Automatic Analysis of Malware Behavior using Machine Learning

Konrad Rieck, Philipp Trinius, Carsten Willems, Thorsten Holz

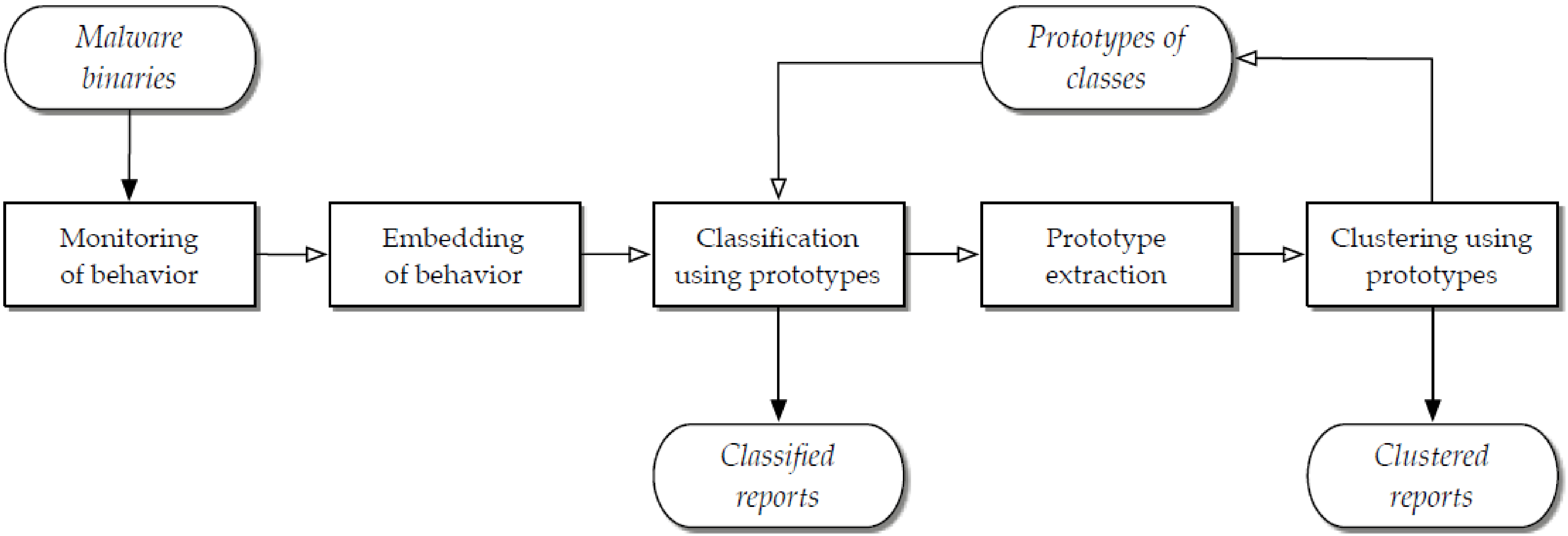
Peng Su

CISC850
Cyber Analytics

Automatic Analysis of Malware Behavior

- Malware threaten the Internet
- Dynamic VS Static
- binary packers, encryption, or self-modifying code, to obstruct analysis.
- behavior of malicious software during run-time.

Automatic Analysis of Malware Behavior

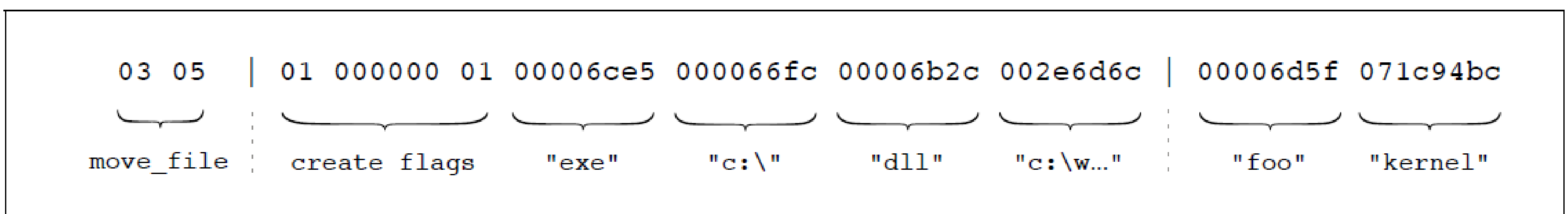


Monitoring of Malware Behavior

- Malware Sandboxes --CWSandbox
- Malware Instruction Set

```
<move_file srcfile="c:\foo.exe" dstfile="c:\windows\system32\kernel32.dll"
        filetype="file" creationdistribution="CREATE_NEW" />
```

(a) CWSandbox representation of system call



(b) MIST representation of system call

Malware Instruction Set

- MIST instruction keep the stable and discriminative patterns such as directory and mutex name at the beginning.

$$\underbrace{\underbrace{\underbrace{\texttt{CATEGORY OPERATION}}_{\textit{Level 1}} \mid \texttt{ARGBLOCK1}}_{\textit{Level 2}} \mid \texttt{ARGBLOCK2}}_{\textit{Level 3}} \mid \ldots \mid \texttt{ARGBLOCK}\mathit{N}$$

Embedding of Malware Behavior

- Embedding using Instruction Q-grams
- Comparing Embedding reports

Embedding using Instruction Q-grams

$$\varphi(x) = (\varphi_s(x))_{s \in \mathcal{S}} \text{ with } \varphi_s(x) = \begin{cases} 1 & \text{if report } x \text{ contains } q\text{-grams } s, \\ 0 & \text{otherwise.} \end{cases}$$

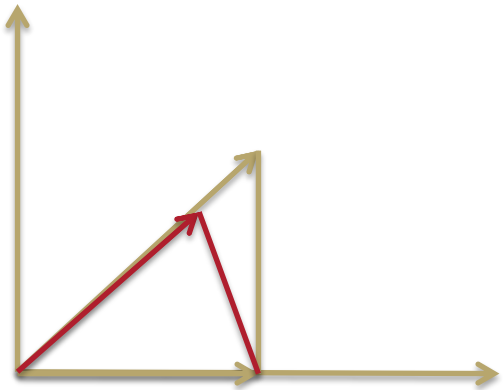
- For example, if report x='1|A 2|A 1|A 2|A', A={1|A, 2|A }, the q for q-grams is 2.

$$\varphi(\texttt{`1|A 2|A 1|A 2|A'}) \longmapsto \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} \begin{array}{l} \texttt{`1|A 1|A'} \\ \texttt{`1|A 2|A'} \\ \texttt{`2|A 1|A'} \\ \texttt{`2|A 2|A'.} \end{array}$$

Embedding using Instruction Q-grams

- Normalization
- Redundancy of behavior, considered alphabet, length of reports

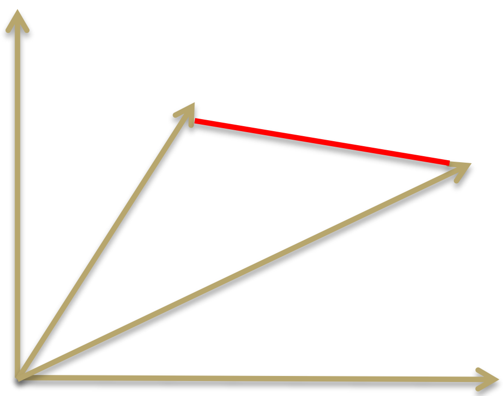
$$\hat{\varphi}(x) = \frac{\varphi(x)}{||\varphi(x)||}$$



Comparing Embedding reports

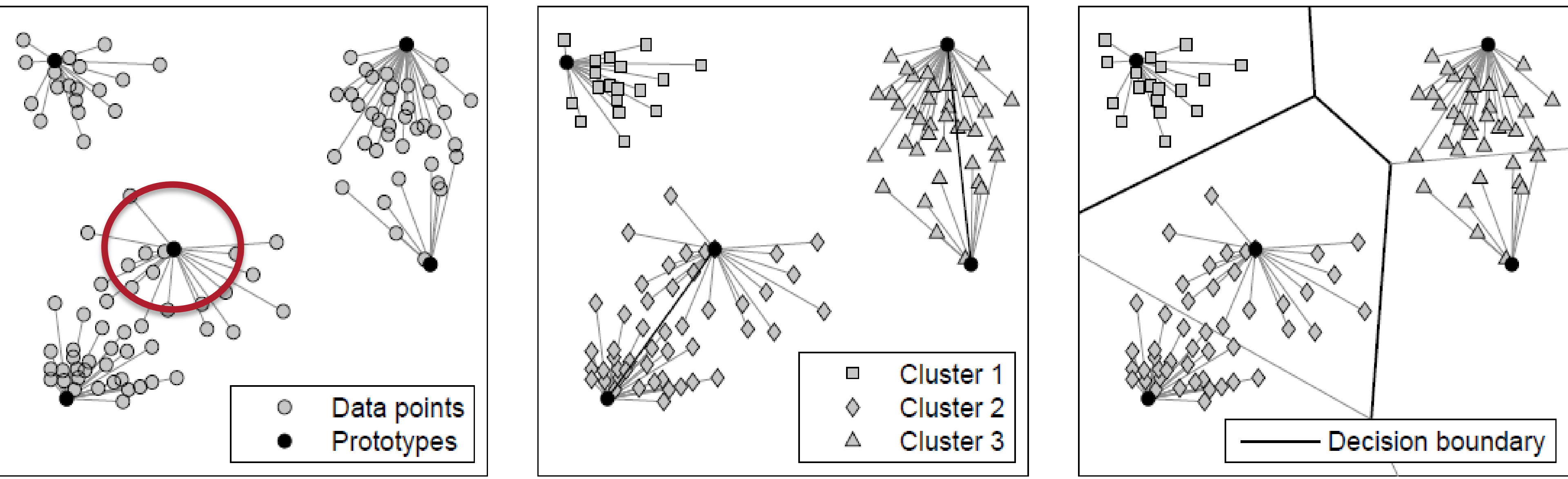
- Euclidean distance

$$d(x,z) = ||\hat{\varphi}(x) - \hat{\varphi}(z)|| = \sqrt{\sum_{s \in \mathcal{S}} (\hat{\varphi}_s(x) - \hat{\varphi}_s(z))^2}$$



Clustering and Classification

- Prototypes->Clustering-> Classification

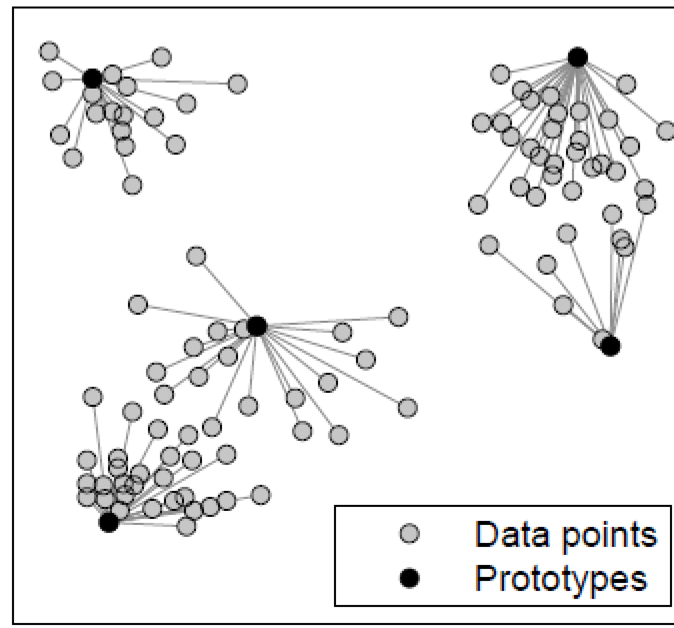


(a) Prototypes (b) Clustering (c) Classification

Prototype Extraction

Algorithm 1 Prototype extraction

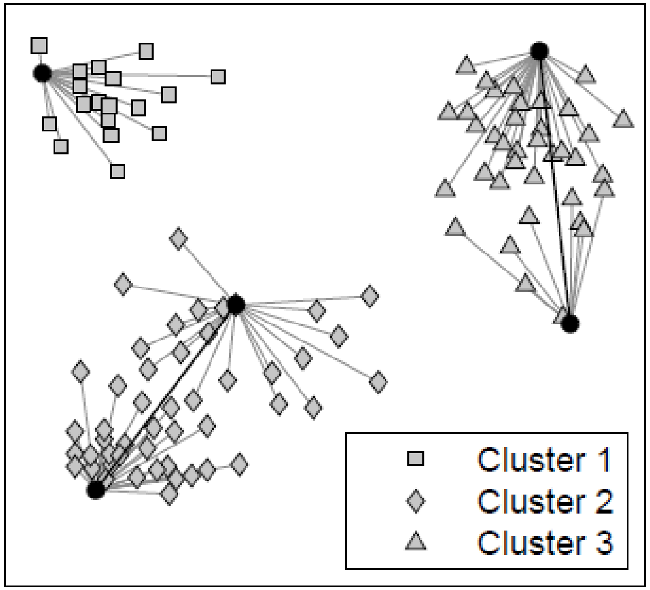
$prototypes \leftarrow \emptyset$
$distance[x] \leftarrow \infty$ for all $x \in reports$
while $\max(distance) > d_p$ **do**
    choose z such that $distance[z] = \max(distance)$
    for $x \in reports$ and $x \neq z$ **do**
        if $distance[x] > ||\hat{\varphi}(x) - \hat{\varphi}(z)||$ **then**
            $distance[x] \leftarrow ||\hat{\varphi}(x) - \hat{\varphi}(z)||$
    add z to $prototypes$



Clustering using Prototypes

Algorithm 2 Clustering using prototypes

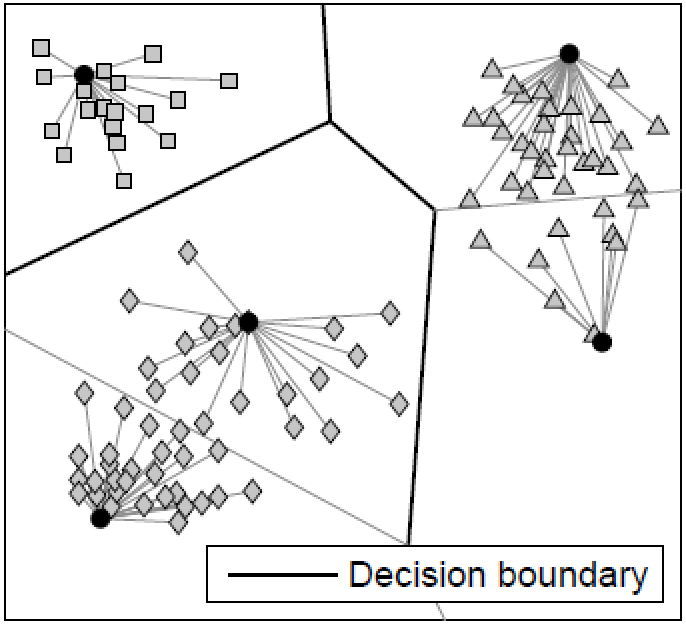
1: **for** $z, z' \in prototypes$ **do**
2: $\quad distance[z, z'] \leftarrow ||\hat{\varphi}(z) - \hat{\varphi}(z')||$
3: **while** $\min(distance) < d_c$ **do**
4: $\quad$ merge clusters z, z' with minimum $distance[z, z']$
5: $\quad$ update $distance$ using complete linkage
6: **for** $x \in reports$ **do**
7: $\quad z \leftarrow$ nearest prototype to x
8: $\quad$ assign x to cluster of z
9: reject clusters with less than m members



Classification using Prototypes

Algorithm 3 Classification using prototypes

1: **for** $x \in \mathit{reports}$ **do**
2: $\quad z \leftarrow$ nearest prototype to x
3: $\quad$ **if** $||\hat{\varphi}(z) - \hat{\varphi}(x)|| > d_r$ **then**
4: $\quad\quad$ reject x as unknown class
5: $\quad$ **else**
6: $\quad\quad$ assign x to cluster of z



Incremental Analysis

Algorithm 4 Incremental Analysis

```
1: rejected ← ∅, prototypes ← ∅
2: for reports ← data source ∪ rejected do
3:     classify reports to known clusters using prototypes
4:     extract prototypes from remaining reports
5:     cluster remaining reports using prototypes
6:     prototypes ← prototypes ∪ prototypes of new clusters
7:     rejected ← rejected reports from clustering
```

Experiments & Application

- Evaluation Data
- Three parameters to decide
- Evaluation of Components
- How to select the best parameters d_p, d_c, d_r

Evaluation Data

- A reference data set
- Evaluate and calibrate the framework
- An application data set
- See the performance on unknown malwares

Reference Data Set

	Malware class	#		Malware class	#
a	AdultBrowser	262	*m*	PornDialer	98
b	Allaple*	300	*n*	Rbot	101
c	Bancos	48	*o*	Rotator*	300
d	Casino	140	*p*	Sality	85
e	Dorfdo	65	*q*	Spygames	139
f	Ejik	168	*r*	Swizzor	78
g	Flystudio	33	*s*	Vapsup	45
h	LdPinch	43	*t*	VikingDll	158
i	Looper	209	*u*	VikingDz	68
j	MagicCasino	174	*v*	Virut	202
k	Podnuha*	300	*w*	Woikoiner	50
l	Posion	26	*x*	Zhelatin	41

Application Data Set

Data set description			
Collection period			August 1–7, 2009
Collection location			Sunbelt Software
Data set size (kilobytes)			21,808,644
Number of reports			33,698
Data set statistics	*min.*	*avg.*	*max.*
Reports per day	3,760	4,814	6,746
Instructions per report	15	11,921	103,039
Size per report (kilobytes)	1	647	5,783

Evaluation of Components

- Precision and recall

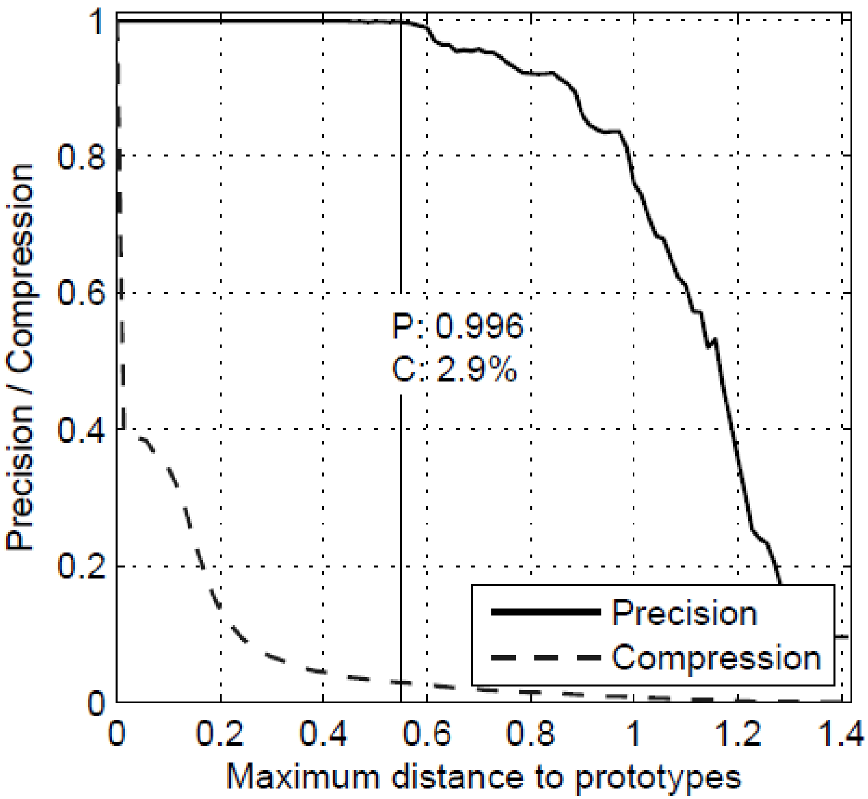
$$P = \frac{1}{n} \sum_{c \in C} \#_c \quad \text{and} \quad R = \frac{1}{n} \sum_{y \in Y} \#_y$$

Evaluation of Components

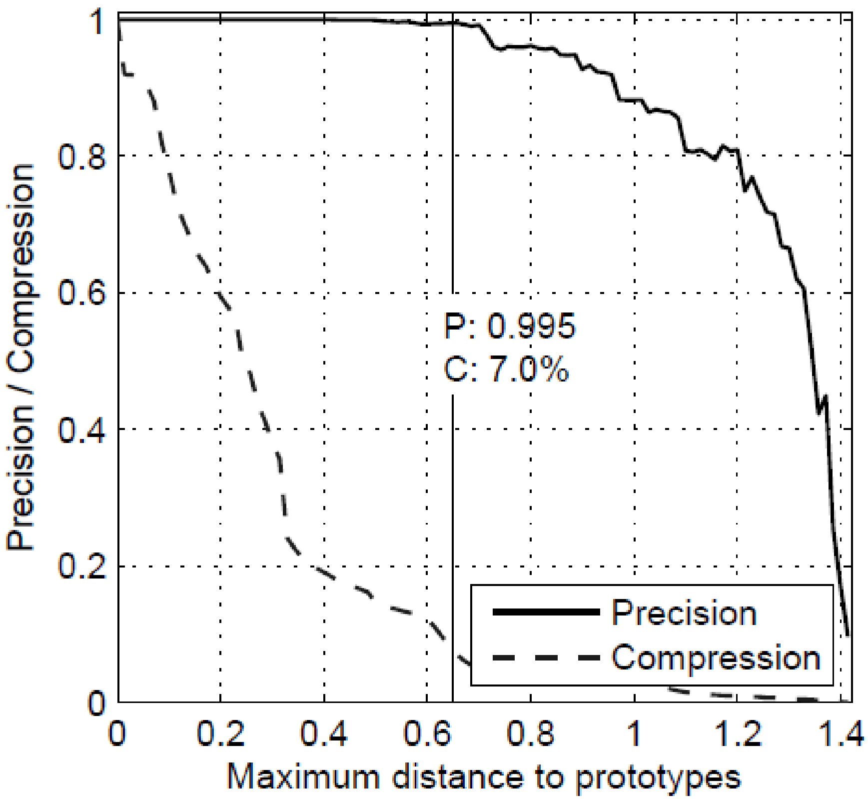
- F-measure

$$F = \frac{2 \cdot P \cdot R}{P + R}$$

Evaluation of Components--d_p

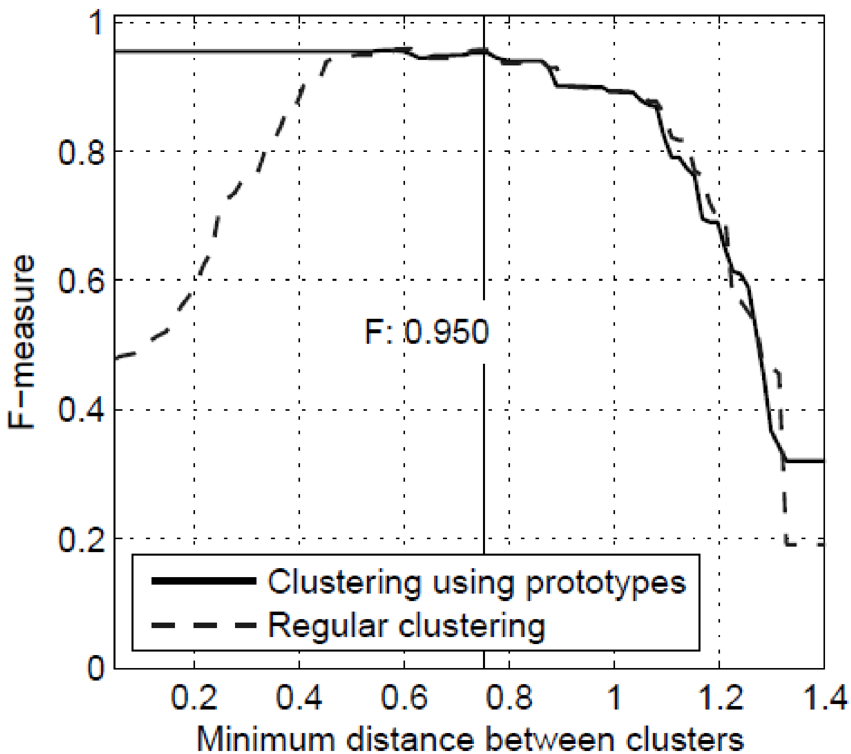


(a) MIST level 1

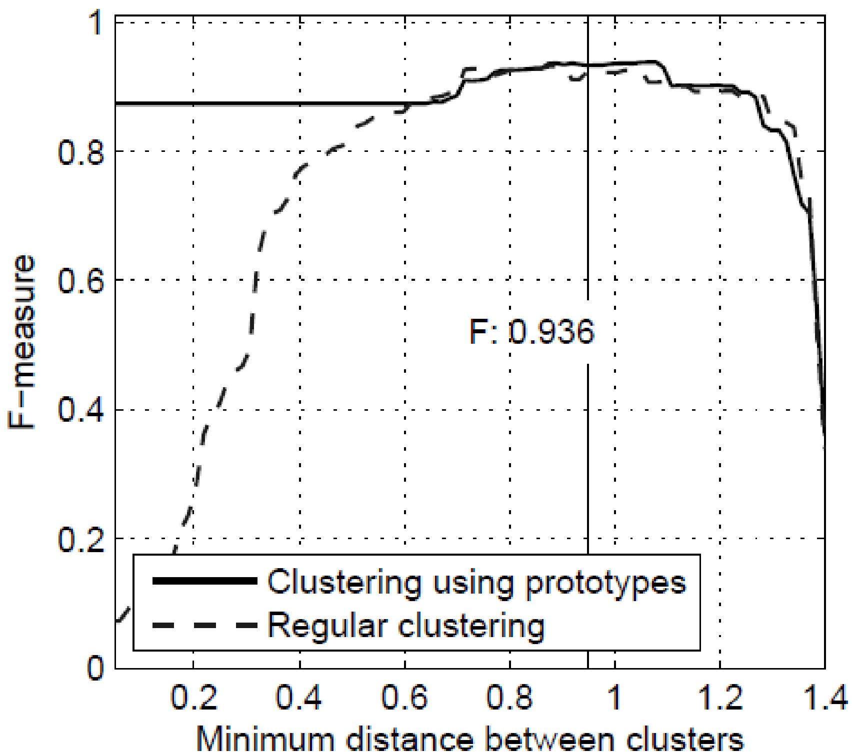


(b) MIST level 2

Evaluation of Components--d_c

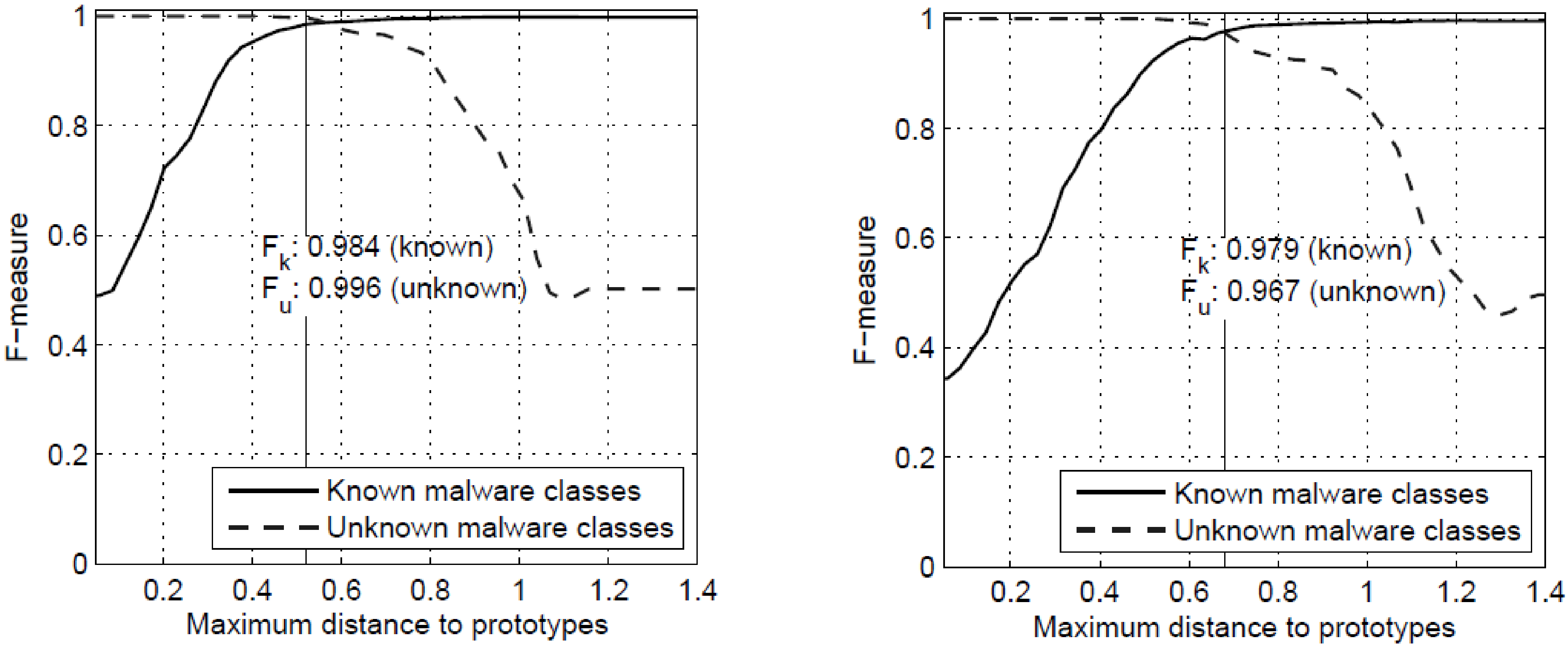


(a) MIST level 1



(b) MIST level 2

Evaluation of Components--d_r

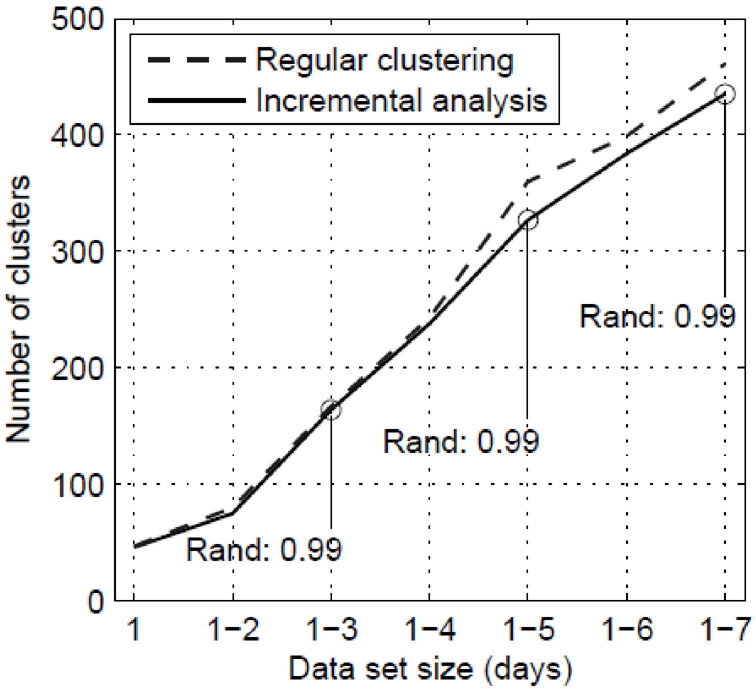


(a) MIST with level 1 (b) MIST with level 2

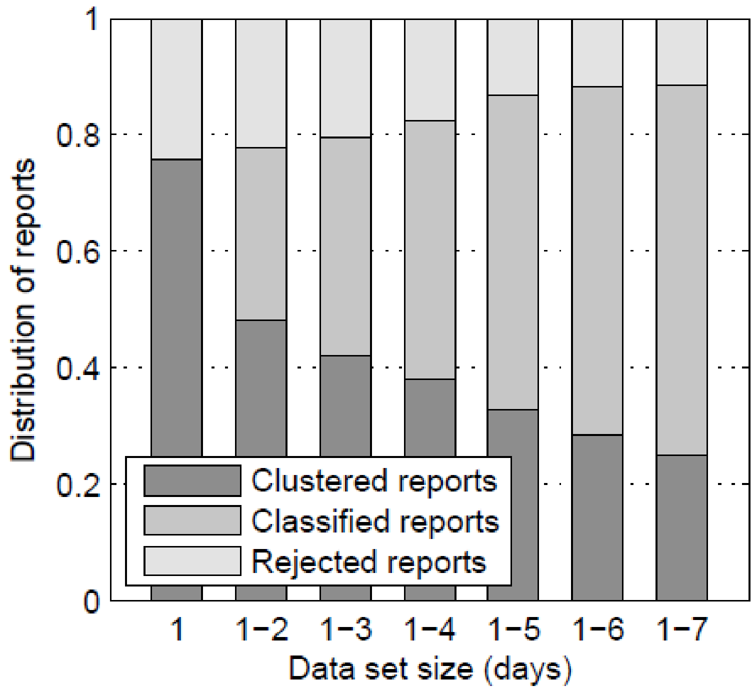
Comparative Evaluation with State-of-the-Art

Clustering methods	F-measure	
Clustering using prototypes with MIST level 1	0.950	
Clustering using prototypes with MIST level 2	0.936	
Clustering using LSH with Anubis features (Bayer et al., 2009a)	0.881	
Classification methods	F_k	F_u
Classification using prototypes with MIST level 1	0.981	0.997
Classification using prototypes with MIST level 2	0.972	0.964
Classification using SVM with XML features (Rieck et al., 2008)	0.807	0.548

An Application Scenario



(a) Comparison with regular clustering



(b) Clustering-classification ratio