Improving accuracy of malware detection by filtering evaluation dataset based on its similarity

Junichi Murakami
Director of Advanced Development

FFRI, Inc.
http://www.ffri.jp

Preface

- This slides was used for a presentation at CSS2013
  - http://www.iwsec.org/css/2013/english/index.html
- Please refer the original paper for the detail data
  - http://www.ffri.jp/assets/files/research/research_papers/MWS2013_paper.pdf
    (Written in Japanese but the figures are common)
- Contact information
  - research-feedback@ffri.jp
  - @FFRI_Research (twitter)

Agenda

- Background
- Problem
- Scope and purpose
- Experiment 1
- Experiment 2
- Experiment 3
- Consideration
- Conclusion

Background – malware and its detection

Malware generators

Obfuscators

Increasing malware

Targeted Attack (Unknown malware)

Limitation of signature matching

other methods

Bigdata

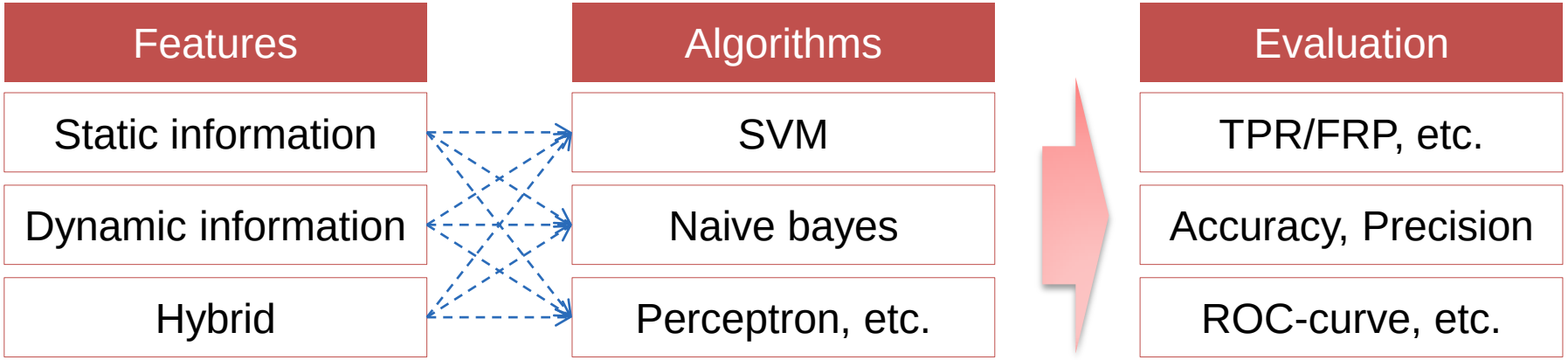
Heuristic

Machine learning

Could reputation

Background – Related works

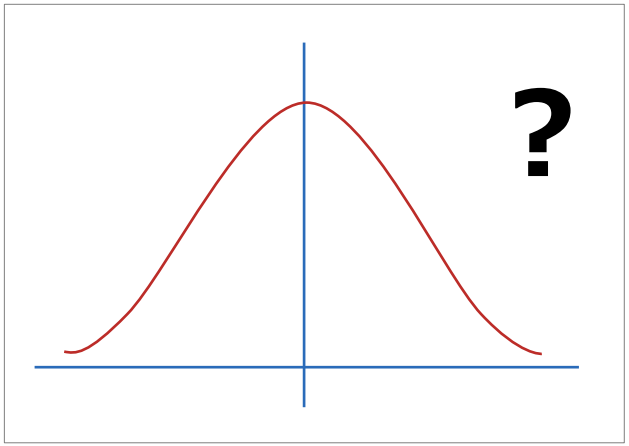
- Mainly focusing on a combination of the factors below
  - Features selection and modification, parameter settings
- Some good results are reported (TRP:90%+, FRP:1%-)

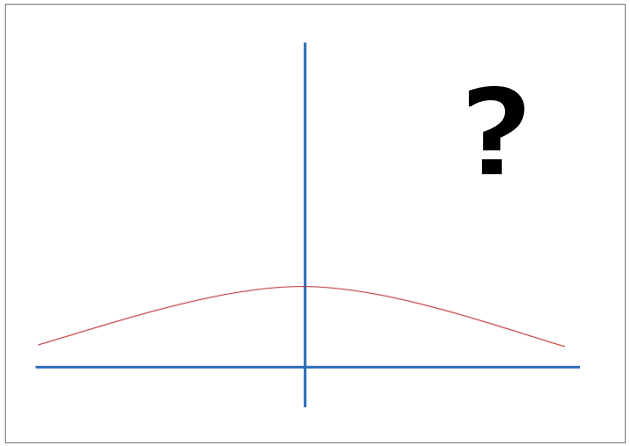


Problem

- General theory of machine learning:
  - Accuracy of classification declines
    if trends of training and testing data are different

- How about malware and benign files





Scope and purpose

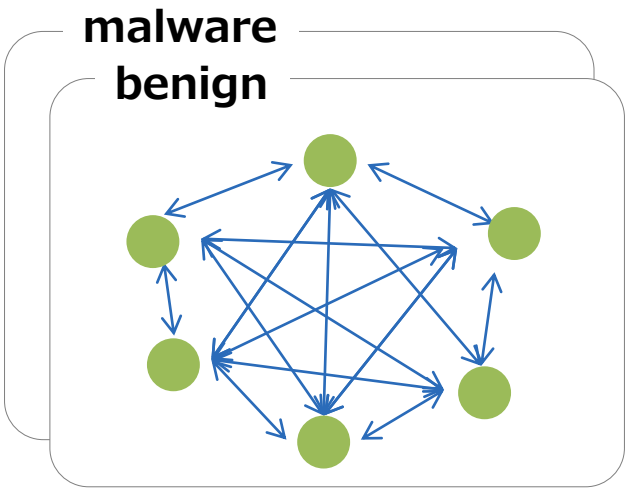
① Investigating differences between similarities of malware and benign files*(Experiment-1)*

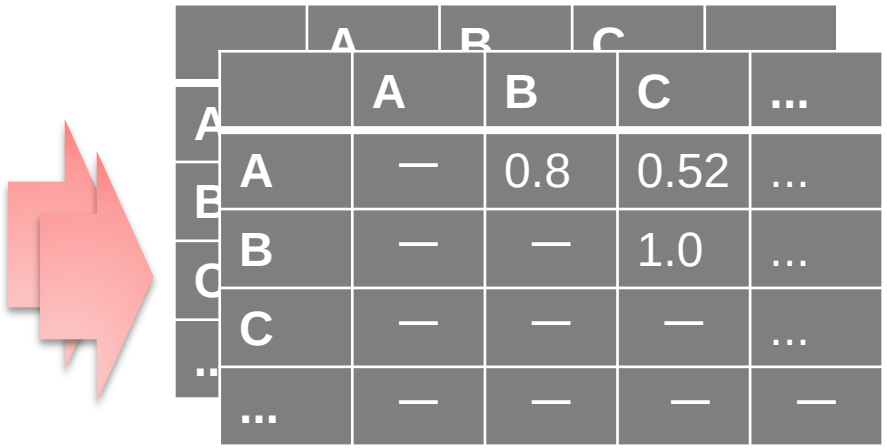
② Investigating an effect for accuracy of classification by the difference*(Experiment-2)*

③ Based on the result above, confirming an effect of removing data whose similarity with a training data is low *(Experiment-3)*

Experiment-1(1/3)

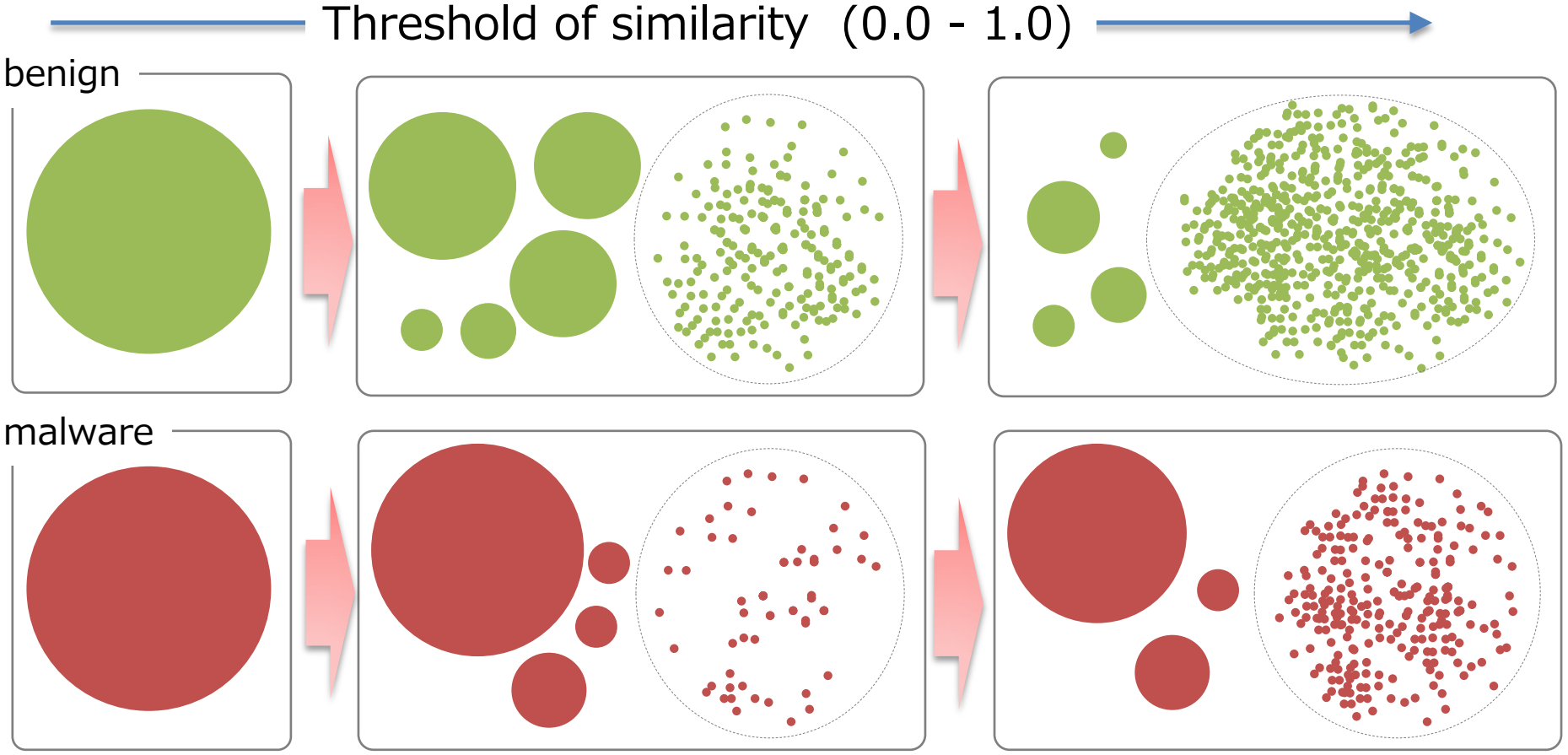
- Used FFRI Dataset 2013 and benign files we collected as datasets
- Calculated the similarity of each malware and benign files (Jubatus, MinHash)
- Feature vector: A number of 4-gram of sequential API calls
  - *ex: NtCreateFile_NtWriteFile_NtWriteFile_NtClose: n times*
    NtSetInformationFile_NtClose_NtClose_NtOpenMutext: m times





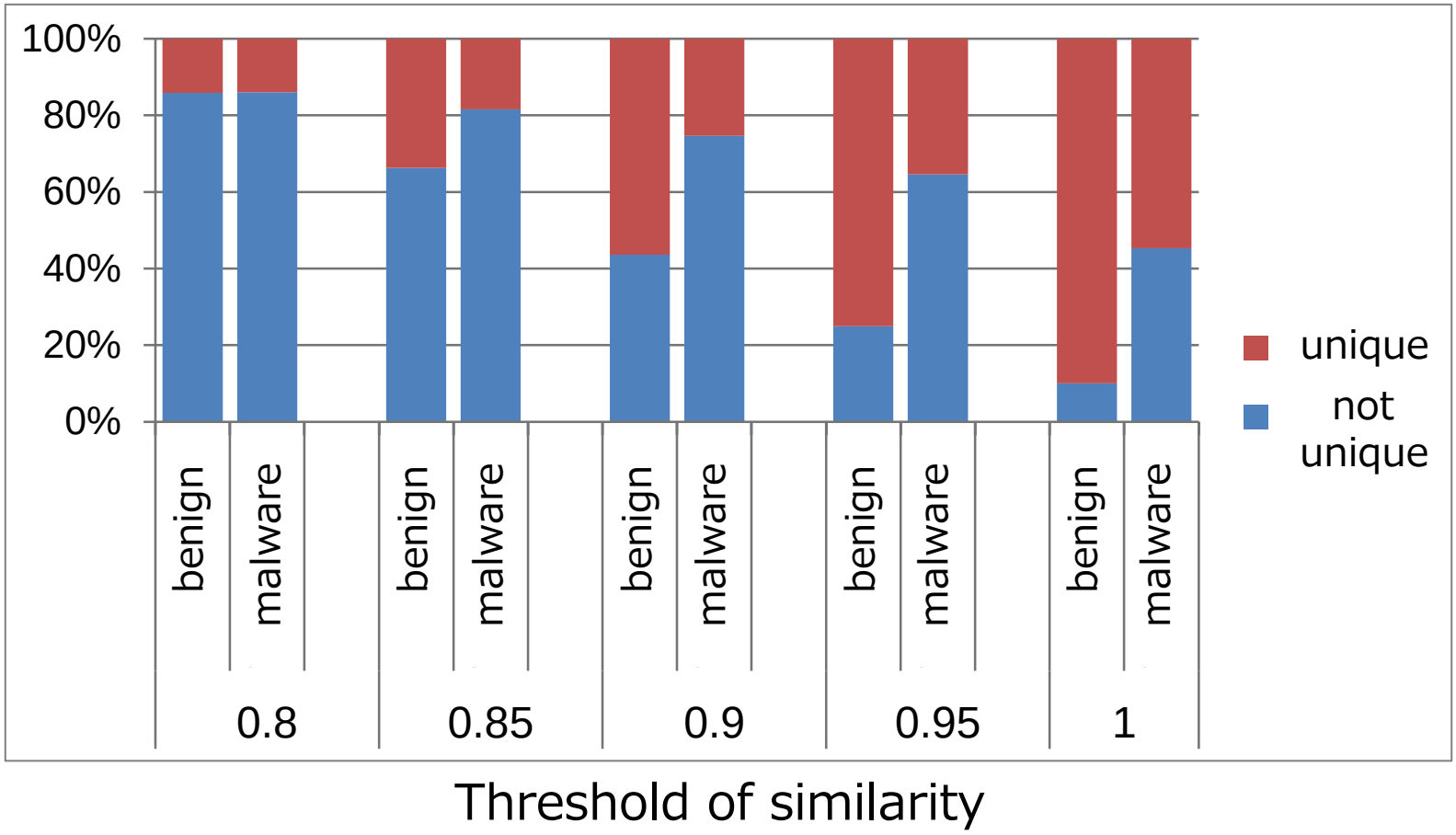
Experiment-1(2/3)

Grouping malware and benign files based on their similarities



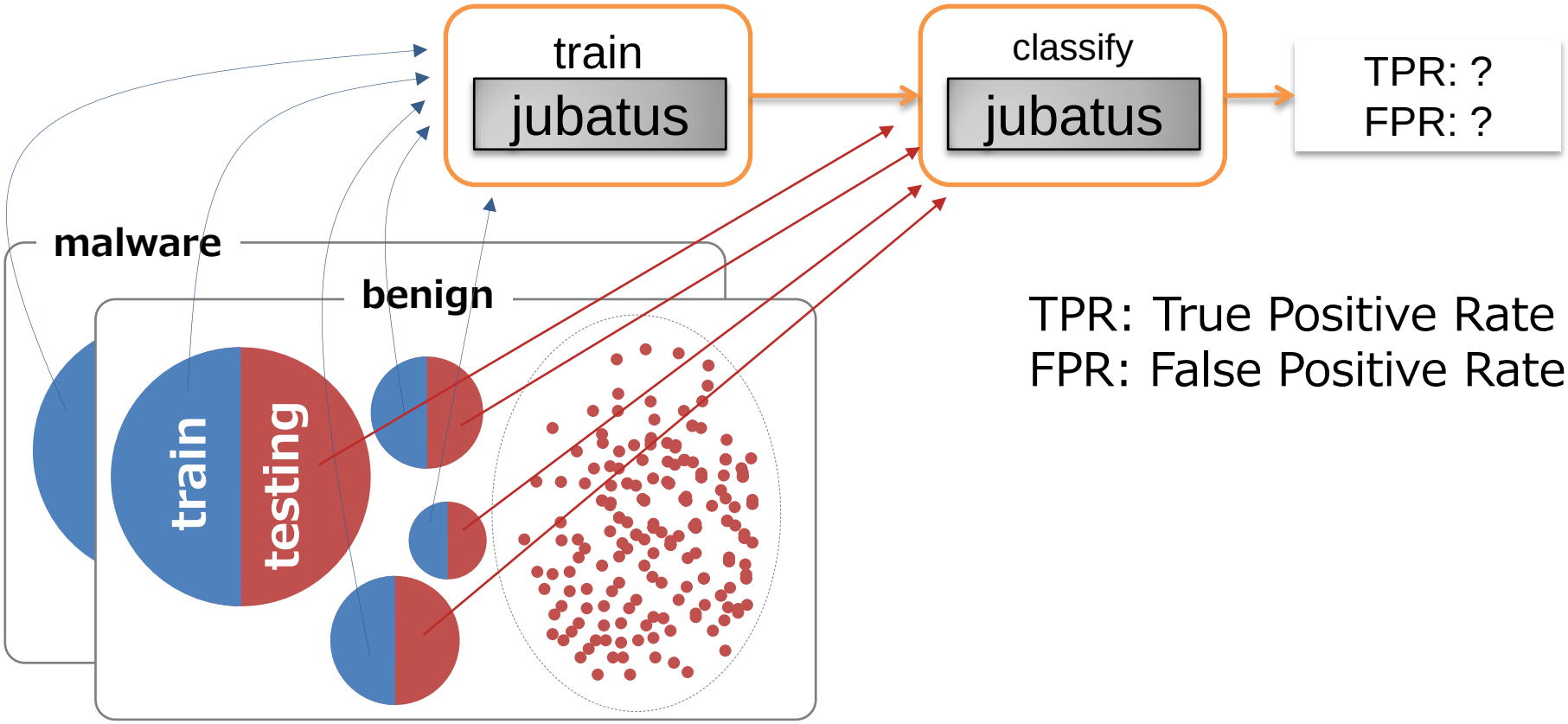
Experiment-1(3/3)

It is more difficult to find similar benign files compared to malware



Experiment-2(1/3)

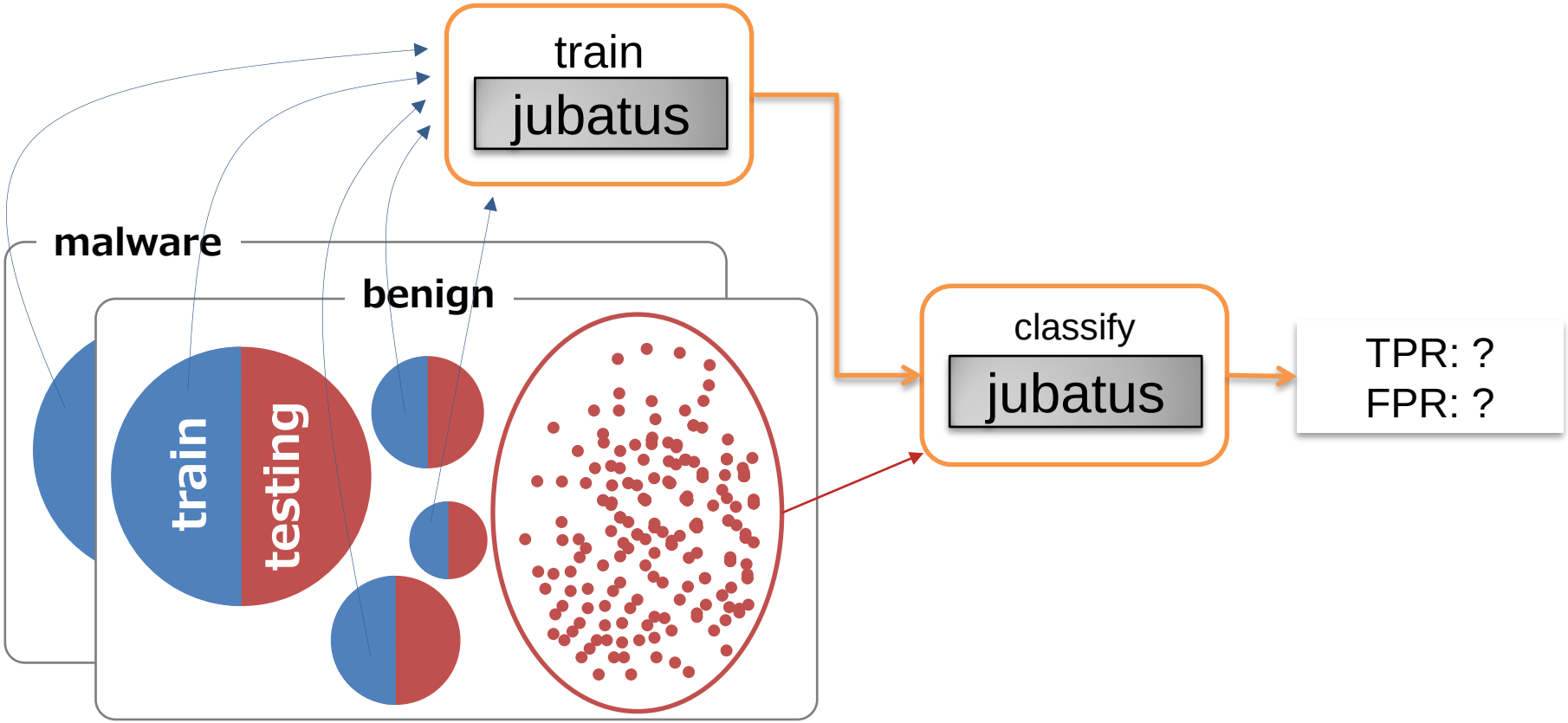
- How much does the difference affect a result?
- 50% of malware/benign are assigned to a training, the others are to a testing dataset(Jubatus, AROW)



TPR: True Positive Rate
FPR: False Positive Rate

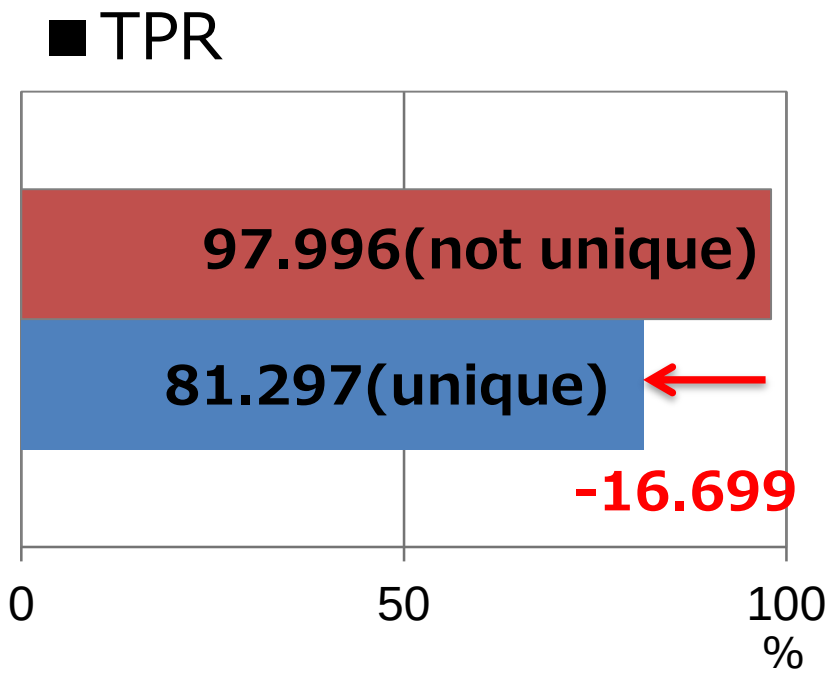
Experiment-2(2/3)

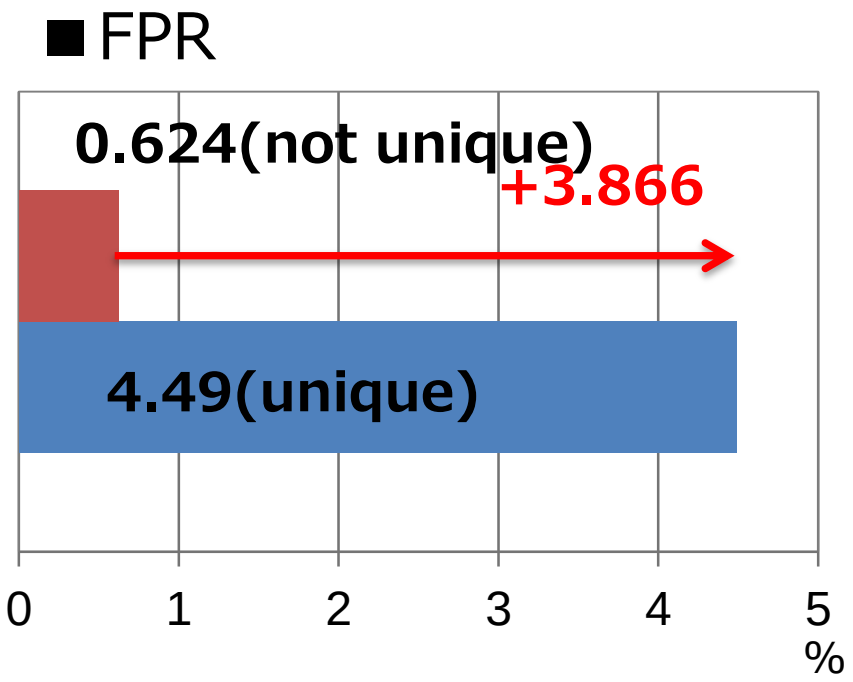
- How much does the difference affect a result?
- 50% of malware/benign are assigned to a training, the others are to a testing dataset(Jubatus, AROW)



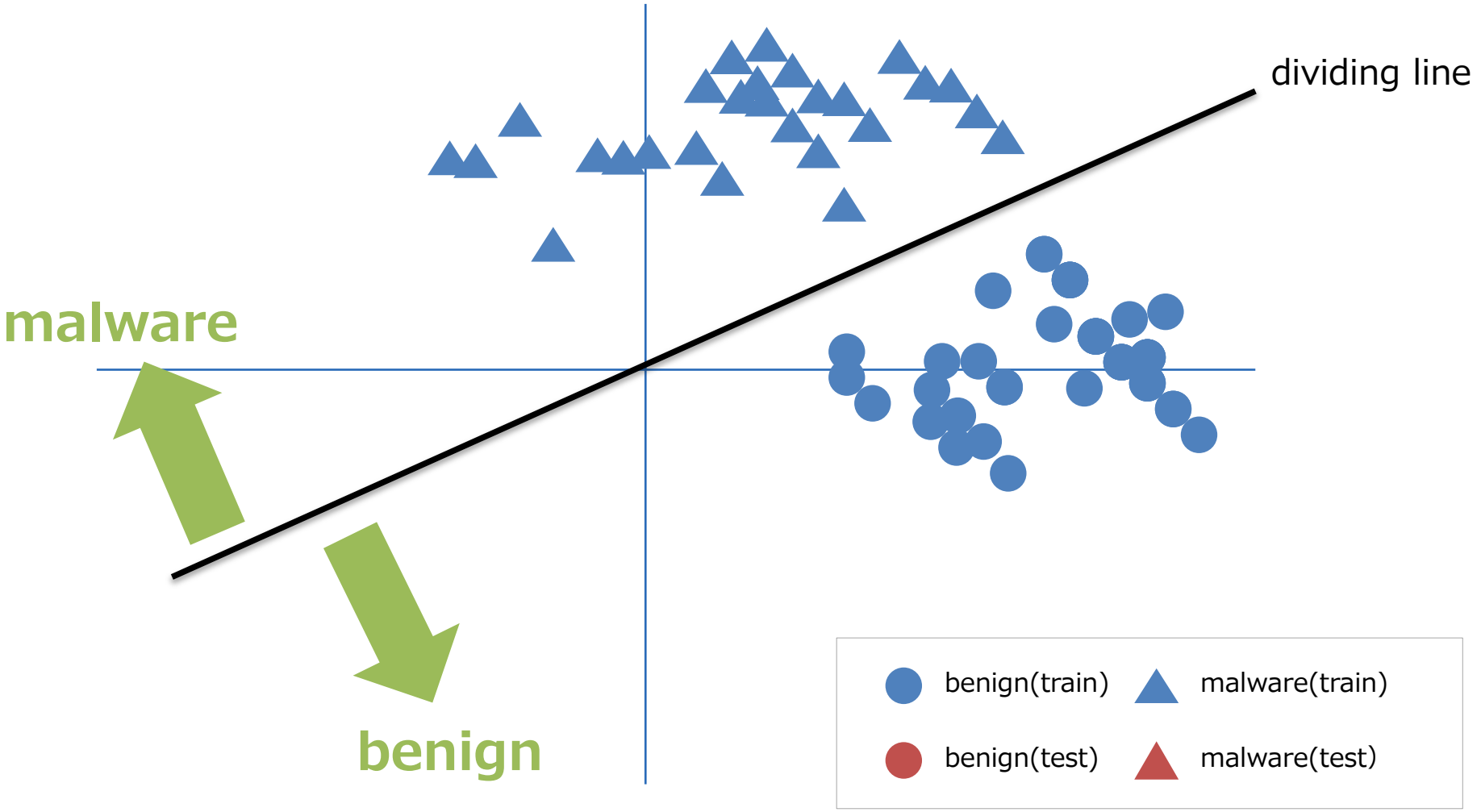
Experiment-2(3/3)

The accuracy declines if trends of training and testing data are different

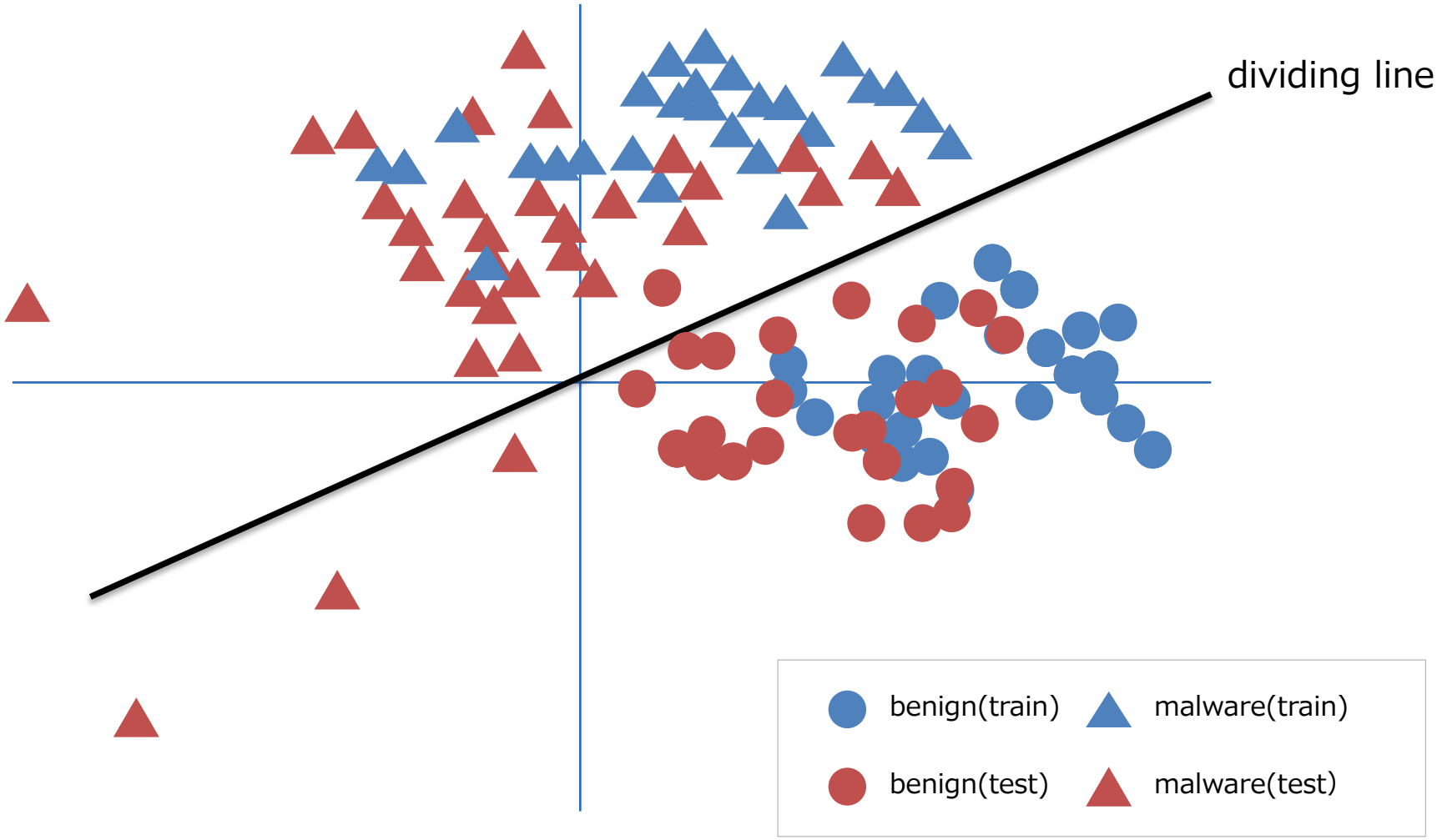




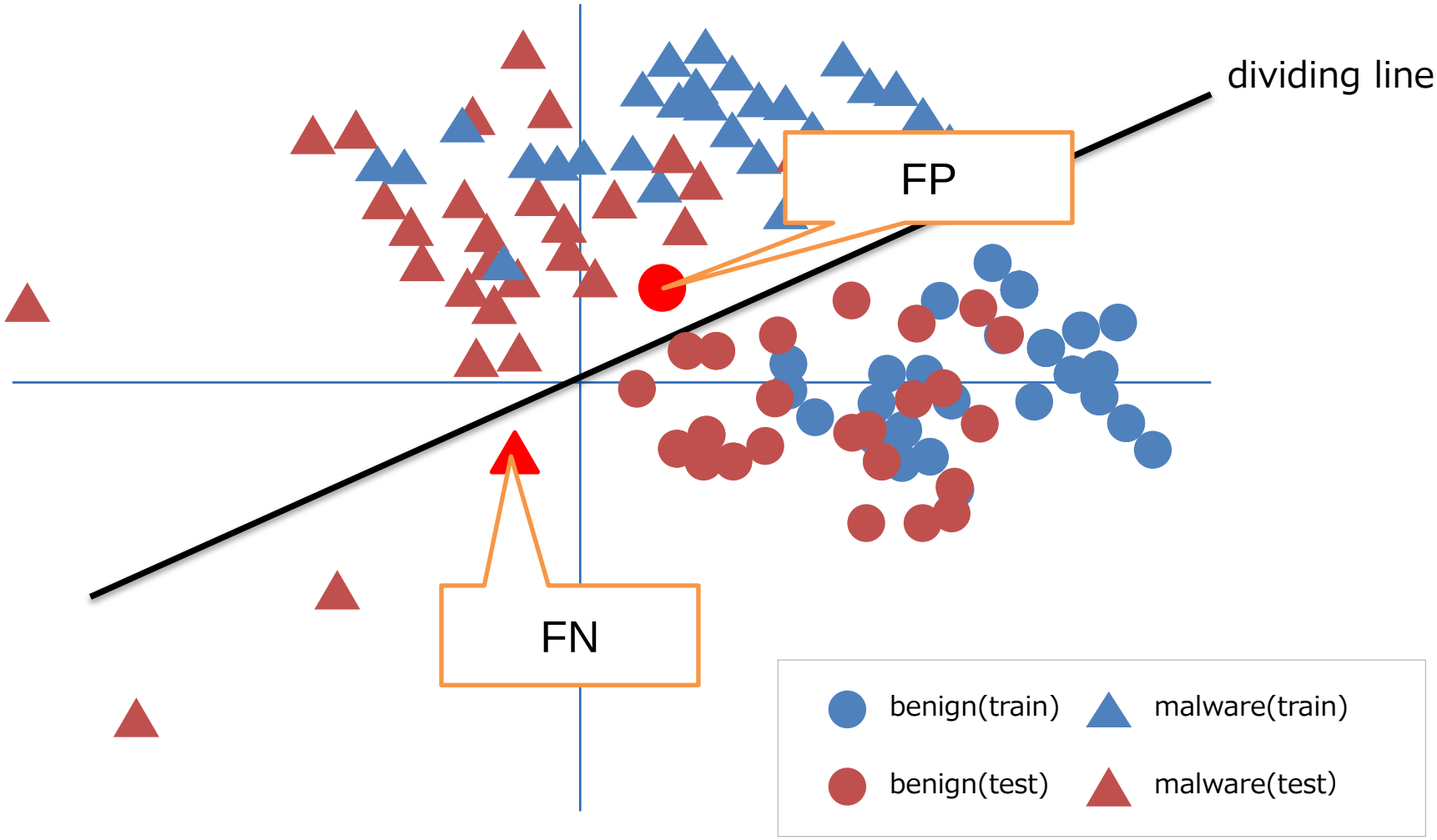
Experiment-3(1/6) – After a training



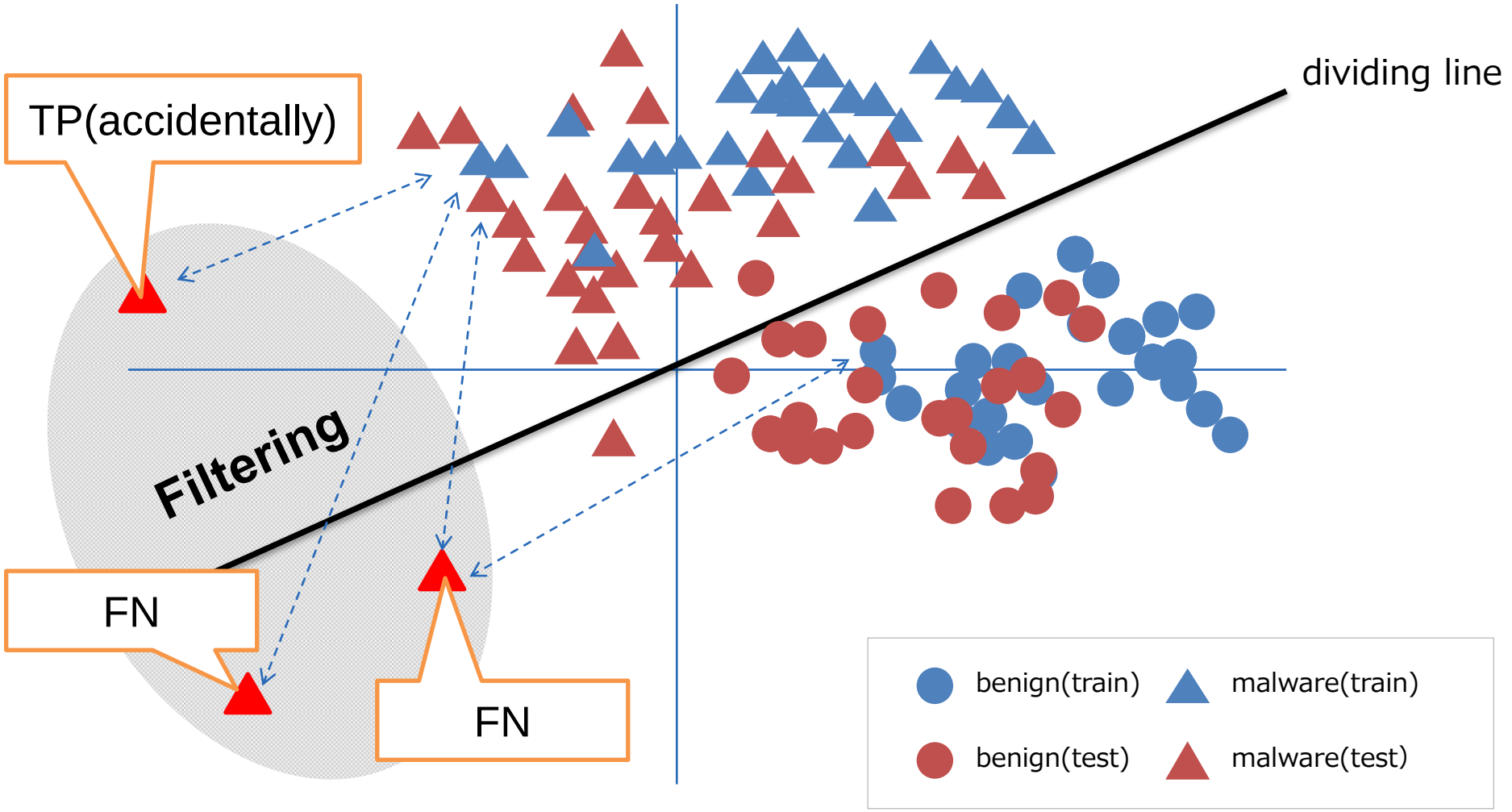
Experiment-3(2/6) – After a classification



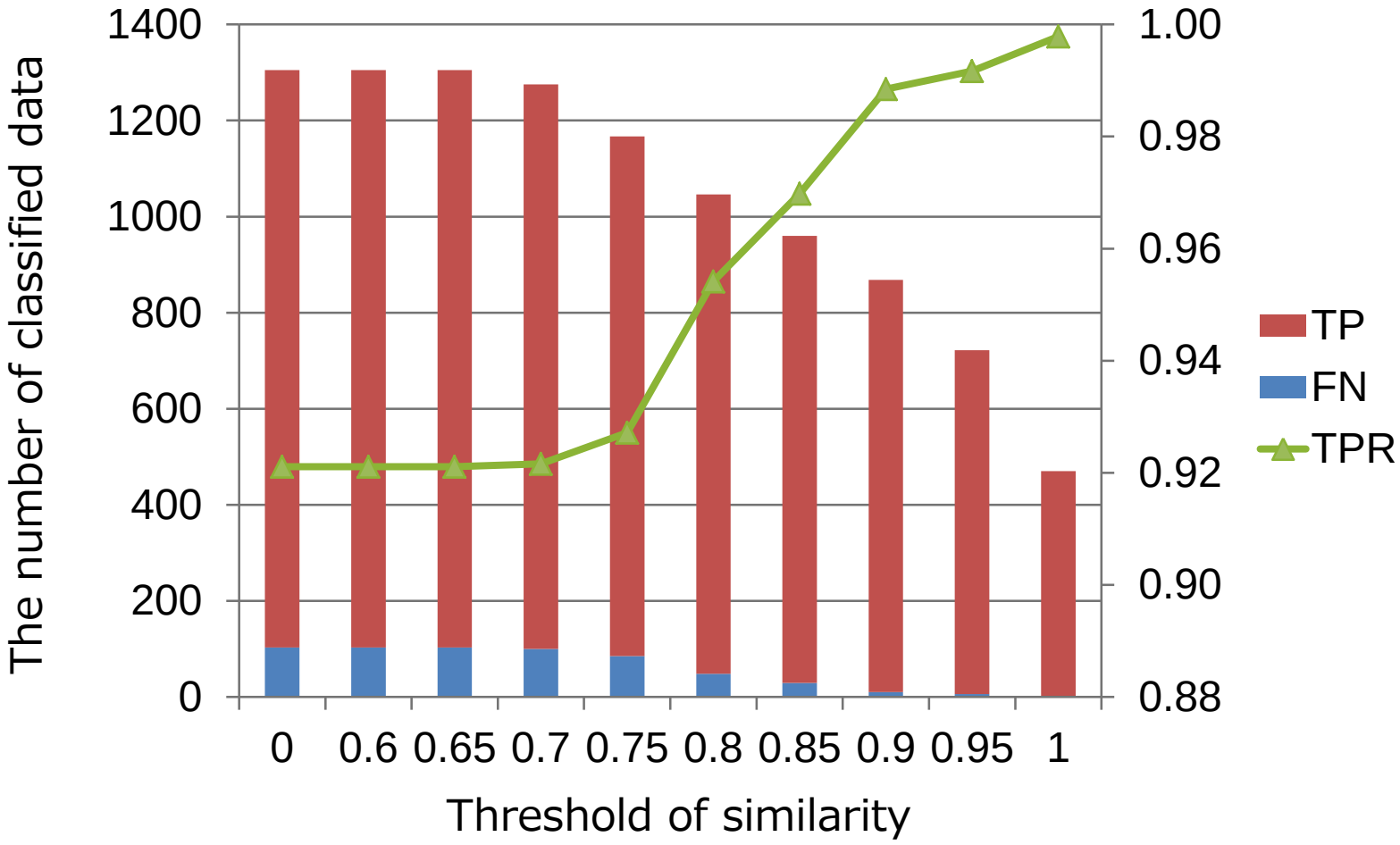
Experiment-3(2/6) – After a classification



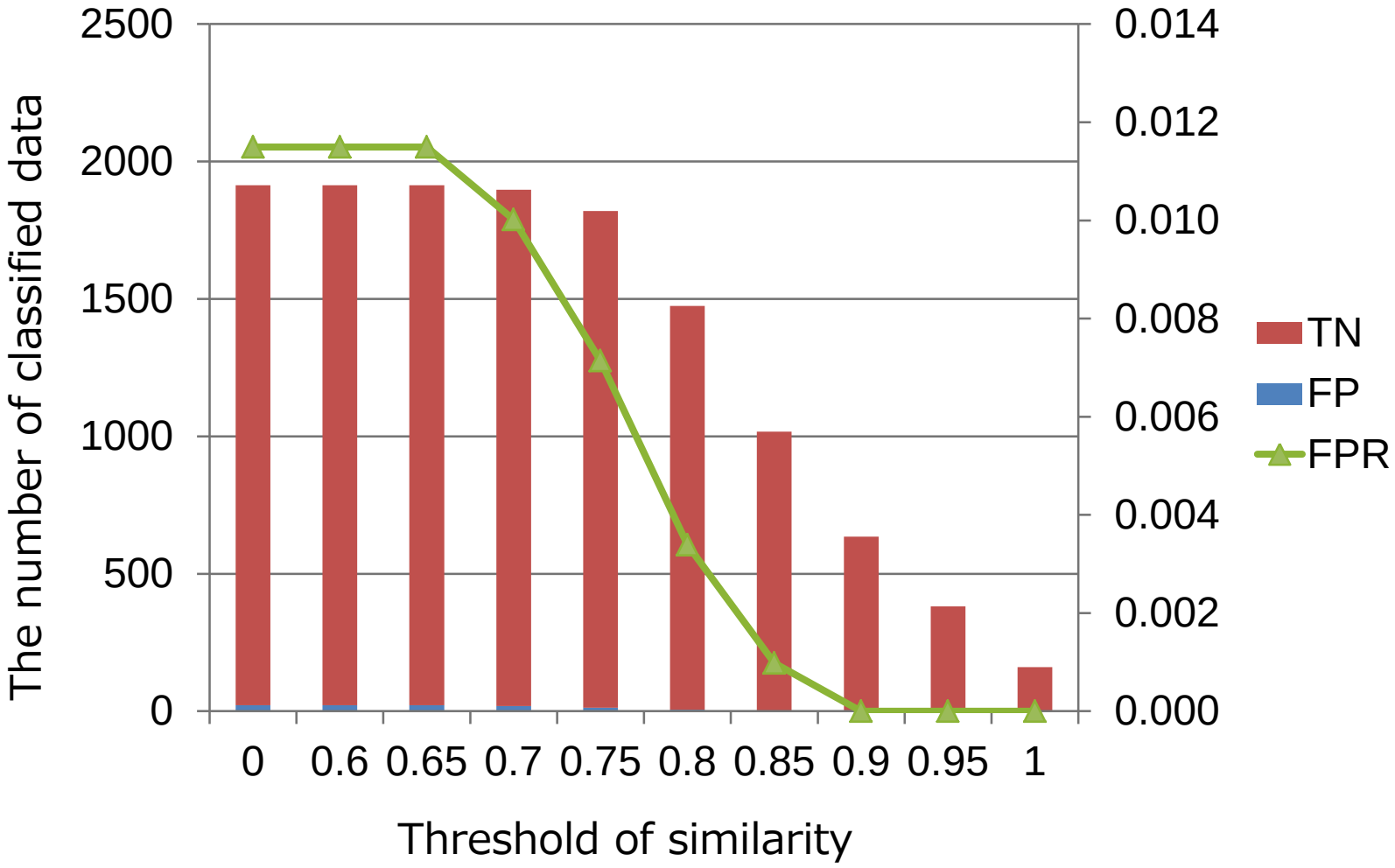
Experiment-3(3/6) – Low similarity data



Experiment-3(4/6) – Effect to TPR

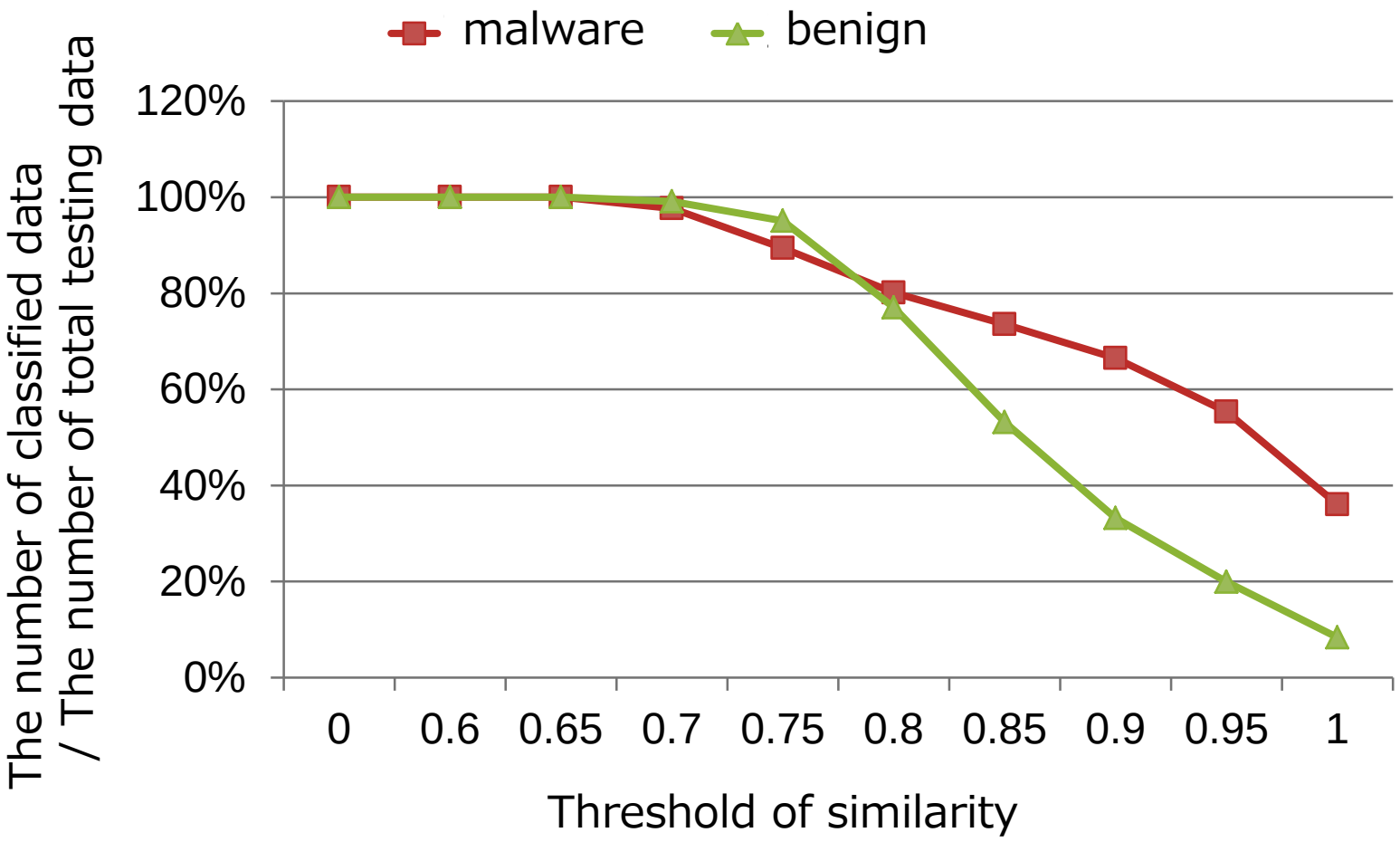


Experiment-3(5/6) – Effect to FPR



Experiment-3(6/6)

Transition of the number of classified data



Consideration(1/3)

- In real scenario:
  - trying to classify an unknown file/process whether it is benign files or not

- If we apply Experiment-3:
  - Files are classified only if similar data is already trained
  - If not, files are not classified which results in
    - FN if the files is malware
    - TF if the files is benign (All right as a result)

- Therefore it is a problem about **“TPR for unique malware”** (Unique malware is likely to be undetectable)

Consideration(2/3)

- If malware have many variants as the current
  - ML-based detection works well
- Having many variants ∝ malware generators/obfuscators
- We have to investigate
  - Trends of usage of the tools above
  - Possibility of anti-machine learning detection

Consideration(3/3)

- How to deal with unclassified (filtered) data

  1. Using other feature vectors
  2. Enlarging a training dataset (Unique → Not unique)
  3. Using other methods besides ML

Conclusion

- Distribution of similarity for malware and benign are difference *(Experiment-1)*
- Accuracy declines if trends of training and testing data are different *(Experiment-2)*
- TPR of unique malware declines when we remove low similarity data *(Experiment-3)*
- Continual investigation for trends of malware and related tools are required
- (Might be necessary to develop technology to determine benign files)