Attacking DBSCAN for Fun and Profit

Jonathan Crussell, Philip Kegelmeyer
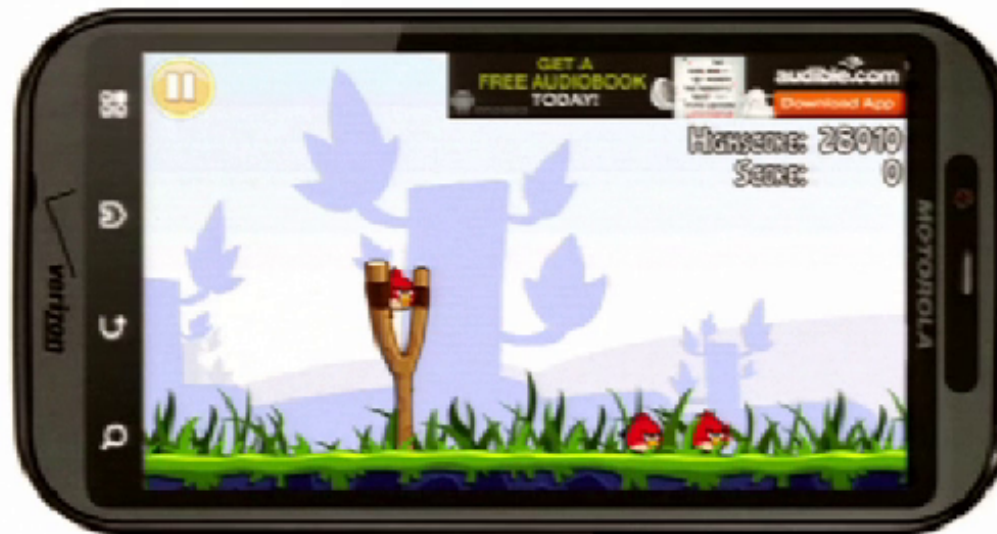
Sandia National Laboratories, California[1]

April 30th, 2015
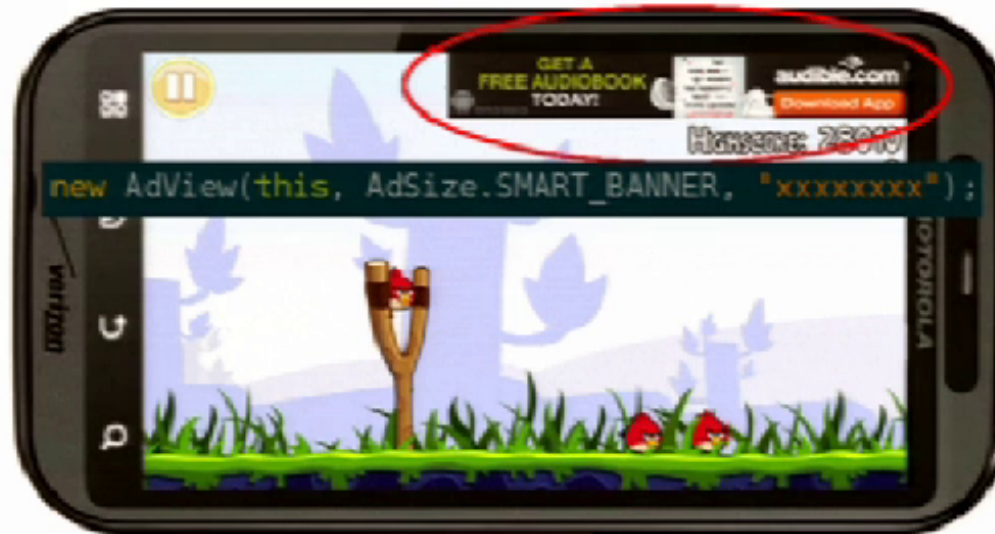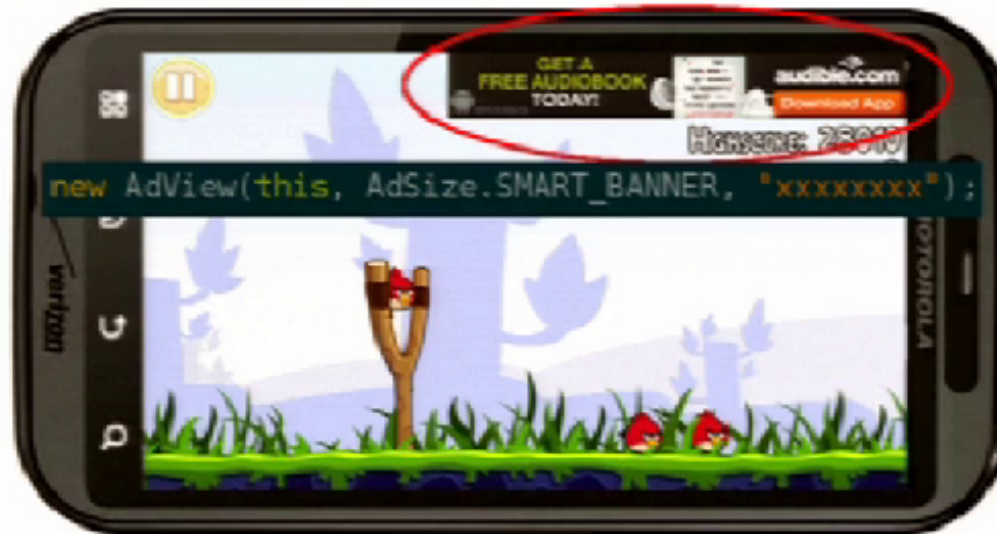
App Plagiarism

App Plagiarism

App Plagiarism

Miscreants copy apps to siphon ad revenue

- Gibler et al. (MobiSys'13) estimate losses of 14%

AnDarwin

AnDarwin (Crussell et al., ESORICS'14):

- Crawled 265K apps from 17 Android markets

AnDarwin

AnDarwin (Crussell et al., ESORICS'14):

- Crawled 265K apps from 17 Android markets
- Detected copied apps via clustering based on DBSCAN
- One application: plagiarism detection

AnDarwin

AnDarwin (Crussell et al., ESORICS'14):

- Crawled 265K apps from 17 Android markets
- Detected copied apps via clustering based on DBSCAN
- One application: plagiarism detection
- Designed to be robust to attacks against data representation

AnDarwin

AnDarwin (Crussell et al., ESORICS'14):

- Crawled 265K apps from 17 Android markets
- Detected copied apps via clustering based on DBSCAN
- One application: plagiarism detection
- Designed to be robust to attacks against data representation
- *Not* designed to be robust to attacks against data analysis

AnDarwin
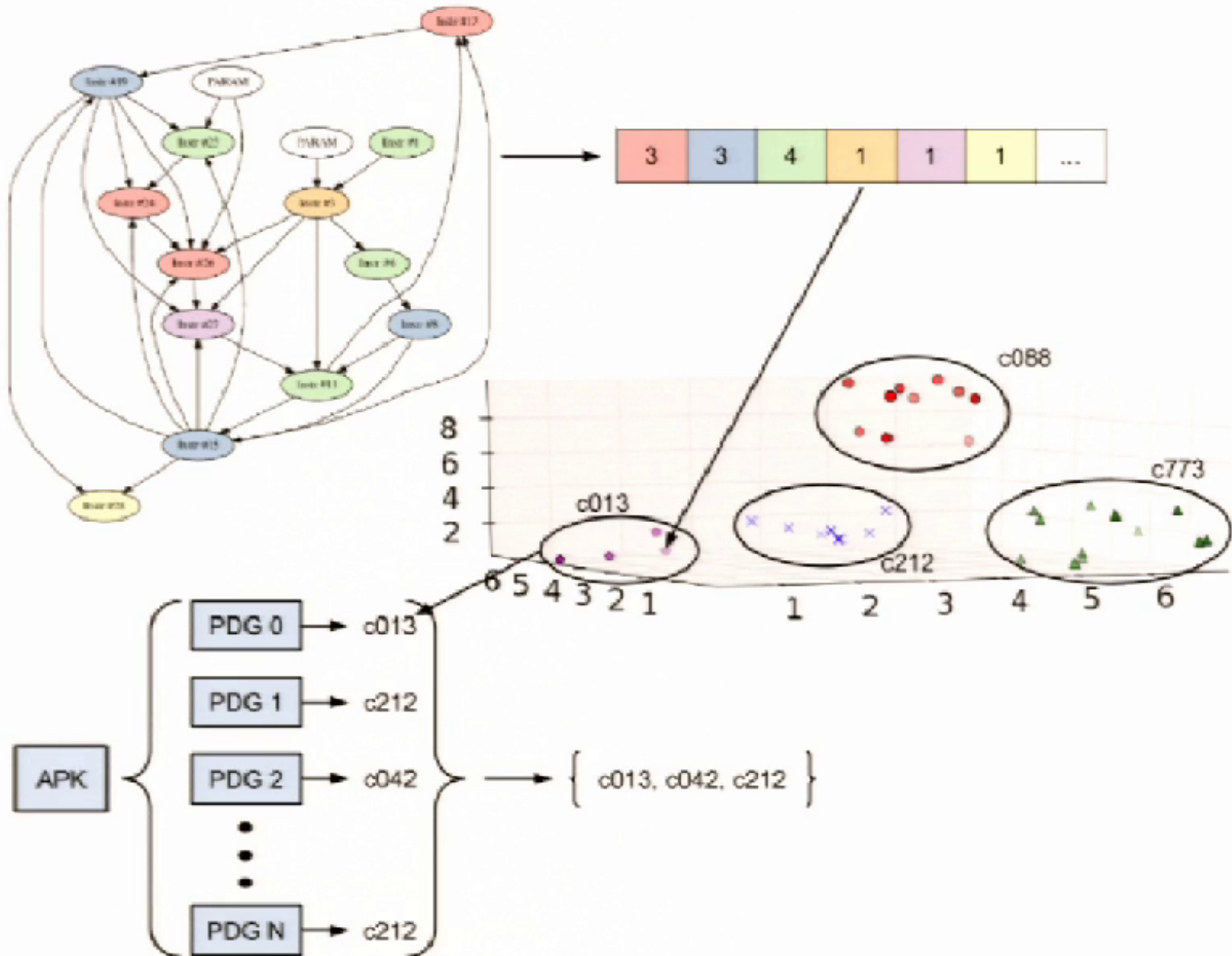
Thinking like an Adversary

What goals might an adversary have?

- Avoid being clustered with similar apps
- Favorably alter clustering structure
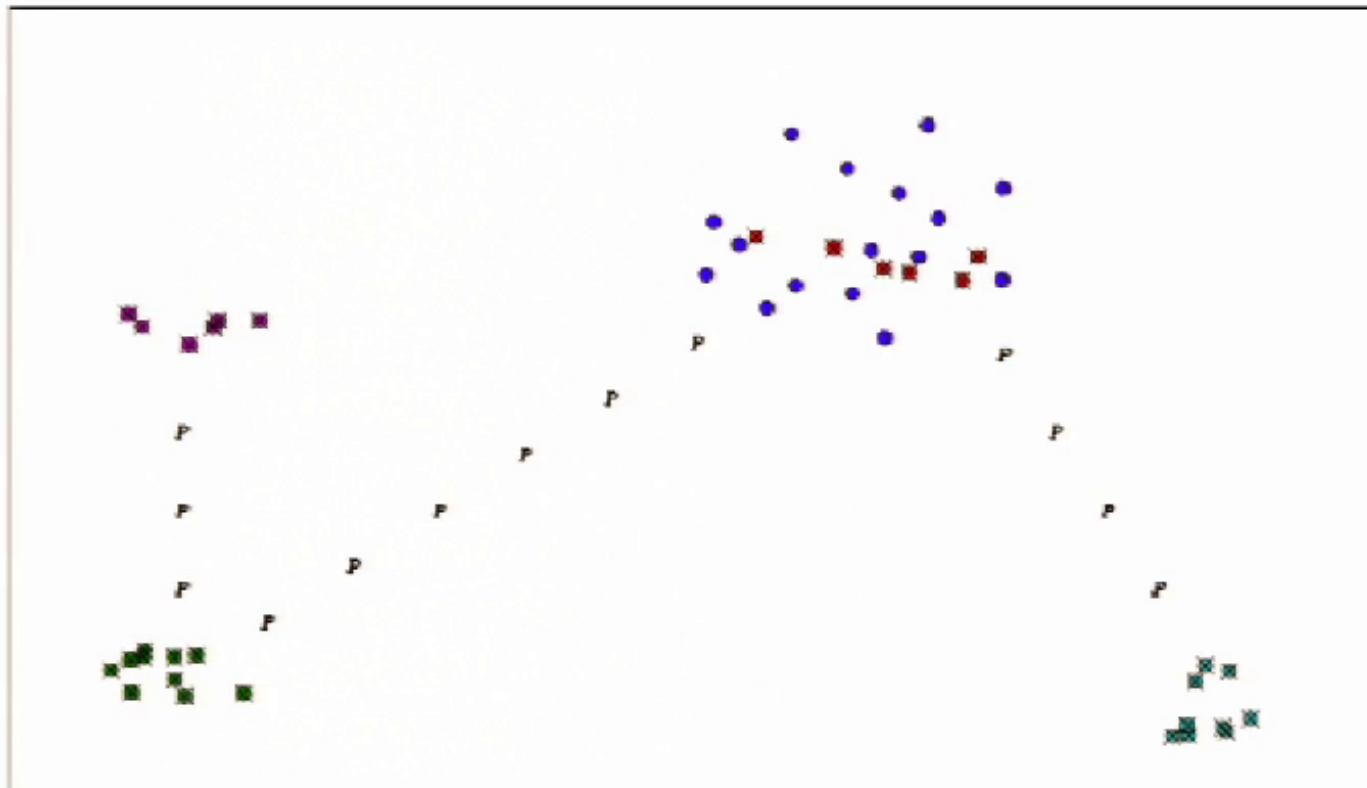- ...

Thinking like an Adversary

What goals might an adversary have?

- Avoid being clustered with similar apps
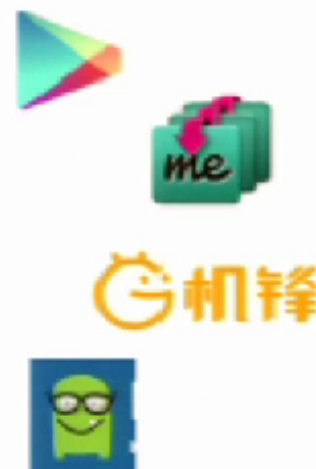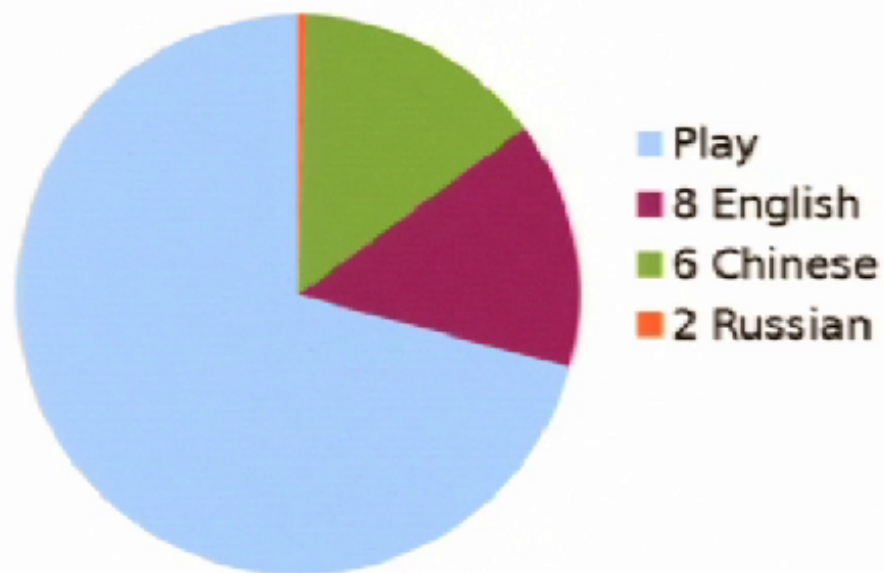- Favorably alter clustering structure
- ...

Confidence Attack

- Inject new points into dataset to poison the clustering
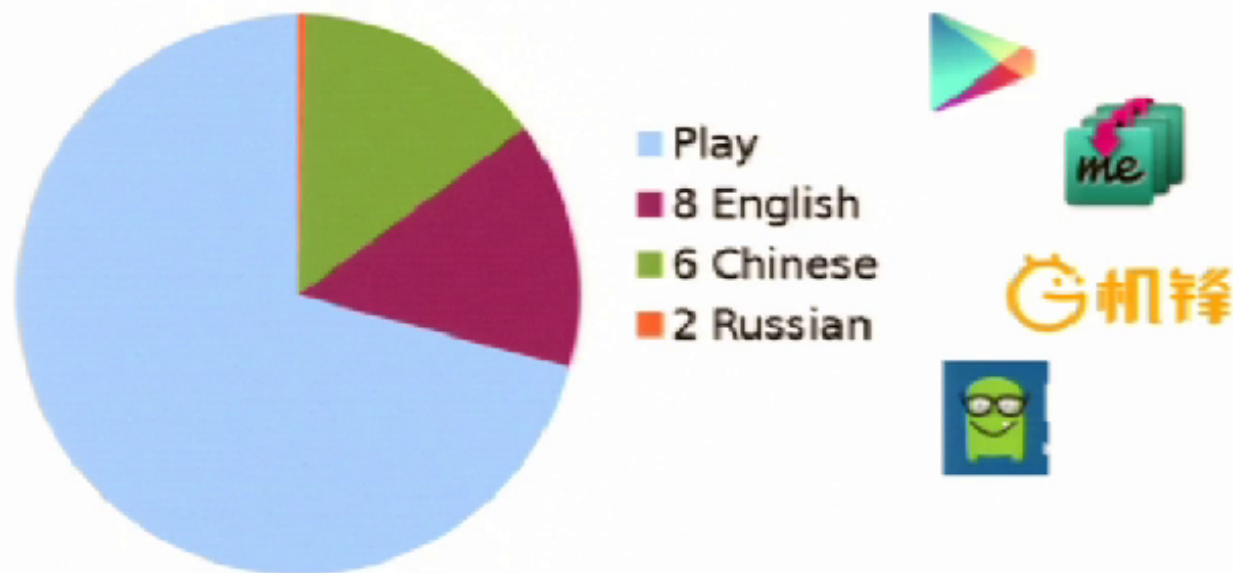
Confidence Attack

Is this Feasible?

In most cases, we analyze "found data:"

- Play
- 8 English
- 6 Chinese
- 2 Russian

Is this Feasible?

In most cases, we analyze "found data:"

- Play
- 8 English
- 6 Chinese
- 2 Russian

Semantic Gap (Jana and Shmatikov, IEEE S&P'12)

- Program analysis vs program execution

Attack Methodology

1. Pick two clusters to merge

Attack Methodology

1. Pick two clusters to merge
2. Generate series of optimal data mines between two clusters
3. Goto 1 until all desired merges completed

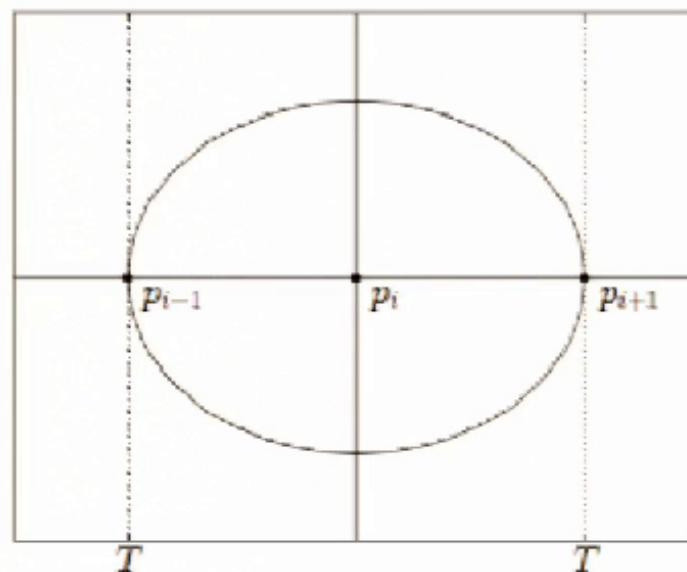# Generating Data Mines

AnDarwin represents apps as sets

- Minimum Jaccard similarity threshold T

Generating Data Mines

AnDarwin represents apps as sets

- Minimum Jaccard similarity threshold T

Generate points exactly T-width apart:

Generating Data Mines

DBSCAN (Ester et al., KDD'96):

- Core point has $>=$ *MinPts* neighbors in T-neighborhood
- Clusters form around a core point:
  - Other core points that are at least T similar to a core point already in the cluster
  - Points in the T-neighborhood of a core point

Generating Data Mines

DBSCAN (Ester et al., KDD'96):

- Core point has $>=$ *MinPts* neighbors in *T*-neighborhood
- Clusters form around a core point:
  - Other core points that are at least *T* similar to a core point already in the cluster
  - Points in the *T*-neighborhood of a core point

Generate points to match *MinPts*:
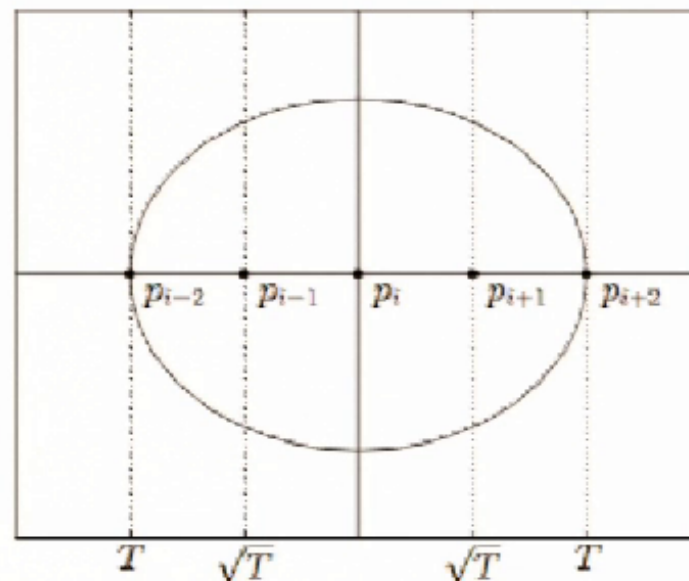
Which Clusters to Merge?

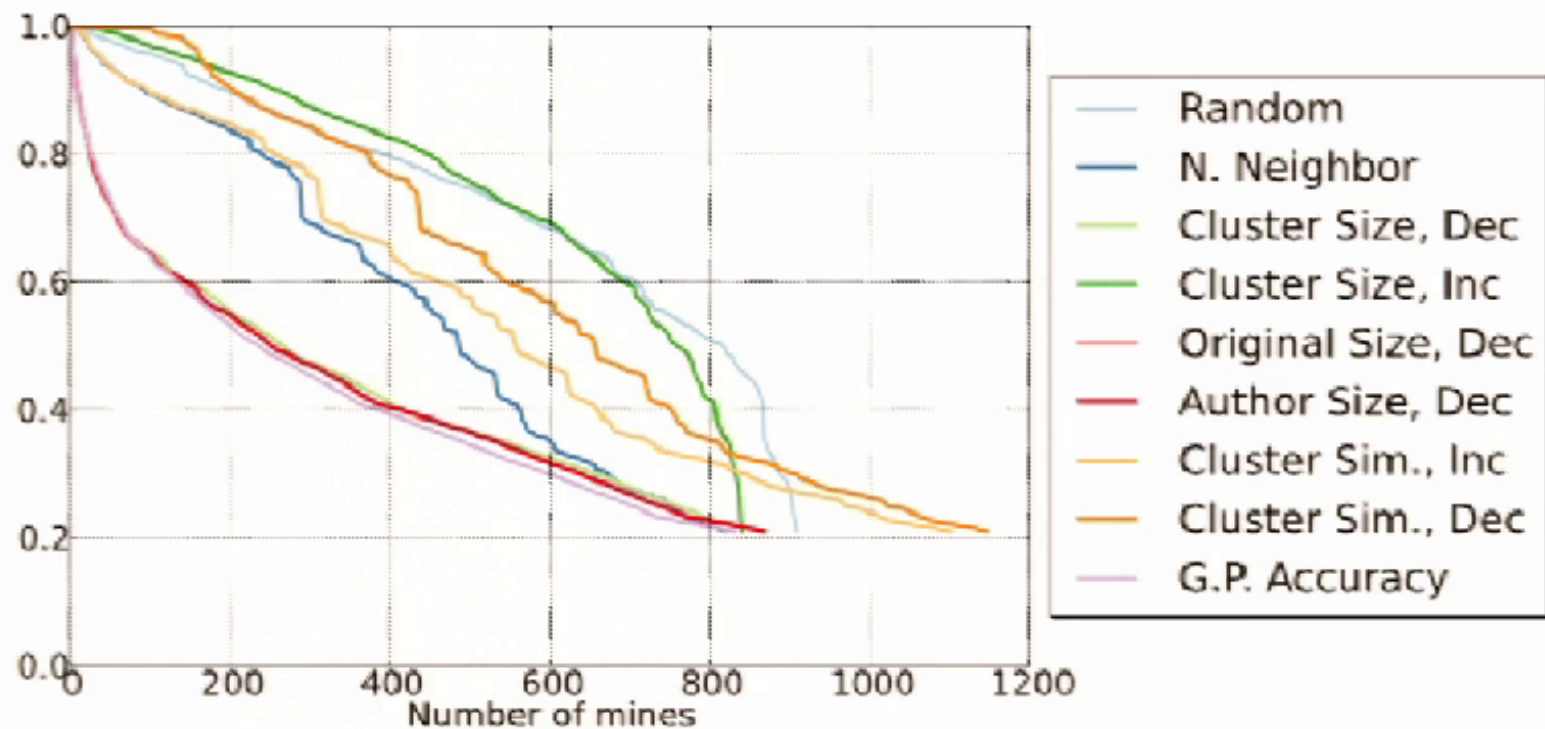Depends on adversary goals (and, perhaps, budget)

Which Clusters to Merge?

Depends on adversary goals (and, perhaps, budget)

- Maximally degrade plagiarism detection accuracy

Which Clusters to Merge?

Depends on adversary goals (and, perhaps, budget)

- Maximally degrade plagiarism detection accuracy

Dataset: 273 randomly selected clusters (1,394 apps total)

Defenses?

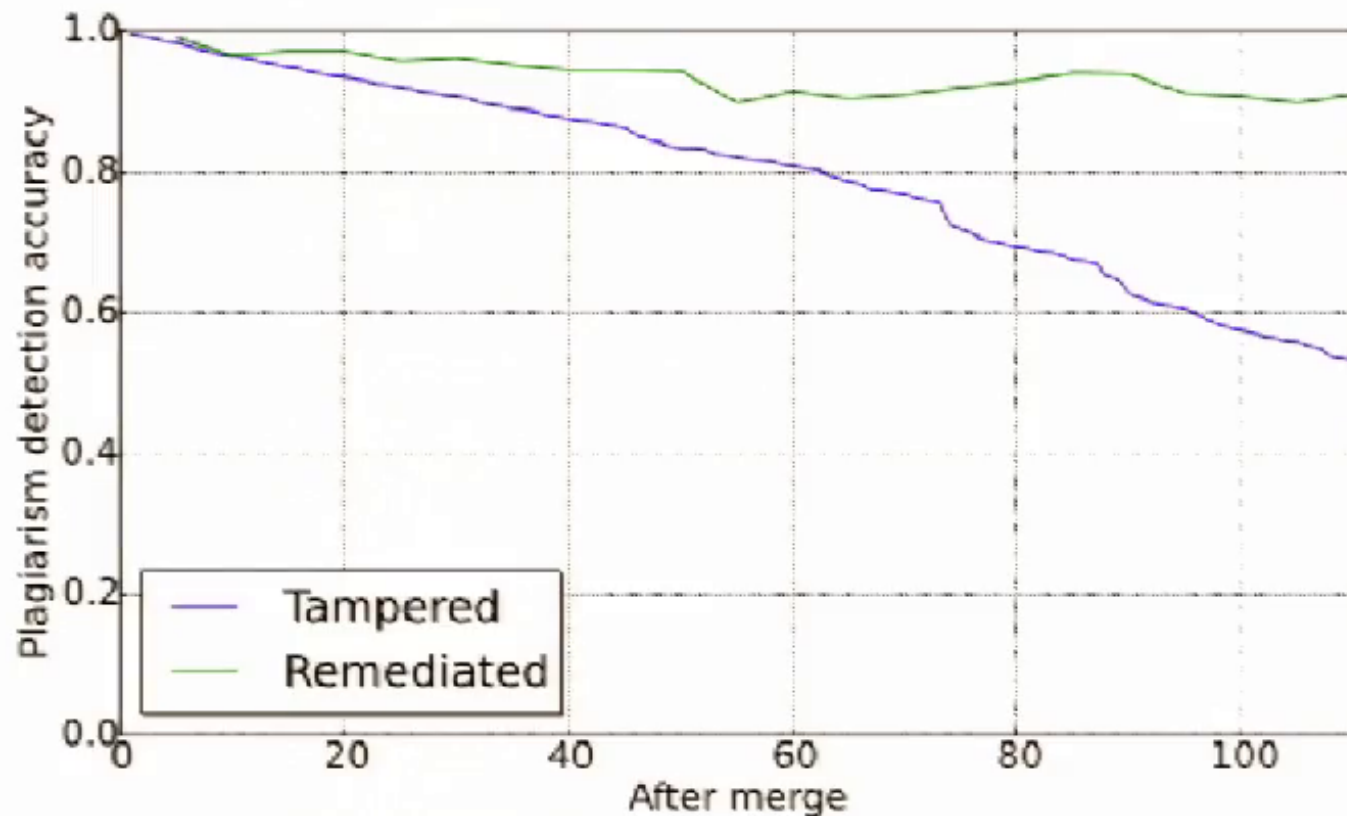Increasing T and *MinPts* may cause us to miss plagiarizing apps

Defenses?

Increasing T and *MinPts* may cause us to miss plagiarizing apps

Instead, can we detect and remove data mines?

Defenses?

Increasing *T* and *MinPts* may cause us to miss plagiarizing apps

Instead, can we detect and remove data mines?

Conclusion

Contributions:

- Methodology for selecting and then merging arbitrary clusters
- Evaluate effectiveness in a real-world scenario
- Show DBSCAN's vulnerability to the chaining phenomenon
- Propose and evaluate outlier-based remediation

Questions/Comments?

Presenter: Jonathan Crussell
jcrusse@sandia.gov