Automatic Malware Categorization Using Cluster Ensemble

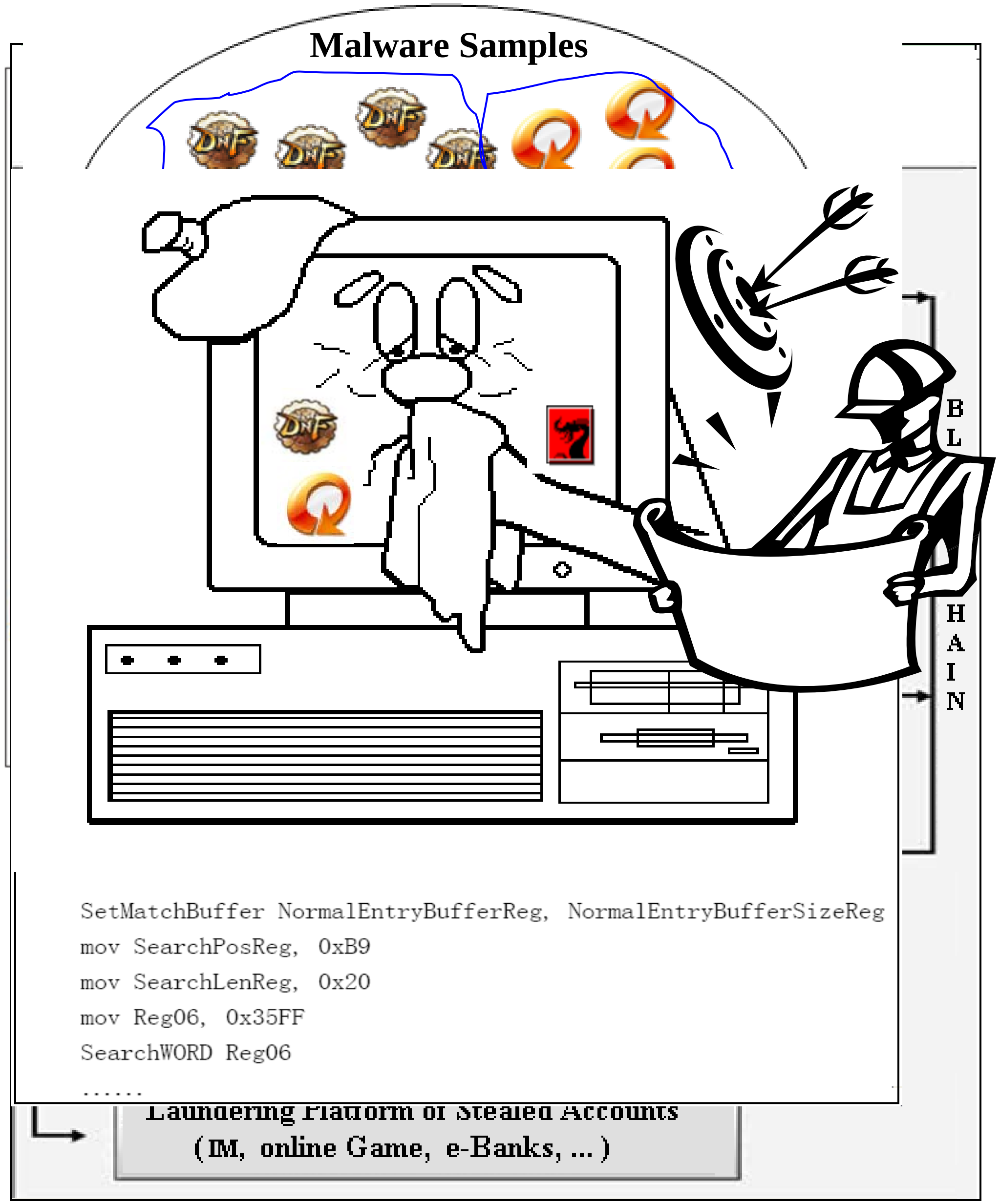
Yanfang Ye: Xiamen University & Internet Security R&D Center, Kingsoft Corporation
Tao Li: School of Computer Science, Florida International University
Yong Chen: Internet Security R&D Center, Kingsoft Corporation
Qiangshan Jiang: Xiamen University

Talk Outline

- ***Introduction and Motivation***
- **System Architecture and Description**
- **Experimental Results and Case Studies**

Motivation

- ➢Huge number of malware samples
- ➢"Black Chain" is driven by the economic benefit
- ➢Defense against malware
- ➢Effective methods for automatic malware categorization is in need



Automatic Malware Categorization

The process of automatic malware categorization:

1. Feature Extraction
   - ➢ Application Programming Interface (API) calls
   - ➢ Instruction sequences
   - ➢ Program behaviors
   - ➢ ……
2. Categorization
   - ➢ Hierarchical Clustering
   - ➢ K-medoids
   - ➢ ……

Limitation of current automatic malware categorization methods:

1. Limited effectiveness and efficiency
2. Few have been applied in real anti-malware industry

Automatic Malware Categorization using Cluster Ensemble

- Cluster Ensemble methods are popular in overcoming instability and increasing performance in machine learning tasks

- Our goal is developing an Automatic Malware Categorization System (AMCS for short) for automatically grouping malware samples into families that share some common characteristics using a cluster ensemble by aggregating the clustering solutions generated by different base clustering algorithms, while the domain knowledge in the form of sample-level constraints can be naturally incorporated in the ensemble framework.

- Our case studies on large and real data collections collected by the Anti-malware Lab of Kingsoft corporation demonstrate the usefulness of AMCS, which has already been incorporated into the scanning tool of Kingsoft's anti-malware software.

Talk Outline

- Introduction and Motivation
- **System Architecture and Description**
- Experimental Results and Case Studies

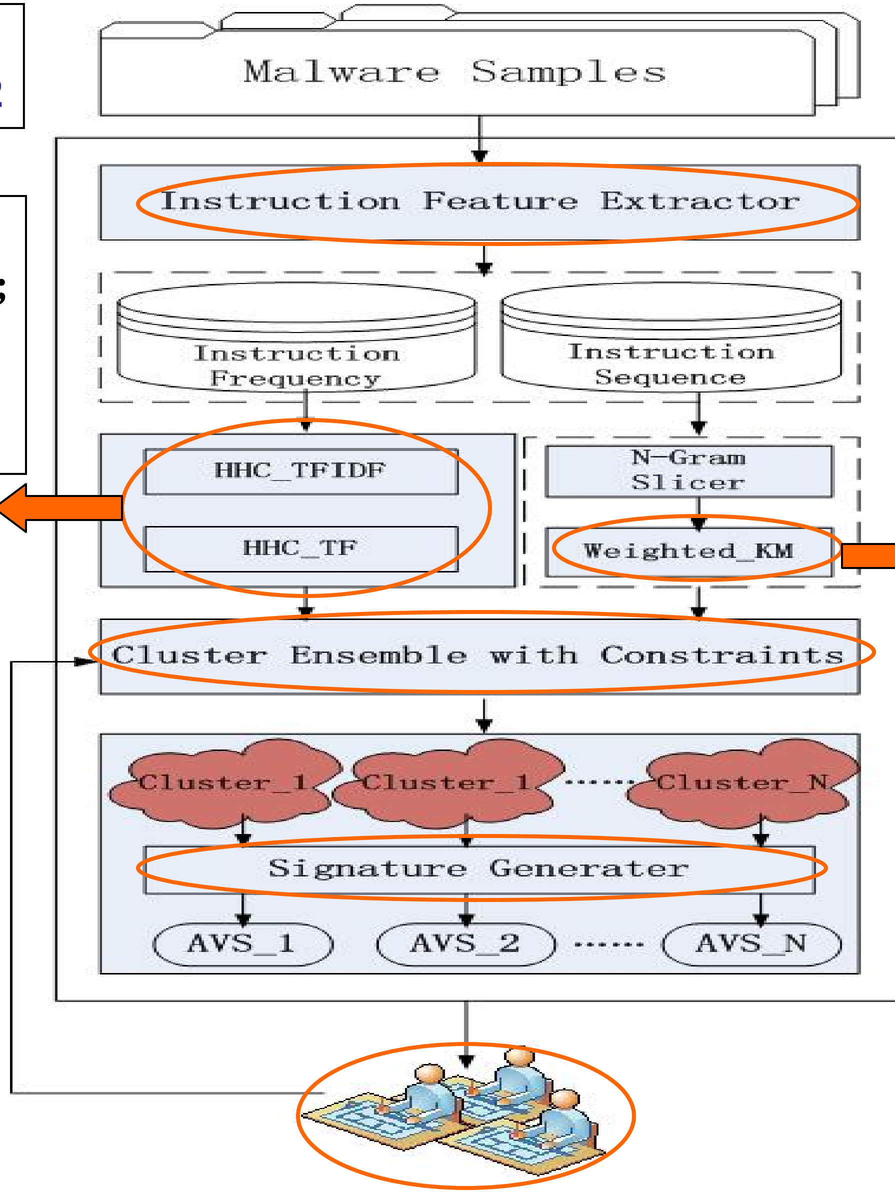
System Architecture

AdWare.Win32.Nodef.kp_7409
edaae06e6763dd23c93101ff4662

sub_10001000:
mov,dec,jnz,mov,mov,mov,retn;
sub_100010e0:
sub,push,mov,or,xor, …, ret;
…….

Hybrid Hierarchical Clustering algorithm with the tf-idf & tf weighting schemes

id	filename	instructionseq	invectorofinstruction
1	Adware.Clicker.vb__df4f57...	[illegible]	...10,10,10,10,10,10,43,10,9
2	Adware.Dodolook.ic__a1a6aff06822e...	sub_100010b7,push,mov,sub,push,push,push,and,push,...	5,2,7,5,5,5,31,5,22,9,8,425,426,427,2,2,5,22...
3	AdWare.Win32.Nodef.kp__037afc073...	sub_1001000,mov,dec,jnz,mov,mov,mov,retn;sub_1000...	2,27,12,2,2,2,21,5,5,2,5,2,20,4,2,20,4,5,5,5,6...
10	AdWare.Win32.Zwangi.a__053badb5e...	sub_1001010,push,mov,mov,add,test,mov,jnz,xor,pop,re...	5,2,2,10,20,2,12,9,22,21,2,2,10,20,434,2,5,2...
31	Adware.WSearch.a__ac7943f1e1750...	sub_1001000,ret;sub_10001001,mov,movsx,push,push,...	424,2,24,5,5,5,6,10,424,2,2,424,2,2,9,11,46,2
32	Adware.WSearch.a__bc1e983c9431d...	sub_1001000,mov,movsx,push,push,push,call,add,ret,su...	2,24,5,5,5,6,10,424,2,2,424,2,2,9,11,46,2,21...
33	Adware.WSearch.a__f6b0cecc483304...	sub_1001000,ret;sub_10001001,mov,movsx,push,push,...	424,2,24,5,5,5,6,10,424,2,2,424,2,2,9,11,46,2
34	Backdoor-DVC.trojan__b0ca175d4c98...	sub_00401000,push,nop,nop,mov,push,push,push,mov,p...	5,23,23,2,5,5,5,2,5,23,5,23,9,9,23,23,22,50,2...
35	Backdoor.Agent.mt__365a041413a01...	sub_004004b0,mov,mov,lea,push,push,push,push,push,p...	2,2,8,5,5,5,5,5,5,5,5,5,6,20,12,2,5,2,3,490,49...
36	Backdoor.Agent.mt__a30362fa5b294...	sub_1001000,push,push,push,xor,push,mov,push,mov,p...	5,5,5,9,5,2,5,2,5,5,5,2,6,5,5,5,5,2,6,2,2,2,2,2...
37	Backdoor.Huigezi.ec__0ad0799bab03...	sub_1001000,push,mov,xor,mov,mov,mov,lea,push,mov,...	5,2,9,2,2,2,8,5,2,6,2,22,424,5,2,2,2,20,4,5,5,5

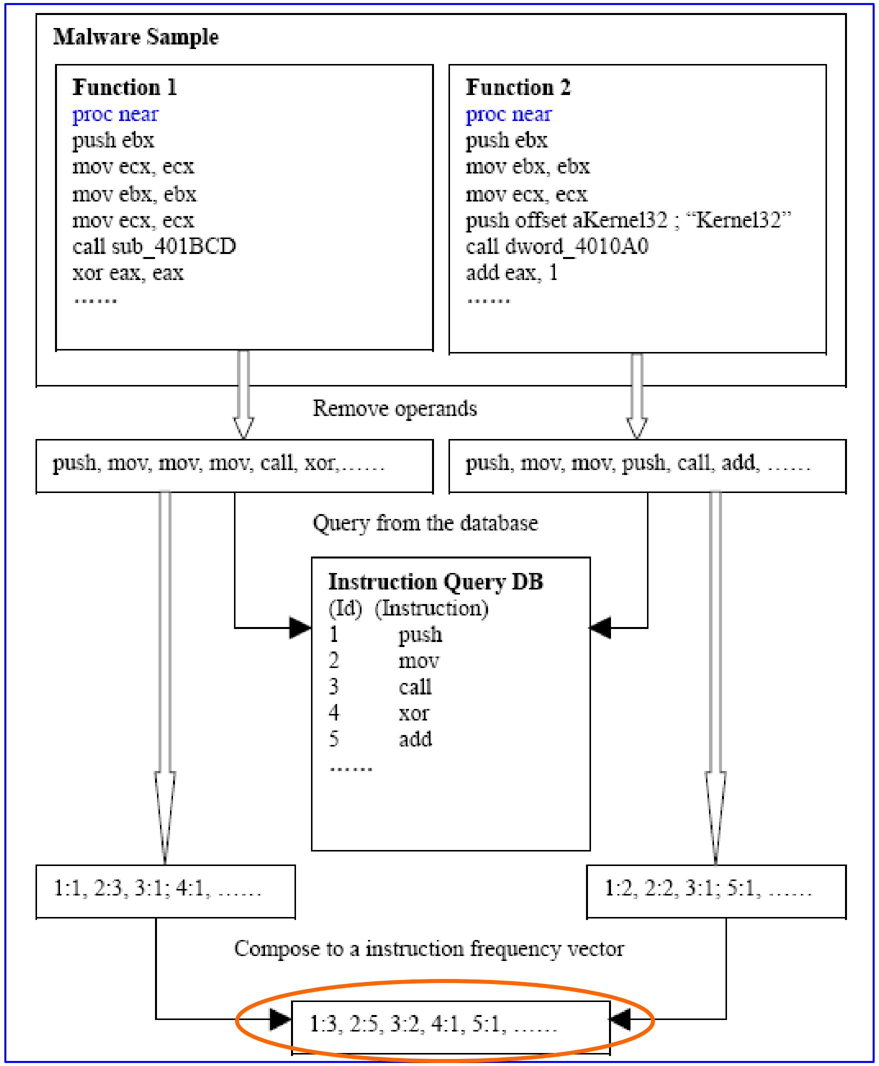


Weighted K-medoids Partitioned algorithm

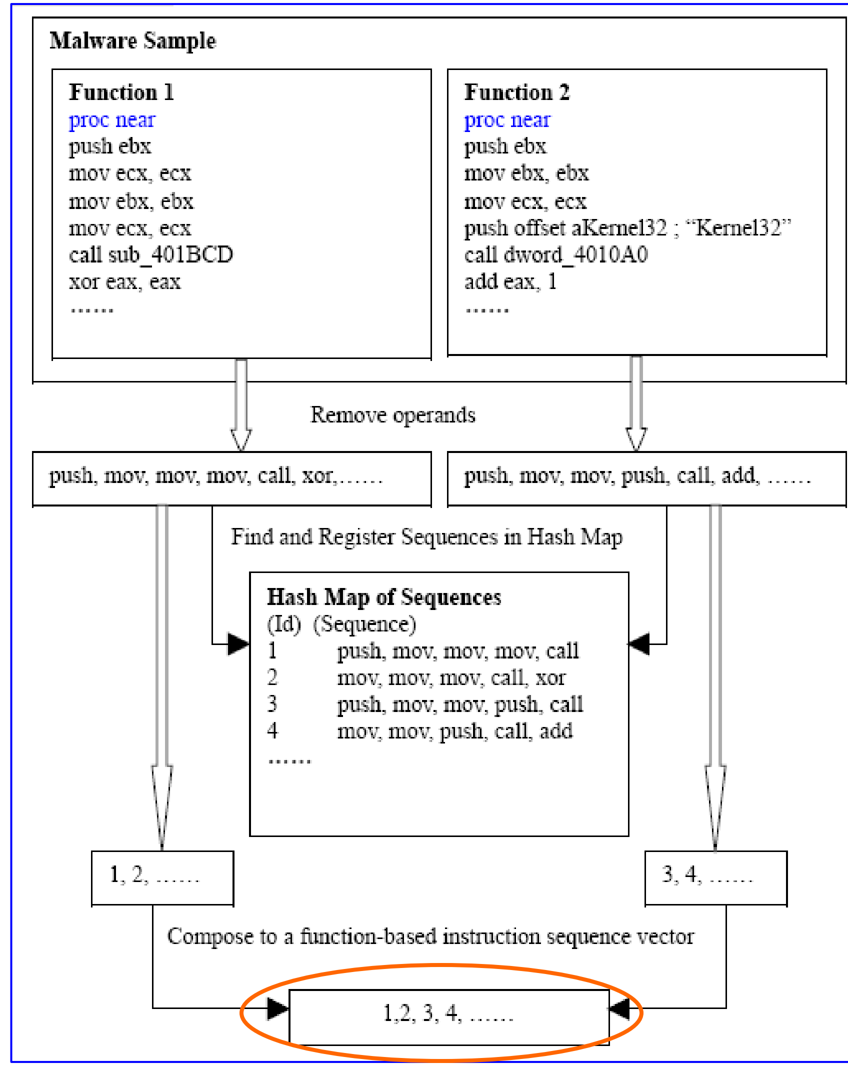
Characteristics of AMCS

- Well-Chosen Feature Representations
- Carefully-Designed Base Clusterings
- A Principled Cluster Ensemble Scheme
- Natural Application for Signature Generation
- Human-in-the-loop

Feature Representations



The extraction of instruction frequency



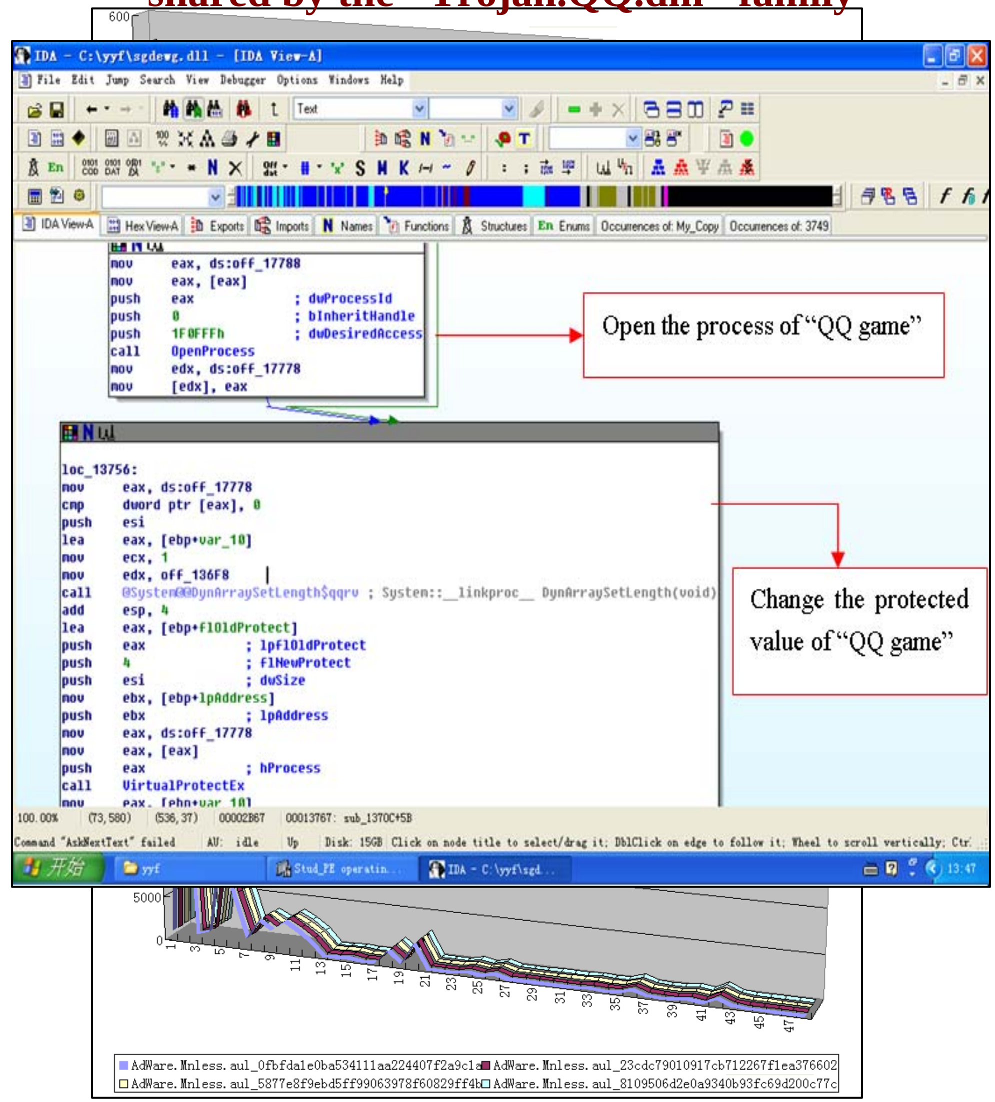
The extraction of function-based 5-gram instruction sequence segments

Advantages of the Feature Representations for Malware Categorization

- ➢Great ability for representing variants of a malware family
- ➢Easy for generating signatures for a malware family to detect its variants
- ➢High coverage rate of malware samples
- ➢Semantic Implications
- ➢High efficiency for feature extraction

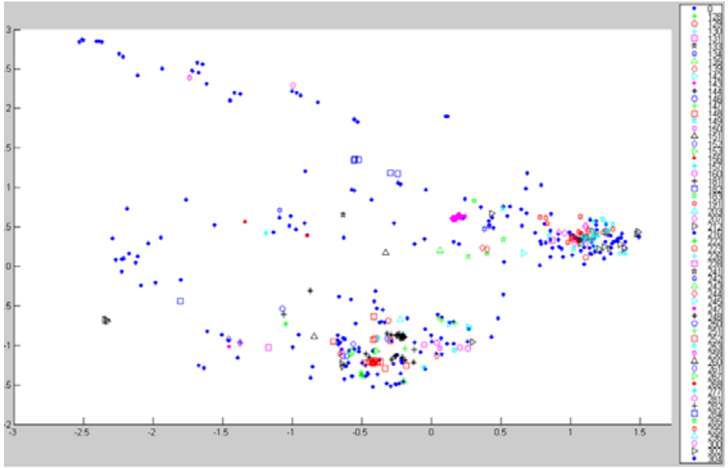
Shapes of instruction frequency patterns are shared by same malware family and differ between different families

The function-based instruction sequences shared by the "Trojan.QQ.dll" family

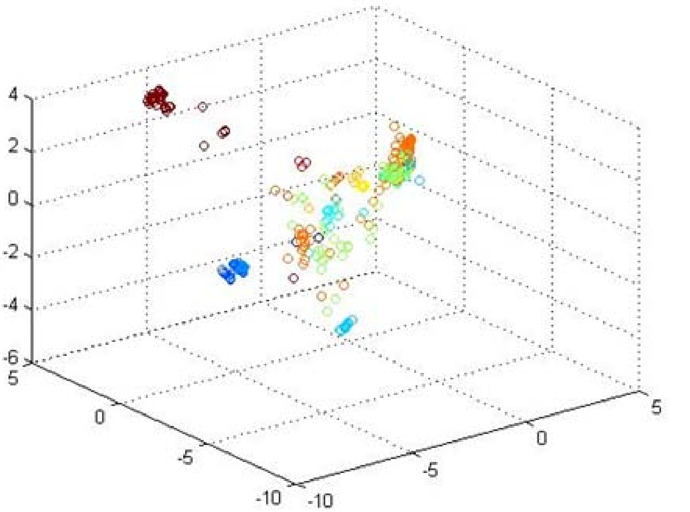


Base Clusterings ---- Hybrid Hierarchical Clustering (HHC)

DB: 1,434 malware with 1,222 dimensions



2 Dimensions transformed by PCA



3 Dimensions transformed by PCA

Input: The data set D
Output: The best K and data clusters

Set each sample as a singleton cluster;
for $K \leftarrow N - 1$ to 1 **do**
  Merge two clusters with closest medoids;
  Generate the new medoids of the merged clusters;
  Run K-medoids to obtain a partition;
  Calculate the validity index;
  Compare and keep the best K and corresponding clusters until now ;
end
Return the best K and corresponding clusters.

The algorithm description of HHC

$$FS = \sum_{i=1}^{N} \sum_{j=1}^{nc} u_{ij}^{m} \left(\|x_i - v_j\|_A^2 - \|v_j - v\|_A^2 \right)$$

The *Fukuyama-Sugeno* index (FS)

Base Clusterings ---- Weighted K-Medoids (WKM)

Input: N points in d-dimensional space, number of clusters k
Output: k clusters and the corresponding weight vector

Randomly choose k cluster medoids;
set initial weights to be $\frac{1}{d}$;
repeat
    Assign each points to the nearest cluster;
    Update the cluster medoids;
    Update the weight vector using Eq.(1);
    Calculate the validity index;
until *the weight vectors and the medoids do not change*;

$$w_{ij} = \begin{cases} \dfrac{\sum_{l=1}^{d} D_{il} - D_{ij} + E_{ij}}{(d-1)\sum_{l=1}^{d} D_{il} + \sum_{l=1}^{d} E_{il}}, & \sum_{l=0}^{d} D_{il} > 0 \\ \dfrac{1}{d}, & \sum_{l=0}^{d} D_{il} = 0 \wedge E_{ij} = 0 \end{cases}$$

$$D_{ij} = \sum_{x_t \in C_i} w_{ij}(x_{tj} - m_{ij})^2$$

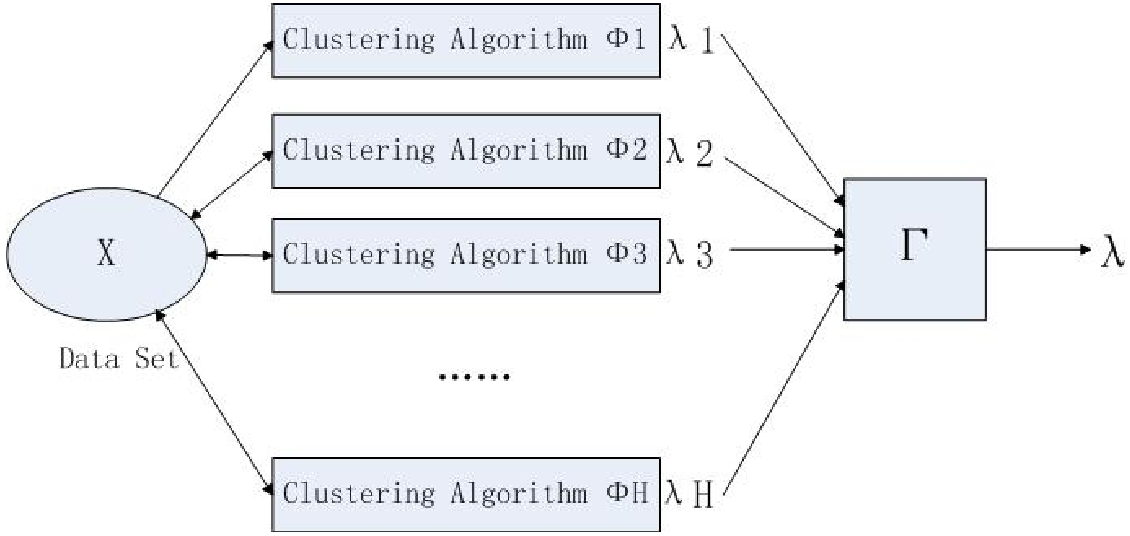
$$E_{ij} = \sum_{x_t \notin C_i} w_{ij}(x_{tj} - m_{ij})^2$$

Alg2. The algorithm description of WKM

Eq. (1) The weight of the *j*-th feature for cluster *i*

Cluster Ensemble

Cluster Ensemble: **aggregate the clustering solutions generated by different both hierarchical and partitioned clustering algorithms.**



Cluster ensemble incorporating sample-level constraints

- **Define the connectivity matrix** $M(P^T)$ **for the partition** P^t **as:**

$$M_{ij}(P^T) = \begin{cases} 1 & if \ x_i \ and \ x_j \ belong\ to\ same\ cluster\ C^t \\ 0 & else \end{cases} \Rightarrow \tilde{M}_{ij} = \frac{1}{T}\sum_{t=1}^{T} M_{ij}(P^t)$$

- **Incorporate sample-level constraints:**

1. **Must-link constraints:** $A = \{(x_{i1}, x_{j1}), \ldots, (x_{ia}, x_{ja})\}, a = |A|$
2. **Cannot-link constraints:** $B = \{(x_{p1}, x_{q1}), \ldots, (x_{pb}, x_{qb})\}, b = |B|$

- **A convex optimization problem with linear constraints:**

$$\begin{aligned} &\min_{P^*} J = \frac{1}{T}\sum_{t=1}^{T}\sum_{i,j=1}^{n}[M_{ij}(P^t) - M_{ij}(P^*)]^2 \\ &s.t. \\ &M_{ij}(P^*) = 1, \quad if\,(x_i, x_j) \in A \\ &M_{ij}(P^*) = 0, \quad if\,(x_i, x_j) \in B \end{aligned}$$

$$\Rightarrow M_{i_s j_s}(P^*) = \begin{cases} \frac{1}{T}\sum_{t=1}^{T} M_{ij}(P^t) & if\,(i_s, j_s)\, not\, in\, C \\ \mathrm{b_s} & else \end{cases}$$

Talk Outline

- Introduction and Motivation
- System Architecture and Description
- Experimental Results and Case Studies

Experimental Results and Analysis

1. Comparisons of Clustering Methods Based on Instruction Frequency
2. Comparisons of Clustering Methods Based on Instruction Sequences
3. Evaluation of Cluster Ensemble with Constraints
4. Comparisons with Different AV Venders

(1) Comparisons of Malware Clustering Methods Based on Instruction Frequency

- Measures:

$$Macro-F1=\frac{\sum_{i=1}^{C}F_{1i}}{n}\qquad Micro-F1=\frac{\sum_{i=1}^{C}2*\mathrm{Re}call_i*\Pr ecision_i}{\sum_{i=1}^{C}\mathrm{Re}call_i+\Pr ecision_i}\qquad F_1=\frac{2*\mathrm{Re}call*\Pr ecision}{\mathrm{Re}call+\Pr ecision}$$

Day	Num	D	F	Alg	*Macro*	*Micro*
1	3546	1226	88	KM_TFIDF	0.6925	0.7376
1	3546	1226	88	HC_TFIDF	0.7501	0.7134
1	3546	1226	88	HHC_TFIDF	0.8218	0.8015
1	3546	1226	88	KM_TF	0.6279	0.6802
1	3546	1226	88	HC_TF	0.7228	0.7237
1	3546	1226	88	HHC_TF	0.8162	0.8128
2	3005	1178	102	KM_TFIDF	0.7033	0.7661
2	3005	1178	102	HC_TFIDF	0.787	0.7921
2	3005	1178	102	HHC_TFIDF	0.8263	0.8101
2	3005	1178	102	KM_TF	0.5687	0.602
2	3005	1178	102	HC_TF	0.6605	0.6845
2	3005	1178	102	HHC_TF	0.7655	0.7468
3	5162	2375	324	KM_TFIDF	0.5942	0.5905
3	5162	2375	324	HC_TFIDF	0.6365	0.6591
3	5162	2375	324	HHC_TFIDF	0.722	0.7335
3	5162	2375	324	KM_TF	0.6398	0.6126
3	5162	2375	324	HC_TF	0.7436	0.7228
3	5162	2375	324	HHC_TF	0.7895	0.7957

Table 1: Based on instruction frequency, the categorization results of different categorizers on the real daily new malware collection from Jan 11th, 2010 to Jan 13th, 2010.

Remark: "Num"-the total number of the malware samples, "D"-Dimensions of the data set, "F"-the real malware families, "Macro"-Macro-F1 measure and "Micro"-Micro-F1 measure.

(2) Comparisons of Malware Clustering Methods Based on Instruction Sequences

- Measures:

$$Macro-F1=\frac{\sum_{i=1}^{C} F_{1i}}{n} \qquad Micro-F1=\frac{\sum_{i=1}^{C} 2*\mathrm{Re}\,call_i * \mathrm{Pr}\,ecision_i}{\sum_{i=1}^{C} \mathrm{Re}\,call_i + \mathrm{Pr}\,ecision_i} \qquad F_1=\frac{2*\mathrm{Re}\,call*\mathrm{Pr}\,ecision}{\mathrm{Re}\,call+\mathrm{Pr}\,ecision}$$

Day	Num	D	F	Alg	*Macro*	*Micro*
1	3546	7208	88	KM	0.6135	0.6747
1	3546	7208	88	WKM	0.8196	0.8559
2	3005	6923	102	KM	0.6882	0.6421
2	3005	6923	102	WKM	0.8071	0.8015
3	5162	11054	324	KM	0.6279	0.6252
3	5162	11054	324	WKM	0.7874	0.8147

Table 2: Based on function-based instruction sequences, the categorization results of different categorizers on the real daily new malware collection from Jan 11th, 2010 to Jan 13th, 2010.

(3) Evaluation of Cluster Ensemble with Constraints

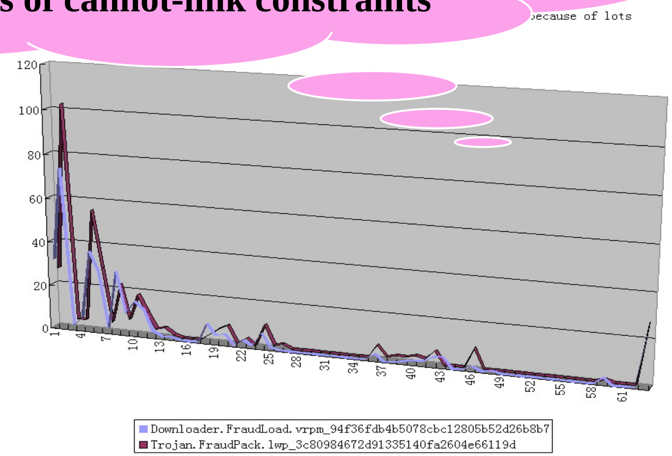
- 1,385 pairs of must-link constraints
- 1,078 pairs of cannot-link constraints

- Effectiveness

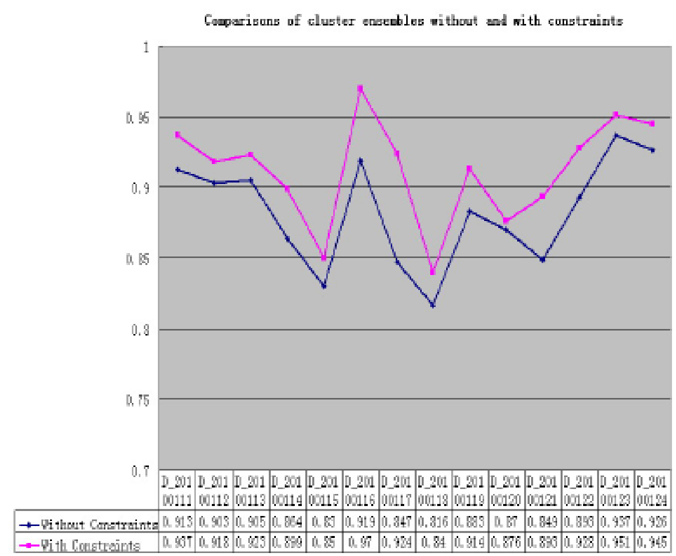
Day	Num	F	Alg	*Macro*	*Micro*
1	3546	88	HHC_TFIDF	0.8218	0.8015
1	3546	88	HHC_TF	0.8162	0.8128
1	3546	88	WKM	0.8196	0.8559
1	3546	88	NCE	0.9017	0.9137
1	3546	88	CE	0.9302	0.9437
2	3005	102	HHC_TFIDF	0.8263	0.8101
2	3005	102	HHC_TF	0.7655	0.7468
2	3005	102	WKM	0.8071	0.8015
2	3005	102	NCE	0.8989	0.8669
2	3005	102	CE	0.9245	0.9113
3	5162	324	HHC_TFIDF	0.722	0.7335
3	5162	324	HHC_TF	0.7895	0.7957
3	5162	324	WKM	0.7874	0.8147
3	5162	324	NCE	0.8605	0.8896
3	5162	324	CE	0.9183	0.9181

Table 2: Evaluation of the malware categorization results of clustering ensemble.

Remark: "NCE"-cluster ensemble without constraints, "CE"-cluster ensemble with constraints.



Example of sample-level Cannot-link constraints



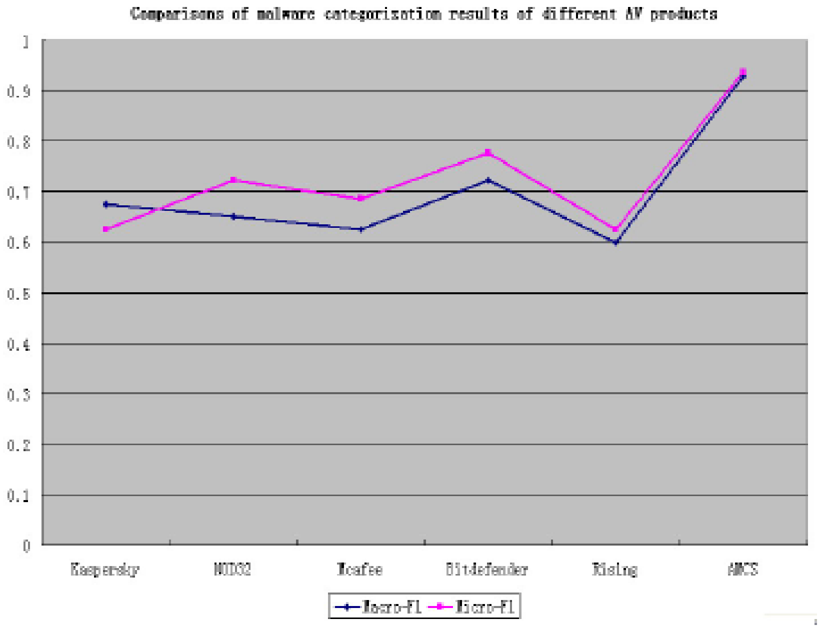
Comparisons of malware categorization results of cluster ensembles without and with constraints

(4) Comparisons with Different AV Venders

- Effectiveness

The categorization results of different AV software on the whole data collection of 42,180 malware samples.

AV.	Detected	Families	$MacroF1$	$MicroF1$
Kasp	35,433	1,998	0.6736	0.6246
Nod32	33,634	1,598	0.6498	0.7233
Mcafee	35,500	1,859	0.6253	0.6856
BD	39,723	1,916	0.7215	0.7761
Rising	35,712	1,885	0.5983	0.6257
AMCS	41,623	2,271	0.9274	0.9369



- Efficiency

(1) Categorizing 3,546 malware samples by our AMCS system including feature extraction needs 3 minutes;

(2) The whole process of 42,180 malware samples needs 15 minutes.

(4) Automatic Signature Generation

- Signature: the instruction sequences (with operands) frequently appeared within a malware family but rarely appeared in other families.

The signature with id “72237142”:

```
[ScriptScan]
    mov Reg00, 0x00
    mov Reg01, 0x01
    mov Reg02, 0x02
    mov Reg03, 0x03
    mov Reg04, 0x04
    Call PE.GetImageBase Reg05
    ReturnIfFalse

    SetMatchBuffer NormalEntryBufferReg, NormalEntryBufferSizeReg
    mov SearchPosReg, 0xB9
    mov SearchLenReg, 0x20
    mov Reg06, 0x35FF
    SearchWORD Reg06
    ......
```

signature id	malware family name	result check analyst	malware variants
72237142	Win32.Troj.YmdfiveT.xb.151557	lixue	28935
72396381	Win32.AgeTCT.ex.210944	lixue	15544
5144351	Win32.LoaderT.ek.367616	lixue	5753
5143478	Win32.Troj.InjectT.ia.299044	lixue	3823
5030525	Win32.Troj.StartPageT.ze.180224	lixue	3770
5133803	Win32.Troj.StartPageT.oc.495663	lixue	2554
72215575	Win32.Troj.LeiguoT.al.188421	lixue	2436
5062310	Win32.Troj.StartPageT.ic.192512	raoshuai	2356
5295746	Win32.Troj.FakeTenT.ul.51200	lixue	1890
72133690	Win32.Troj.GuoToolbarT.di.184320	raoshuai	1802
205632057	Worm.AllApleT.cz.67868	chenzhangqun	1761
72182948	Win32.Troj.StartPageT.qw.180224	lixue	1659
72175725	Win32.Troj.StartPageT.mn.241731	lixue	1263
5073955	Win32.Troj.StartPageT.zg.180224	lixue	1173
206229895	Win32.Troj.FakeRegT.ss.1222249	liangfei	1113
474780405	Win32.Troj.ClickerT.hn.212773	raoshuai	1082
4877882	Win32.Troj.FuckCryptT.d.114176	raoshuai	986

System development and operation

- Expense: Kingsoft has spent over $500K in the development of the AMCS system and about $100K on the hardware equipment.

- Usefulness: Over 30 virus analysts at Kingsoft's Anti-Virus lab are utilizing the system on the daily basis. In practice, a virus analyst has to spend at least 10 hours to manually analyze 100 malware samples for categorization. Using the AMCS system, the categorization of about 30,000 malware samples can be performed within 20 minutes. The high efficiency of our AMCS system can greatly save human labors and reduce the staff cost.

- Benefited users: This would benefit over 10 million Internet users of Kingsoft's client anti-malware products.

The AV products of Kingsoft incorporated AMCS

The AV products of Kingsoft incorporating AMCS

The detection of malware using the signature generated by AMCS

Conclusion

- Summary

  - ✓ AMCS categorizes malware samples into families that share some common traits by an ensemble of different clustering solutions generated by different clustering methods.

  - ✓ The domain knowledge in the form of sample-level constraints can be naturally incorporated in the ensemble framework.

  - ✓ The case studies on large and real daily malware sample collections obtained from the Anti-malware Lab of Kingsoft corporation demonstrate the effectiveness and efficiency of our AMCS system.

  - ✓ AMCS is a practical solution for automatic malware categorization from the huge malware sample collections and has already been incorporated into the scanning tool of Kingsoft's Anti-malware software.

- Future Work

  - ✓ Conduct further study to construct a more streamline signature library for better malware detection on the client anti-malware products based on our AMCS system.

Q & A

- E-mail: yeyanfang@yahoo.com.cn