# Cockroach Analysis

A Statistical Analysis of the Flash and Java Files that Infest the Internet

# Who’s this… guy

- 10+ years on the defensive side
- File analysis, RE, network forensics, some IR
- Started to do data analysis and machine learning on security data about 2 years ago

# Why Do We Care About This

- Take the large amount of data you have and turn it into information you can act on
- Gives starting place for hunting and finding new, interesting and unknown malicious files.
- Doesn’t rely on easy to avoid signatures
- Heuristics on nontrivial boundaries in high dimensional space

# Agenda

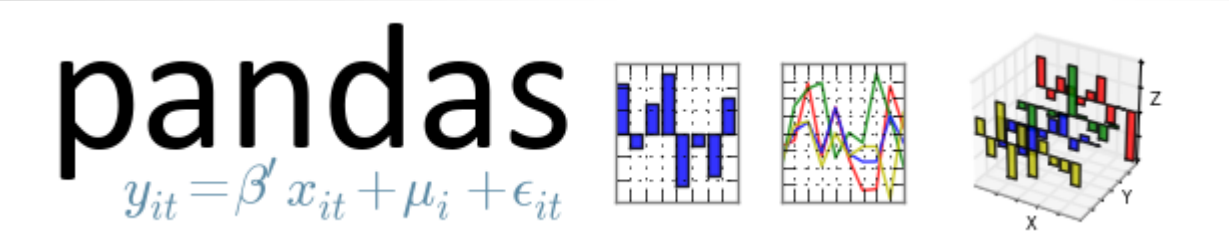
- Tools Used
- Vocabulary
- Java Analysis
- SWF Analysis
- Wrap Up

# IPython

- Interactive Shell
- Browser based
- Great way to collect
- Easy to share results with others

# pandas

- Provides dataframe functionality to Python
  - Dataframe is like a spreadsheet
  - Abstracts out a lot of headaches
- Provides a more friendly interface to data



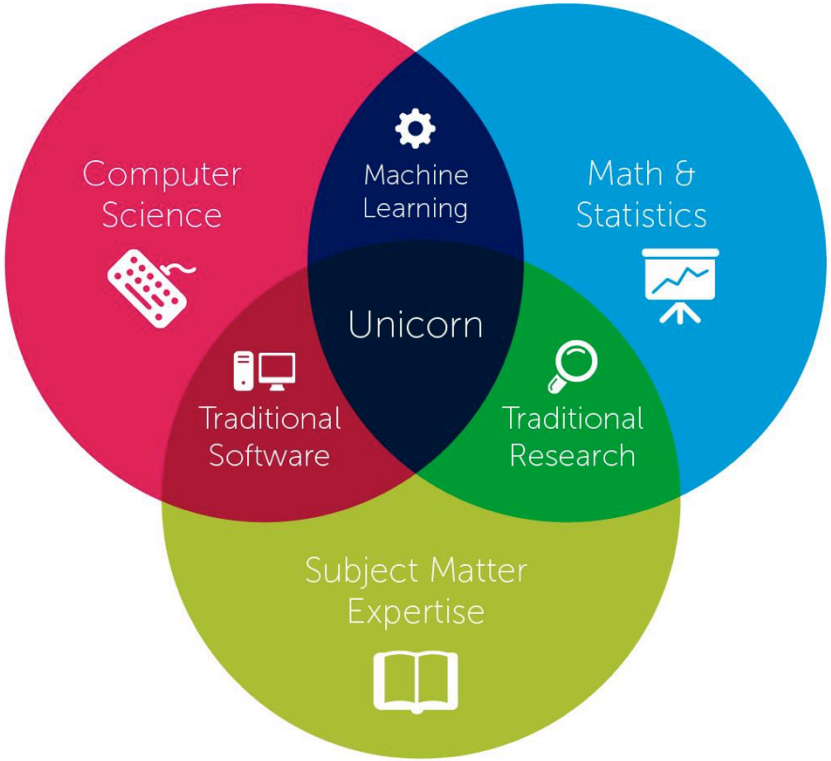
# matplotlib

- 2D plotting library
- Tries to make easy things easy and hard things possible
- Generate plots, histograms, bar charts, scatterplots and more

# scikit-learn

- Machine learning in Python
- Simple and efficient tools for data mining and data analysis
- Does not support Pandas dataframes
  - Really easy to transform from a dataframe to a supported data type

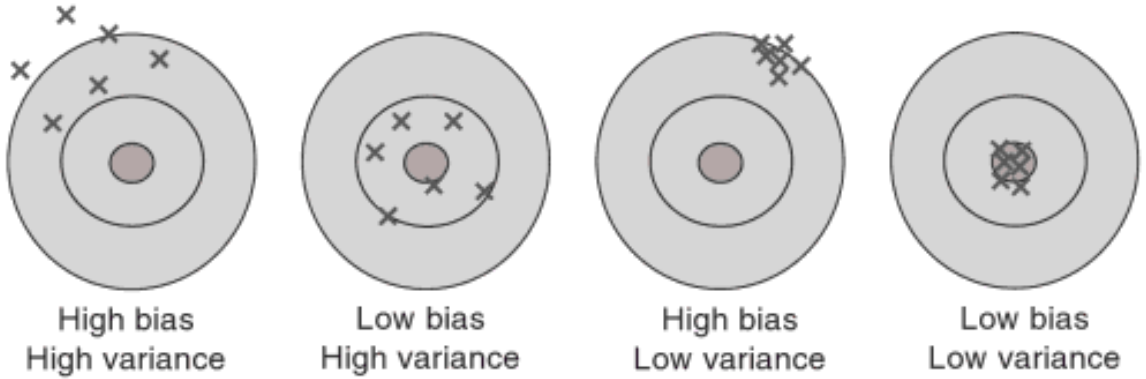
# Data Science



# Ground Truth

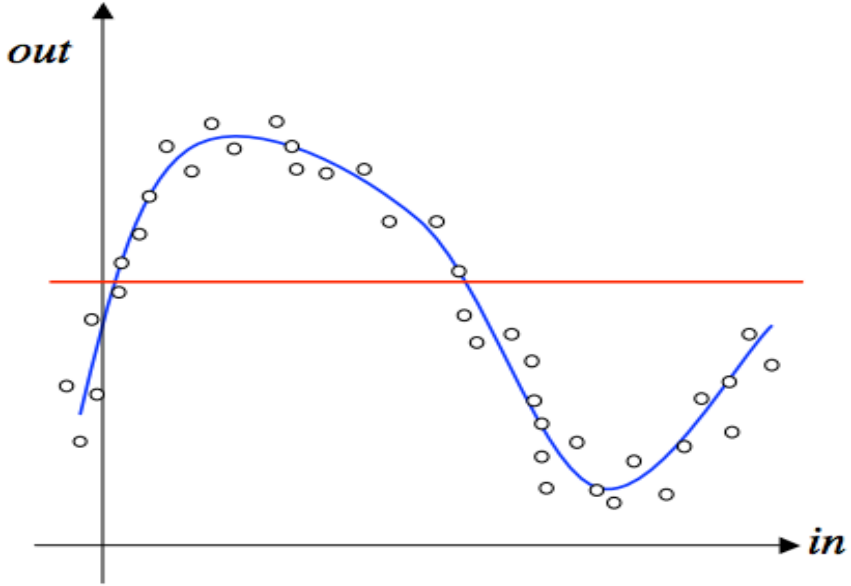
- Accuracy in labels
- Ideally, 100% accuracy in your labels

# Bias And Variance

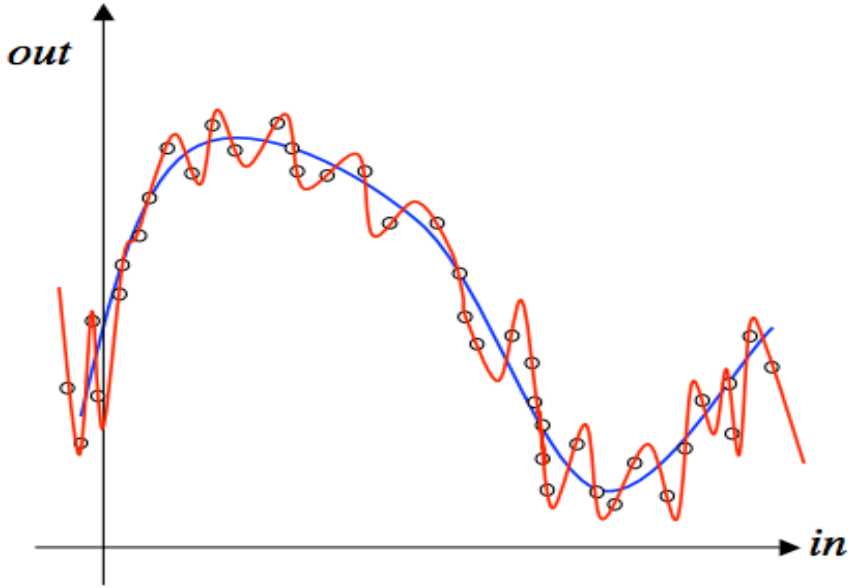



**Bias Variance Decomposition. Figure 1. The bias-variance decomposition is like trying to hit the bullseye on a dart-board. Each dart is thrown after training our "dart-throwing" model in a slightly different manner. If the darts vary wildly, the learner is *high variance*. If they are far from the bullseye, the learner is *high bias*. The ideal is clearly to have both low bias and low variance; however this is often difficult, giving an alternative terminology as the bias-variance "dilemma" (*Dartboard analogy*, Moore & McCabe (2002))**

# (Over|Under)fitting



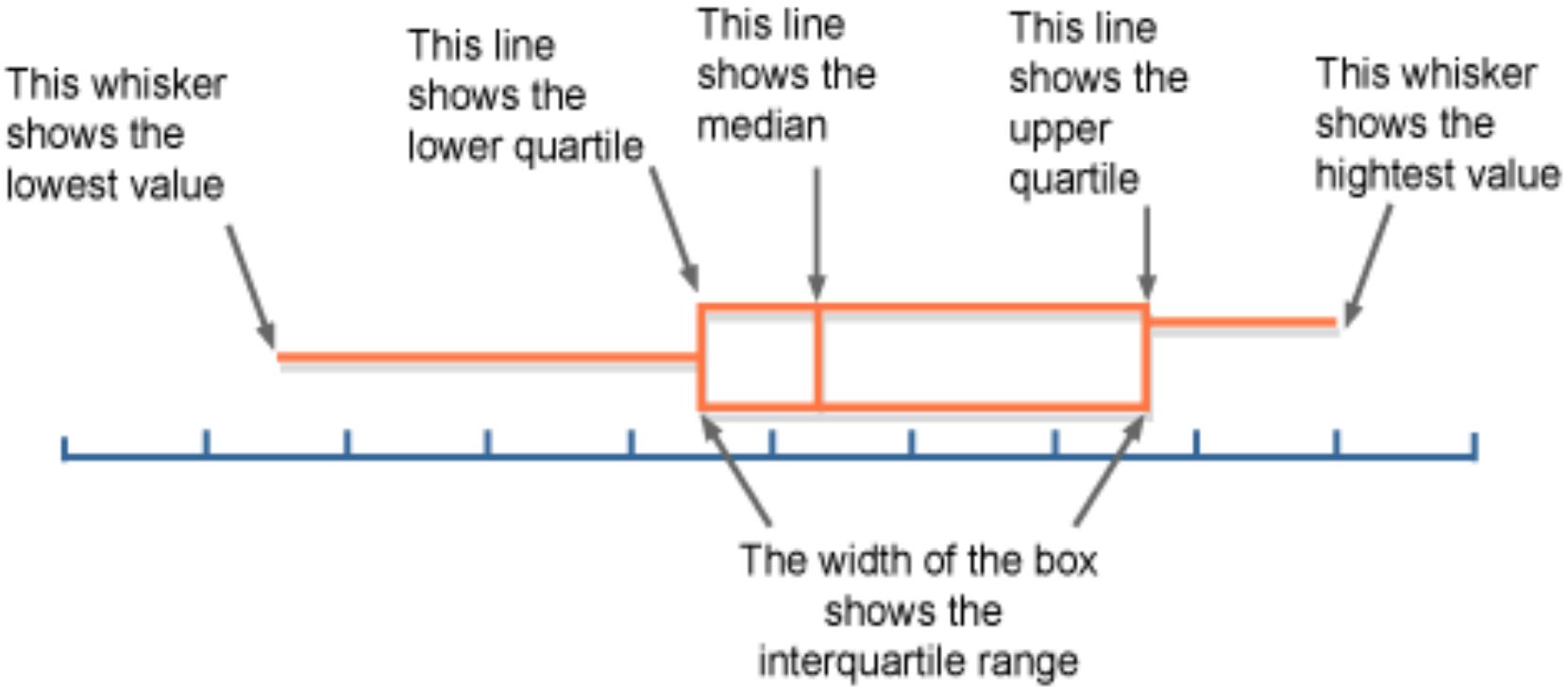

Ignore the data ⇒
Big approximation error (high bias)
No variation between data sets (no variance)




Fit every data point ⇒
No approximation error (zero bias)
Variation between data sets (high variance)

http://www.cs.bham.ac.uk/~jxb/INC/l9.pdf

# Boxplot

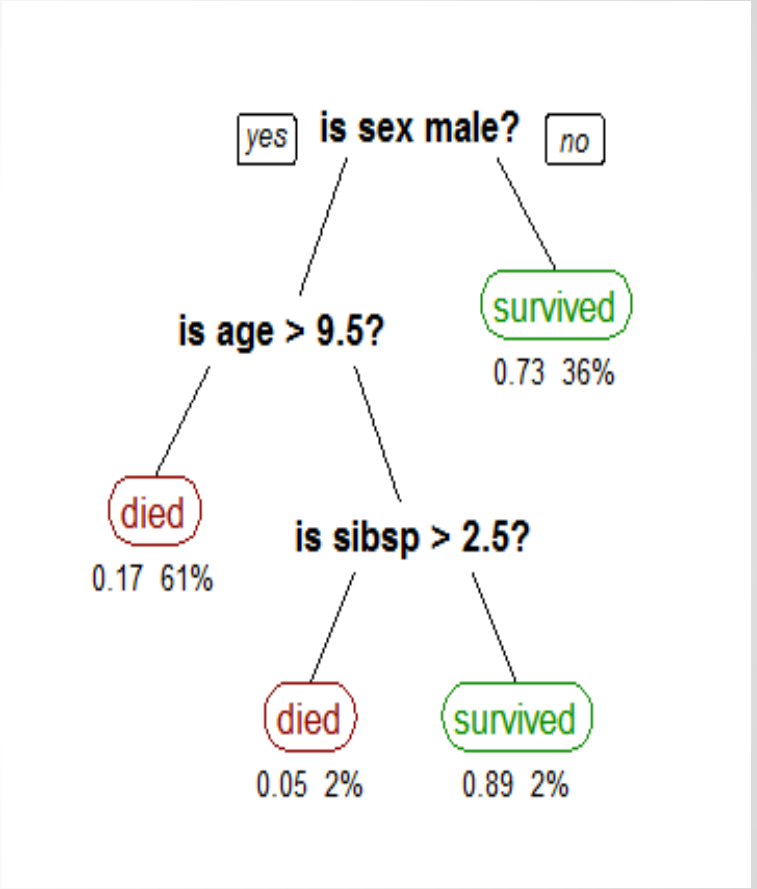

This whisker shows the lowest value
This line shows the lower quartile
This line shows the median
This line shows the upper quartile
This whisker shows the hightest value
The width of the box shows the interquartile range

# Cross Validation

- Validation technique for assessing how the results of a statistical analysis will generalize to an independent data set
- Randomly partition the data
- Train model on one set, test on the other set
- Repeat *k* times

# Decision Trees

- Maps observations about an element to labels
- Leaves are labels and branches are observations that lead to labels

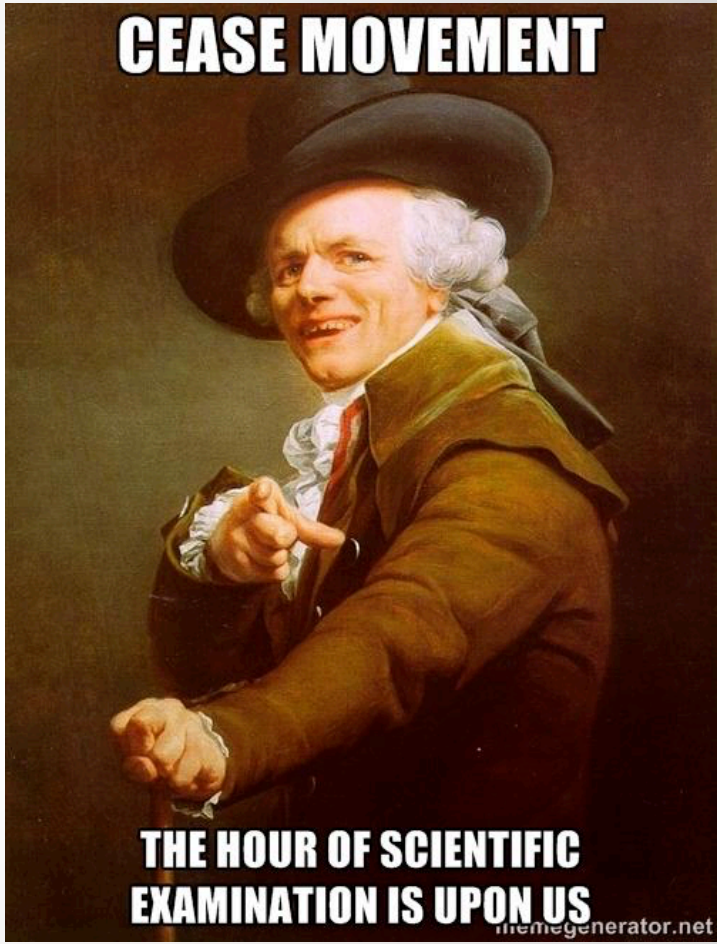


# Random Forest

- Forest (group) of randomly created decision trees
- Random feature selection for trees used to control variance
- Not prone to overfitting
- Classification answer is the result that appears most frequently from the decision trees

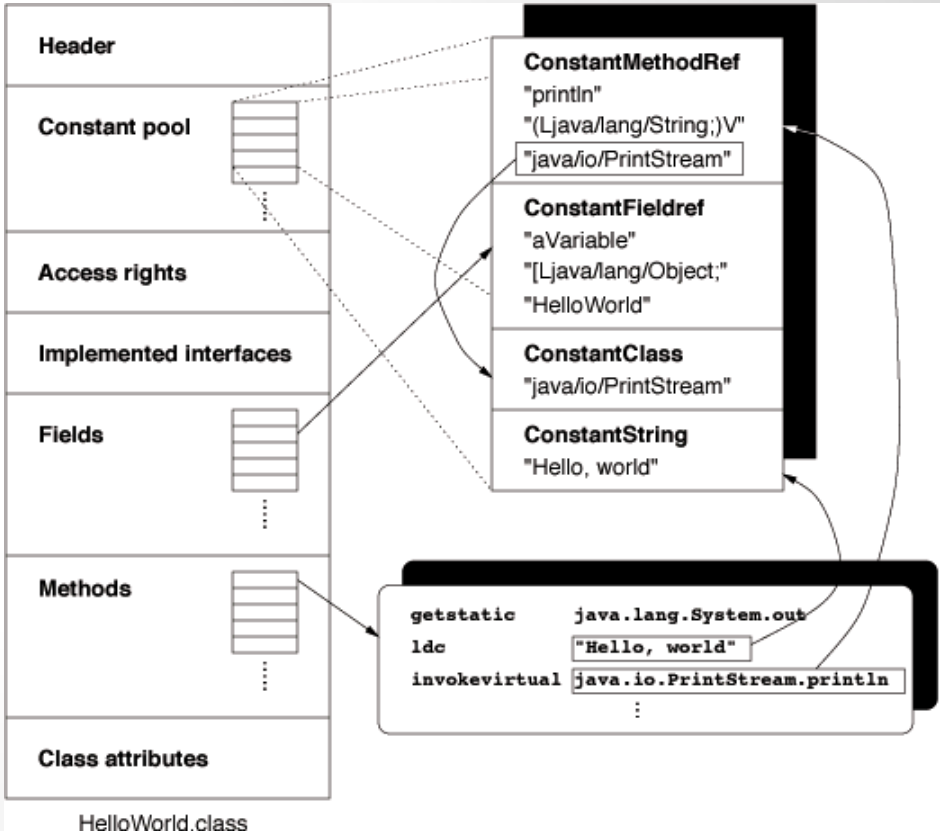
# The Data

- 300K uncompressed SWF
- 370K Java class files
- Can get IPython notebooks at https://github.com/ClickSecurity/data_hacking



# Java Class File Highlights

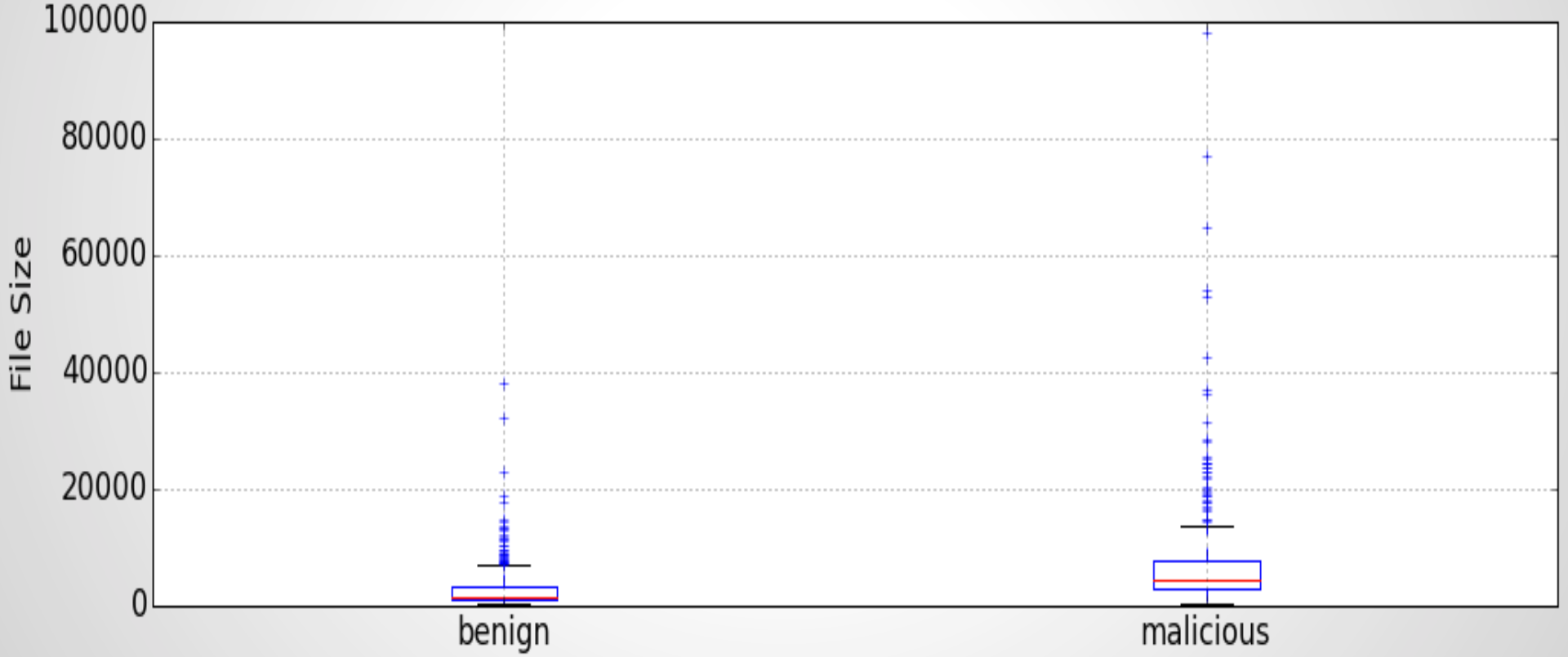
- Version
- Class name
- Super class name
- Constant Pool
- Access Flags
- Interfaces
- Methods
- Attributes



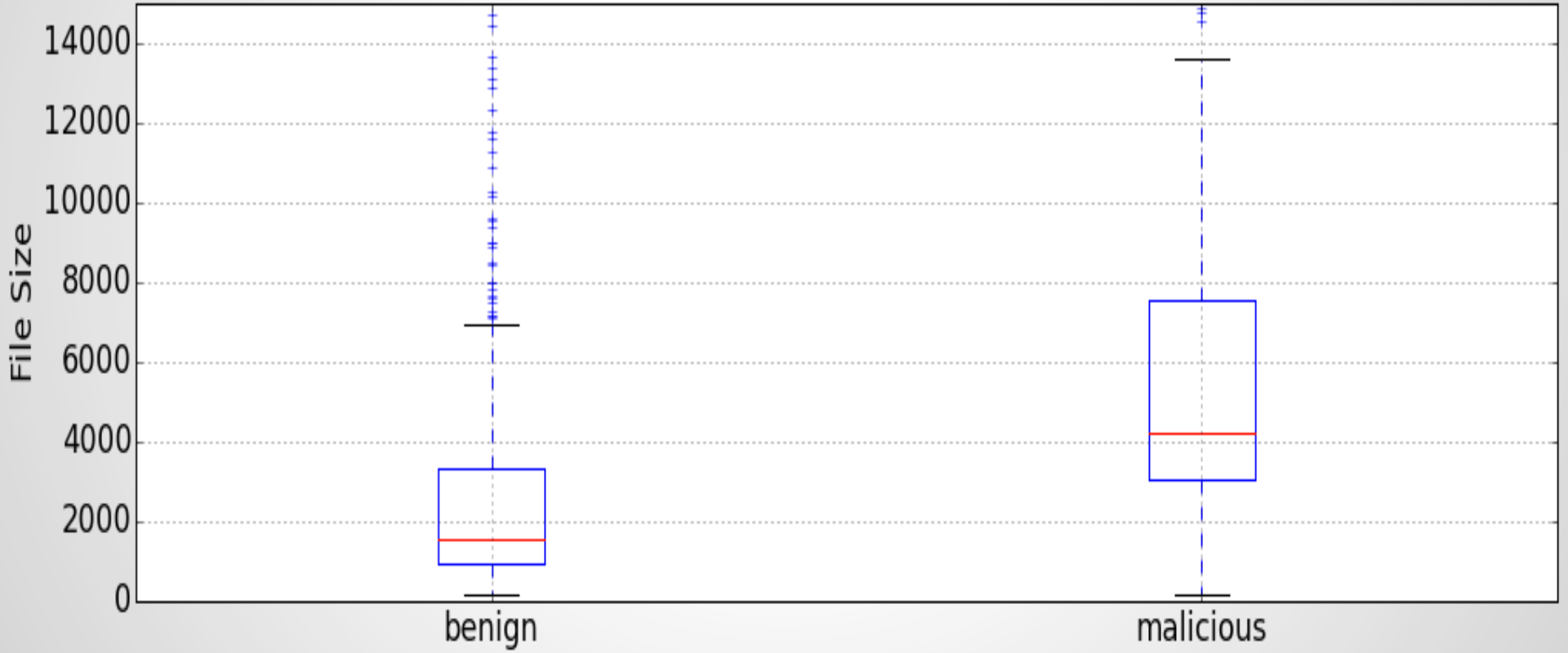
# Java Class Files

- 370K Java class files
  - Probably all benign
- 2K known malicious files
- Start with 500 of each

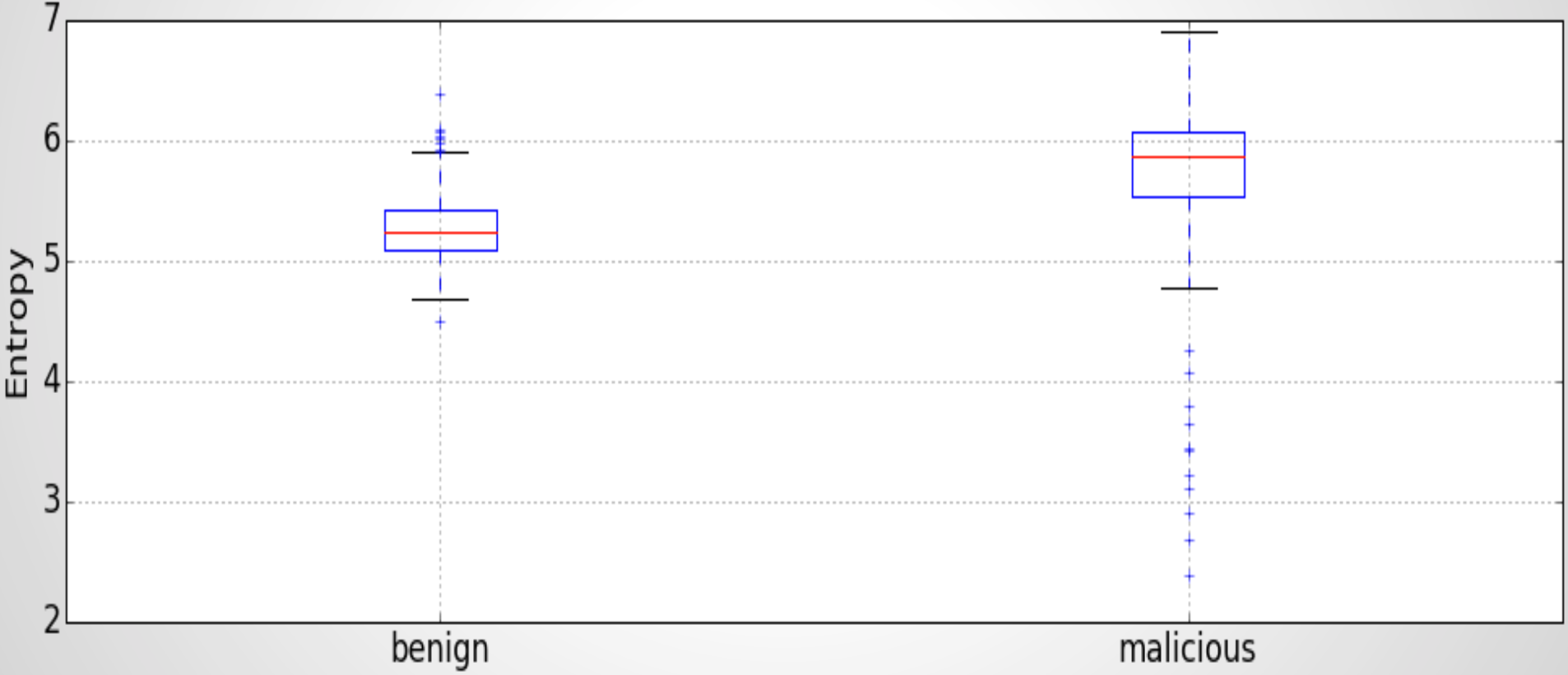
# File Size Comparison



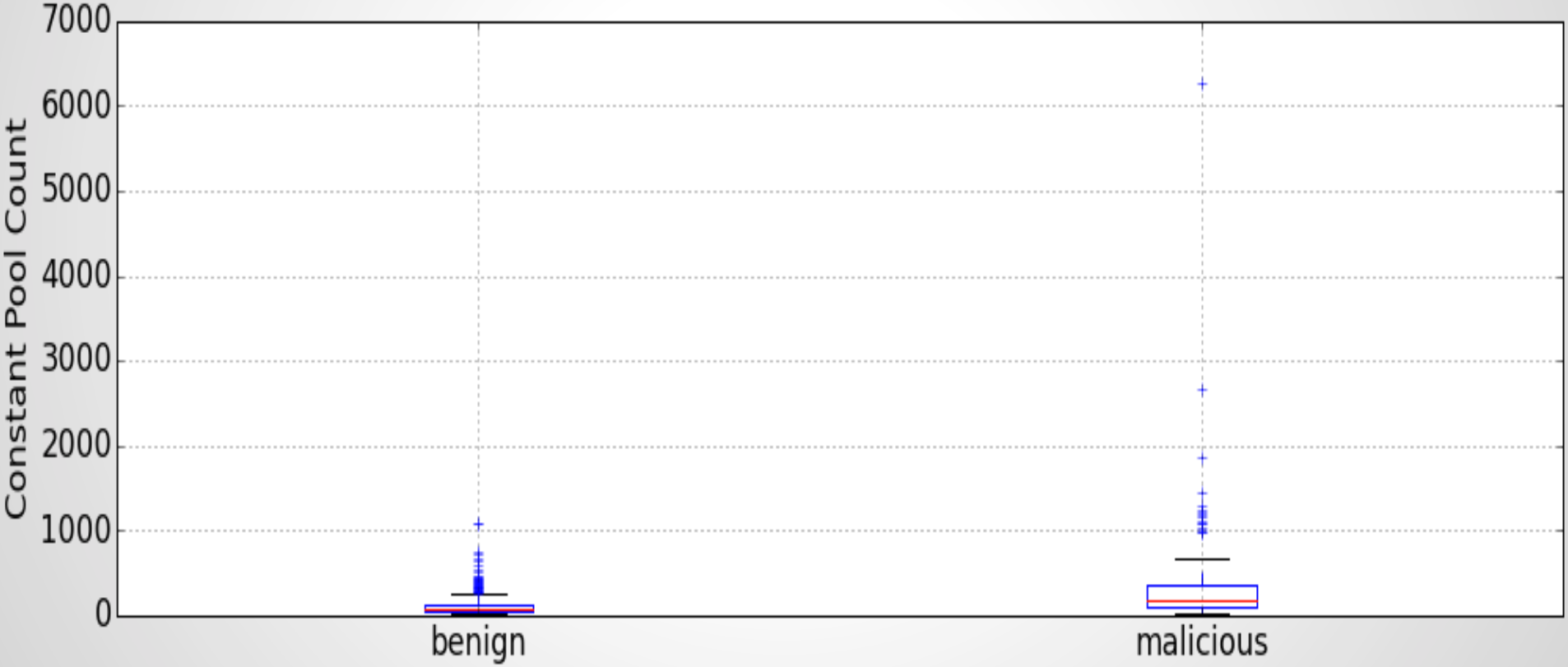
# File Size Comparison (zoomed in)



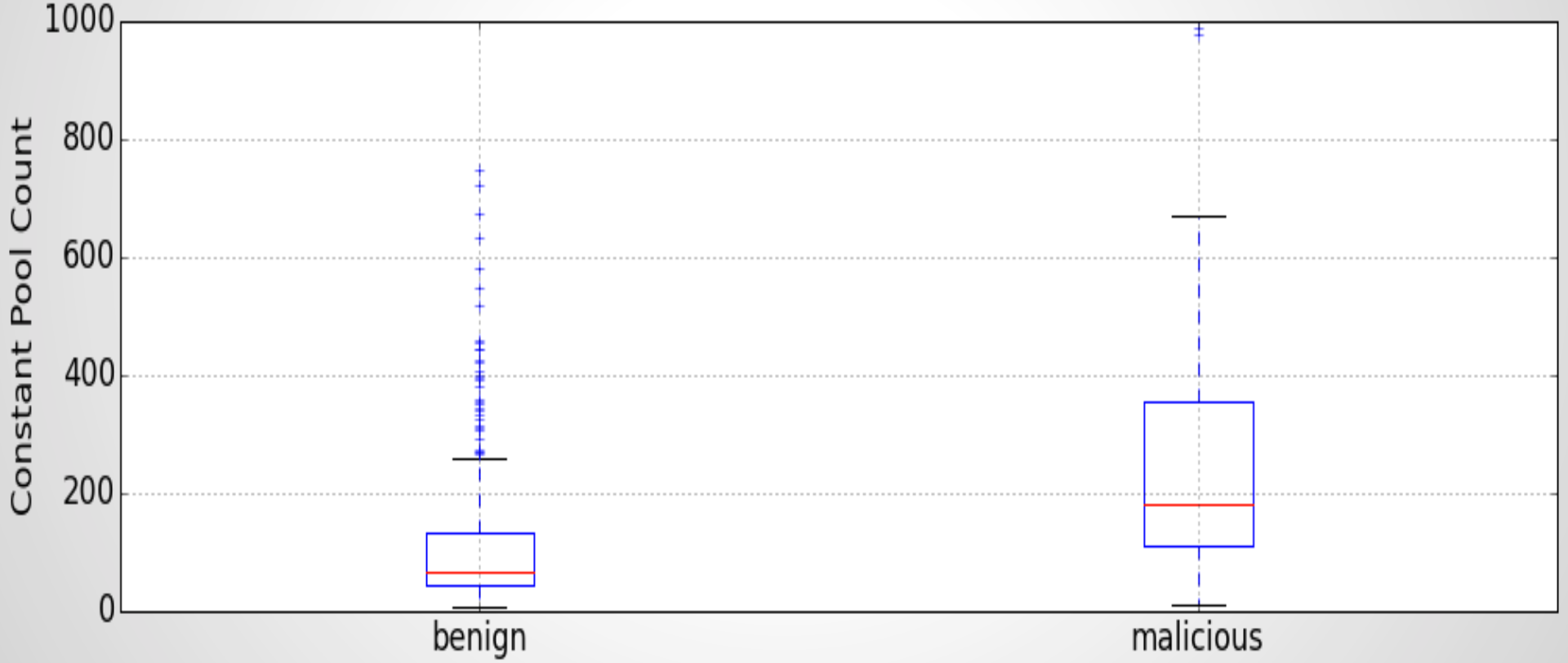
# Entropy Comparison



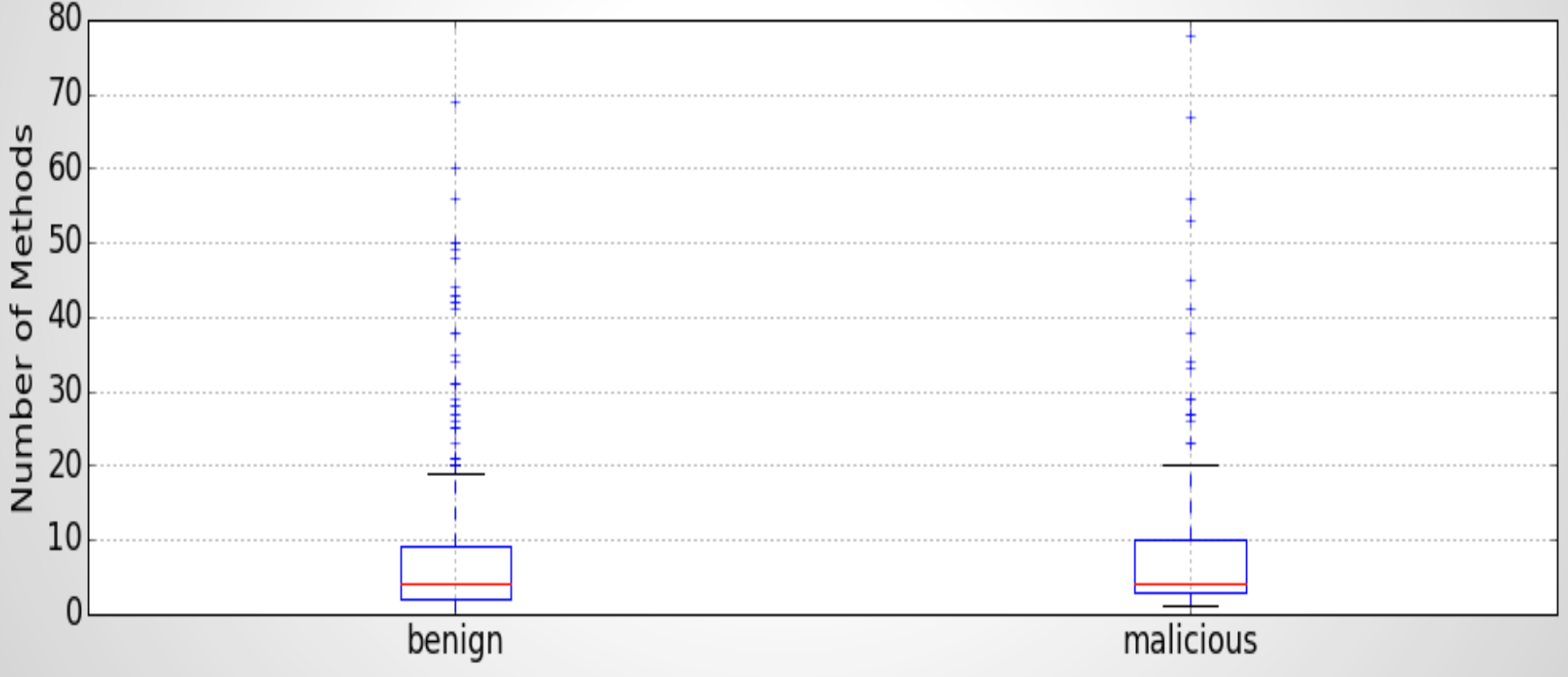
# Constant Pool Count Comparison



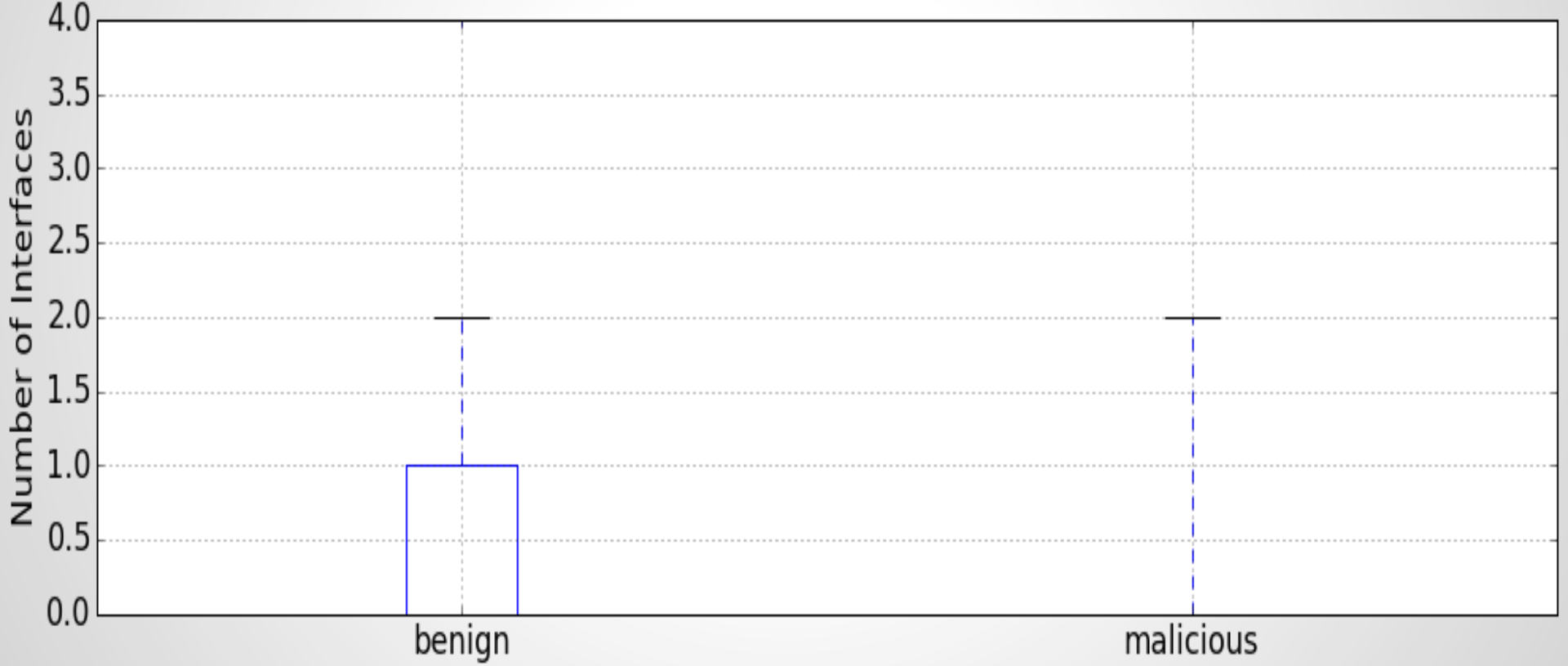
# Constant Pool Count Comparison (zoomed in)



# Number of Methods Comparison



# Number of Interfaces Comparison



# First Pass Classifier Feature Set

- Each flag from access rights
- Number of access flags set
- Major & minor version
- Entropy
- Size
- Constant pool size
- Number of methods
- Number of interfaces

# Cross Validation Results

- 10 fold
- Accuracy: 92.7% +/- 4.6

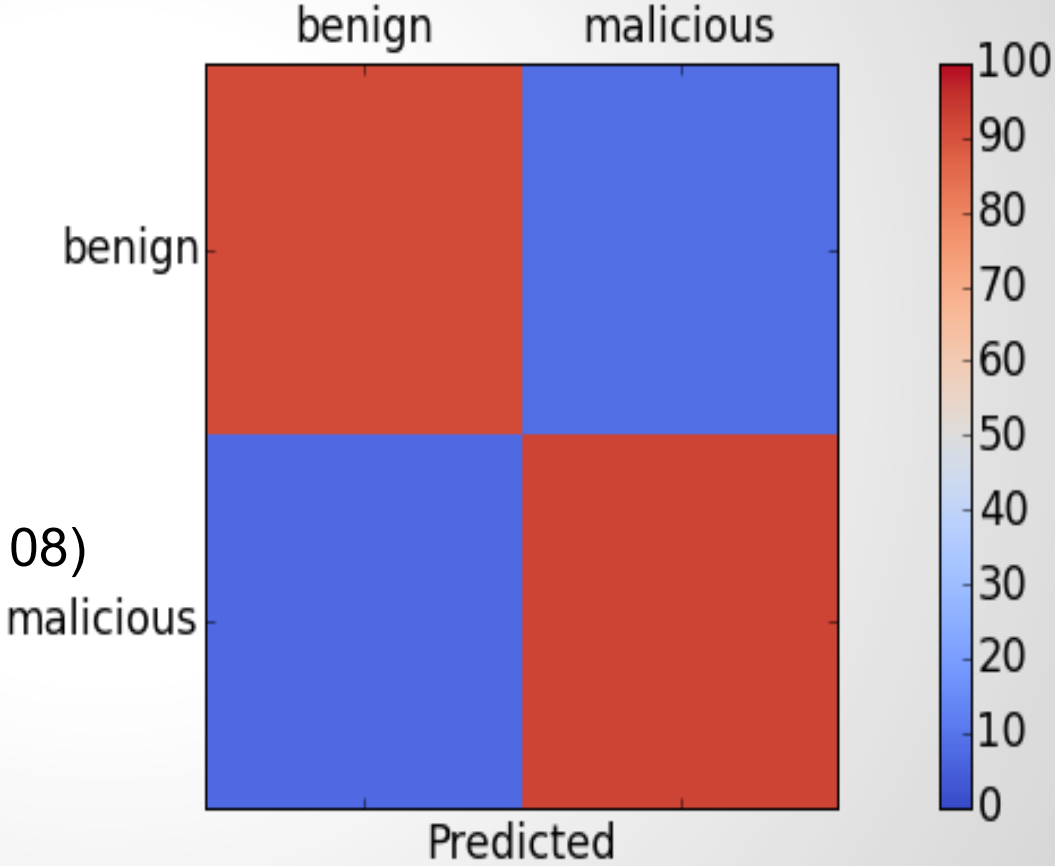
# Confusion Matrix

benign/benign: 91.67% (88/96)

benign/malicious: 8.33% (8/96)

malicious/benign: 7.41% (8/108)

malicious/malicious: 92.59% (100/108)



# Feature Importance

| | |
|---|---|
| Entropy | 0.30666 |
| Constant pool size | 0.22264 |
| File size | 0.15842 |
| Number of methods | 0.08960 |
| Number of interfaces | 0.05482 |
| Major version | 0.05328 |
| acc_public | 0.03754 |
| Number of access flags | 0.03316 |

# Need More Features

- Method names and attributes
- Class name
- Super class name

# Benign Method Names

| | |
|---|---|
| 0 | <init> |
| 1 | delegate |
| 2 | putIfAbsent |
| 3 | remove |
| 4 | replace |
| 5 | resetState |
| 6 | comparator |
| 7 | createElementSet |
| 8 | elementSet |
| 9 | descendingMultiset |
| 10 | firstEntry |

| | |
|---|---|
| 11 | lastEntry |
| 12 | pollFirstEntry |
| 13 | pollLastEntry |
| 14 | headMultiset |
| 15 | subMultiset |
| 16 | tailMultiset |
| 17 | invoke |
| 18 | hasNext |
| 19 | isValidLine |
| 20 | next |
| 21 | nextLine |

| | |
|---|---|
| 22 | close |
| 23 | closeQuietly |
| 24 | exec |
| 25 | getInitial |
| 26 | getIntermed |
| 27 | getFinal |
| 28 | max |
| 29 | outputSchema |
| 30 | estimateLength |
| 31 | appendTo |
| 32 | getXPath |

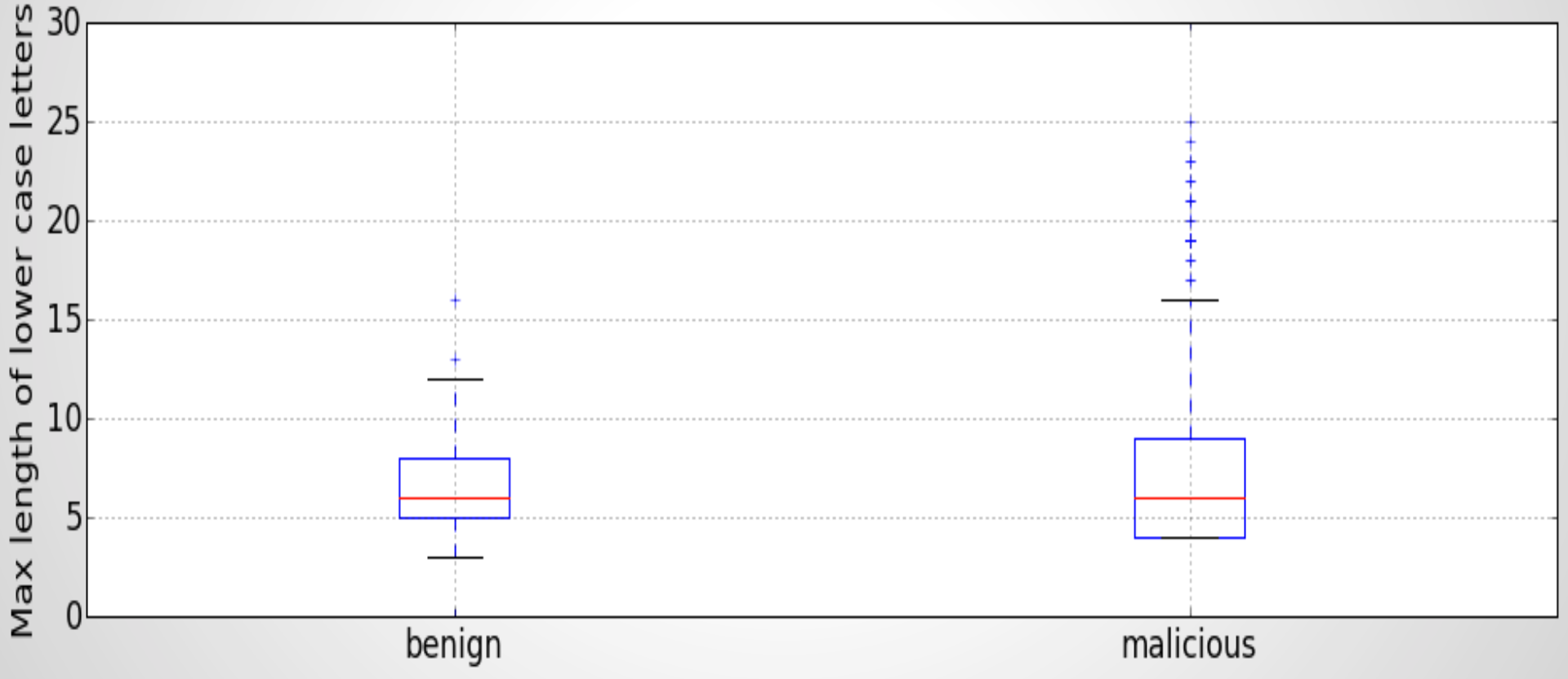
# Malicious Method Names

| | |
|---|---|
| 0 | <init> |
| 1 | init |
| 2 | ktCgxlqo |
| 3 | <clinit> |
| 4 | gvuNr |
| 5 | eiaxyercdfvbgscpbv |
| 6 | yginlmcynkyuohnfhe |
| 7 | mtyvzetsjhvnbyz |
| 8 | fxxhgjttqfavlooxcb |
| 9 | wyjgamzmowywjihkuuf |
| 10 | kgthsnqdqutacivcptong |
| 11 | qgasjqrogibkblyzourtq |
| 12 | glfouhczfxzyskaystx |
| 13 | mikczoanebdkwpyb |
| 14 | bwssduenvebnvgix |
| 15 | wafrcwijizypmitodmb |
| 16 | bfznyeevclzzxxqbw |
| 17 | jmzisxwtxhekbkl |
| 18 | szivddjiptybevduli |
| 19 | forwnxmgnutbtdwvptj |
| 20 | mwwmrvljafpkwzdiy |
| 21 | vvpbdzrhvvnzaieyi |
| 22 | qkkxoygluwwlnwbxu |
| 23 | dvvwse |
| 24 | c |
| 25 | k |
| 26 | main |
| 27 | writeEmbeddedFile |
| 28 | bootstrap |
| 29 | getJreExecutable |
| 30 | addExtension |
| 31 | findInDir |
| 32 | normalize |

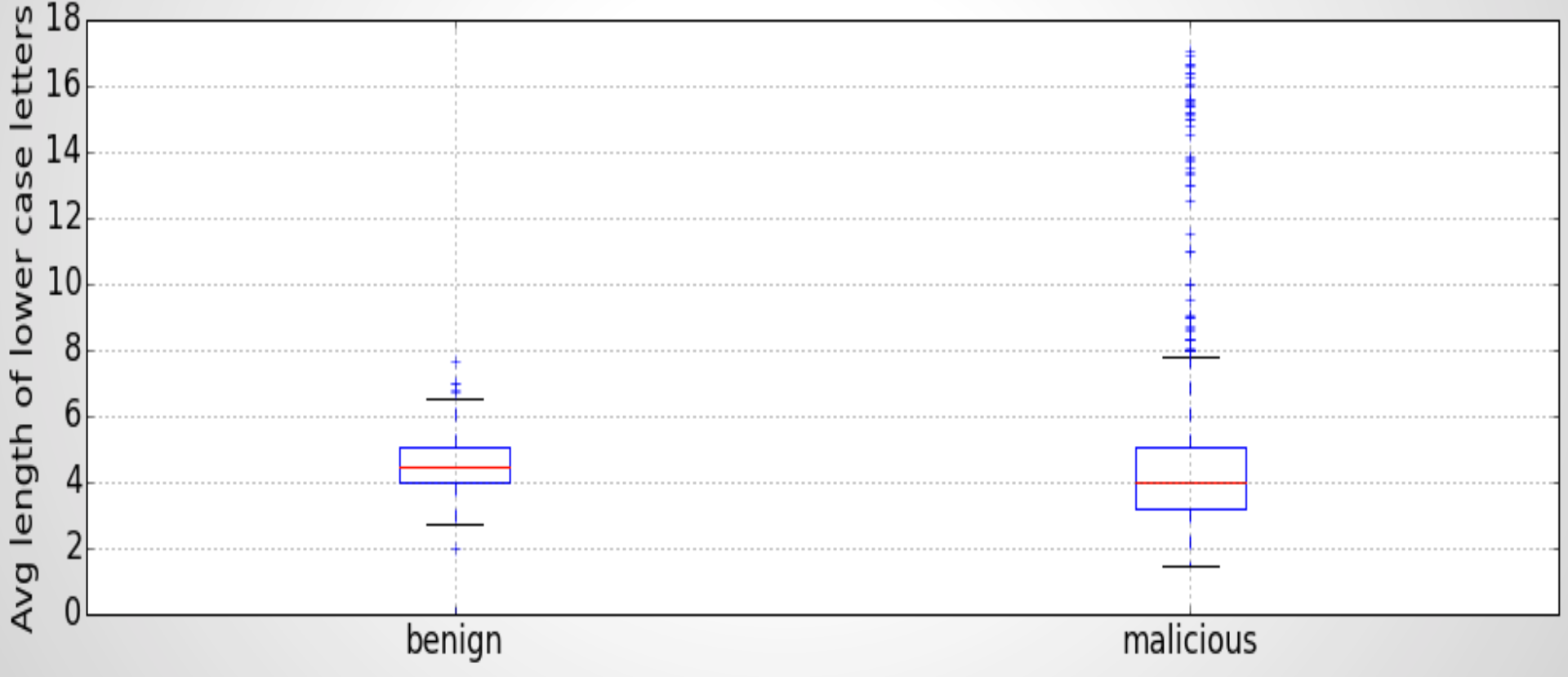
# Feature Creation

- Took inspiration from spam filter
  - http://thinktostart.com/build-a-spam-filter-with-r/
- Max length of consecutive digits, lower case, and upper case letters
- Average length of consecutive digits, lower case, and upper case letters

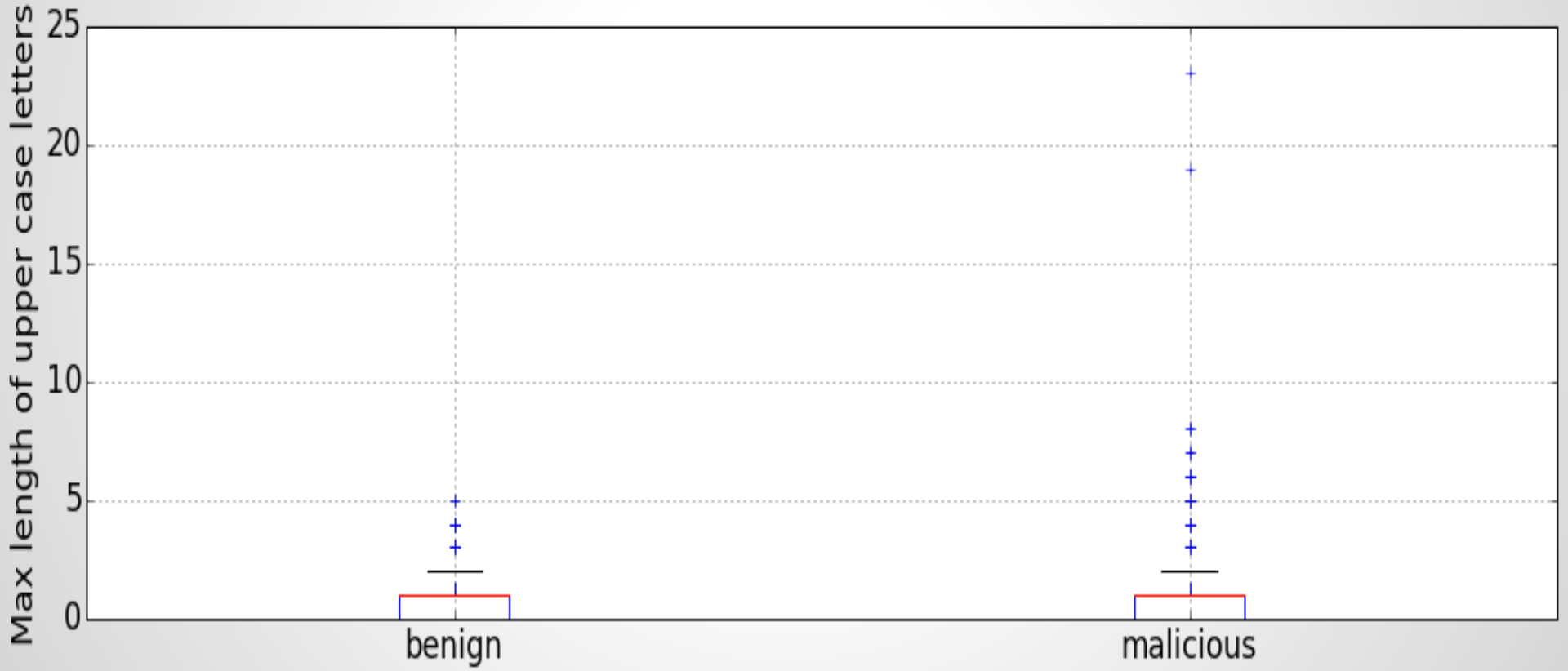
# Max Length of Consecutive Lower Case Letters



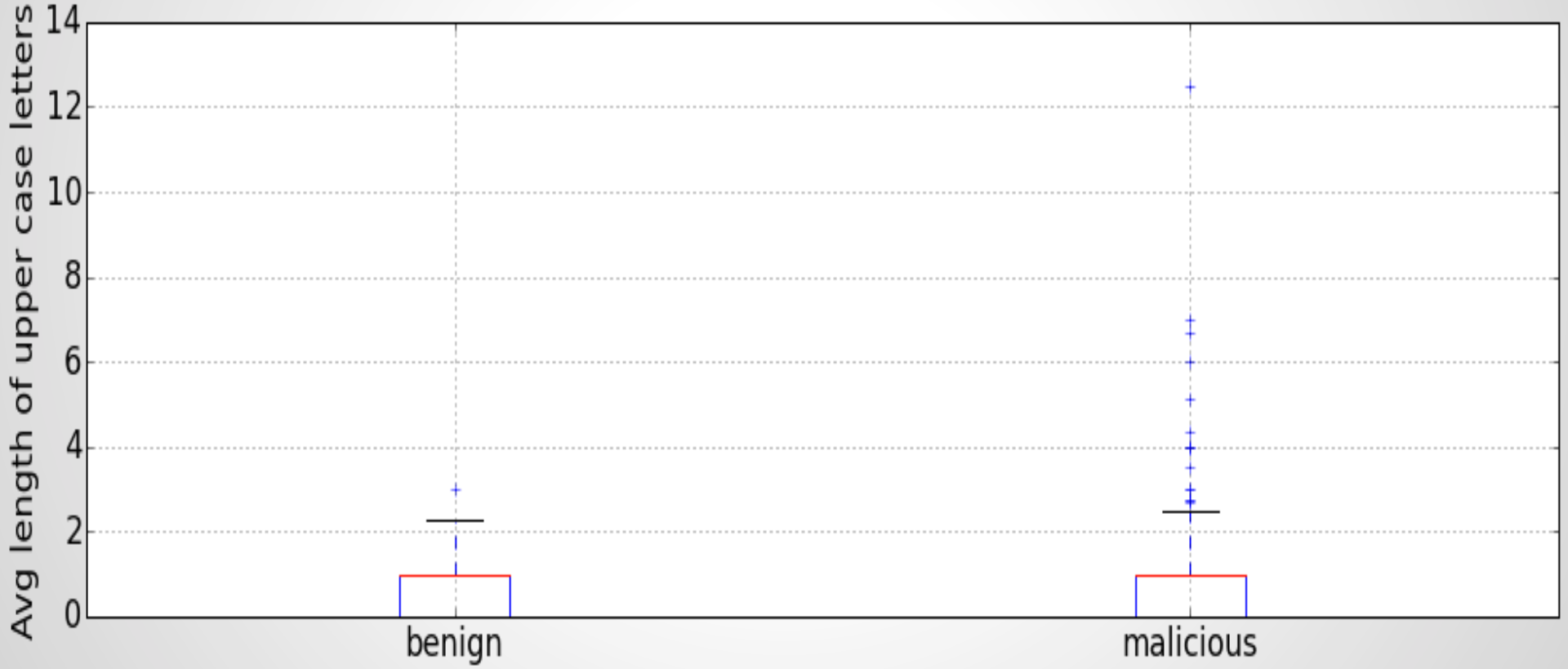
# Average Length of Consecutive Lower Case Letters



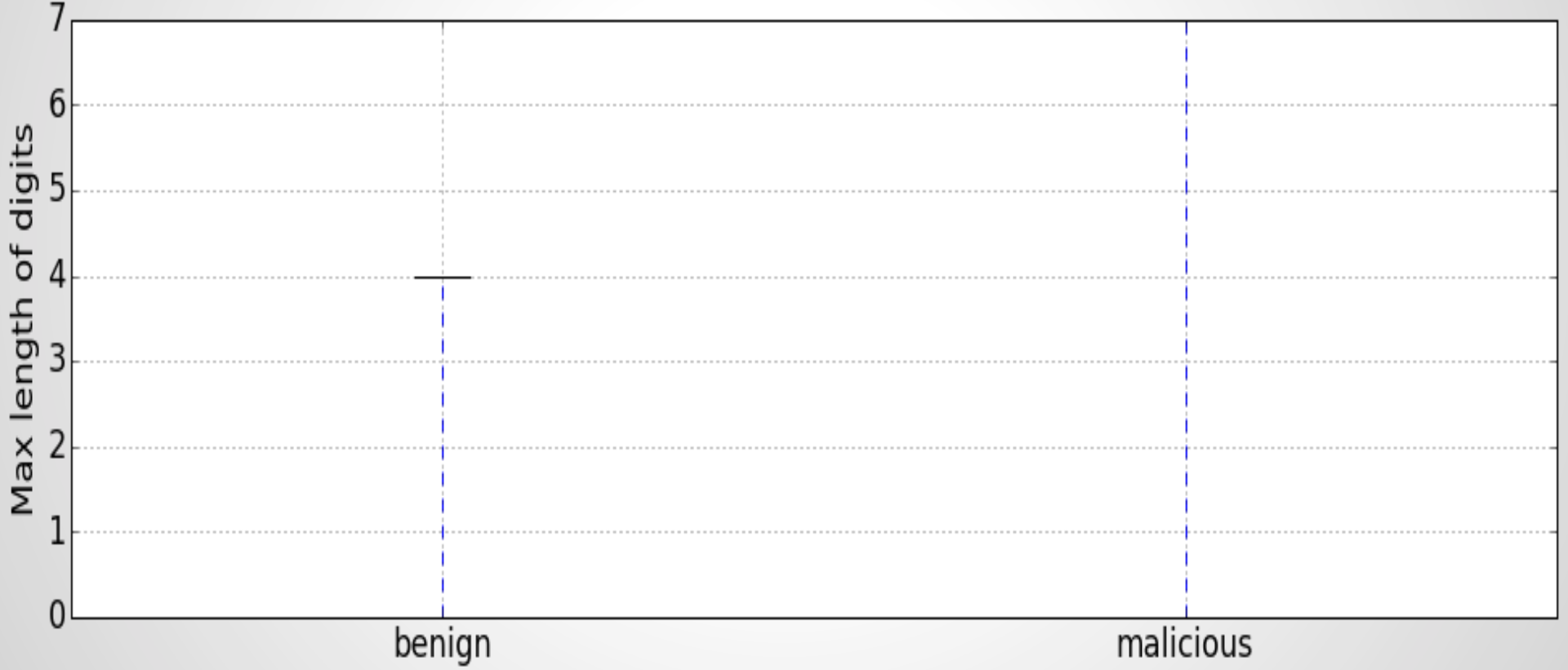
# Max Length of Consecutive Upper Case Letters



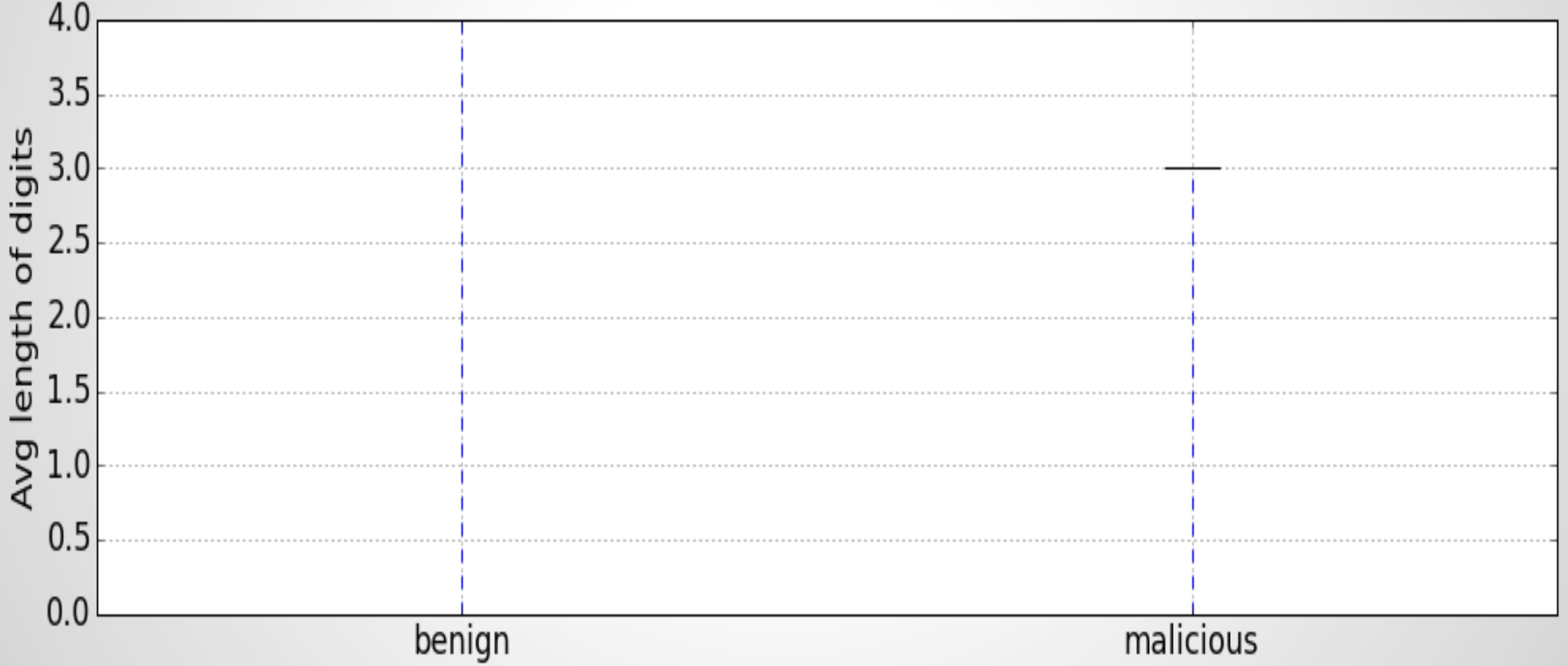
# Average Length of Consecutive Upper Case Letters



# Max Length of Consecutive Digits



# Average Length of Consecutive Digits



# Cross Validation Results

- 10 fold
- Accuracy: 94.9% +/- 4.0

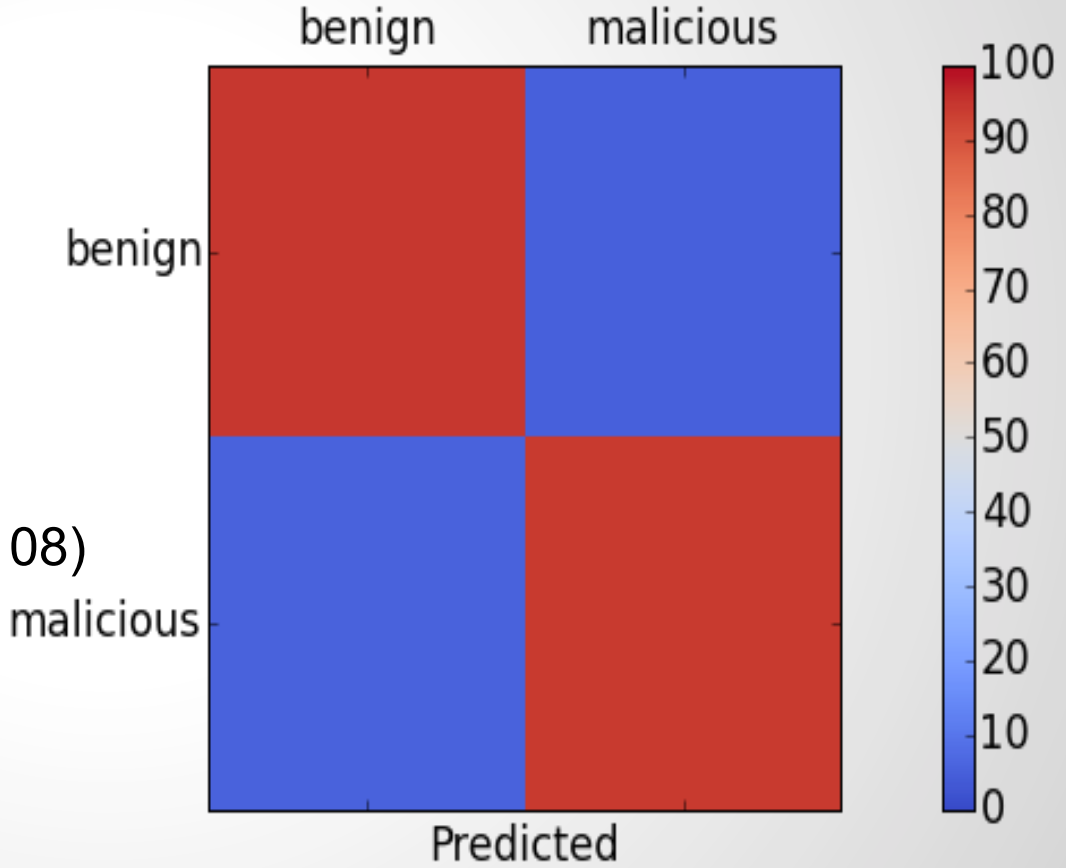
# Confusion Matrix

benign/benign: 94.79% (91/96)

benign/malicious: 5.21% (5/96)

malicious/benign: 5.56% (6/108)

malicious/malicious: 94.44% (101/108)



# Feature Importance

| | |
|---|---|
| Entropy | 0.25620 |
| Constant pool size | 0.13497 |
| File size | 0.12721 |
| Method name lower case avg | 0.08061 |
| Method name lower case max | 0.05628 |
| Method name upper case avg | 0.05525 |
| Number of interfaces | 0.05329 |
| Number of methods | 0.05135 |

# Still Need More Features

- Still have class name
- Still have super class name
- Method attributes
- Class attributes
- Interface

# Benign Class Names

```
com/google/common/collect/ForwardingConcurrentMap
org/apache/hadoop/io/compress/GzipCodec$GzipOutput
com/google/common/collect/Multisets$UnmodifiableS
hu/openig/mechanics/StaticDefensePlanner$1
org/apache/commons/io/LineIterator
org/apache/pig/builtin/IntMax
org/apache/commons/lang/time/FastDateFormat$String
hu/openig/screen/items/ResearchProductionScreen$1
org/dom4j/XPathException
threadWordlistExec
org/odlabs/wiquery/ui/autocomplete/AbstractAutocor
org/xml/sax/ContentHandler
```

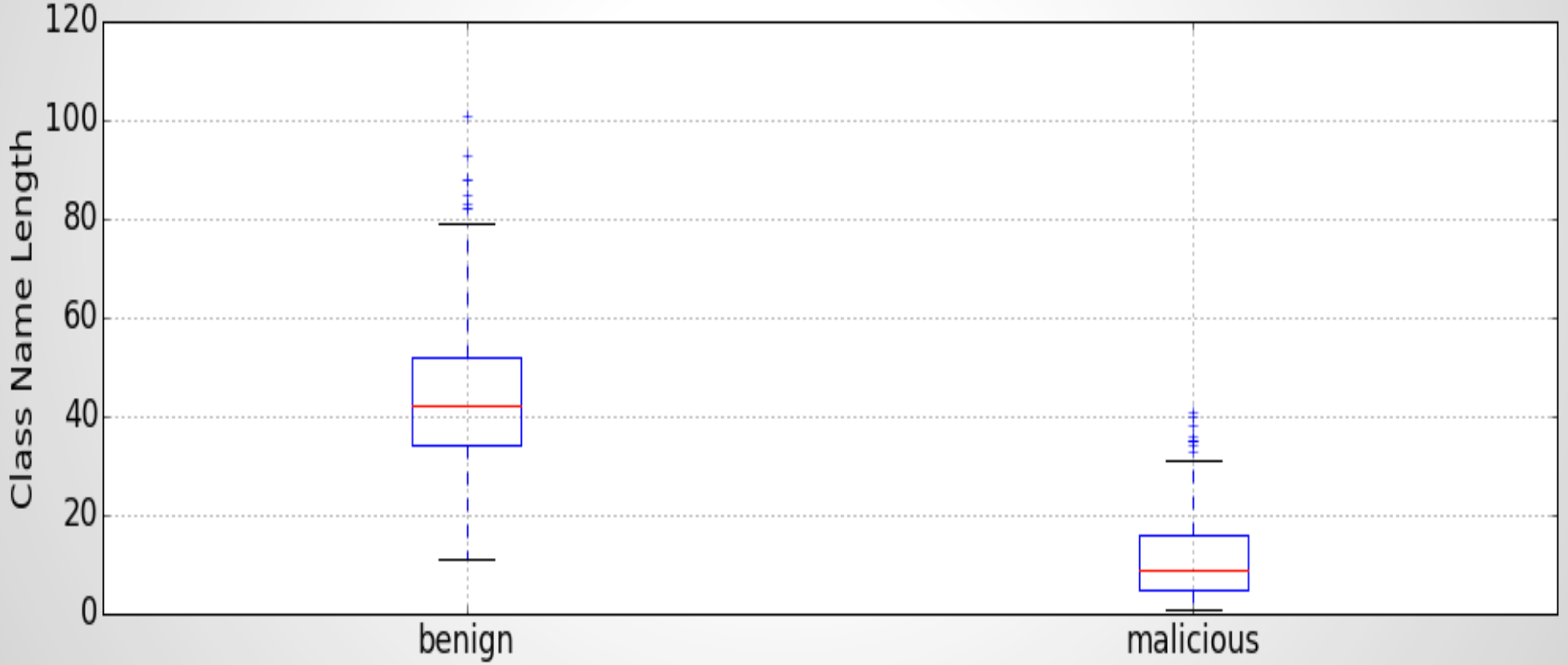
# Malicious Class Names

```
Main
YdCdHX/VcZaXVjyy
aOcMSp
a/zylasqwjlpbqyrwrr
tljpjunbjwtqlywm/sdnrybknlf
Mainer
Main
mNIJnGIOkm/Payload
aHrMCrboe/chspSxY
Main
OSrAfQWThe/SHLeanN
hhIji/XQDODV
```

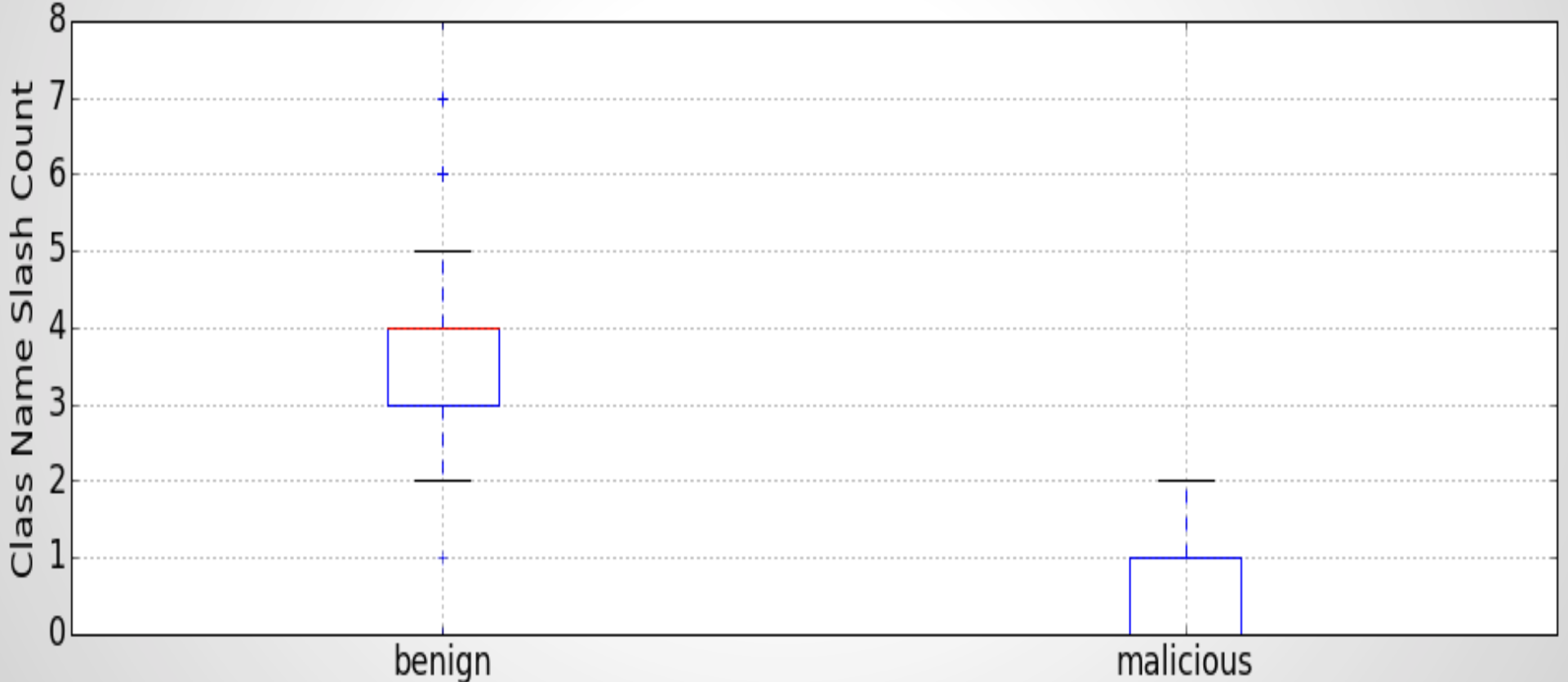
# More Feature Creation

- Length of class name
- Number of slashes in class name
- Max length of consecutive digits, lower case, and upper case letters
- Average length of consecutive digits, lower case, and upper case letters

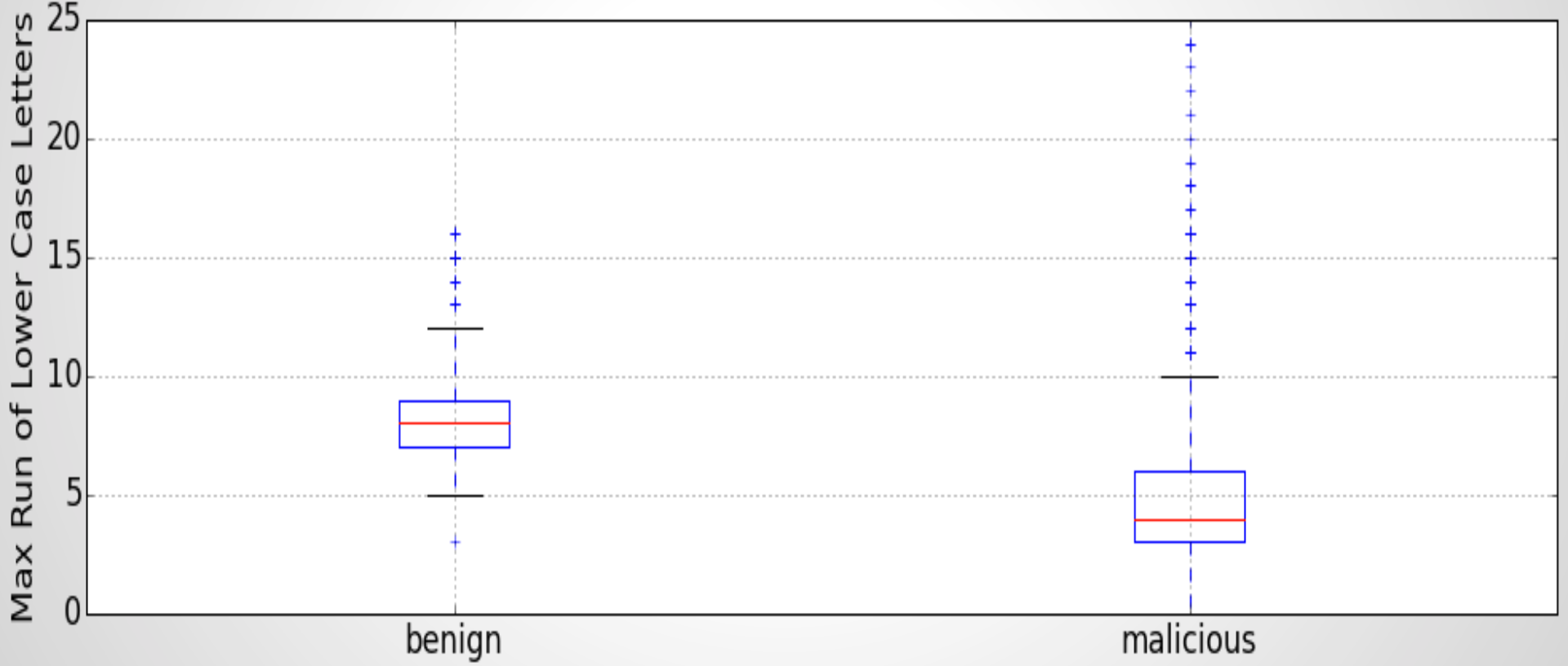
# Class Name Length



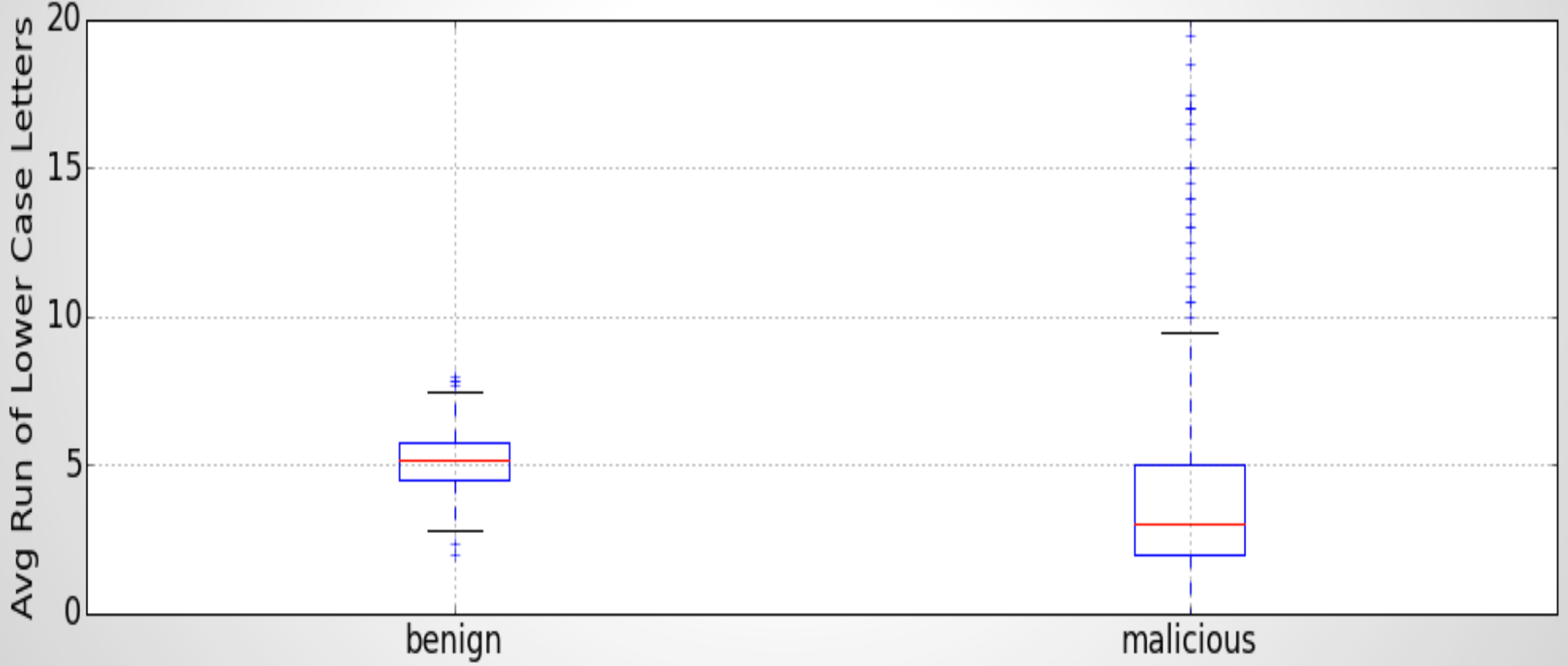
# Class Name Slash Count



# Longest Run of Consecutive Lowercase Letters



# Average Run of Consecutive Lowercase Letters



# Cross Validation Results

- 10 fold
- Accuracy: 99.4% +/- 1.3

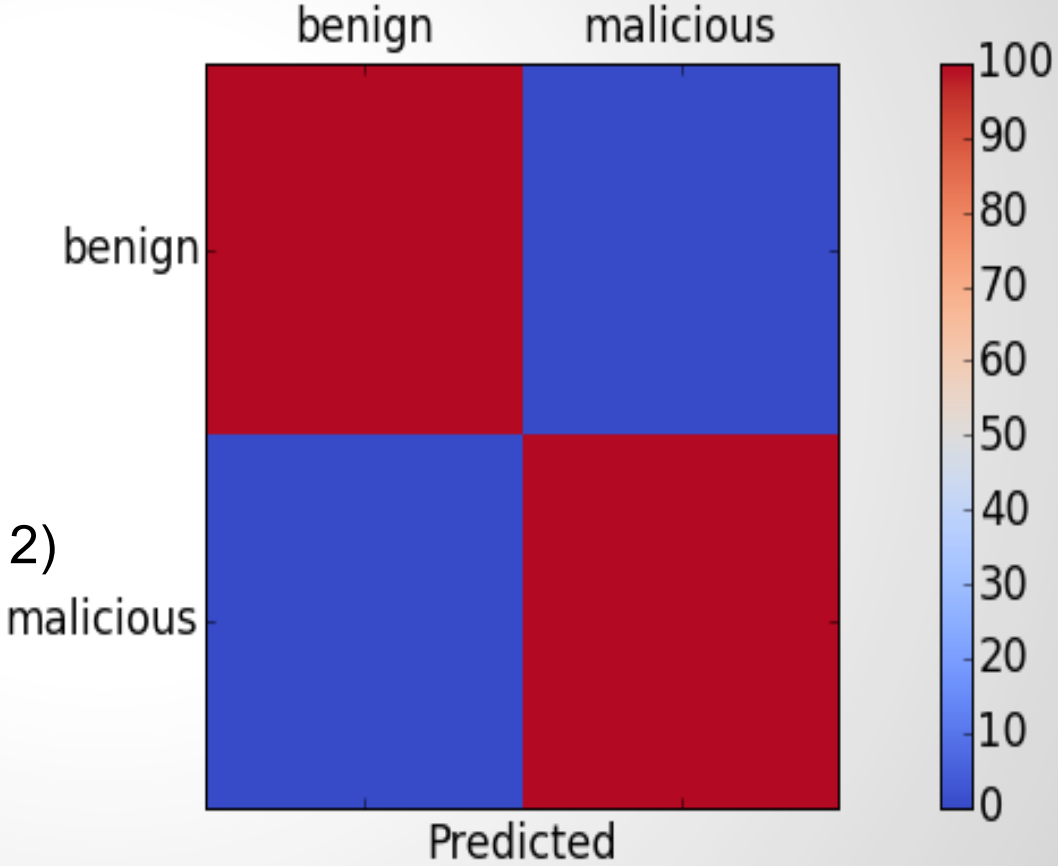
# Confusion Matrix

benign/benign: 100.00% (92/92)

benign/malicious: 0.00% (0/92)

malicious/benign: 0.0% (0/112)

malicious/malicious: 100.00% (0/112)



# Knobs To Turn

- Random Forest allows you to modify the threshold for classification
- The default threshold in 2 label case is 50%
- We can increase the threshold to decrease FP rate
- We can decrease the threshold to increase TP rate

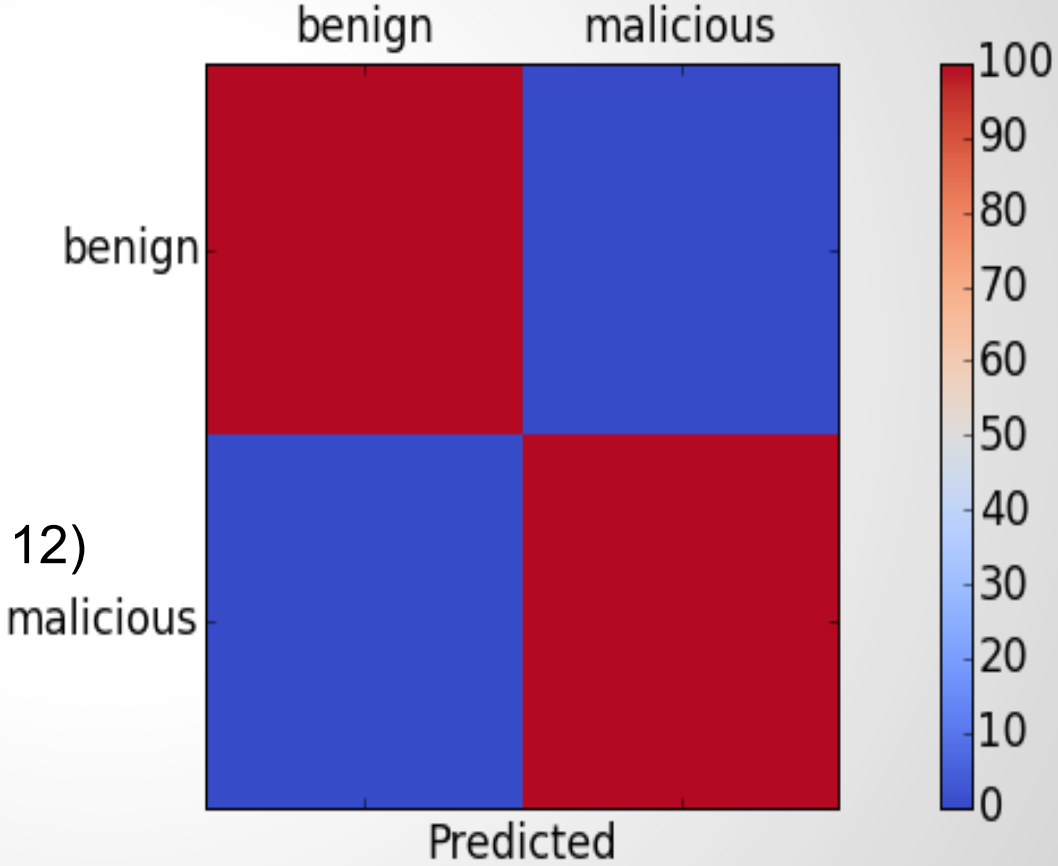
# 80% Confusion Matrix

benign/benign: 100.00% (92/92)

benign/malicious: 0.00% (0/92)

malicious/benign: 2.68% (3/112)

malicious/malicious: 97.32% (109/112)



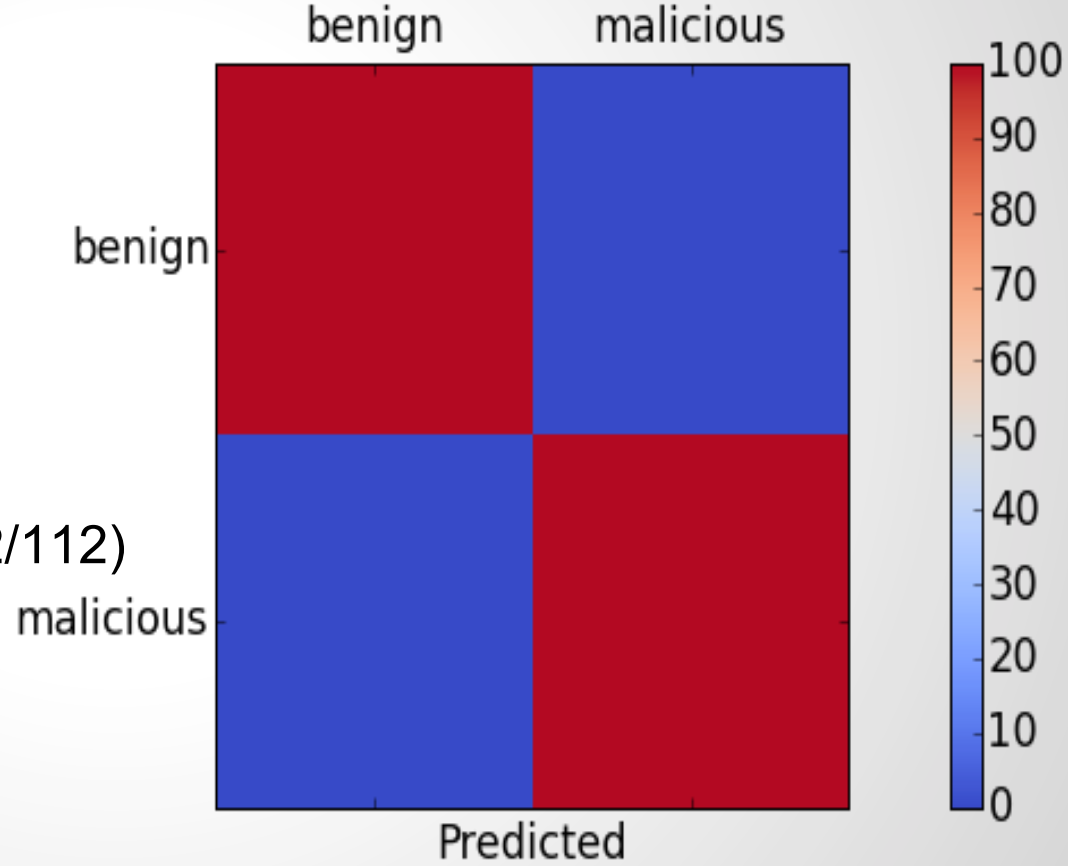
# 20% Confusion Matrix

benign/benign: 98.91% (91/92)

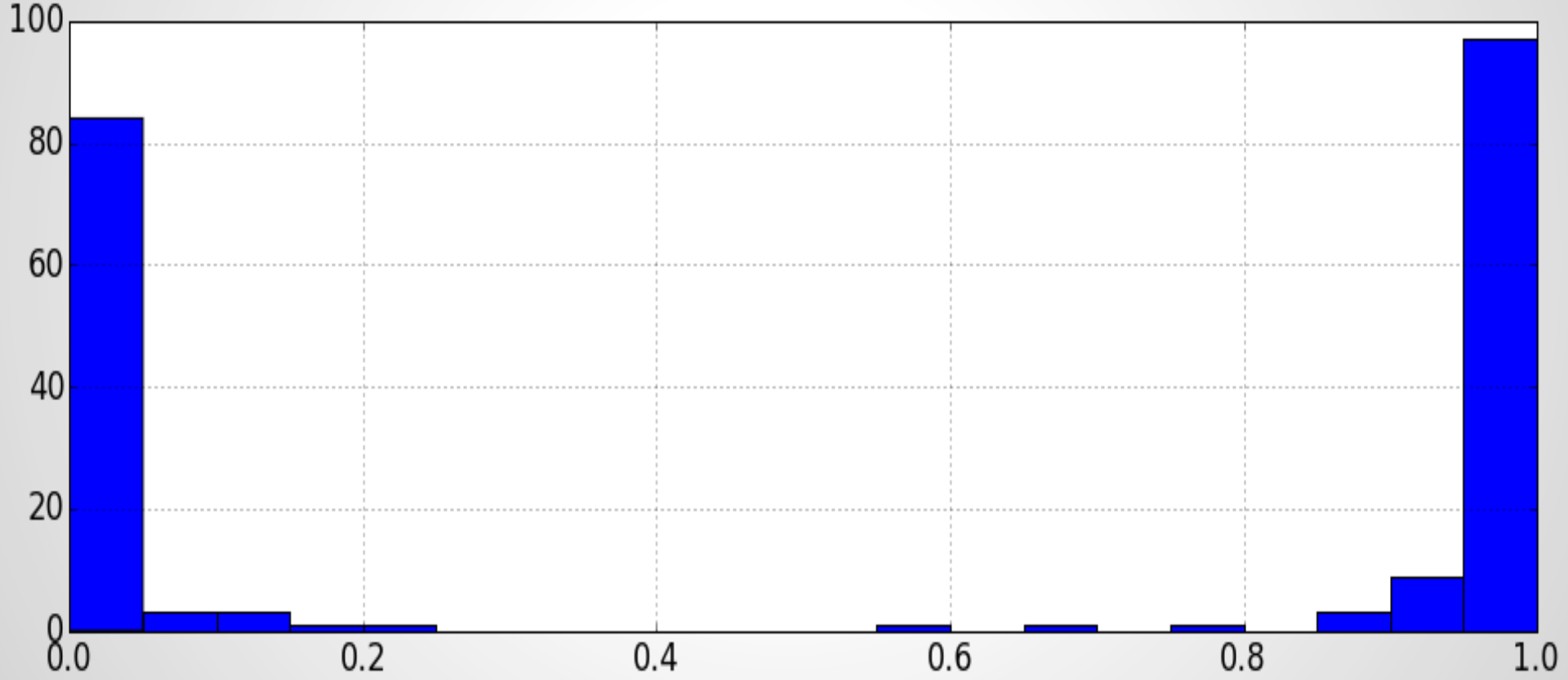
benign/malicious: 1.09% (1/92)

malicious/benign: 0.00% (0/112)

malicious/malicious: 100.00% (112/112)



# Score Distribution



# Feature Importance

| | |
|---|---|
| Class Name Slash Count | 0.30258 |
| Class Name Length | 0.27841 |
| Class Name Lower Case Max | 0.07751 |
| Entropy | 0.06485 |
| Class Name Lower Case Avg | 0.05893 |
| Constant Pool Count | 0.04767 |
| Class Name Upper Case Max | 0.03479 |
| Size | 0.02790 |

# Classifier Results

- How does the classifier work on the 366K files?

- 340K files marked as benign (score < 0.5)
- 11K files in gray area (0.5 <= score < 0.8)
- 15K files marked as malicious (0.8 <= score)

# #iago

- Not really a surprise
- 500 files of each is really not enough
- Let’s use more data!

# A New Classifier

- 2000 labeled malicious files
- Label 2000 randomly chosen files as benign
- Cross Validation Score: 99.0% +/- 1.1

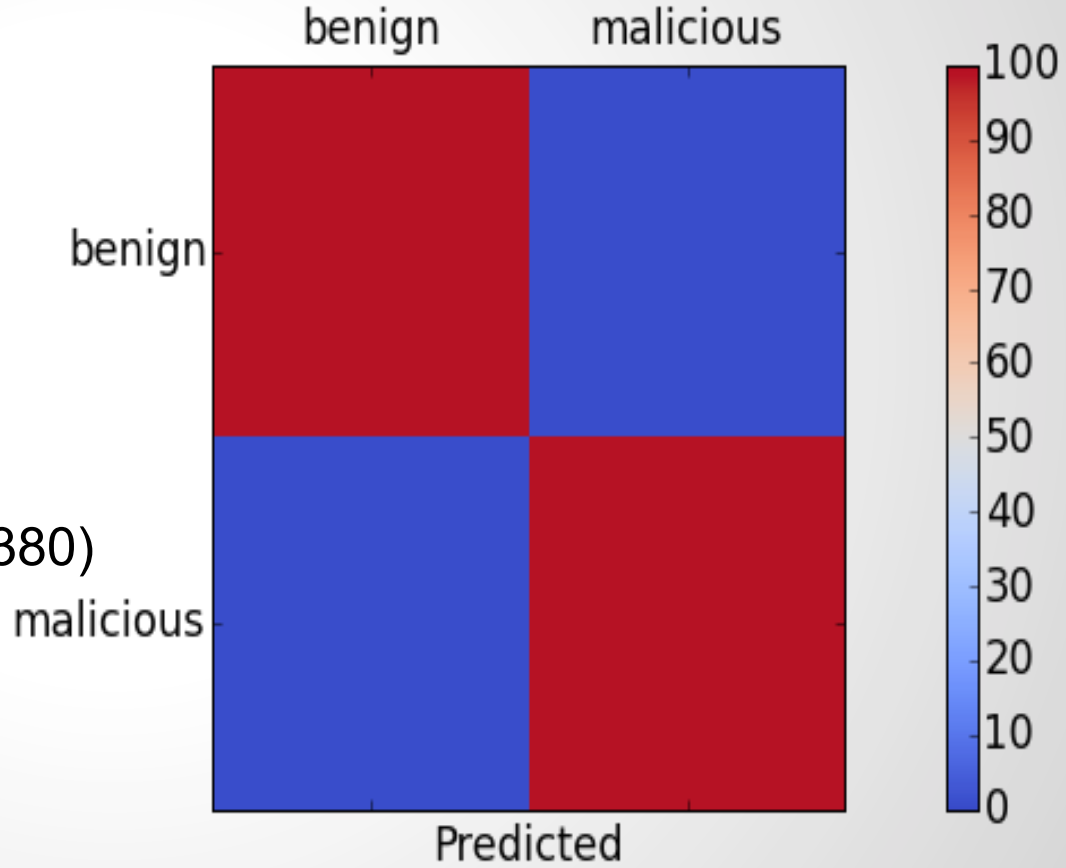
# Confusion Matrix

benign/benign: 99.01% (400/404)

benign/malicious: 0.99% (4/404)

malicious/benign: 1.05% (4/380)

malicious/malicious: 98.95% (376/380)



# Classifier Results - Part 1

- How does the classifier work on the 364K files?

- 299K files marked as benign (score < 0.5)
- 64K files in gray area (0.5 <= score < 0.8)
- 840 files marked as malicious (0.8 <= score)

# Classifier Results - Part 2

- Train another classifier using a different random 2000 benign files

- 360K files marked as benign (score < 0.5)
- 3.3K files in gray area (0.5 <= score < 0.8)
- 1.3K files marked as malicious (0.8 <= score)

# Classifier Results - Part 3

- Train yet another classifier using a different random 2000 benign files

- 359K files marked as benign (score < 0.5)
- 3.7K files in gray area (0.5 <= score < 0.8)
- 1.3K files marked as malicious (0.8 <= score)

# Java Summary

- Can detect malicious files using data science
- Variability in results on large pile of files depending on test data
- A classifier that seems good at classifying class names
- More data seems to give better results

# Next Steps

- More malicious files
- Better labeled data
  - Not just good/bad
  - Exploit Kit
  - CVE
- New features
  - Better features from method and class names
  - More features about methods
  - N-Gram analysis on actual Java bytecode

# Adobe Flash

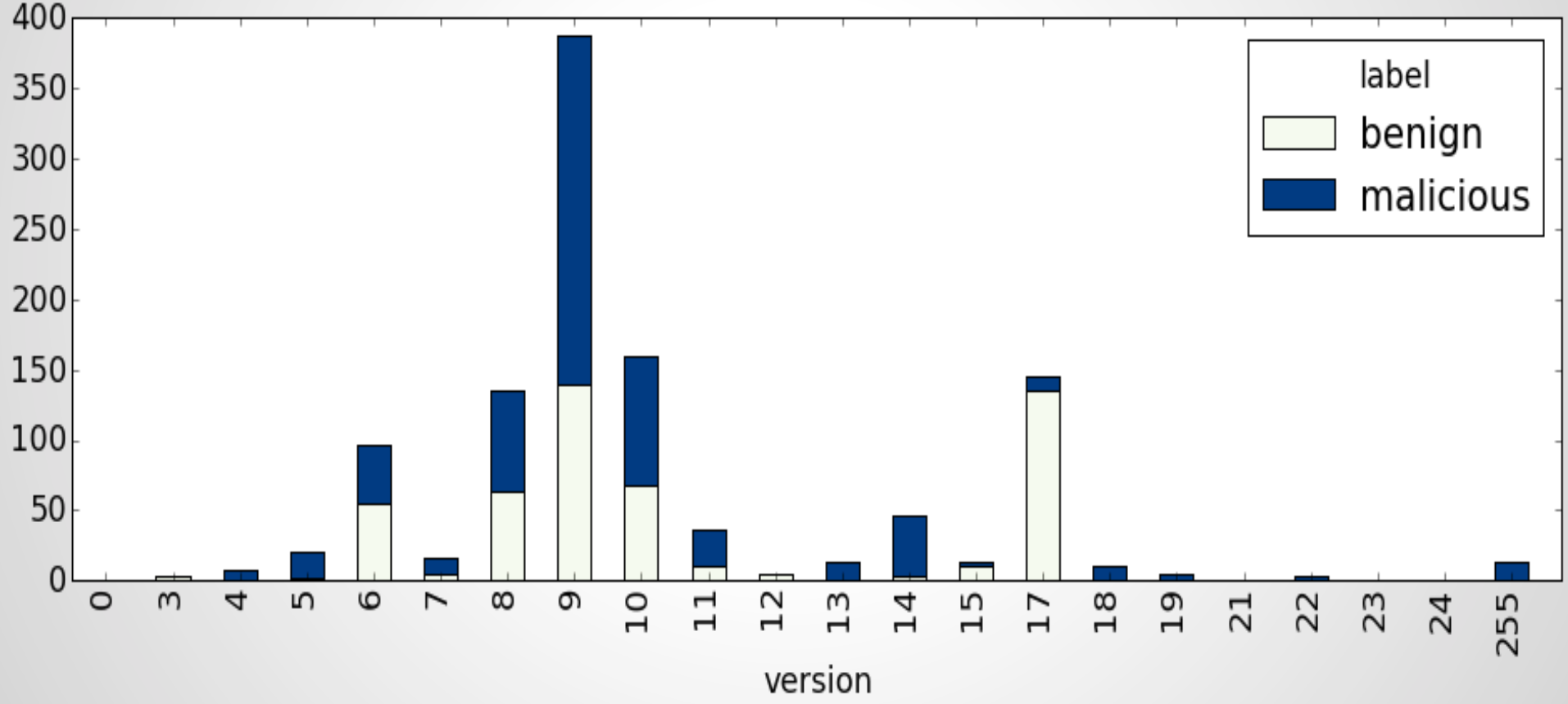
- 288K Uncompressed SWF
  - Probably all benign
- 628 known malicious files
- Start with 500 benign and all the malicious

# SWF File Highlights

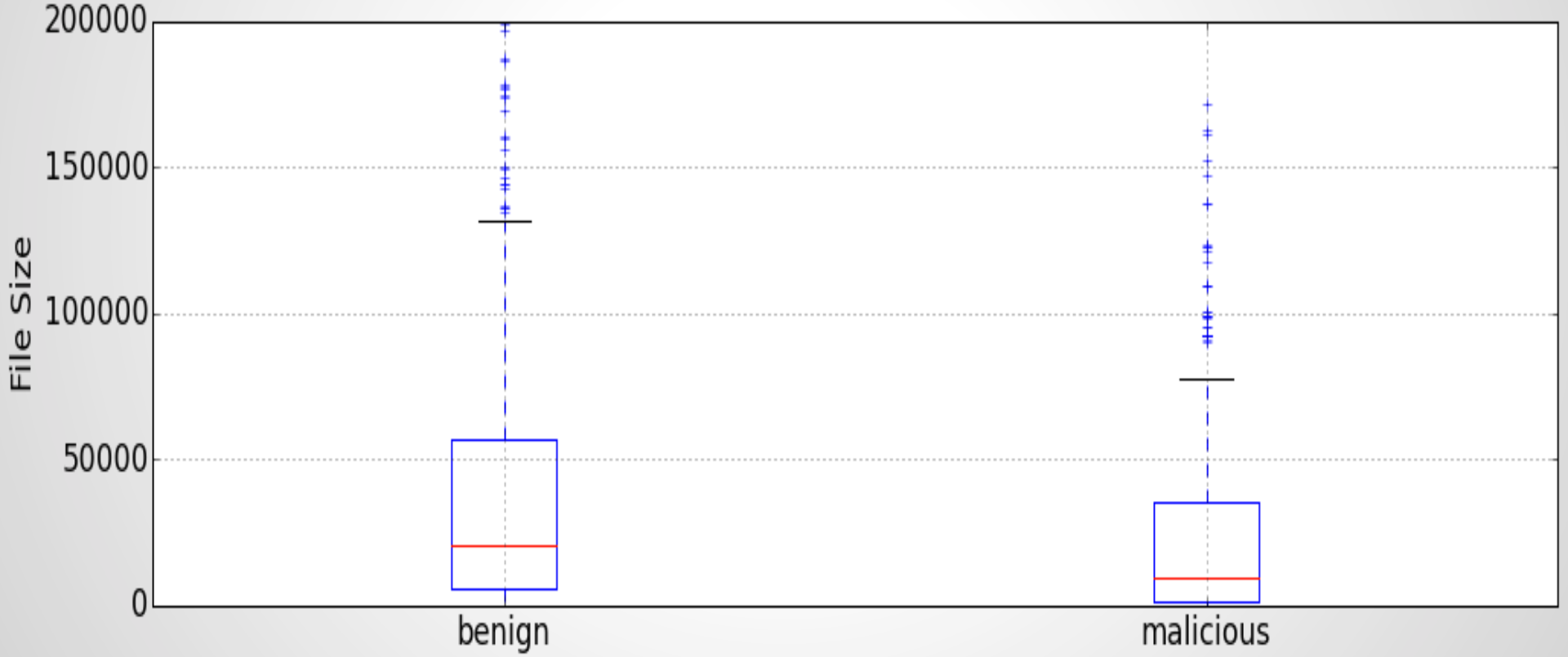

- Version
- Frame Count
- Frame Rate
- Frame Size

- Number Of Tags
- Tag Information

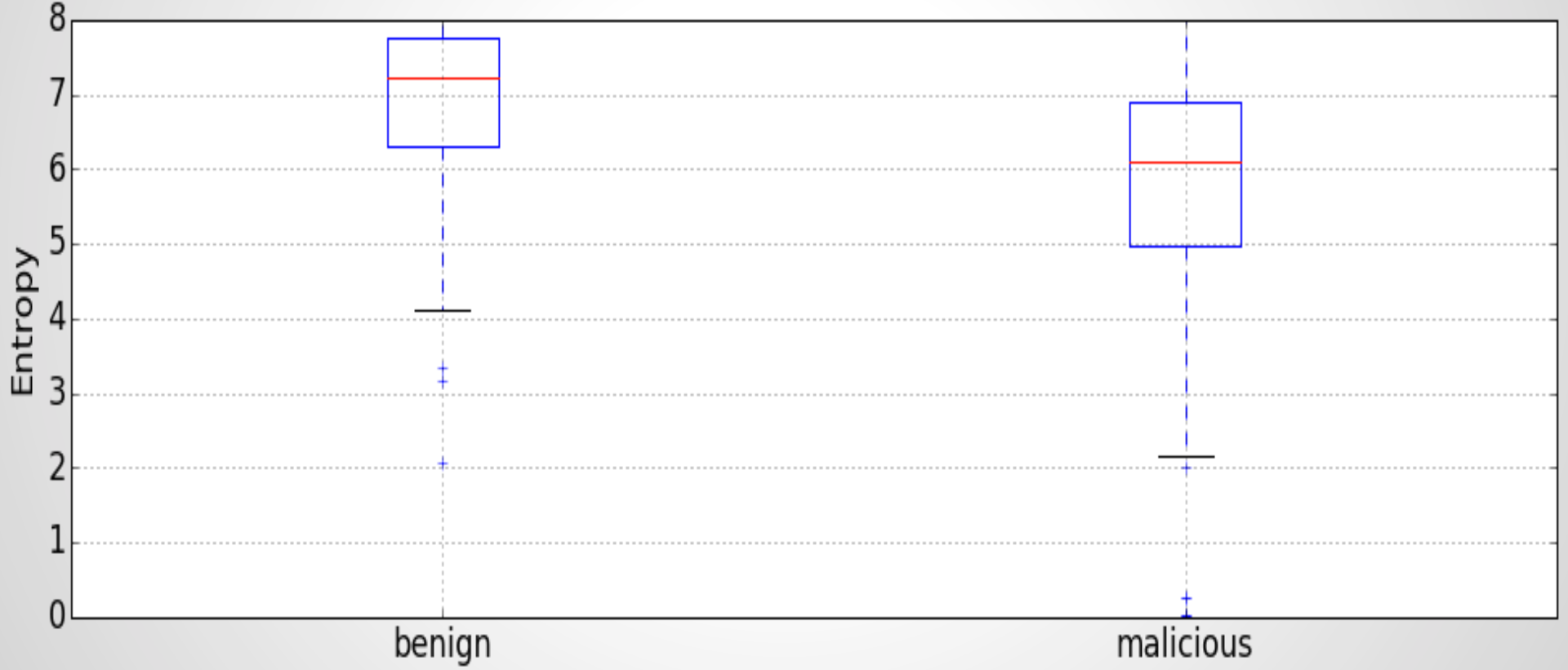
# Versions Seen



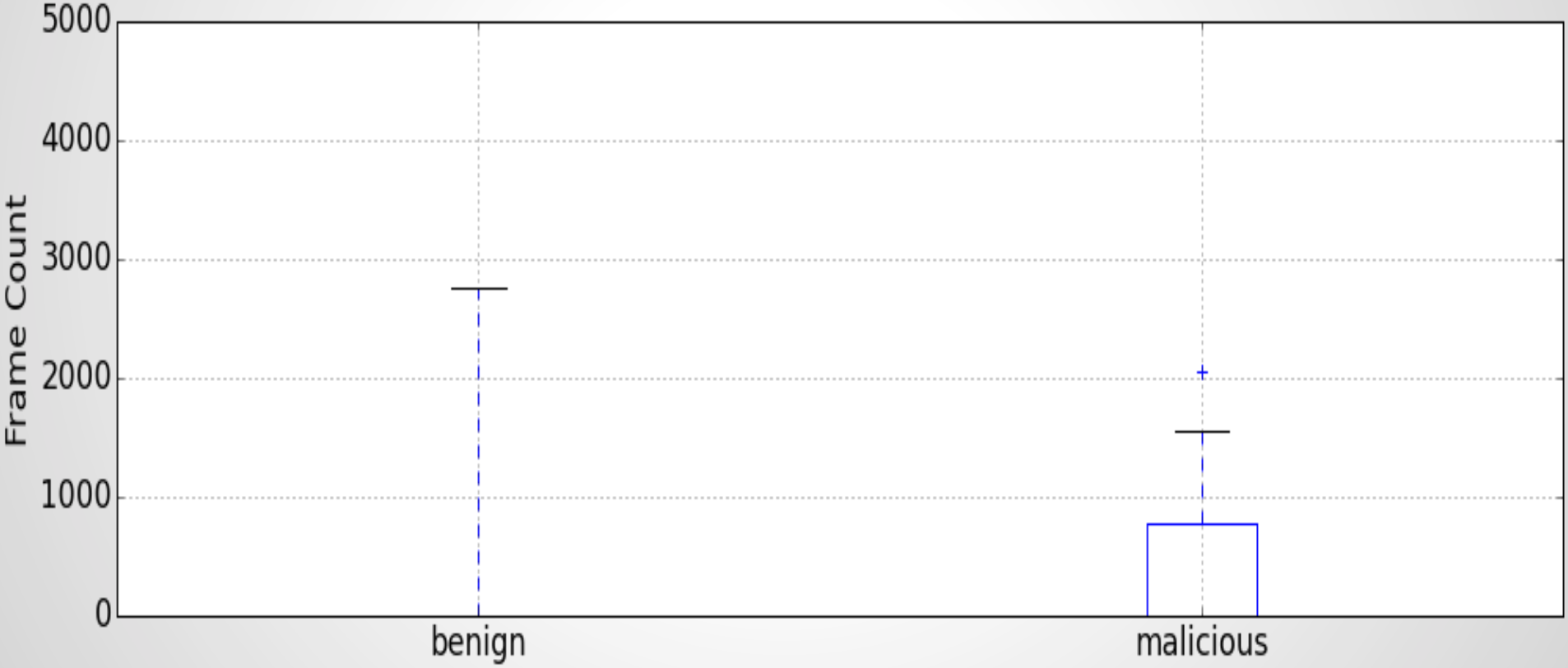
# File Size Comparison



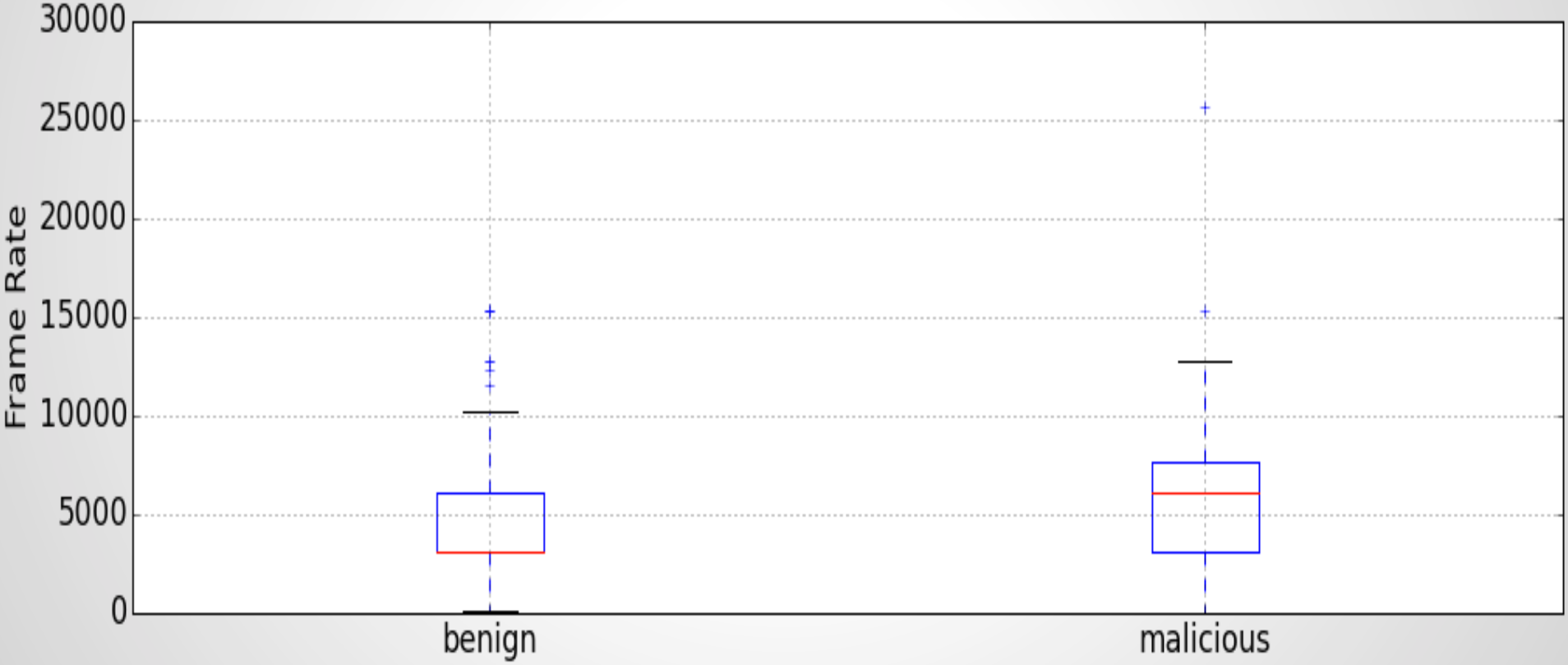
# Entropy Comparison



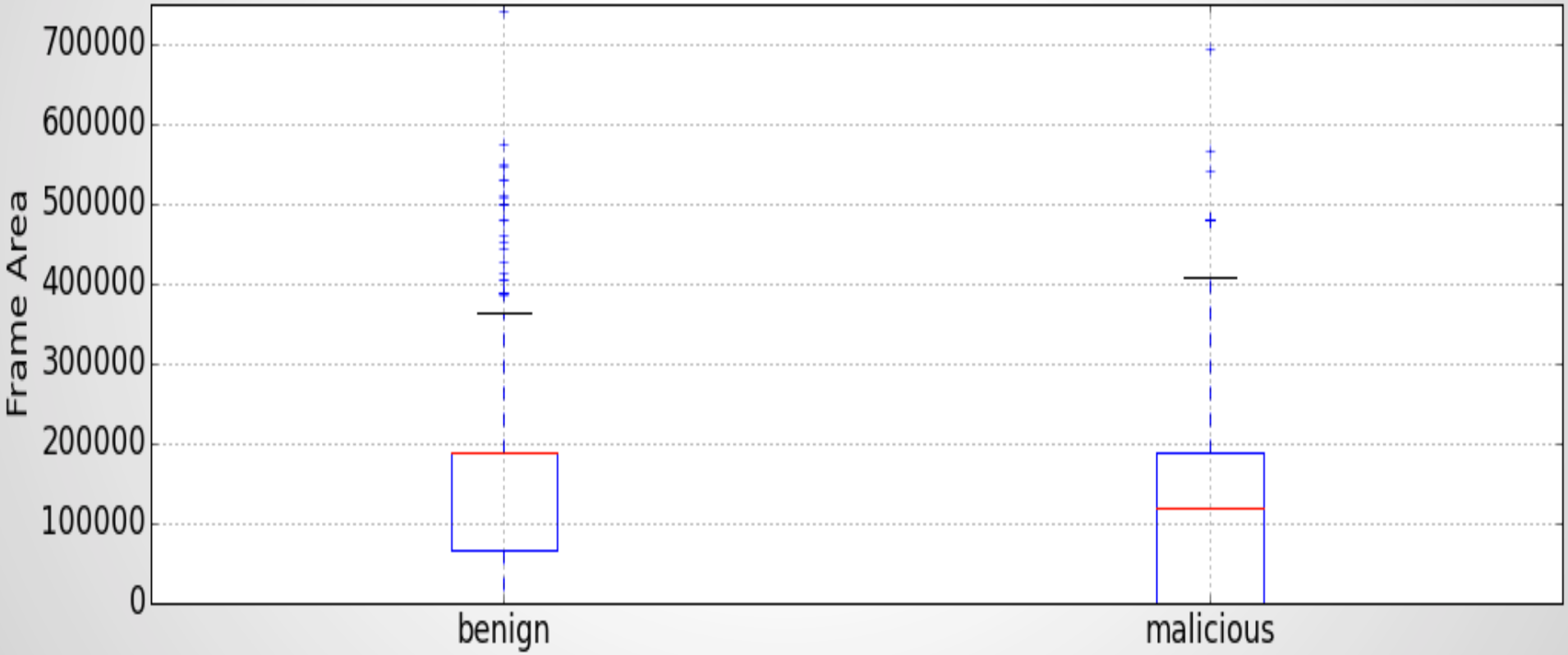
# Frame Count Comparison



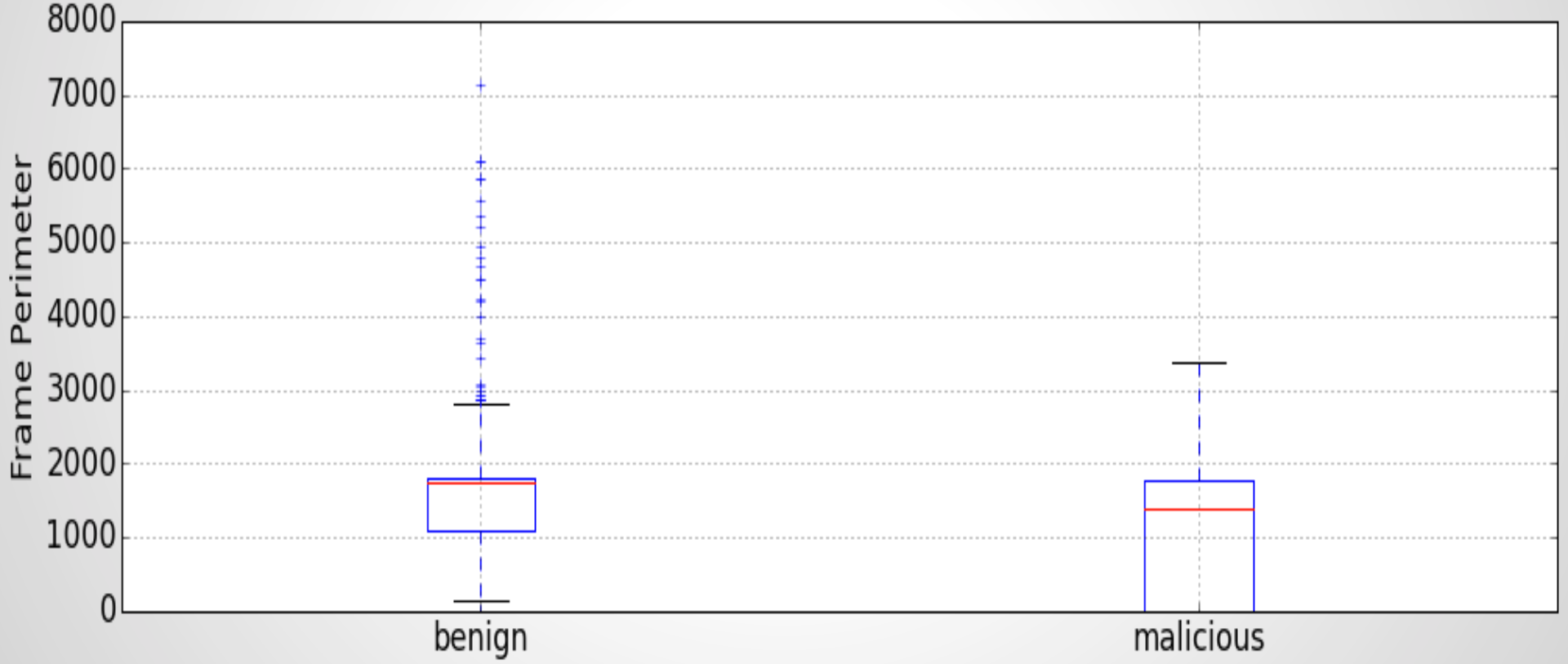
# Frame Rate Comparison



# Frame Area Comparison



# Frame Perimeter Comparison



# Cross Validation Results

- 10 fold
- Accuracy: 96.1% +/- 3.2

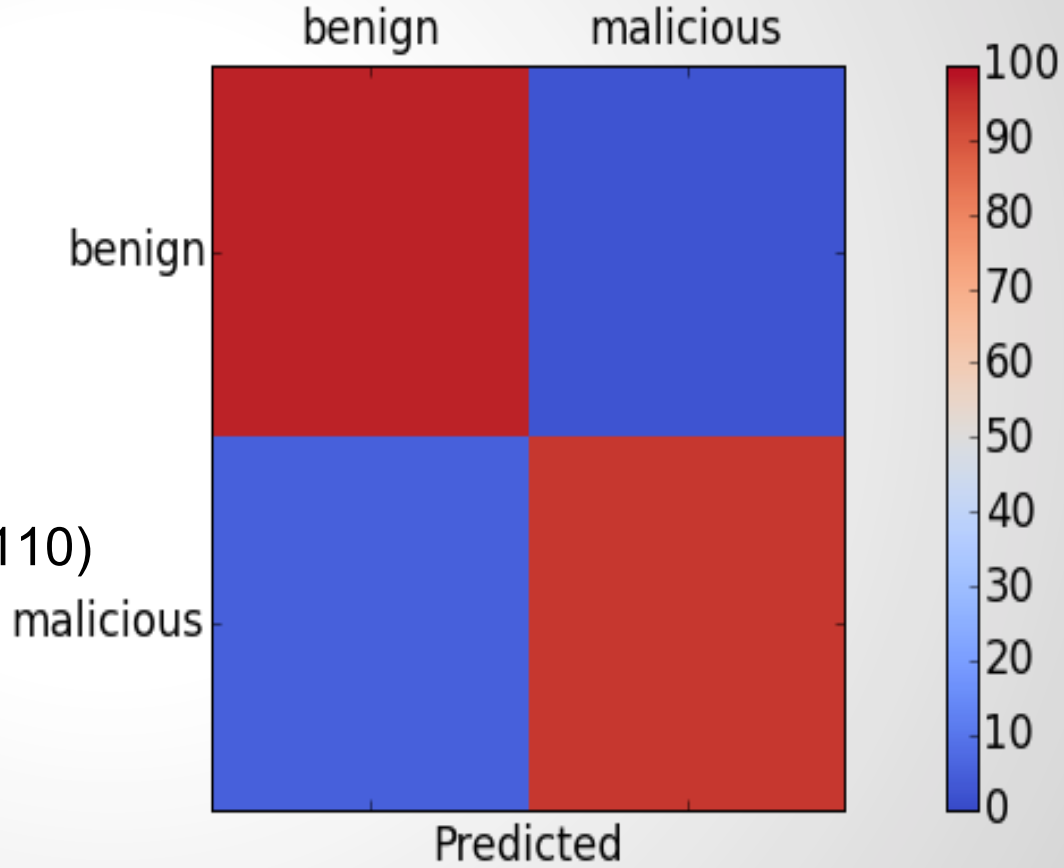
# Confusion Matrix

benign/benign: 97.37% (111/114)

benign/malicious: 2.63% (3/114)

malicious/benign: 5.45% (6/110)

malicious/malicious: 94.55% (104/110)



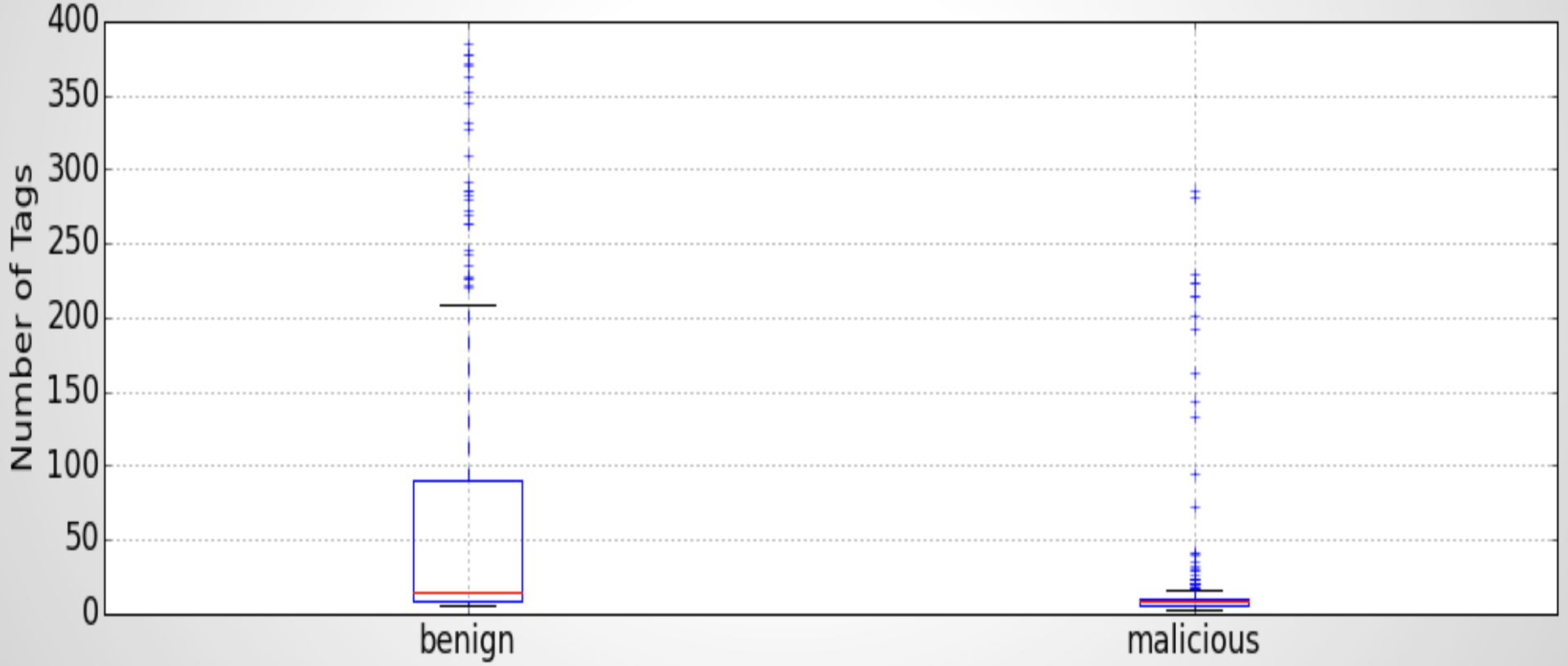
# Feature Importance

| | |
|---|---|
| Entropy | 0.24429 |
| Frame Area | 0.17804 |
| Frame Perimeter | 0.15848 |
| Size | 0.12977 |
| Version | 0.12801 |
| Frame Rate | 0.11892 |
| Frame Count | 0.04248 |

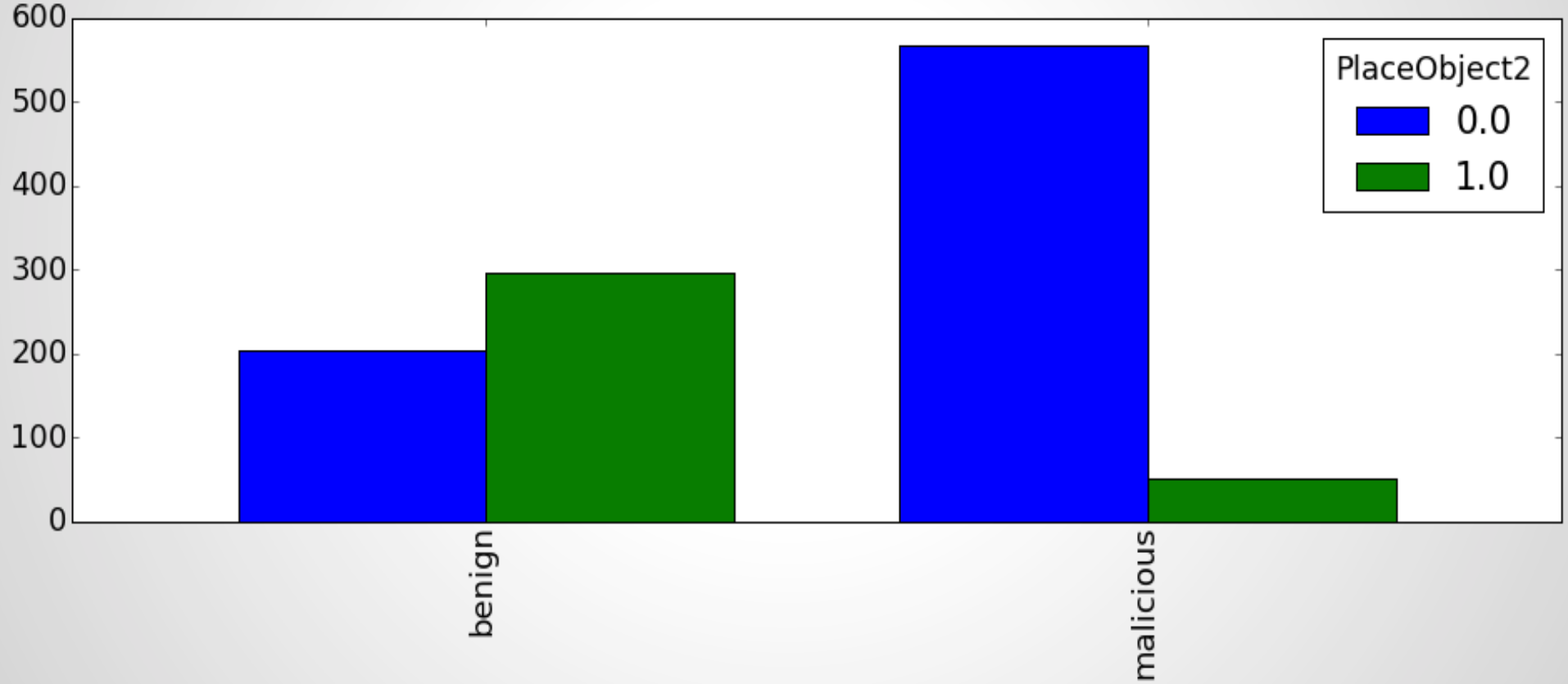
# Tag Information

- A tag can exist multiple times
- Only take into consideration the existence of a tag in the file, not the number of them
- An End tag should be the last tag

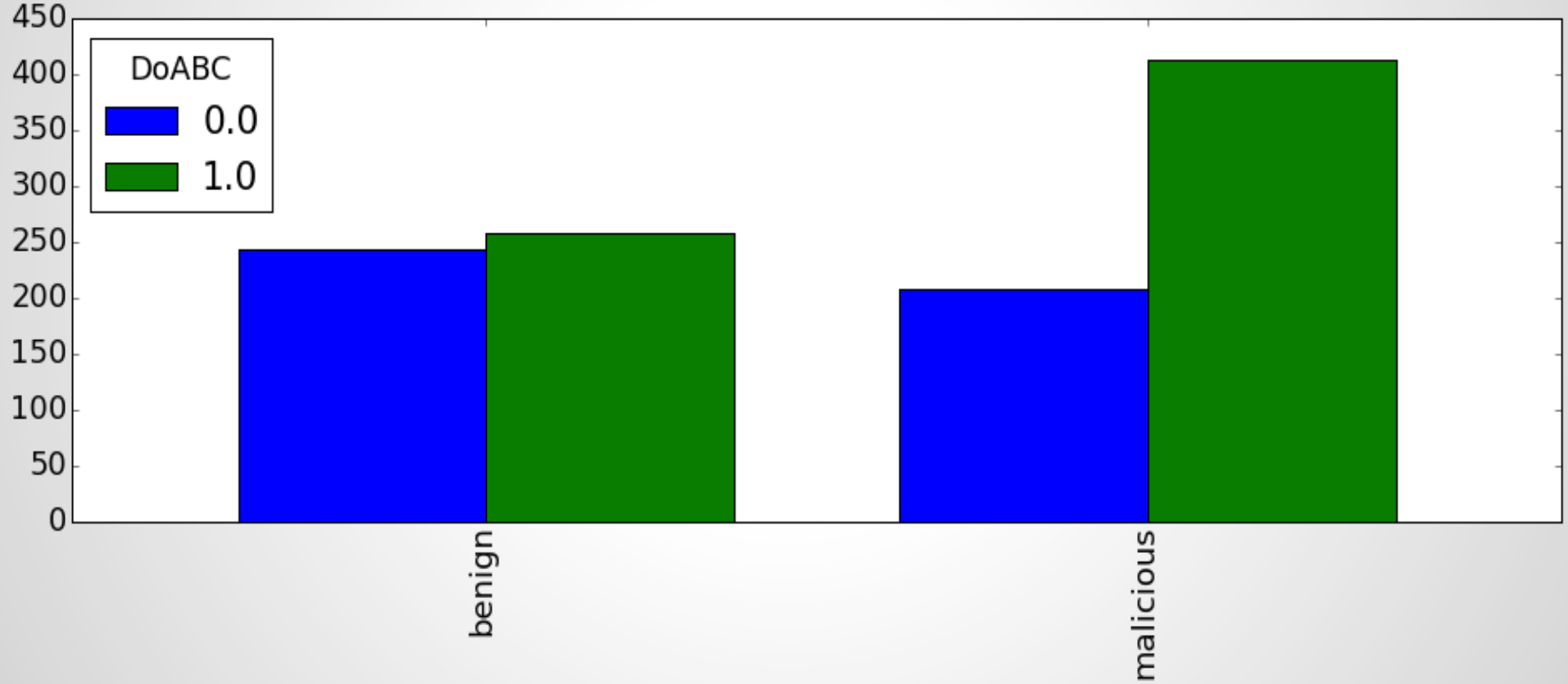
# Number Of Tags Comparison



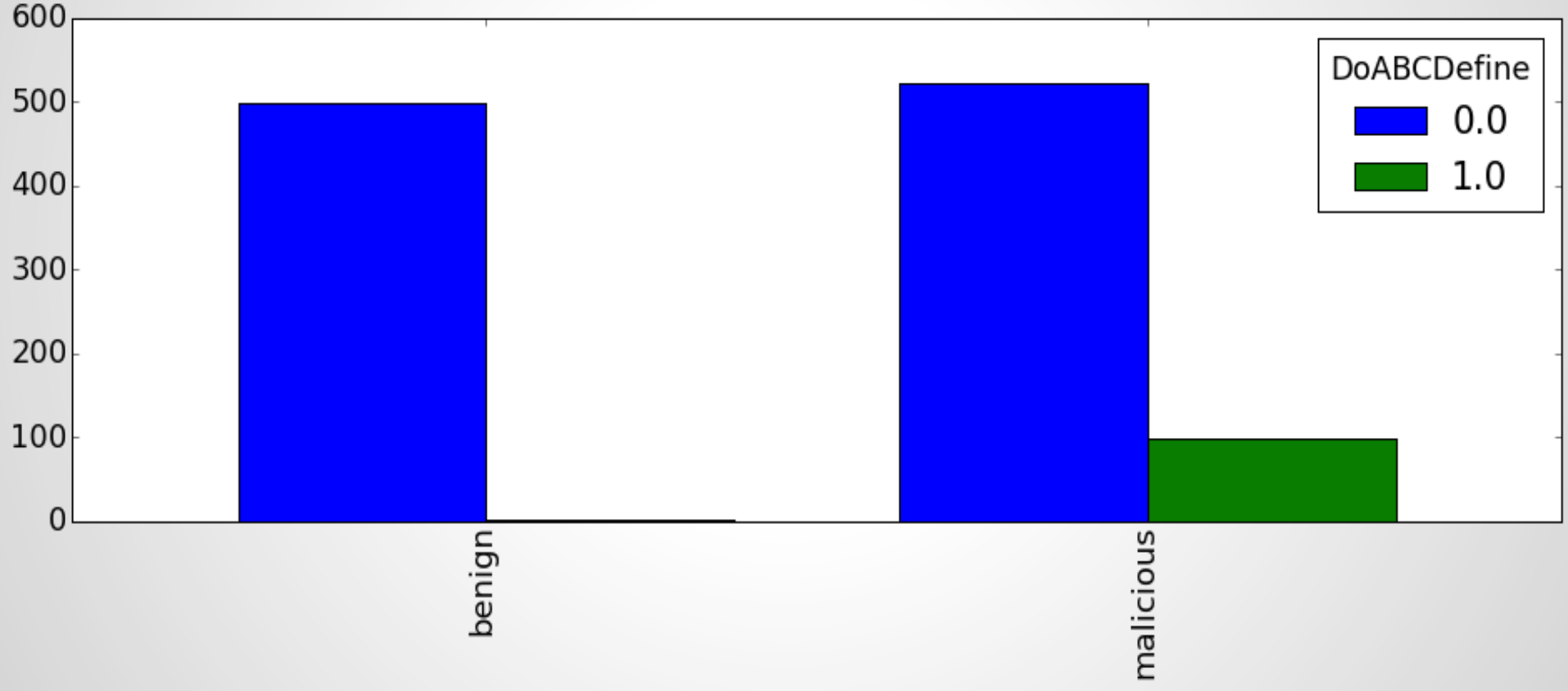
# PlaceObject2 Comparison



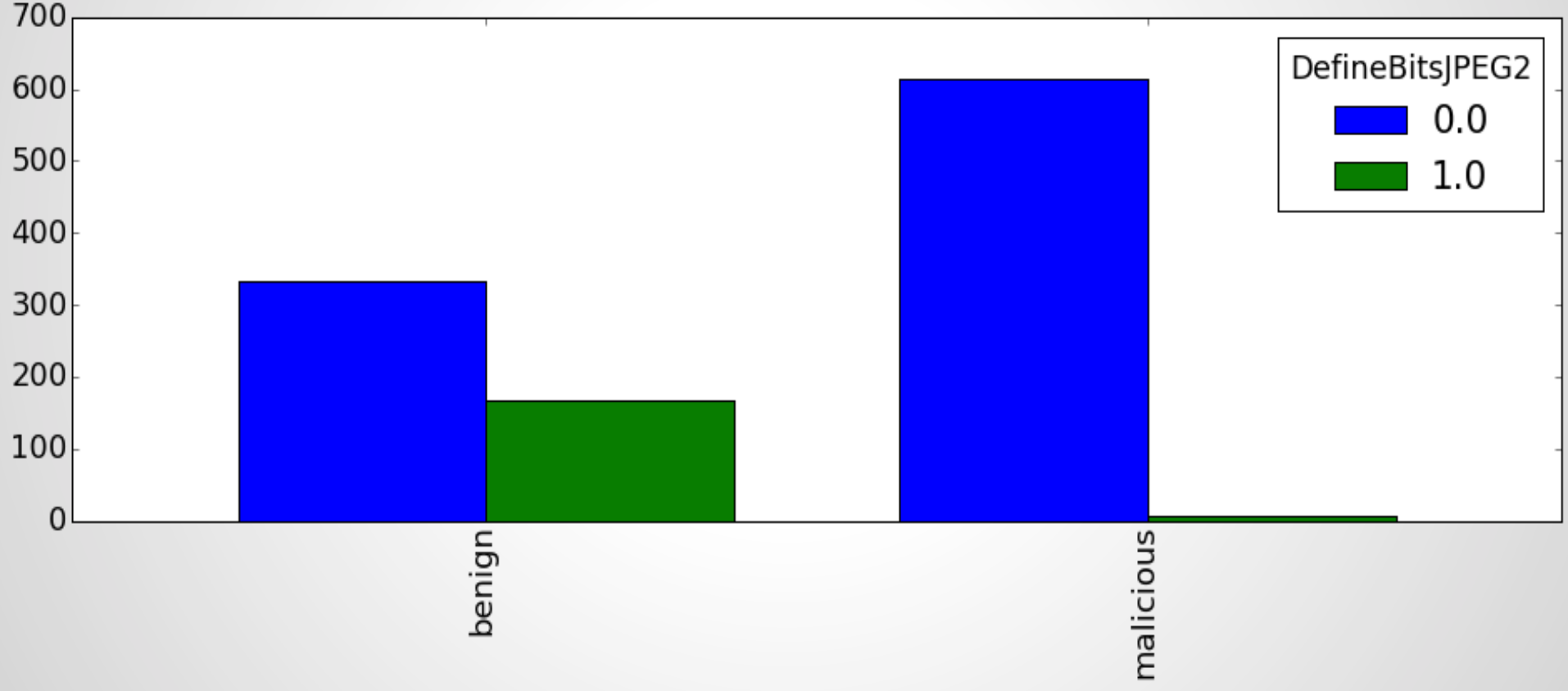
# DoABC Comparison



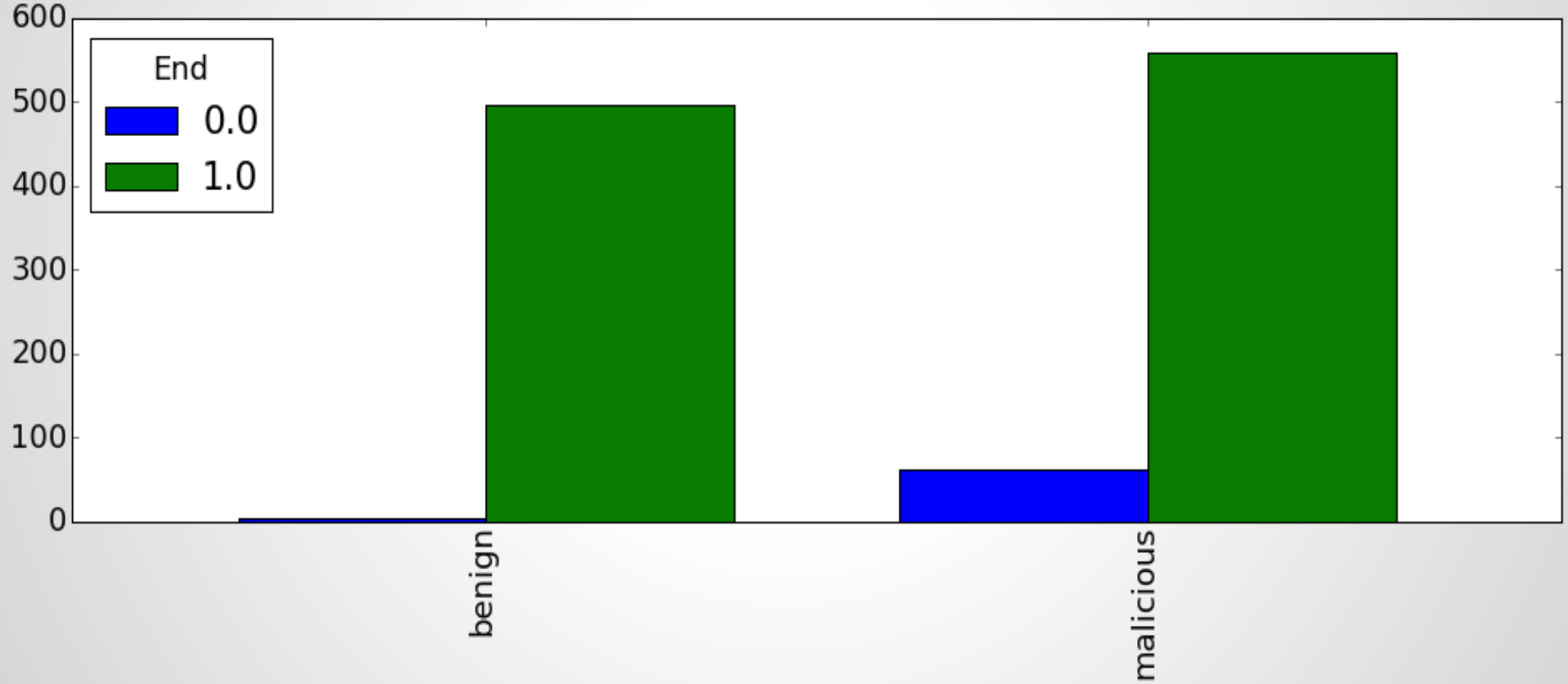
# DoABCDefine Comparison



# DefineBitsJPEG2 Comparison



# End Comparison



# Cross Validation Results

- 10 fold
- Accuracy: 96.8% +/- 2.7

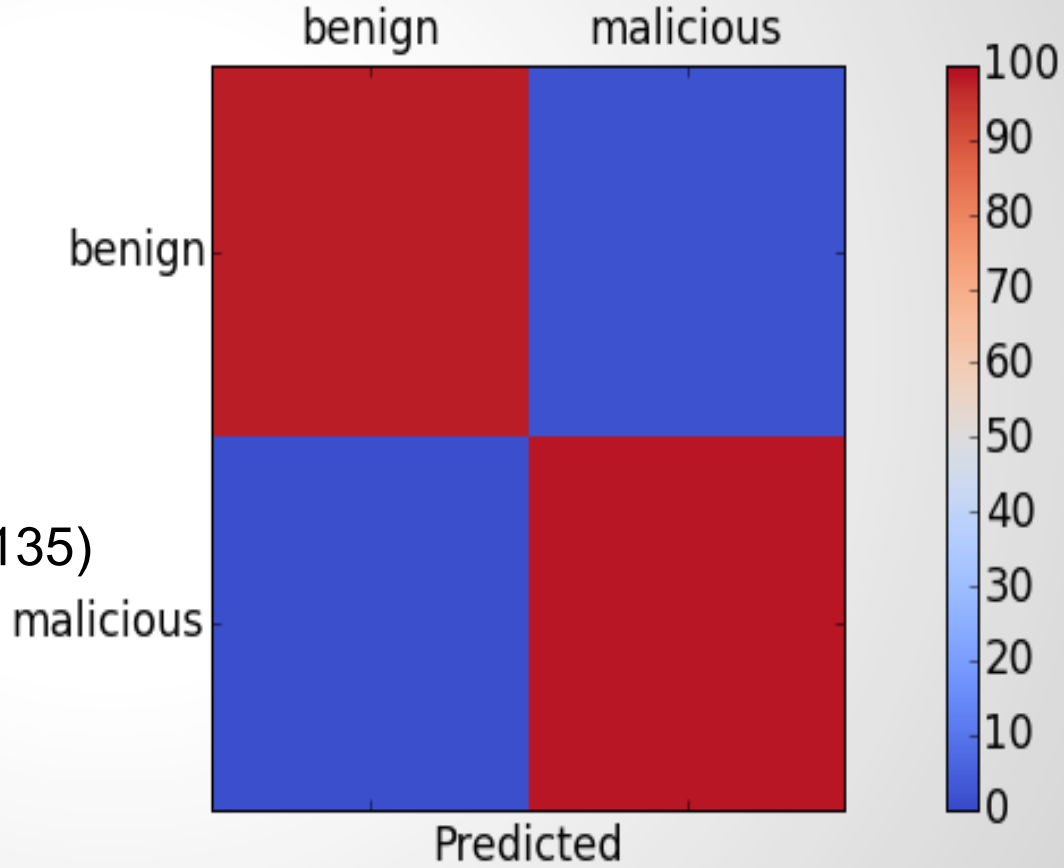
# Confusion Matrix

benign/benign: 97.75% (87/89)

benign/malicious: 2.25% (2/89)

malicious/benign: 1.48% (2/135)

malicious/malicious: 98.52% (133/135)



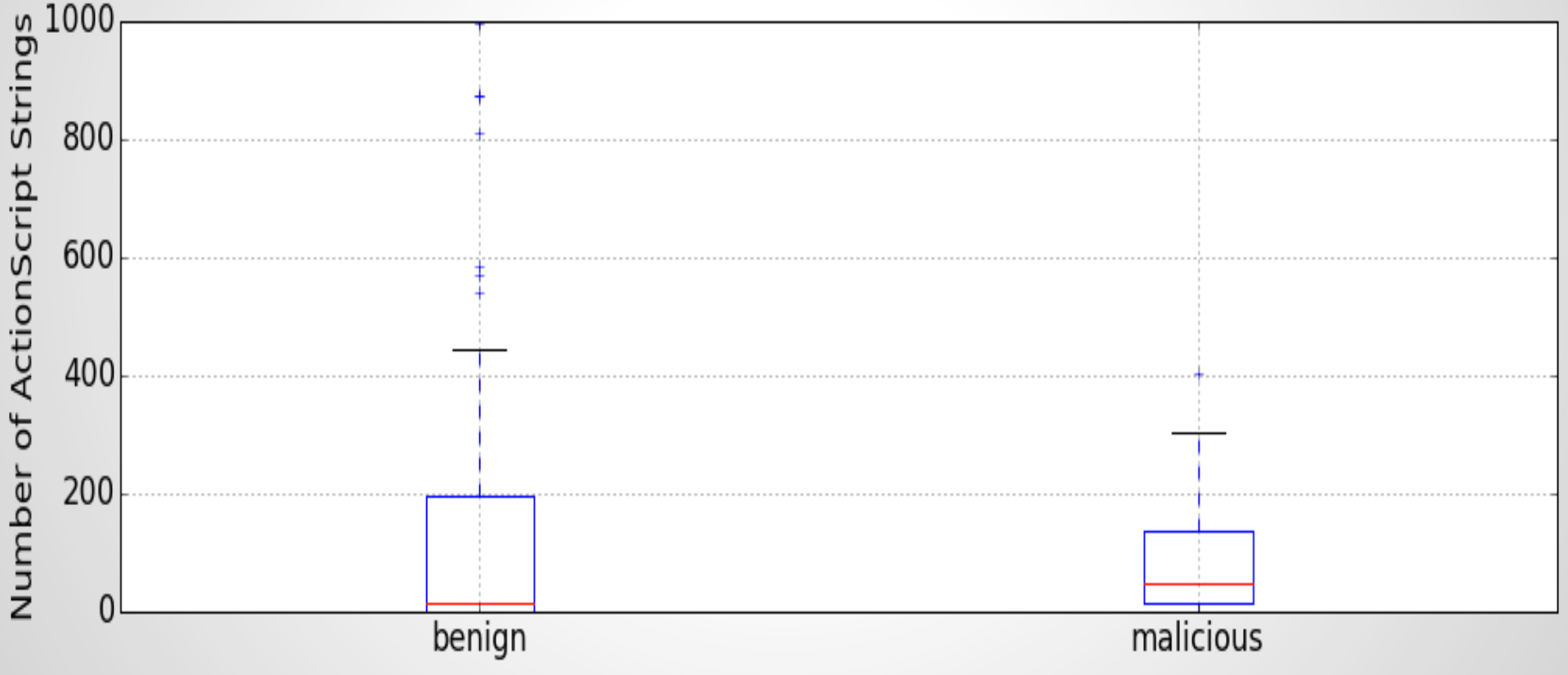
# Feature Importance

| | |
|---|---|
| Entropy | 0.10226 |
| Number Of Tags | 0.09275 |
| Frame Rate | 0.07430 |
| Version | 0.07110 |
| Frame Area | 0.06588 |
| Frame Perimeter | 0.06246 |
| PlaceObject2 | 0.05839 |
| Size | 0.05102 |
| DefineShape3 | 0.03744 |

# ActionScript

- Allows Flash to be interactive
- Multiple Versions
- Structured similarly to Java Class files
- The bane of my existence

# ActionScript String Count Comparison



# Mean to Median String Length Ratio

- Divide the mean line length by the median line length
- Higher ratio is more suspicious

"TOP_LEFT"

"Loader"

"ldr"

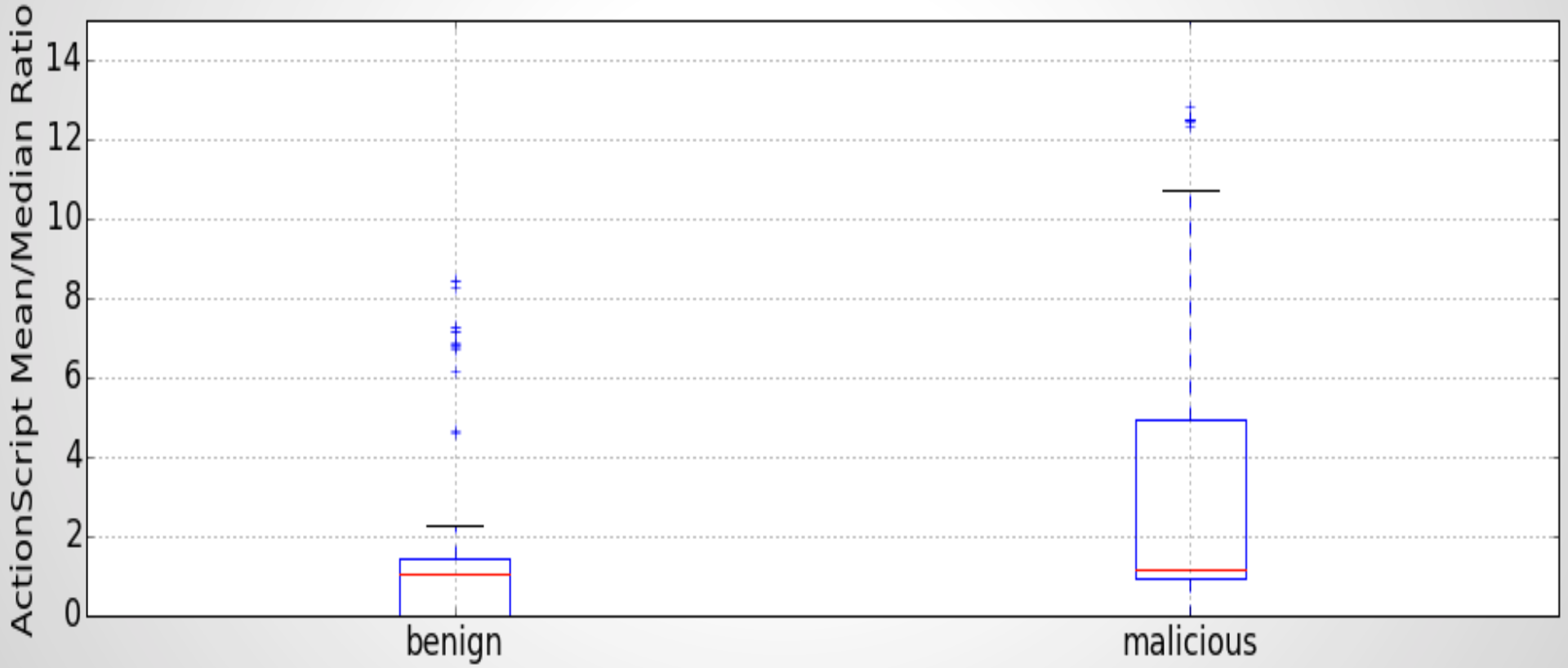
"flash.events"

"Event"

"completeHandler"

"4657530825060000300A00A0000C0303441108000000430 2FFFFFFC13F20BF01000400

# ActionScript Mean/Median Ratio Comparison



# Cross Validation Results

- 10 fold
- Accuracy: 97.1% +/- 2.9

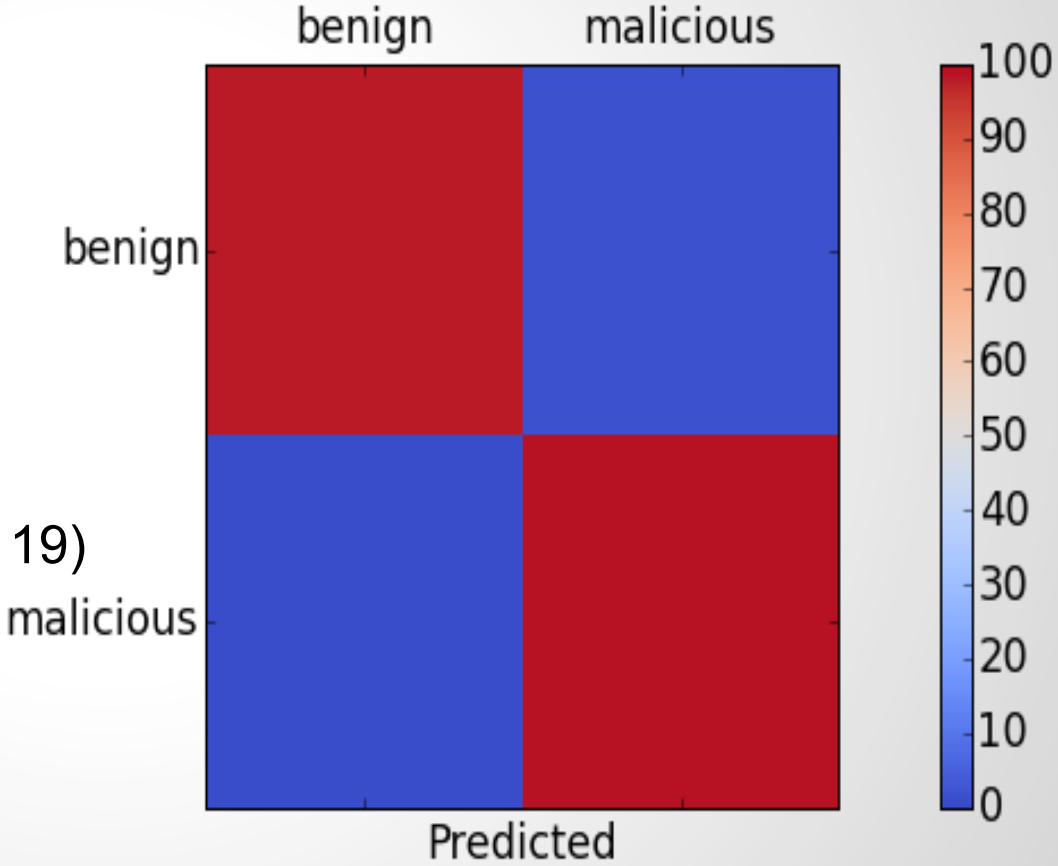
# Confusion Matrix

benign/benign: 98.10% (103/105)

benign/malicious: 1.90% (2/105)

malicious/benign: 0.84% (1/119)

malicious/malicious: 99.16% (118/119)



# Feature Importance

| | |
|---|---|
| Entropy | 0.08949 |
| PlaceObject2 | 0.07960 |
| ActionScript String Count | 0.06956 |
| Size | 0.05697 |
| Number Of Tags | 0.05196 |
| Frame Rate | 0.05035 |
| Frame Perimeter | 0.04962 |
| Version | 0.04840 |
| Frame Area | 0.04566 |

# Classifier Results

- How does the classifier work on the 290K files?

- 279K files marked as benign (score < 0.5)
- 8K files in gray area (0.5 <= score < 0.8)
- 2K files marked as malicious (0.8 <= score)

# A New Classifier

- 620 labeled malicious files
- Label 5000 randomly chosen files as benign
- Cross Validation Score: 99.0% +/- 0.4

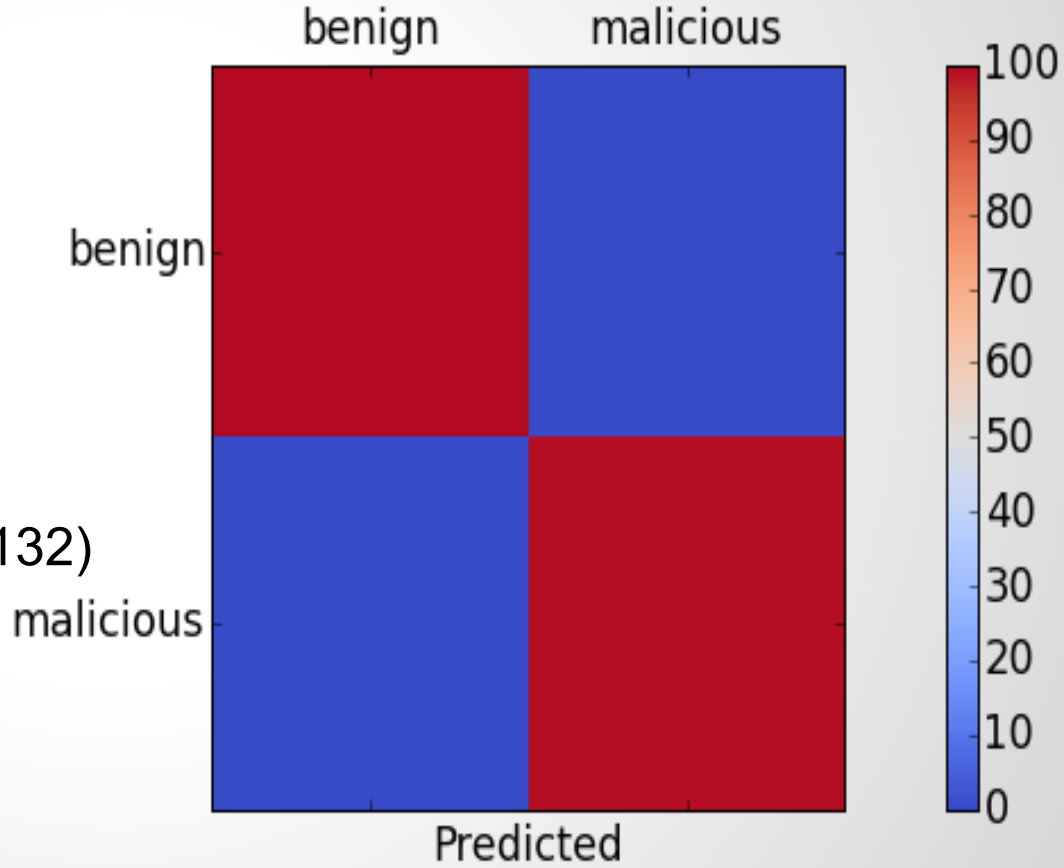
# Confusion Matrix

benign/benign: 99.90% (992/993)

benign/malicious: 0.10% (1/993)

malicious/benign: 0.76% (1/132)

malicious/malicious: 99.14% (131/132)



# Feature Importance

| | |
|---|---|
| Entropy | 0.08452 |
| Frame Perimeter | 0.08293 |
| Frame Area | 0.08219 |
| ActionScript String Count | 0.06897 |
| Frame Count | 0.06435 |
| ActionScript String Size Ratio | 0.06046 |
| Size | 0.05533 |
| Frame Rate | 0.04375 |
| End Tag | 0.04332 |

# Classifier Results

- How does the classifier work on the 284K files?

- 283K files marked as benign (score < 0.5)
- 679 files in gray area (0.5 <= score < 0.8)
- 407 files marked as malicious (0.8 <= score)

# Targeted Classifiers

- What happens when we limit the classifier to only train and test on files with ActionScript?

- Cross Validation Score: 99.1% +/- 0.7

# Classifier Results

- How does the classifier work on the 153K files with ActionScript?

- 152K files marked as benign (score < 0.5)
- 546 files in gray area (0.5 <= score < 0.8)
- 284 files marked as malicious (0.8 <= score)

# SWF Summary

- Can detect malicious files using data science
- Classifier isn’t overly dependent on any one feature
- Separating out a specific classifier for SWFs with ActionScript did not increase accuracy
- Classifier seems to perform well on the random set of files

# Next Steps

- More malicious files
- Better labeled data
  - Not just good/bad
  - Exploit Kit
  - CVE
- New features
  - More ActionScript features
  - N-Gram analysis on actual ActionScript code
  - G-Test Analysis on methods called

# Highlights

- Using data science to detect new malicious files is useful
- Can detect malicious files without signatures
- Java classifier is heavily dependent on class name
- SWF classifier is not highly dependent on any one field
- Training with more data can make the classifier better

# Pitfalls

- Try to avoid having one really dominant feature
- Sometimes features you think are great don’t help that much
- Test on a lot of data

# Where Do We Go From Here

- Need more data
- Verify data is labeled appropriately
- More sophisticated labels
- Extract more features

# Play At Home

- Notebooks and data are available at https://github.com/ClickSecurity/data_hacking
- The data I used from the 300K files can be obtained by contacting me



- @trogdorsey
- trogdorsey at gmail . com


- https://github.com/ClickSecurity/data_hacking
- http://docs.oracle.com/javase/specs/jvms/se7/html/jvms-4.html
- http://wwwimages.adobe.com/www.adobe.com/content/dam/Adobe/en/devnet/swf/pdf/swf-file-format-spec.pdf
- http://www.adobe.com/content/dam/Adobe/en/devnet/actionscript/articles/avm2overview.pdf
- http://hooked-on-mnemonics.blogspot.com/2013/02/detecting-pdf-js-obfuscation-using.html