This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

SAND2021-14331C

Sandia National Laboratories

Exceptional service in the national interest

Faster, featureless classification using compression analytics

Christina Ting, Nicholas Johnson,
Uzoma Onunkwo, and J. Derek Tucker

IEEE ICDM 2021

Data Mining and Machine Learning for Cybersecurity (DMC) Workshop, December 7, 2021

SAND2021-13716 C

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

U.S. DEPARTMENT OF ENERGY

NNSA

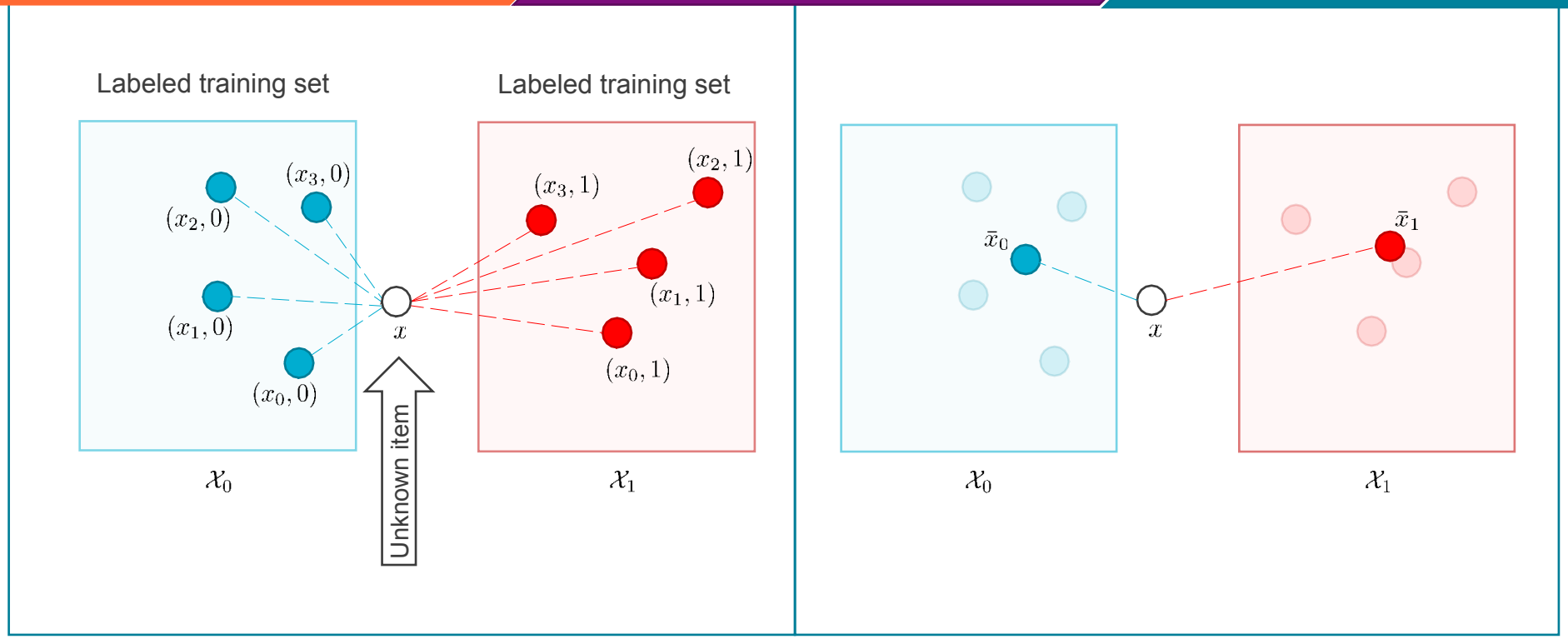
Classification:

Nearest neighbor methods

Featureless classification:
+ compression based distance metrics

Faster, featureless classification:

+ prototype items



Compression-based distance metrics

Compression algorithms can be used as a ***featureless*** approach to compare two items [1].

Lempel-Ziv Set (LZSet):

If you haven't seen the sub-string add it to the dictionary. If you have seen it continue to append characters until you arrive at unseen substring.

ababbaaba → {a, b, ab, bb, aa, ba, aba}

Lempel-Ziv Jaccard Distance[2] (LZJD):

Jaccard distance across LZSets

$$d(X,Y) = \begin{cases} 1 - \frac{|X \cap Y|}{|X \cup Y|} & \text{if } X \cup Y \neq \varnothing, \\ 1 & \text{otherwise.} \end{cases}$$

General distance metric that does not require feature engineering

[1] Li, Ming, et al. "The similarity metric." *IEEE transactions on Information Theory* (2004).

[2] Raff and Nicholas, "An alternative to NCD for large sequences, Lempel-Ziv Jaccard distance," KDD (2017).

Prototype items

Prototypes of classes, rather than all observed data items, can be used for ***faster*** classification

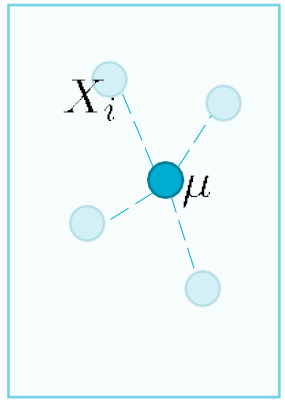
We define a prototype analogous to the arithmetic mean.

Specifically, the Fréchet Mean (FM) is the set that minimizes the sum of squared distances (LZJD) to n observed sets

$$J(M) = \sum_{i=1}^{n} d^2(M, X_i)$$

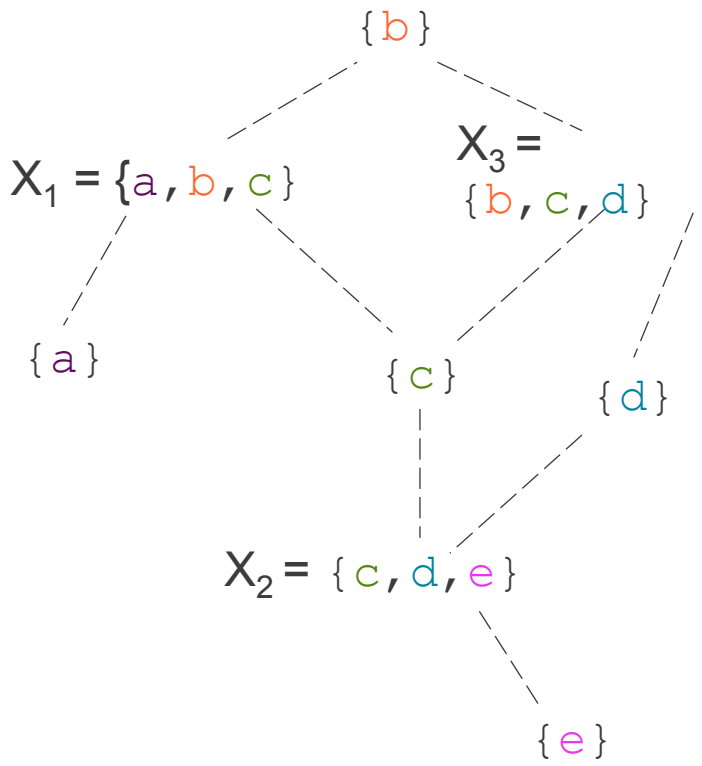
$$= \sum_{i=1}^{n} \left(1 - \frac{|M \cap X_i|}{|M \cup X_i|}\right)^2$$

$$\mu = \arg\min_{M \subset \mathcal{S}} J(M)$$



$|S|$ can be quite large

A greedy algorithm for approximating the Fréchet Mean



1. Identify *elements in the union of observed sets:*

$$U = \{a, b, c, d, e\}$$

2. Sort the elements in order of their increasing sum of squared JD to observed sets:

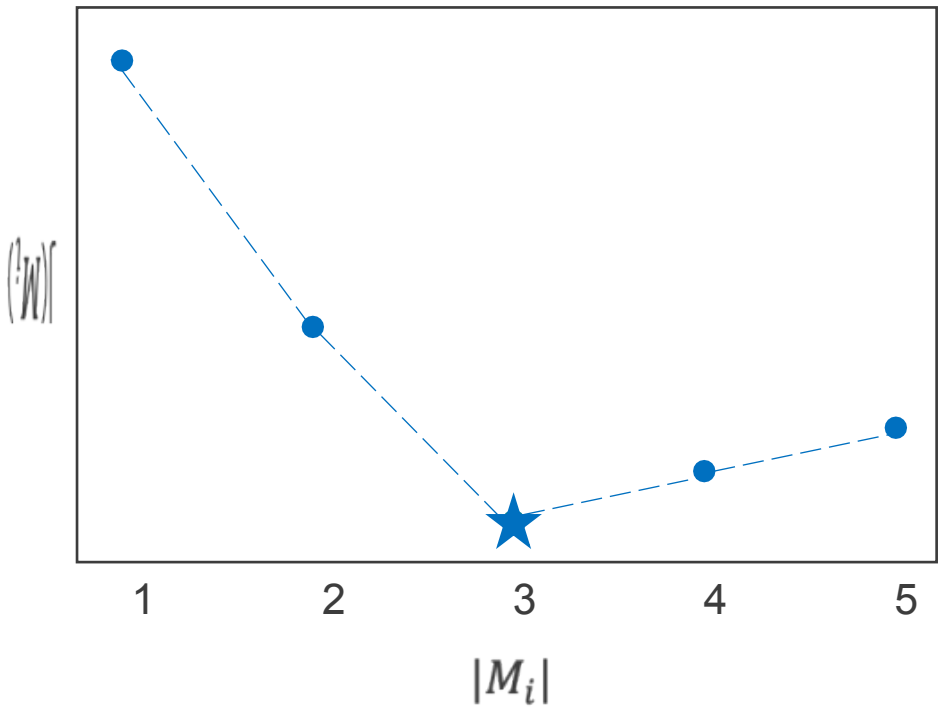
$$\{c\}, \{d\}, \{b\}, \{a\}, \{e\}$$

3. Incrementally build up a list of candidate means by adding the next sorted element to the set:

$$M^1 = \{c\},\ M^2 = \{c, d\}, \ldots,\ M^{|U|} = \{c, d, b, a, e\}$$

Reduced search space to $|U|$ possible candidate means

A greedy algorithm for approximating the Fréchet Mean



1. Identify *elements in the union of observed sets:*

$U = \{a, b, c, d, e\}$

2. Sort the elements in order of their increasing sum of squared JD to observed sets:

$\{c\}, \{d\}, \{b\}, \{a\}, \{e\}$

3. Incrementally build up a list of candidate means by adding the next sorted element to the set:

$M^1 = \{c\}$, $M^2 = \{c, d\}, \ldots,$ $M^{|U|} = \{c, d, b, a, e\}$

4. This becomes our new search space so that

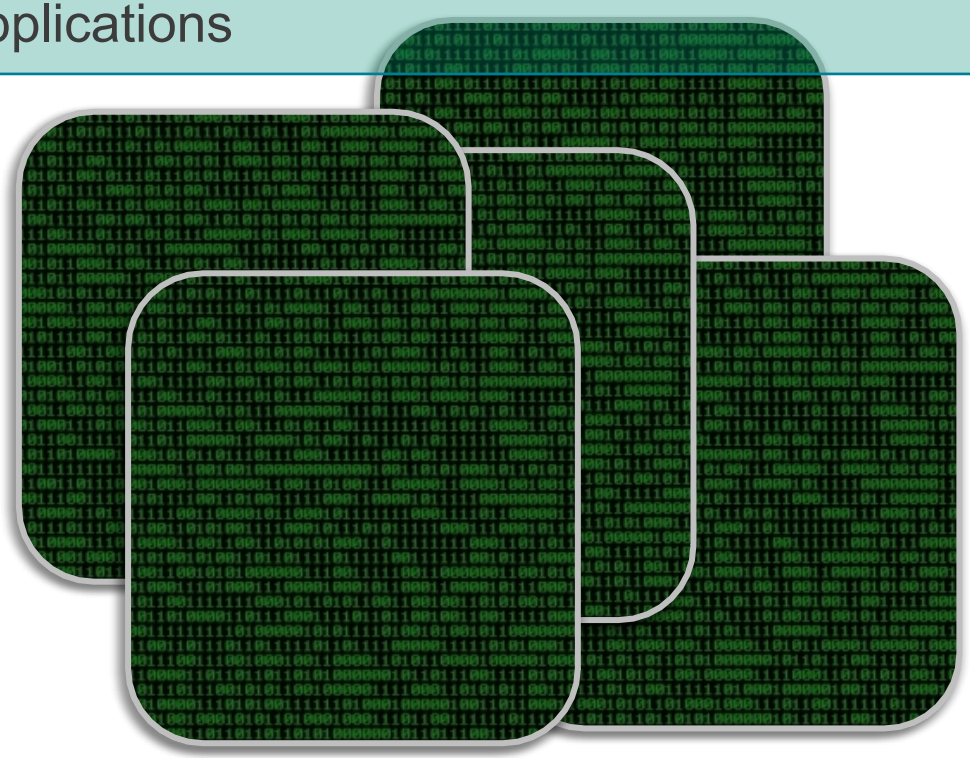
$$\mu = \arg\min_{M \subset \mathcal{S}} J(M) \quad \Rightarrow \quad \mu = \arg\min_{M \in \{M^1, \ldots M^{|U|}\}} J(M)$$

Reduced search space to $|U|$ possible candidate means

Experimental results

Datasets and experiments

Goal: Evaluate **classification performance** and **timing** of a featureless classification approach across a variety of practical applications



1) File fragment classification

- Predict file type from {.doc, .gif, .html, .jpg, .pdf}
- Random selection of 512 contiguous bytes from file

2) Authorship attribution based on Java source code

- Predict which author from a set of 40

3) Microsoft malware family classification

- Predict which malware family out of 9 options
- About 10,000 labeled malware binaries

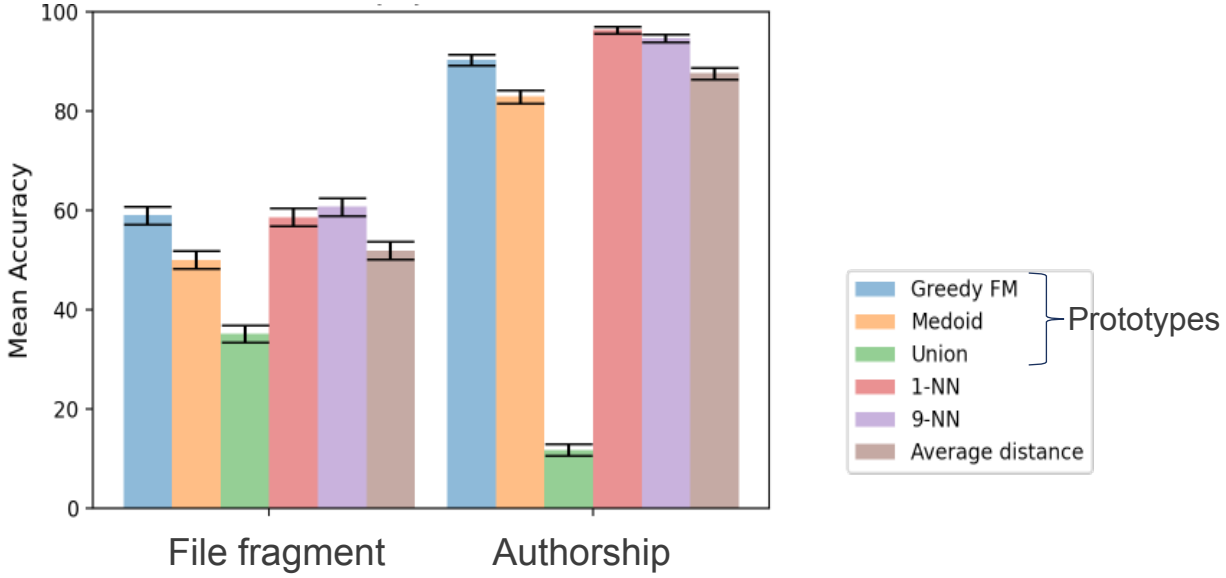
4) Drebin malware family classification

- Predict which malware family out of 20 options
- About 5,000 labeled Android malware binaries

[1] Digital corpora govdocs1 dataset, https://digitalcorpora.org/corpora/files
[2] X. Yang, et al "Authorship attribution of source code by using back propagation neural network based on particle swarm optimization," PloS one, 2017.
[3] R. Ronen, et al, "Microsoft malware classification challenge," 2018.
[4] D. Arp, et al, "Drebin: Effective and explainable detection of android malware in your pocket." in NDSS, 2014

File fragment classification and authorship attribution

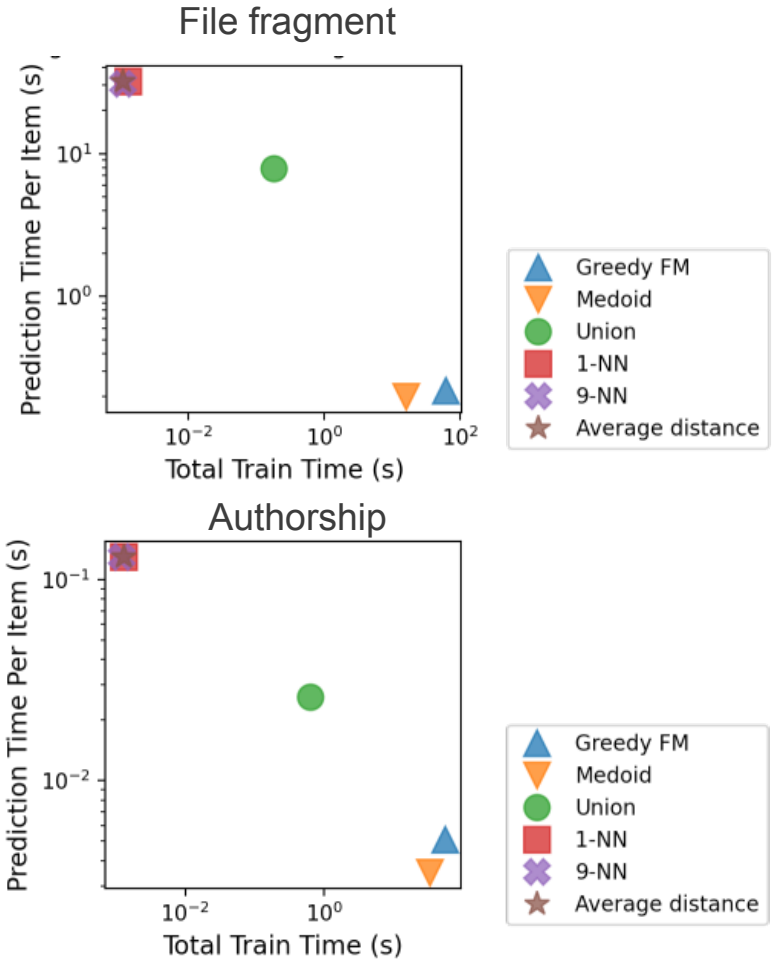
Classification performance



97% accuracies on authorship attribution using highly engineered path based models and neural networks [1]

[1] Bogolomov et al., *Authorship Attribution of Source Code: A Language-Agnostic Approach and Applicability in Software Engineering* (2017)

Timing

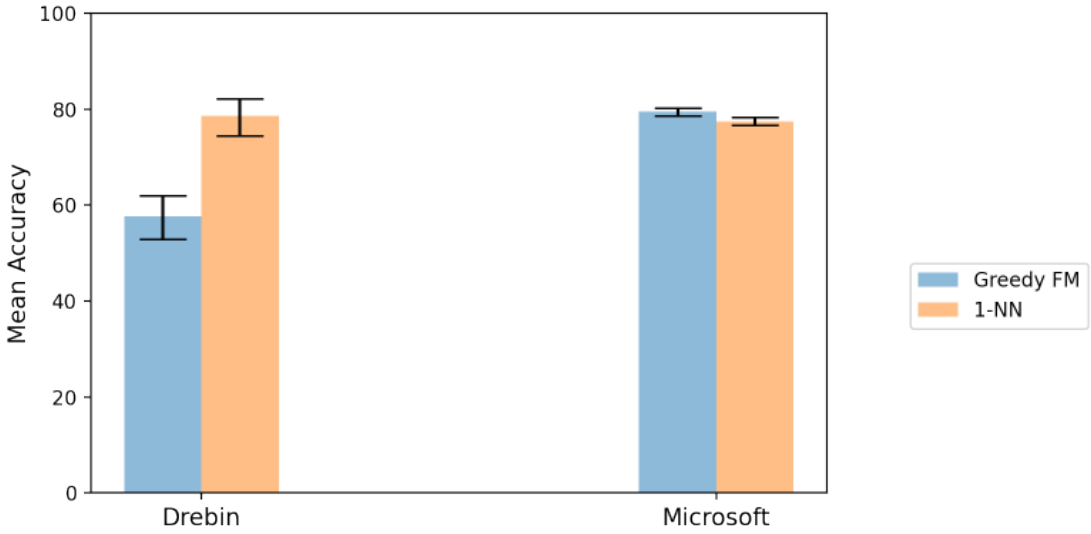


Similar accuracy; faster predictions but slower training

- LZSet creation time not included in comparisons
- 10 simulations for each experiment

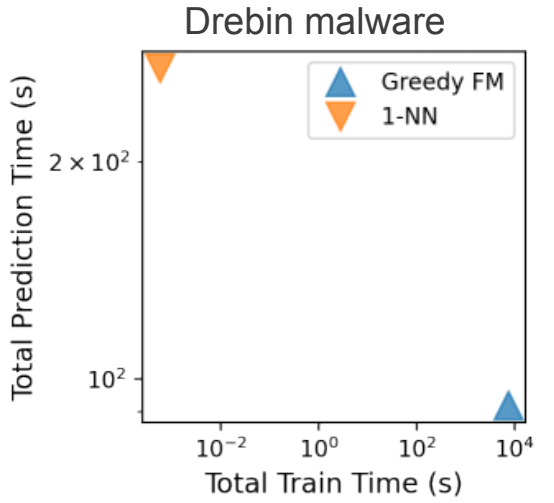
Malware classification

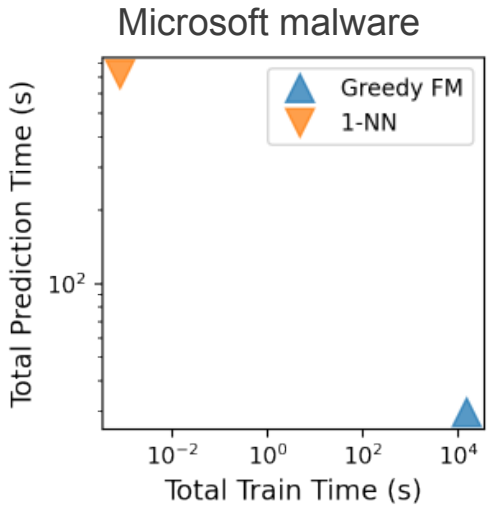
Classification performance



67.2% (Drebin) and 58% (Microsoft) balanced accuracy using NCD [1]

Timing





Developed a general, featureless prototype classification method
with superior prediction times (at the expense of increased training time)

- Malware files with lengths in MBs. Very long sequences
- Used 10% of available data
- [1] Raff and Nicholas, “An alternative to NCD for large sequences, Lempel-Ziv Jaccard distance,” KDD (2017).

Future work

Additional implementation improvements of the greedy algorithm to reduce runtime during training, especially with respect to finding the minimizing set.

Extensions to allow for more than one prototype per class.

Incorporate the FM into k-means clustering.

Contact: clting@sandia.gov