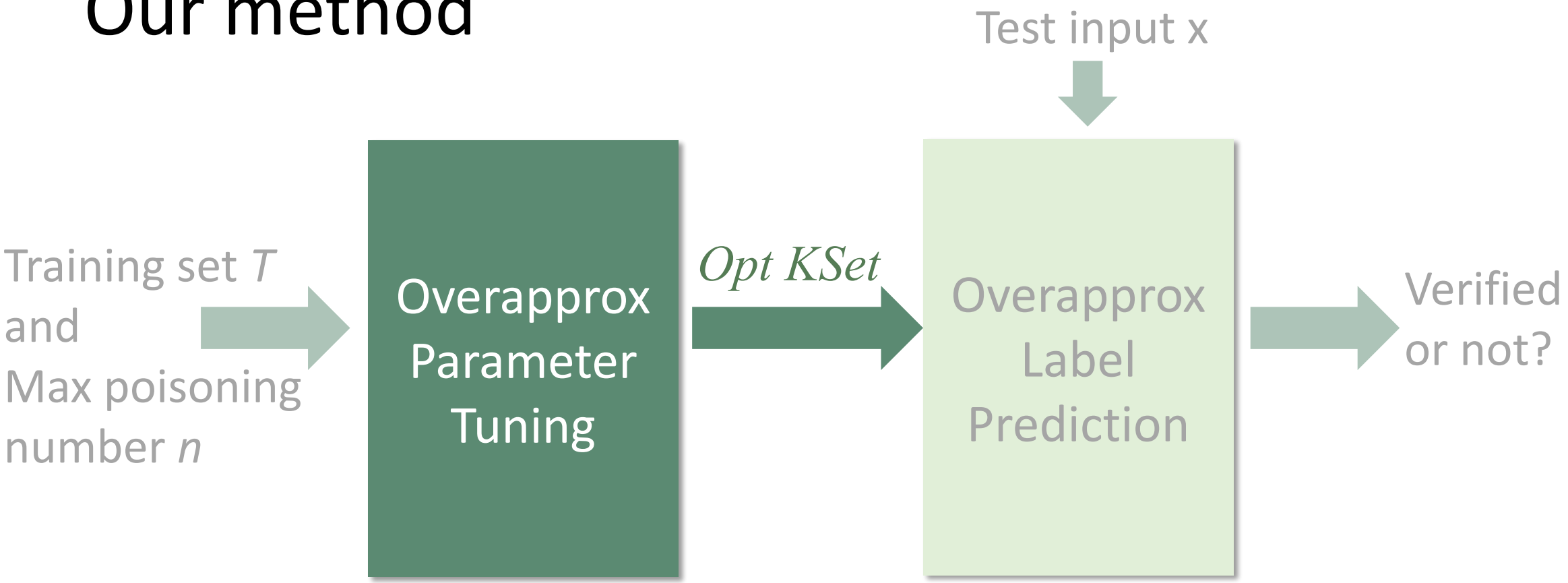
Outline

- Background
  - Data Poisoning Attacks
  - KNNs (k-nearest neighbors)
- Data Poisoning Robustness of KNNs
- Our Method
- Evaluation
- Conclusion

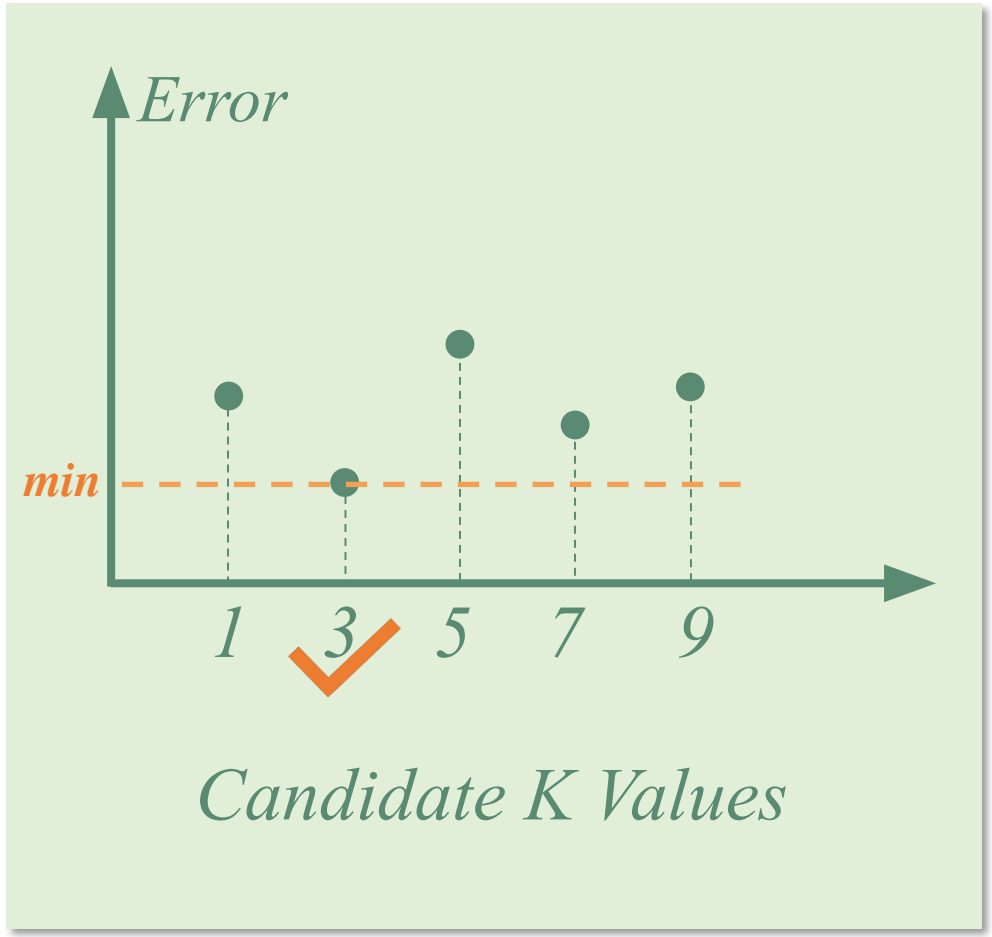
Our method


Test input x
Training set *T*
and
Max poisoning
number *n*
Overapprox
Parameter
Tuning
Opt KSet
Overapprox
Label
Prediction
Verified
or not?

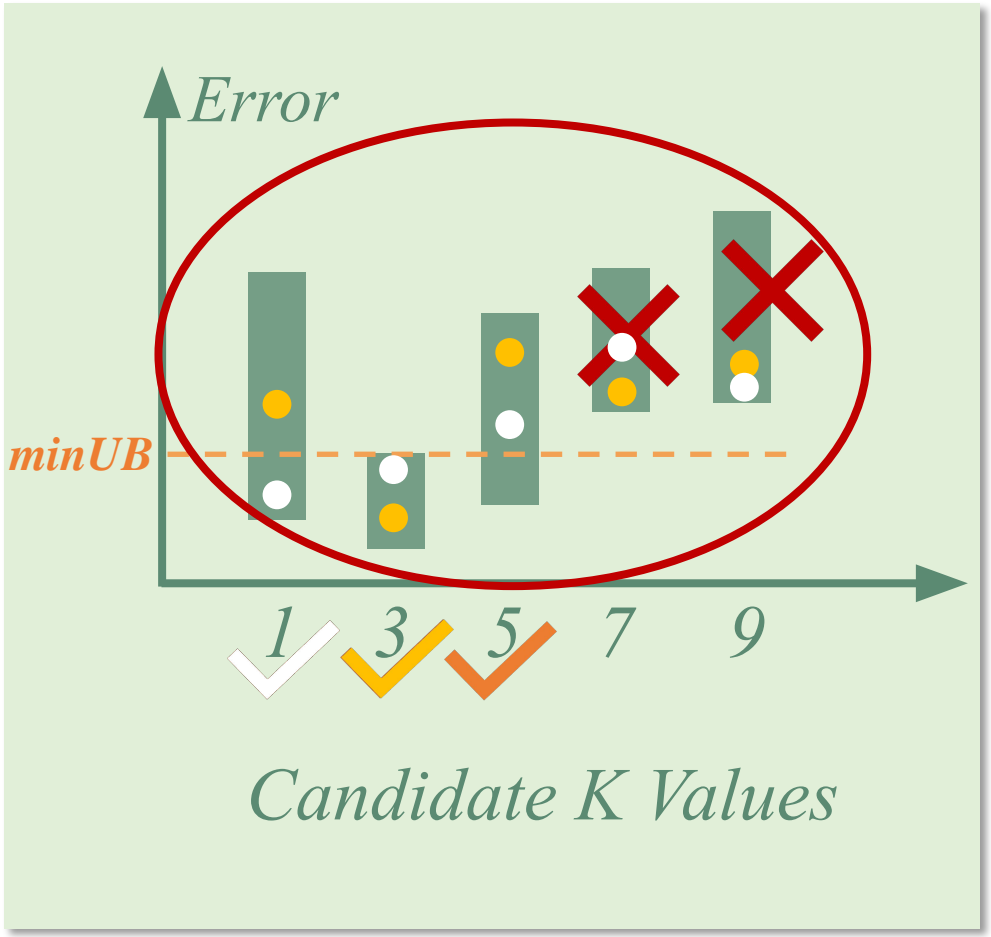
Our method


Test input x
Training set *T*
and
Max poisoning
number *n*
Overapprox
Parameter
Tuning
Opt KSet
Overapprox
Label
Prediction
Verified
or not?

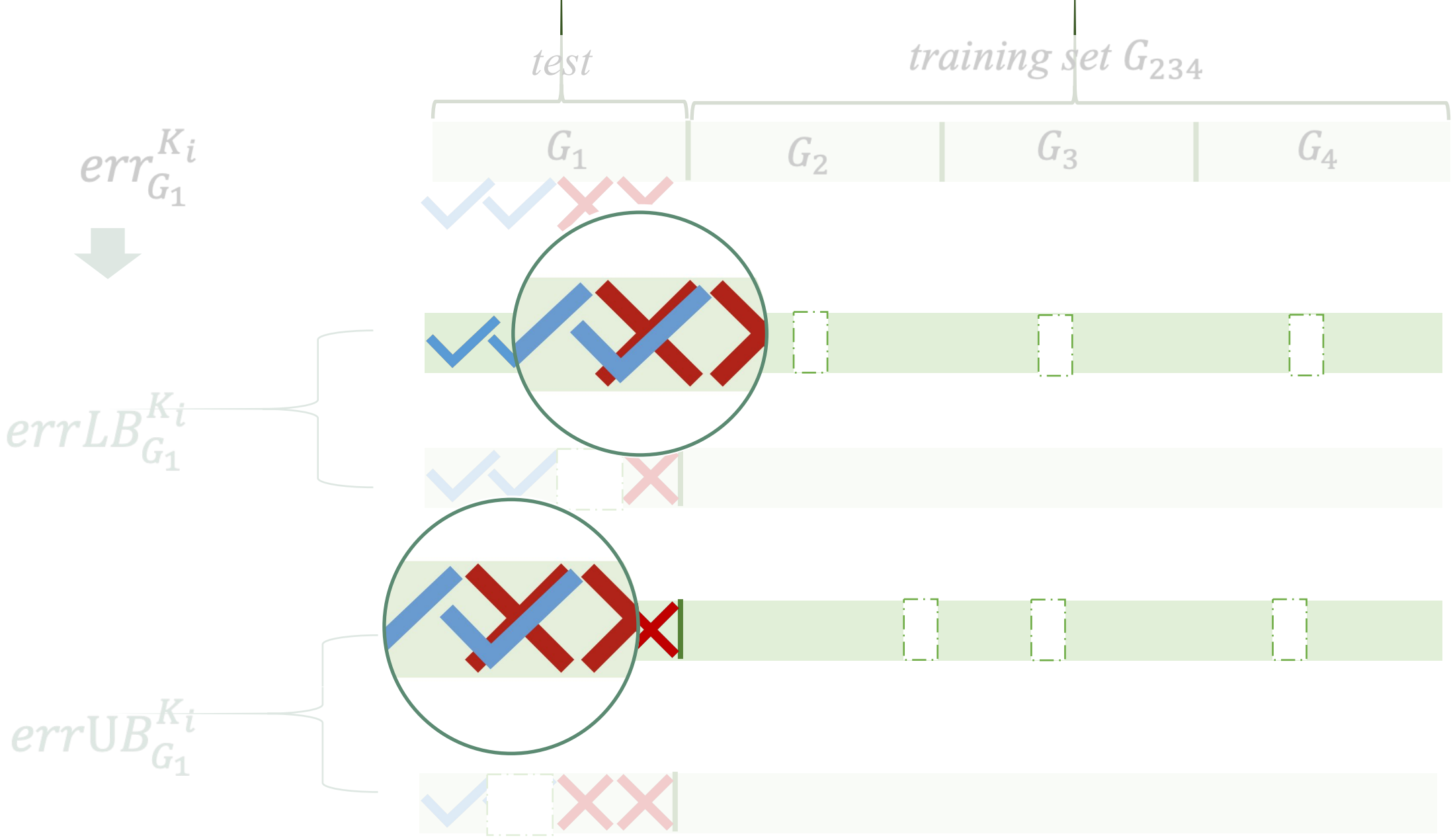
Our method – *Overapprox Parameter Tuning*



(Original) Parameter Tuning



Overapprox Parameter Tuning


test
training set G_{234}
G_1
G_2
G_3
G_4
$err_{G_1}^{K_i}$
$errLB_{G_1}^{K_i}$
$errUB_{G_1}^{K_i}$

Our method – *label changes via removal*

Remove 1 neighbors: Consider K+1 neighbors \ 1 points

Remove 2 neighbors: Consider K+2 neighbors \ 2 points

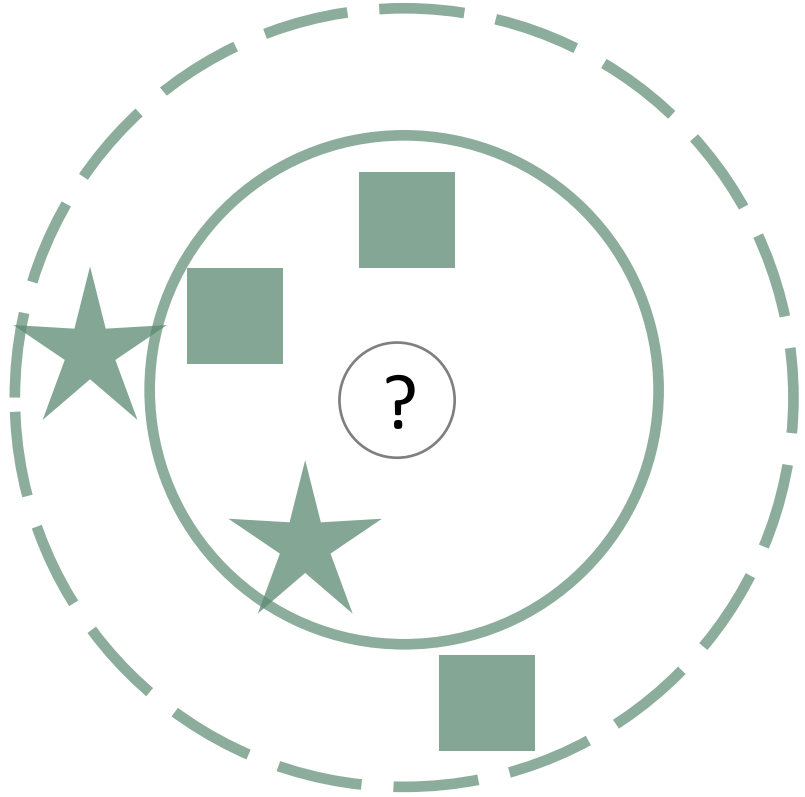
Remove 3 neighbors: Consider K+3 neighbors \ 3 points

…

Theorem: Just need to consider removing ≤n points from K+n nearest neighbors.

Our method – *"Misclassified" becomes "Correctly Classified"*

Intuition: Remove other labels



Current Label: Square (Misclassified)

New Label: Star (Correct)

K=3, n=2, and star being correct

Our method – *"Correctly Classified" becomes "Misclassified"*

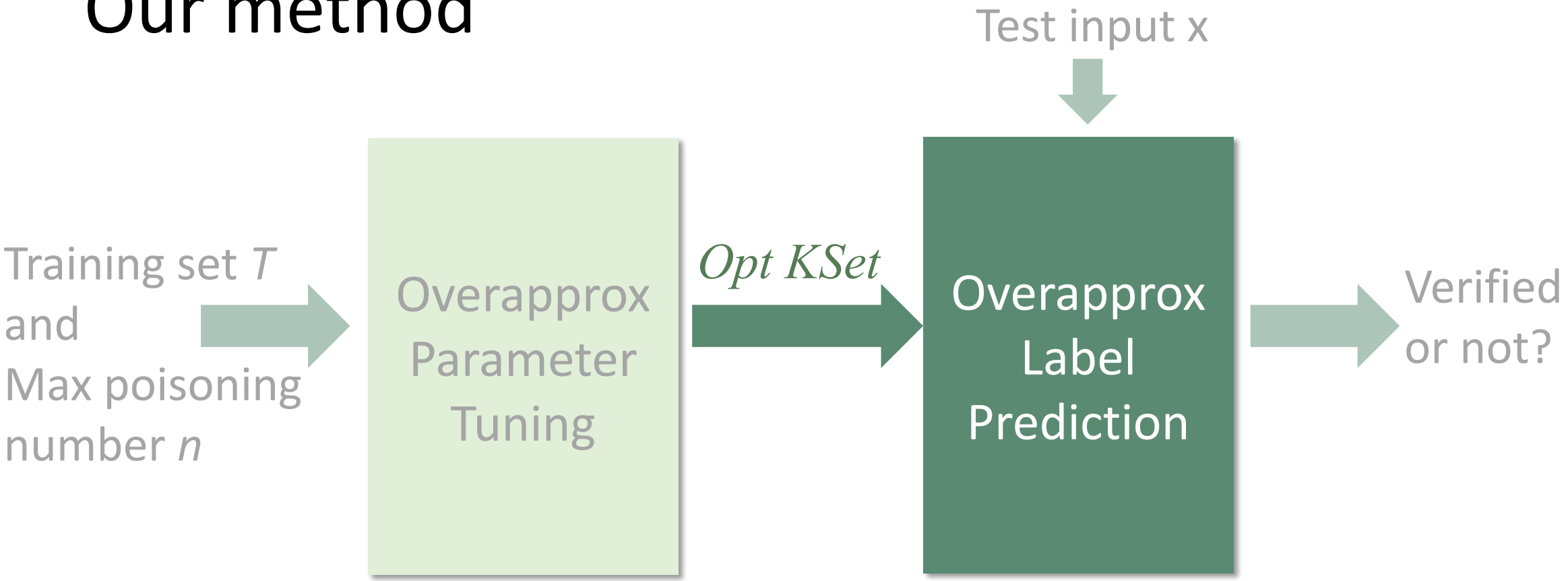
Intuition: Remove correct labels

Current Label: Star (Correct)

New Label: Star (Correct)

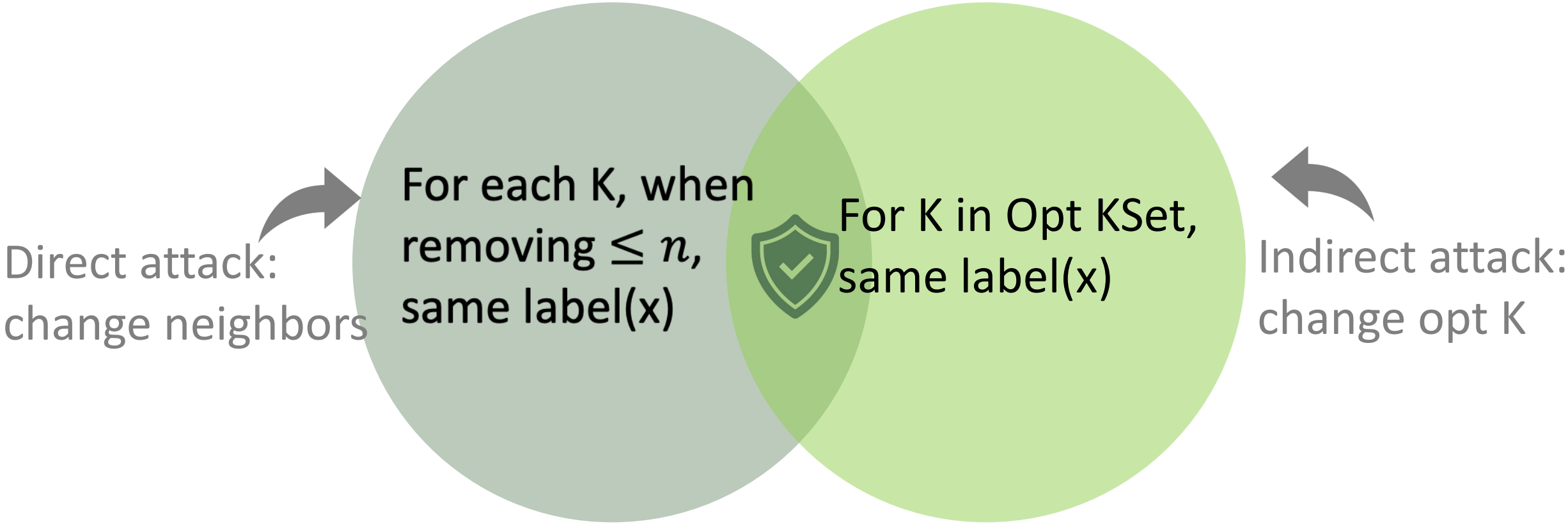
K=3, n=2, and *star* being correct

Our method


Test input x
Training set *T*
and
Max poisoning
number *n*
Overapprox
Parameter
Tuning
Opt KSet
Overapprox
Label
Prediction
Verified
or not?

Our method – *overapprox prediction*

- Input: Optimal KSet, test x, training T, poisoning n
- Output: label(x) remains the same?



Outline

- Background
  - Data Poisoning Attacks
  - KNNs (k-nearest neighbors)
- Data Poisoning Robustness of KNNs
- Our Method
- Evaluation
- Conclusion

Experimental Set Up

- Benchmarks
  - 2 small datasets
  - 4 larger datasets

- Research Questions
  - RQ1: **Accuracy** in proving n-poisoning robustness:
    - Compared to the baseline method (to obtain ground truth on small datasets)
    - Compared to the state of the art [Jia et al, AAAI 2022]
  - RQ2: **Efficiency** in handling realistic datasets:
    - Evaluated using the larger datasets

Benchmarks

	Name	#Training	#Test (x)	#Class (output)	#Feature (in)
Small Datasets	Iris	135	15	3	4
	Digits	1,617	180	10	64
Larger Datasets	HAR	9,784	515	6	561
	Letter	18,999	1,000	10	36
	MNIST	60,000	10,000	10	36
	CIFAR10	50,000	10,000	10	288

Results – *speed and accuracy on small datasets*

Max Poisoning	Baseline Time (s)	Our Time (s)	Accuracy
n = 1	60	1	93.3%
n = 2	4770	1	93.3%
n = 3	>9999	1	-

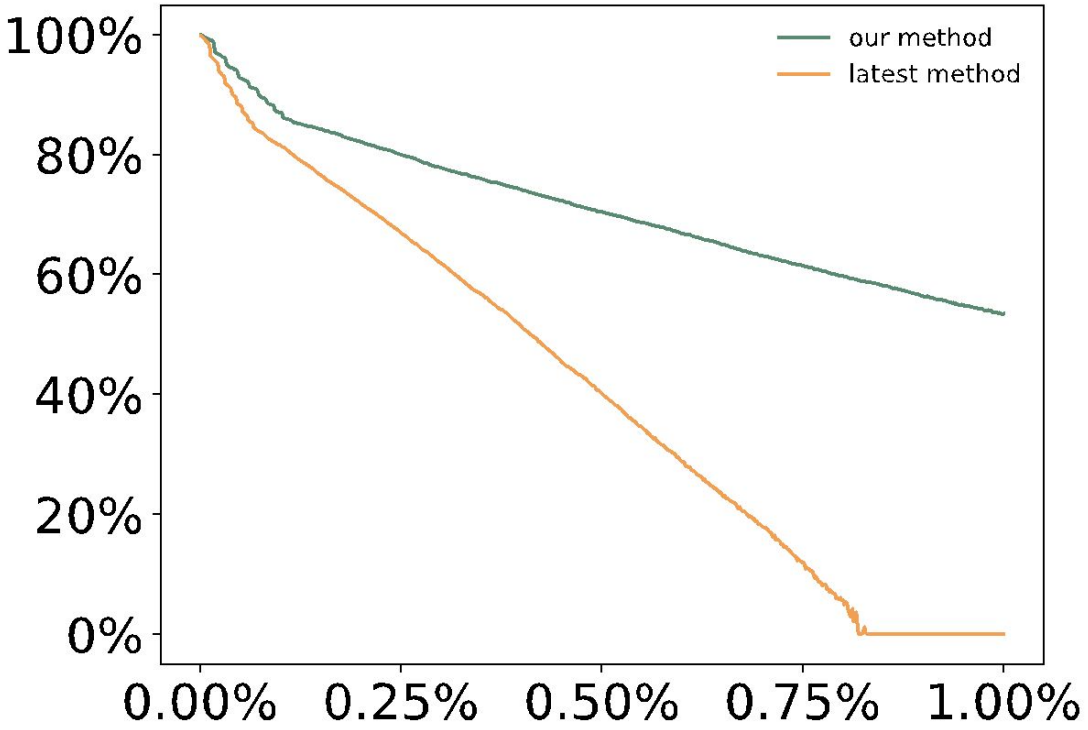
Iris (#training=135, #test = 15, #class=3, #feature=4)

Max Poisoning	Baseline Time (s)	Our Times (s)	Accuracy
n = 1	8032	1	96.1%
n = 2	>9999	1	-

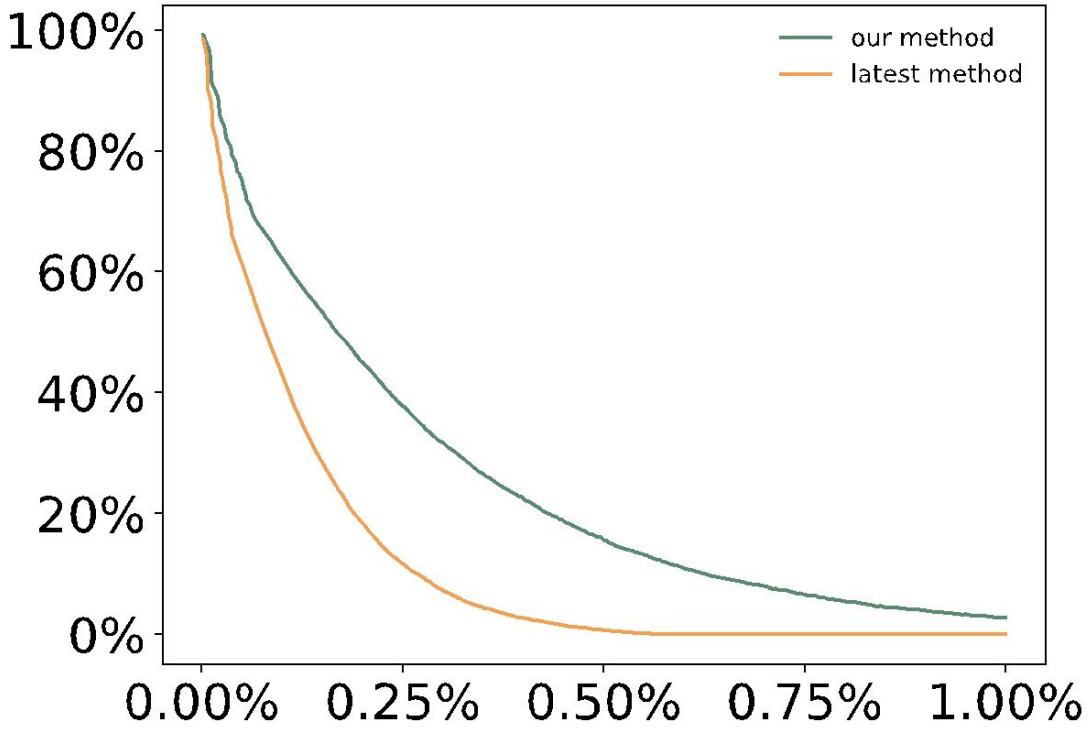
Digits (#training=1617, #test = 180, #class=10, #feature=64)

- Our method is several orders-of-magnitude faster than the baseline
- Accuracy > 93%

Result - *speed and accuracy on large datasets*



MNIST (time=16min, #training=60000, #test = 10000, #class=10, #feature=36)



CIFAR10 (time = 25min, #train=50000, #test = 10000, #class=10, #feature=288)

- Existing method* can only verify prediction phase
- Existing Method* can verify much less percentage

*Jia et al., Certified robustness of nearest neighbors against data poisoning attacks. AAAI 2022.

Conclusion

- The first method for soundly verifying n-poisoning robustness for the entire KNN algorithm
  - parameter tuning step + prediction step

- Demonstrated its accuracy and efficiency on popular supervised-learning datasets
  - small datasets + larger datasets