Evaluation of Data-mining techniques for malware detection using system-call Sequence

Shayan Eskandari
Amirreza Soudi

Anomaly Detection

- Modeling the normal behavior of the system
- Flag the patterns that do not conform to the normal model as Anomaly
- This could be done in different layers, our main focus is on system call level

Anomaly Detection & Data-Mining

- Huge Amount of data to be analyzed
- Different learning and evaluating methods
- Machine learning approach to a security problem
- Analysis of system call sequences
-

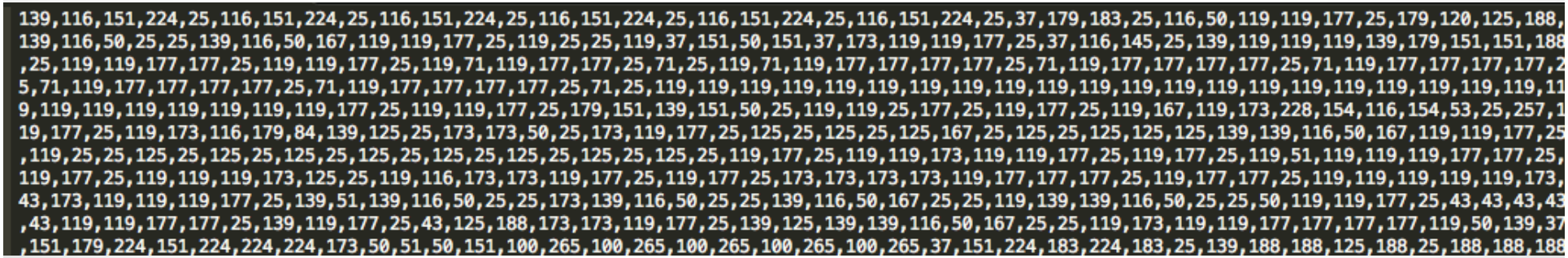
Anubis* Dataset

```
116(desiredaccess: 1048737,filename: "C:\INSTANTLOVENOTE.EXE",shareaccess: 5,openoptions: 96, nstatus:0)
50(desiredaccess: 983071,sectionname: "C:\INSTANTLOVENOTE.EXE",filename: "C:\INSTANTLOVENOTE.EXE", nstatus:0)
119(filename: "HKLM/SYSTEM/CURRENTCONTROLSET/CONTROL/SESSION MANAGER/APPCERTDLLS",desiredaccess: 1, nstatus:0)
119(filename: "HKLM/SYSTEM/CURRENTCONTROLSET/CONTROL/SESSION MANAGER/APPCOMPATIBILITY",desiredaccess: 1, nstatus:0)
177(keyname: "HKLM/SYSTEM/CURRENTCONTROLSET/CONTROL/SESSION MANAGER/APPCOMPATIBILITY",valuename: "DISABLEAPPCOMPAT",informationclass: 2, nstatus:0)
25(filename: "HKLM/SYSTEM/CURRENTCONTROLSET/CONTROL/SESSION MANAGER/APPCOMPATIBILITY", nstatus:0)
179(filename: "C:\INSTANTLOVENOTE.EXE",fsinformationclass: 4, nstatus:0)
120(desiredaccess: 1179649,objectname: "BASENAMEDOBJECTS\SHIMCACHEMUTEX", nstatus:0)
125(desiredaccess: 2,objectname: "BASENAMEDOBJECTS\SHIMSHAREDMEMORY", nstatus:0)
188(mutantname: "BASENAMEDOBJECTS\SHIMCACHEMUTEX", nstatus:0)
139(filename: "C:\WINDOWS\SYSTEM32\APPHELP.DLL", nstatus:0)
116(desiredaccess: 1048608,filename: "C:\WINDOWS\SYSTEM32\APPHELP.DLL",shareaccess: 5,openoptions: 96, nstatus:0)
50(desiredaccess: 14,sectionname: "C:\WINDOWS\SYSTEM32\APPHELP.DLL",filename: "C:\WINDOWS\SYSTEM32\APPHELP.DLL", nstatus:0)
25(filename: "C:\WINDOWS\SYSTEM32\APPHELP.DLL", nstatus:0)
25(filename: "C:\WINDOWS\SYSTEM32\APPHELP.DLL", nstatus:0)
139(filename: "C:\WINDOWS\SYSTEM32\APPHELP.DLL", nstatus:0)
116(desiredaccess: 1048608,filename: "C:\WINDOWS\SYSTEM32\APPHELP.DLL",shareaccess: 5,openoptions: 96, nstatus:0)
50(desiredaccess: 15,sectionname: "C:\WINDOWS\SYSTEM32\APPHELP.DLL",filename: "C:\WINDOWS\SYSTEM32\APPHELP.DLL", nstatus:0)
167(sectionname: "C:\WINDOWS\SYSTEM32\APPHELP.DLL",sectioninformationclass: 1, nstatus:0)
119(filename: "HKLM/SYSTEM/CURRENTCONTROLSET/CONTROL/SAFEBOOT/OPTION",desiredaccess: 3, nstatus:0)
119(filename: "HKLM/SOFTWARE/POLICIES/MICROSOFT/WINDOWS/SAFER/CODEIDENTIFIERS",desiredaccess: 1, nstatus:0)
177(keyname: "HKLM/SOFTWARE/POLICIES/MICROSOFT/WINDOWS/SAFER/CODEIDENTIFIERS",valuename: "TRANSPARENTENABLED",informationclass: 2, nstatus:0)
25(filename: "HKLM/SOFTWARE/POLICIES/MICROSOFT/WINDOWS/SAFER/CODEIDENTIFIERS", nstatus:0)
119(filename: "HKU/USER/SOFTWARE/POLICIES/MICROSOFT/WINDOWS/SAFER/CODEIDENTIFIERS",desiredaccess: 1, nstatus:0)
25(filename: "C:\WINDOWS\SYSTEM32\APPHELP.DLL", nstatus:0)
25(filename: "C:\WINDOWS\SYSTEM32\APPHELP.DLL", nstatus:0)
119(filename: "HKLM/SOFTWARE/MICROSOFT/WINDOWS NT/CURRENTVERSION/IMAGE FILE EXECUTION OPTIONS/APPHELP.DLL",desiredaccess: 2147483648, nstatus:0)
37(desiredaccess: 2148532352,filename: "\SYSTEMROOT\APPPATCH\SYSMAIN.SDB",shareaccess: 1,createoptions: 96, nstatus:0)
151(filename: "\SYSTEMROOT\APPPATCH\SYSMAIN.SDB",fileinformationclass: 5, nstatus:0)
50(desiredaccess: 4,sectionname: "\SYSTEMROOT\APPPATCH\SYSMAIN.SDB",filename: "\SYSTEMROOT\APPPATCH\SYSMAIN.SDB", nstatus:0)
151(filename: "\SYSTEMROOT\APPPATCH\SYSMAIN.SDB",fileinformationclass: 5, nstatus:0)
37(desiredaccess: 2148532352,filename: "\SYSTEMROOT\APPPATCH\SYSTEST.SDB",shareaccess: 1,createoptions: 96, nstatus:0)
173(systeminformationclass: 1, nstatus:0)
119(filename: "HKLM/SYSTEM/WPA/TABLETPC",desiredaccess: 257, nstatus:0)
119(filename: "HKLM/SYSTEM/WPA/MEDIACENTER",desiredaccess: 257, nstatus:0)
177(keyname: "HKLM/SYSTEM/WPA/MEDIACENTER",valuename: "INSTALLED",informationclass: 2, nstatus:0)
25(filename: "HKLM/SYSTEM/WPA/MEDIACENTER", nstatus:0)
37(desiredaccess: 1179926,filename: "\DEVICE\NAMEDPIPE\SHIMVIEWER",shareaccess: 0,createoptions: 0, nstatus:0)
116(desiredaccess: 1048577,filename: "C:\",shareaccess: 3,openoptions: 16417, nstatus:0)
145(filename: "C:\",fileinformationclass: 3, nstatus:0)
25(filename: "C:\", nstatus:0)
139(filename: "C:\INSTANTLOVENOTE.EXE", nstatus:0)
119(filename: "HKLM/SOFTWARE/MICROSOFT/WINDOWS NT/CURRENTVERSION/APPCOMPATFLAGS/LAYERS",desiredaccess: 2147483904, nstatus:0)
119(filename: "HKU/USER/SOFTWARE/MICROSOFT/WINDOWS NT/CURRENTVERSION/APPCOMPATFLAGS/LAYERS",desiredaccess: 2147483904, nstatus:0)
119(filename: "HKLM/SOFTWARE/MICROSOFT/WINDOWS NT/CURRENTVERSION/APPCOMPATFLAGS/CUSTOM/INSTANTLOVENOTE.EXE",desiredaccess: 2147483904, nstatus:0)
139(filename: "C:\INSTANTLOVENOTE.EXE", nstatus:0)
179(filename: "C:\INSTANTLOVENOTE.EXE",fsinformationclass: 4, nstatus:0)
151(filename: "C:\INSTANTLOVENOTE.EXE",fileinformationclass: 4, nstatus:0)
151(filename: "C:\INSTANTLOVENOTE.EXE",fileinformationclass: 5, nstatus:0)
```

* anubis.iseclab.org

Pre-Processing

- Data Cleaning
- Extracting the system calls sequence
- Define a window size and separating the sequences

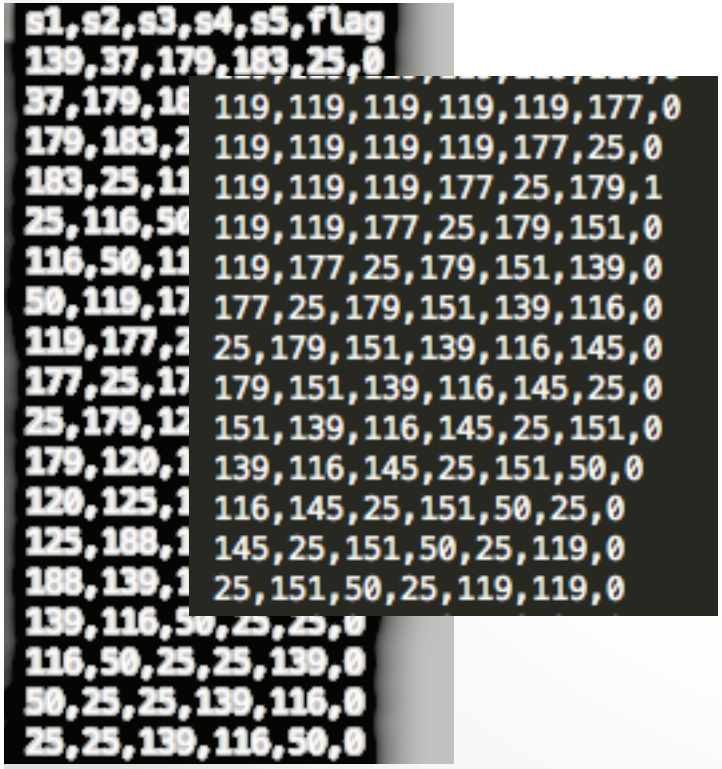


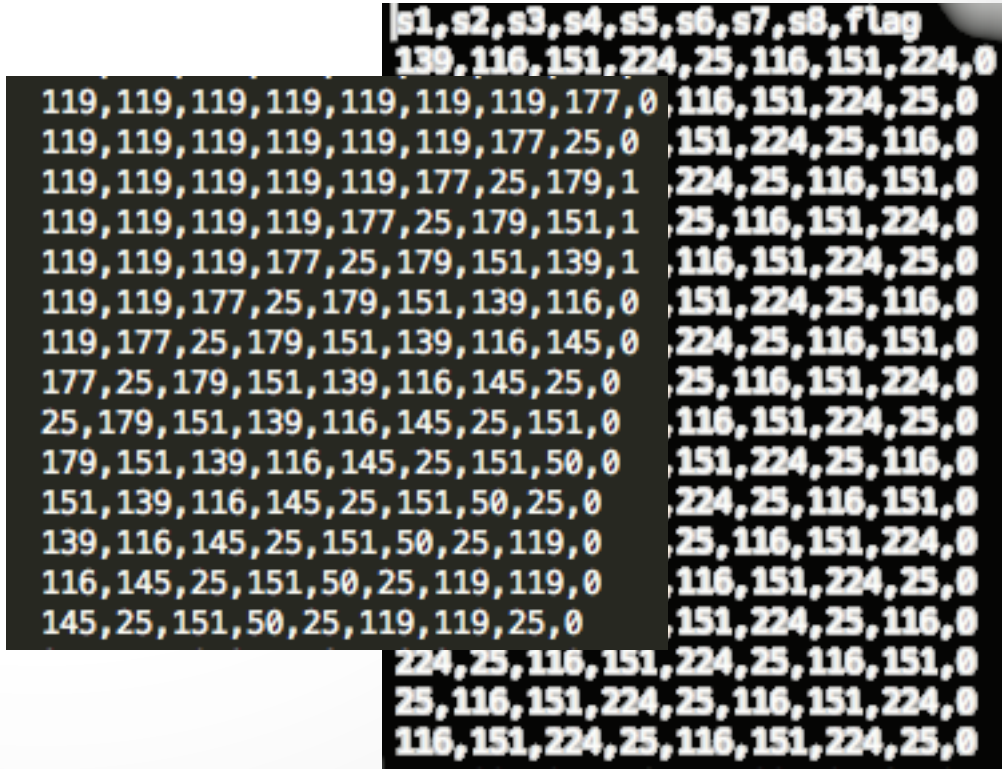
Data Cleaning

```
#!/usr/bin/env python

import fileinput
import os


file = "1grams-seq_Process-"    #name of the raw trace files without the last number
path = "/Users/sbeta/Desktop/PRoject/ISSTA12-mal-data/anubis-good" #path to the raw trace files
W=5     #window size
K=1 #shift size

def parsefile(file):    #parse the files to extract the sequence files
    with open(file) as f:
        for line in f:
            Syscall=line.split("(")
            print Syscall[0]
            if Syscall[0].isdigit():
                seq = open(file + "-seq.txt", "a")
                seq.write(Syscall[0])
                seq.write(",")


def window(seq):    # to extract windows of size W with the shift size of K from the sequence files (-seq)
    with open(seq) as s:
        for line in s:
            sequence=line.split(",")
            end=len(sequence)-W
            for i in xrange(0,end,K):
                for j in xrange(i,W+i):
                    print "K: ", K, "        end: ", end , "        seq: ", sequence[j]
                    norm = open("NormDB.txt", "a")
                    norm.write(sequence[j])
                    norm.write(",")
                norm.write("\n")




os.chdir(path)
for i in xrange(1,36):
    parsefile(file + str(i))        # To extract sequences from the raw files, the i would be the number in the file name
    window(file + str(i) + "-seq.txt")      # to extract windows of size W with the shift size of K from the sequence files (-seq)


```

Window's Size Open Question

- Which window size would be the best size to find the anomaly sequences?
    - For this we tried 5,6,7,8 window sizes and compared the results





Different Models

- Boost
  - boosting algorithms consist of iteratively learning weak classifiers with respect to a distribution and adding them to a final strong classifier.
- Random Forest
  - Each decision tree is constructed by using a random subset of the training data.
- SVM
- Linear Model
  - a family of model-based learning approaches that assume the output can be expressed as a linear algebraic relation with the input attributes
- Neural Network
  - a learning algorithm that is inspired by the structure and functional aspects of biological neural networks

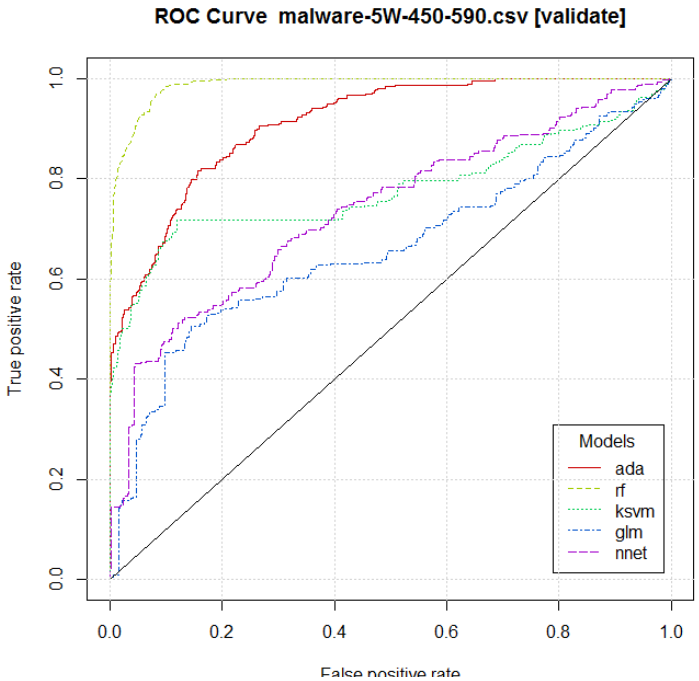
R Code generated by Rattle

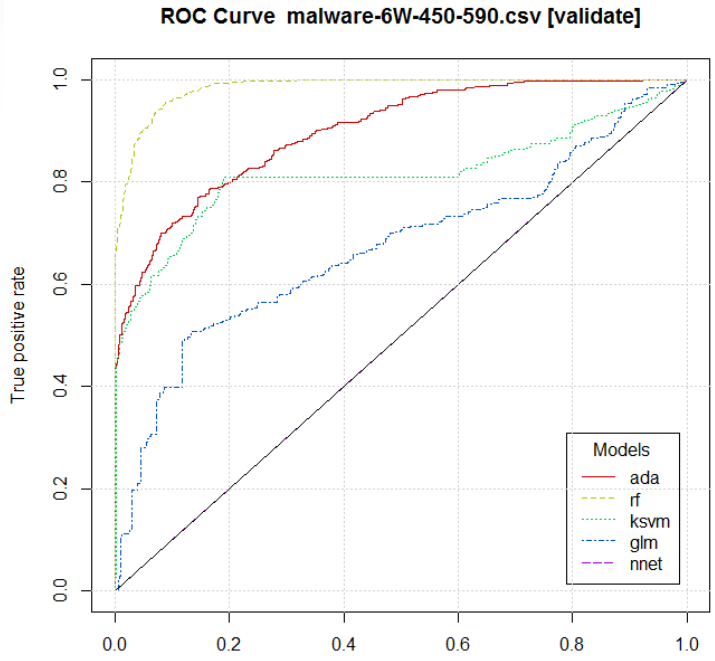
```
# Load the data.

crs$dataset <- read.csv("file:///C:/Users/umroot/Desktop/6180 Project/INSE6180/Real-Data/5W/malware-5W-25-33.csv",
"?"), strip.white=TRUE, encoding="UTF-8")


# Build the training/validate/test datasets.

set.seed(crv$seed)
crs$nobs <- nrow(crs$dataset) # 11999 observations
crs$sample <- crs$train <- sample(nrow(crs$dataset), 0.7*crs$nobs) # 8399 observations
crs$validate <- sample(setdiff(seq_len(nrow(crs$dataset)), crs$train), 0.15*crs$nobs) # 1799 observations
crs$test <- setdiff(setdiff(seq_len(nrow(crs$dataset)), crs$train), crs$validate) # 1801 observations

# The 'kernlab' package provides the 'ksvm' function.

require(kernlab, quietly=TRUE)

# Build a Support Vector Machine model.

set.seed(crv$seed)
crs$ksvm <- ksvm(as.factor(flag) ~ .,
      data=crs$dataset[crs$train,c(crs$input, crs$target)],
      kernel="rbfdot",
      prob.model=TRUE)

# Generate a textual view of the SVM model.

crs$ksvm

require(ROCR, quietly=TRUE)

# Generate an ROC Curve for the rpart model on malware-5W-25-33.csv [validate].

crs$pr <- predict(crs$rpart, newdata=crs$dataset[crs$validate, c(crs$input, crs$target)])[,2]
```

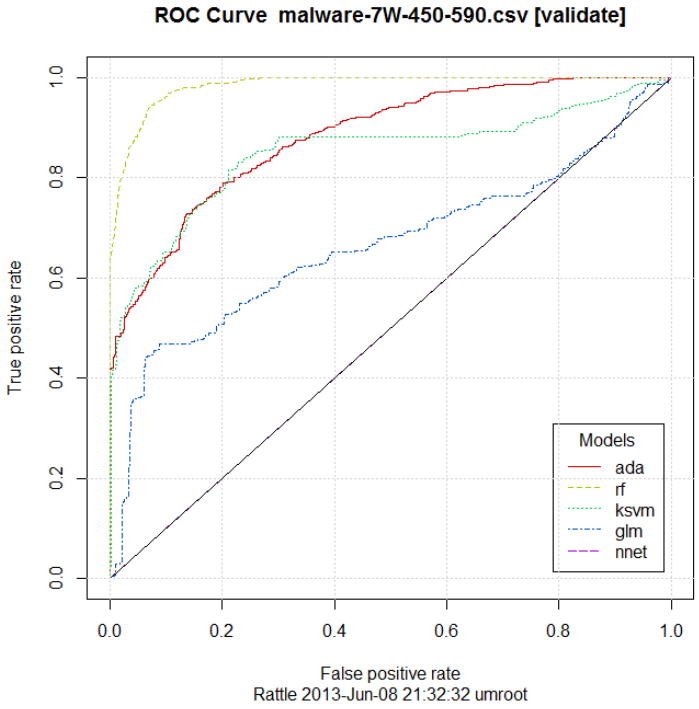
Predicted Flags

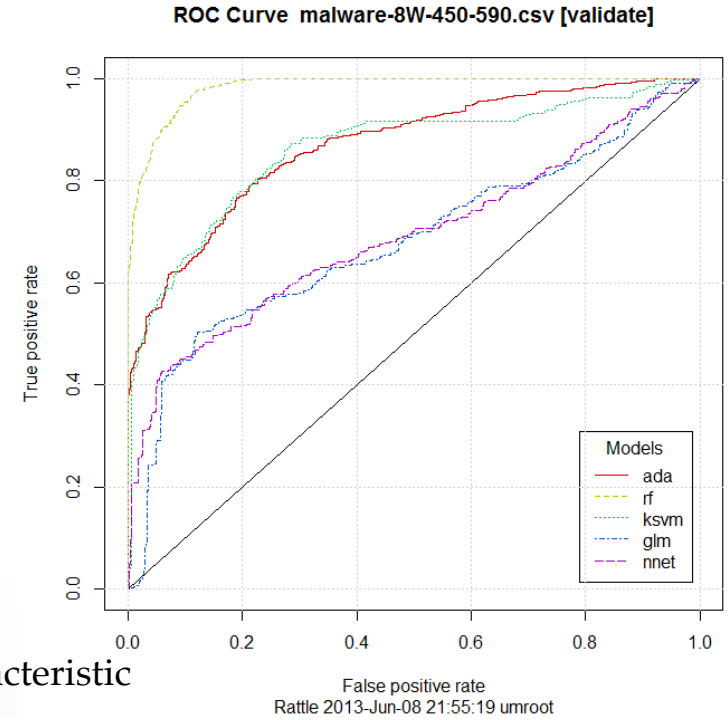
flag	boost	random fore	SVM	Linear	Neural	kmeans
0	0	0	0	0	0	0
0	0	0	0	0	0	1
0	0	0	0	0	0	1
0	0	0	0	0	0	1
0	0	0	0	0	0	0
1	0	0	0	0	0	0
0	0	0	0	0	0	1
0	0	0	0	0	0	1
0	0	0	0	0	0	1
0	0	0	0	0	0	0
1	1	1	0	0	0	0
0	0	0	0	0	0	0
1	1	1	1	0	0	1
0	0	0	0	0	0	0
0	0	0	0	0	0	1
0	0	0	0	0	0	1
0	0	0	0	0	0	1
0	0	0	0	0	0	1
0	0	0	0	0	1	0
1	0	1	0	0	0	0
0	0	0	0	0	0	1
1	0	1	0	0	0	1
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	1
0	0	0	0	0	0	0
0	0	0	0	0	0	1
0	0	0	0	0	0	0
0	0	0	0	0	0	1
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
1	1	1	1	0	1	1
0	0	0	0	0	0	1

ROC*









* Receiver operating characteristic

Accuracy

Window Size = 5

Model	Accuracy (%)
Random Forest	98.79
Boost	91.07
SVM	77.71
Linear Model	65.93
Neural Network	73.62

Window Size = 6

Model	Accuracy (%)
Random Forest	98.43
Boost	89.78
SVM	81.36
Linear Model	67.21
Neural Network	50

Window Size = 7

Model	Accuracy (%)
Random Forest	98.38
Boost	88.06
SVM	84.79
Linear Model	66.41
Neural Network	50

Window Size = 8

Model	Accuracy (%)
Random Forest	98.33
Boost	86.79
SVM	85.92
Linear Model	67.86
Neural Network	68.98

Error Matrix W = 5

Random Forest

Actual/ Predicted	0	1
0	80	1
1	5	15

Overall Error (%) = 5.16

Boost

Actual/ Predicted	0	1
0	80	0
1	11	9

Overall Error (%) = 10.95

SVM

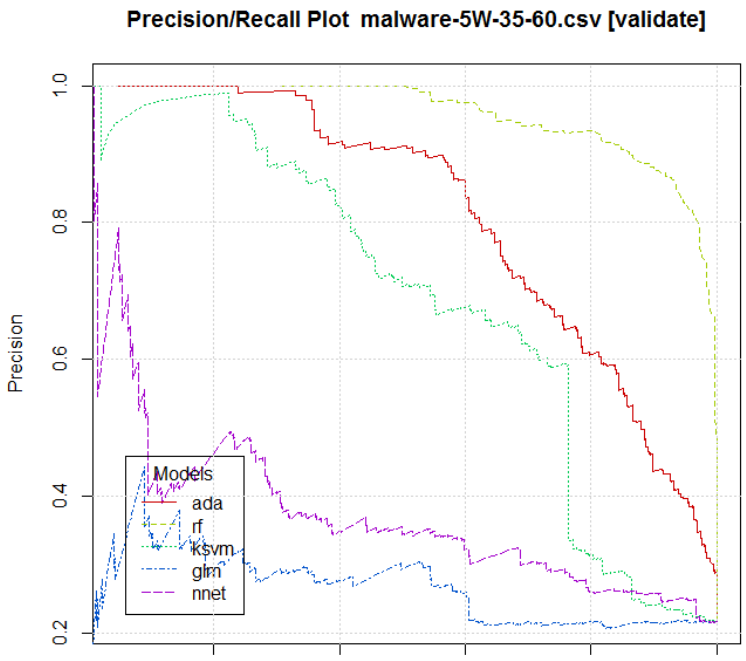
Actual/ Predicted	0	1
0	80	0
1	12	8

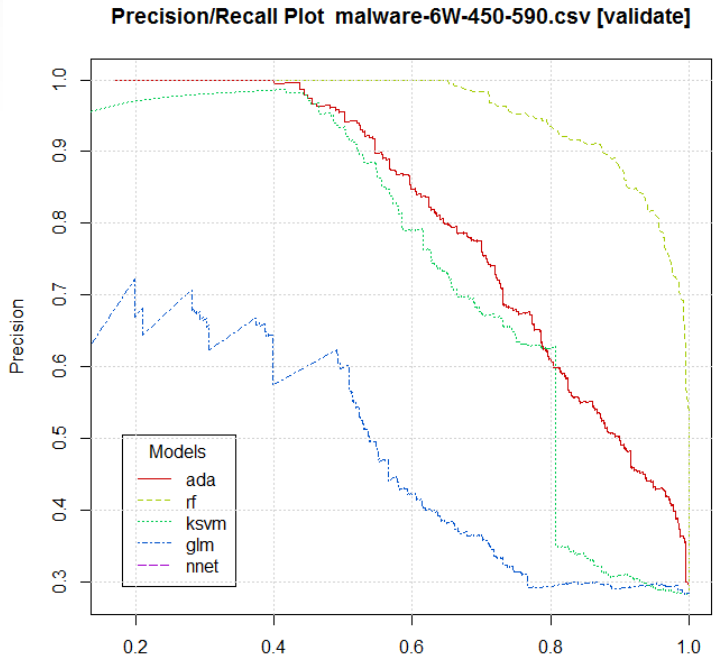
Overall Error (%) = 11.89

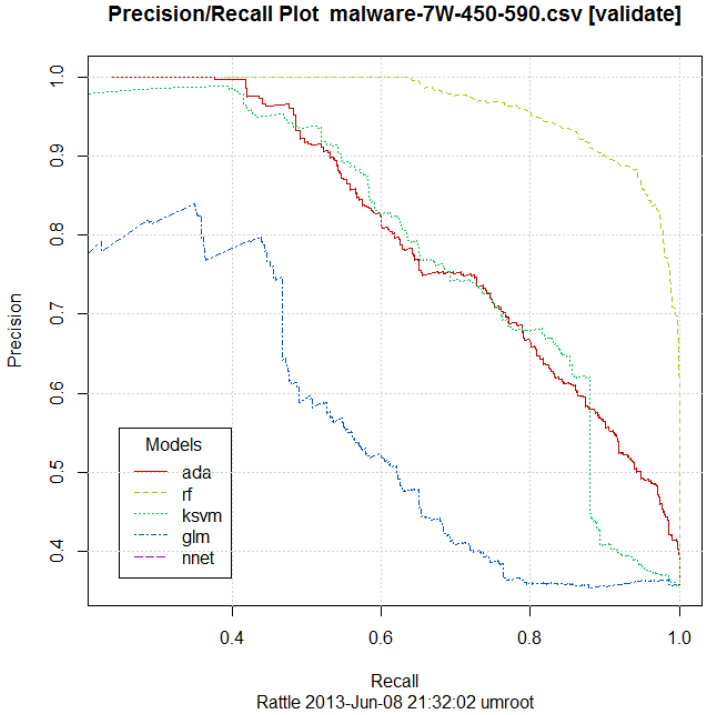
Neural Network

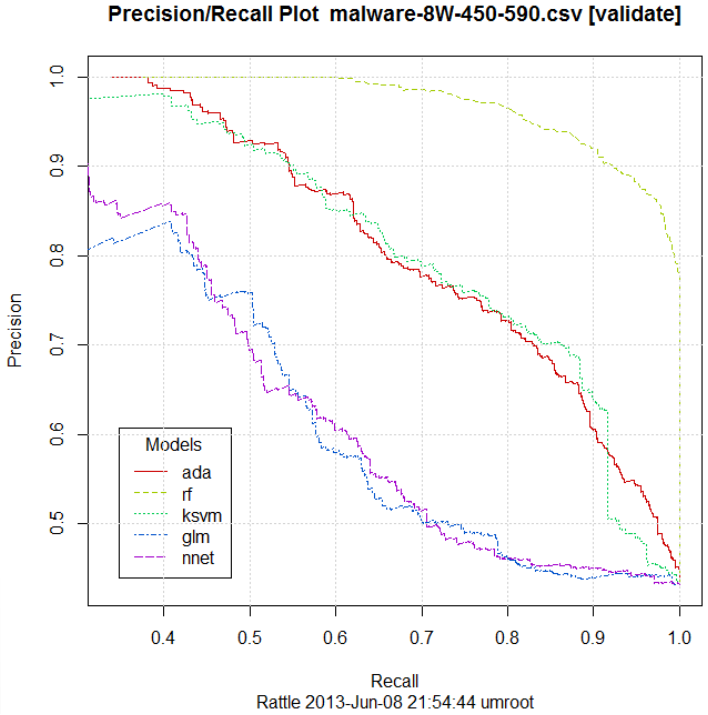
Actual/ Predicted	0	1
0	78	3
1	13	6

Overall Error (%) = 16.12


Precision/Recall Plot malware-5W-35-60.csv [validate]
Precision
Models
ada
rf
ksvm
glm
nnet


Precision/Recall Plot malware-6W-450-590.csv [validate]
Precision
Models
ada
rf
ksvm
glm
nnet


Precision/Recall Plot malware-7W-450-590.csv [validate]
Precision
Models
ada
rf
ksvm
glm
nnet
Recall
Rattle 2013-Jun-08 21:32:02 umroot


Precision/Recall Plot malware-8W-450-590.csv [validate]
Precision
Models
ada
rf
ksvm
glm
nnet
Recall
Rattle 2013-Jun-08 21:54:44 umroot

Conclusion

- Which window size has the overall best result?
  - W = 5 (highest: 98.79 %)
- Which Model has the overall best result?
  - Random Forest (In the 2nd place Boost)
  - RF has the average of 98% and Boost has the average of 87%
- Improvement of accuracy
  - SVM with increasing the window size
    - From 77% to 85%