Kullback–Leibler 情報量を用いた亜種マルウェアの同定

電気通信大学

中村 燎太、松宮 遼、高橋 一志、大山 恵弘

背景

近年、マルウェアの検出数は増加しつつある

- 亜種関係にあるマルウェアも増加傾向にある
  - マルウェア自動生成ツールの台頭が影響
    - ZeuS, SpyEyeといったツールが蔓延
    - 攻撃者のスキルに依らず、亜種の関係にあるマルウェアを自動的かつ大量に生成可能な環境が整えやすい

マルウェアを手作業で分析し、対応するパターンファイルを作成するには人手が掛かる

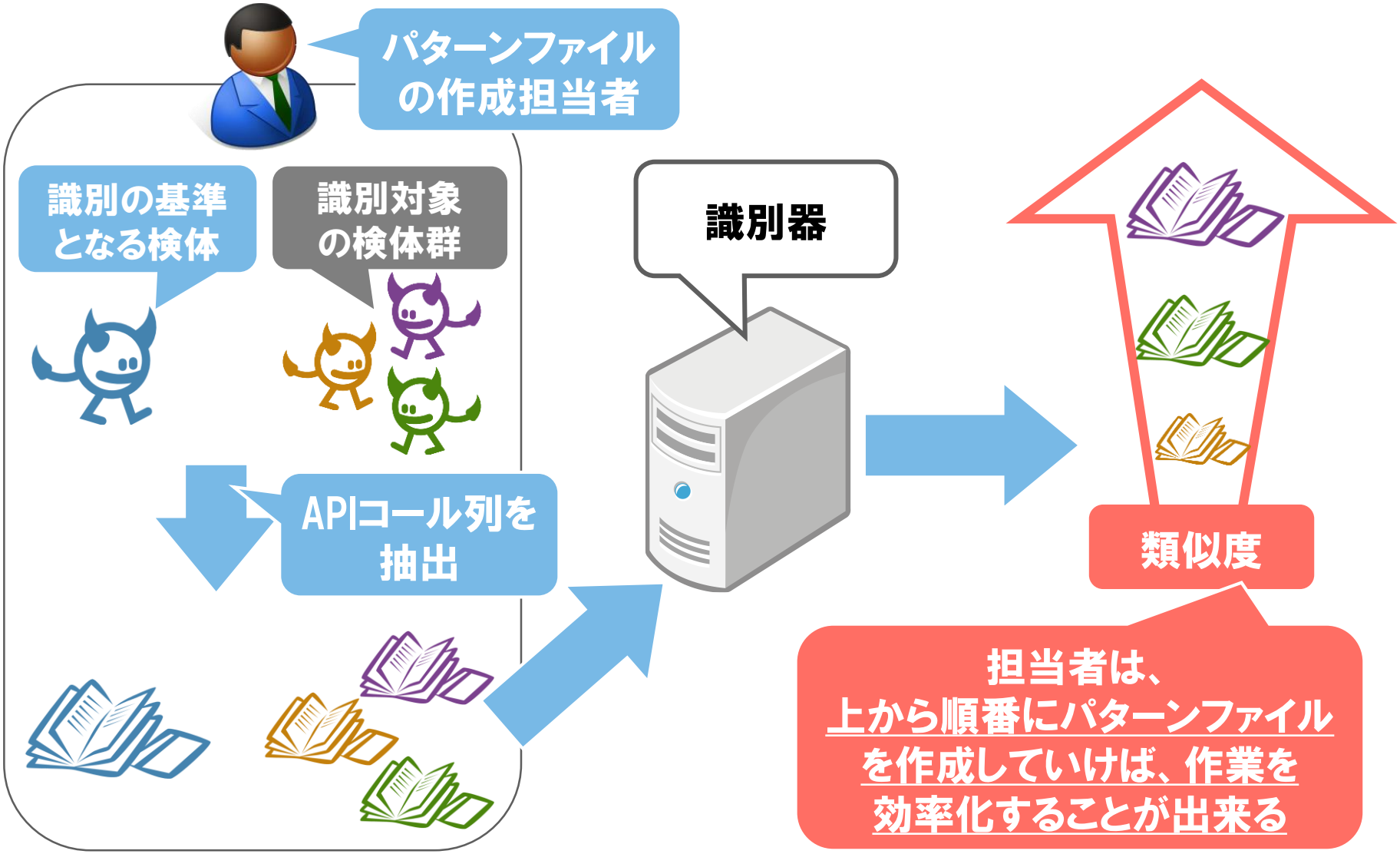
目的と方針

検体同士の類似度を示すシステムを開発し、パターンファイルの作成を補助したい！

- 「基準検体」と「識別対象の検体群」による Windows APIの呼び出し履歴（APIコール列）の「類似度」を判定する識別器を開発
- 類似度の算出によって、検体分析を効率化出来る
  - 未知の検体が既存検体と類似していた場合、当該検体の分析に既存検体の分析結果を活用できるため

① DLLファイルをロード
② レジストリ内のキーをロード
⋮

ユースケース


パターンファイル
の作成担当者
識別の基準
となる検体
識別対象
の検体群
識別器
APIコール列を
抽出
類似度
担当者は、
上から順番にパターンファイル
を作成していけば、作業を
効率化することが出来る

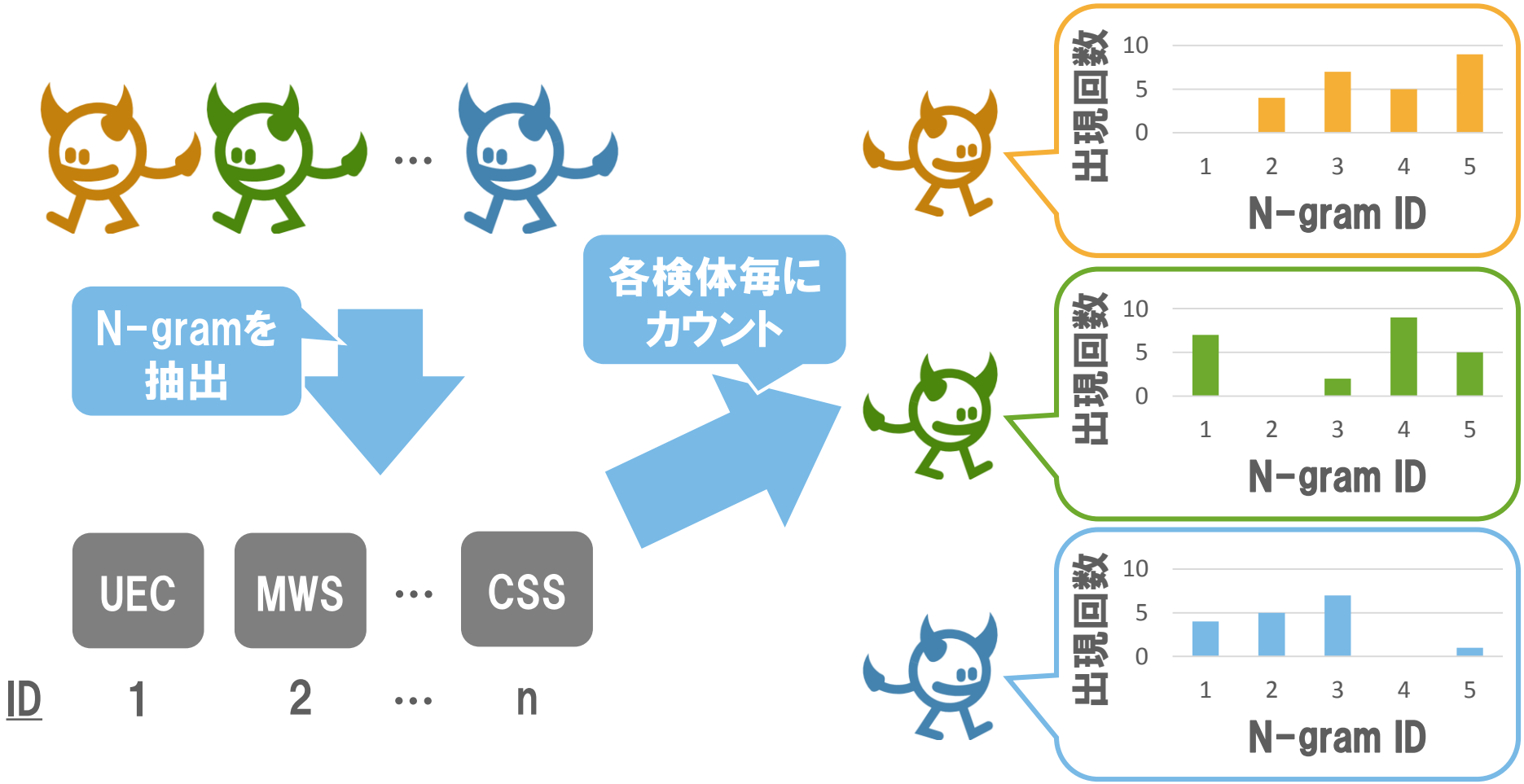
識別器の概要（1/6）

① 全検体のAPIコール列からN-gramを抽出

- N-gramとは？
  - 一塊の情報を，先頭から順に *N* 個の要素（ウィンドウ）群へと分割し直したもの
  - *N* は、ウィンドウサイズと表現される

識別器の概要（2/6）

② 全検体から抽出した各ウィンドウにN-gram IDを振り、各検体に含まれるウィンドウ数をID毎にカウント



識別器の概要（3/6）

③ 各検体内におけるウィンドウの出現数（カウントデータ）をディリクレ分布$Pr(\theta|X)$に置換

- ベイズの公式を利用

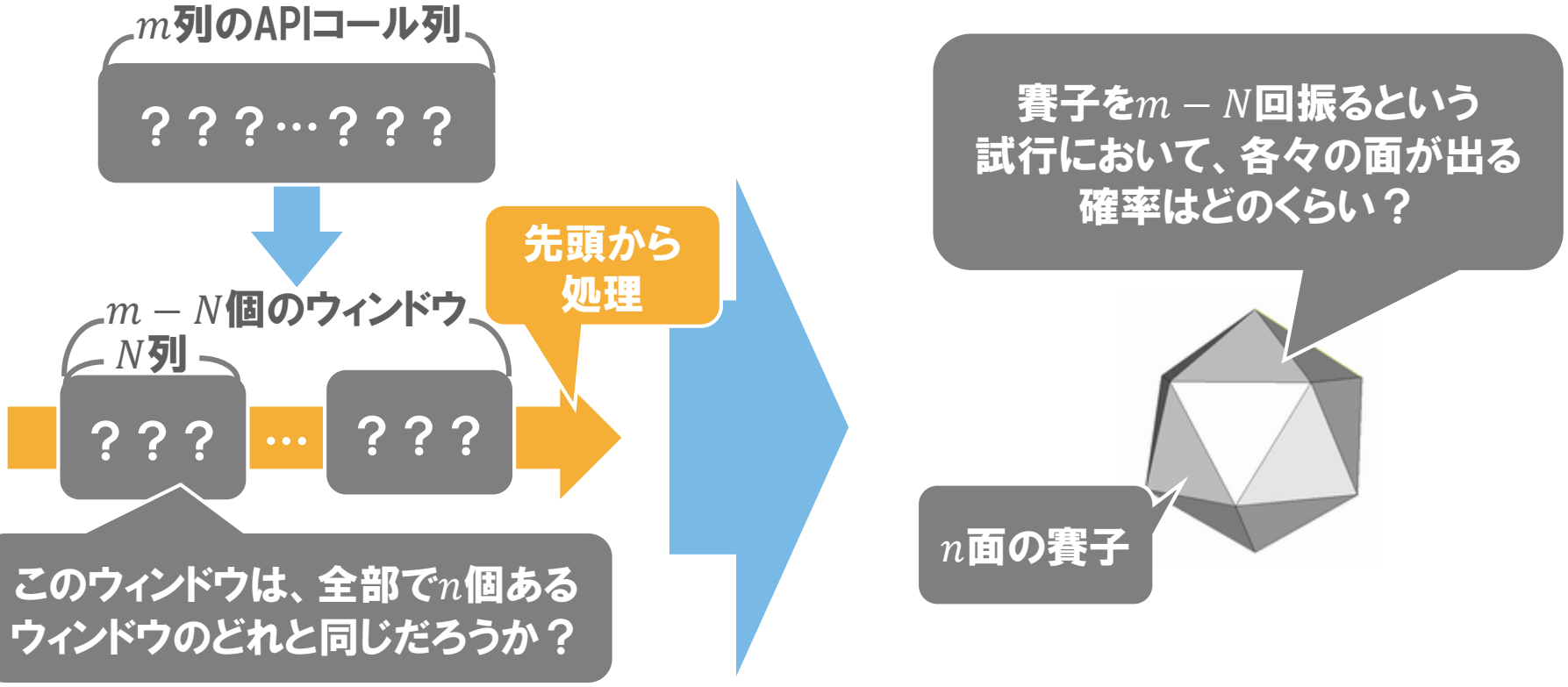
$$Pr(\theta|X) = \frac{Pr(X|\theta)Pr(\theta)}{Pr(X)}$$

$$(X \in [x_1 \ \dots \ x_n],\ \theta \in [0,1])$$

  - 本公式や、関連する数学的議論については省略
  - カウントデータより仮定した多項分布を$Pr(X|\theta)$とする

識別器の概要（4/6）

カウントデータから多項分布を仮定する方法



パターンデータのカウント手法

数学的な問題として再解釈！

識別器の概要（5/6）

③ 各検体内におけるウィンドウの出現数（カウントデータ）をディリクレ分布$Pr(\theta|X)$に置換

- ベイズの公式より求められる分布$Pr(\theta|X)$はディリクレ分布となっている
  - 本分布には、「検体内に各ウィンドウがどの程度含まれているのか」という情報が含まれている
- 本過程を経ることで、1検体につき1つのディリクレ分布が割り当てられることとなる

識別器の概要（6/6）

④ 基準検体及び各識別対象検体の分布間のKullback-Leibler情報量（KL情報量）を導出

- KL情報量は、確率分布Pから同分布Qの統計的距離を表す

$$D_{KL}(P|Q) = \int_{-\infty}^{\infty} p(x) \ln \frac{p(x)}{q(x)} dx$$

  - P, Qの確率密度関数をそれぞれ$p(x), q(x)$とする
  - Pが基準検体、Qが各識別対象検体のディリクレ分布に相当

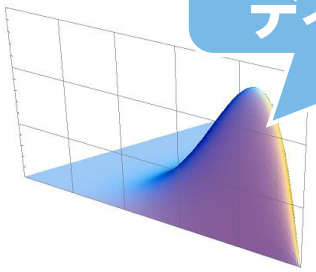
- 検体同士の「類似度」とは、本値のことを指す
  - KL情報量の値が小さい = 類似度が高い

提案手法のフロー

識別器

基準検体のAPIコール列

N-gramの抽出

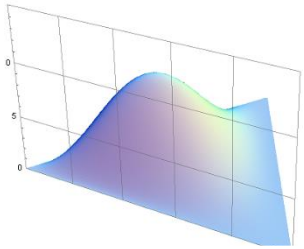


特徴量の出現パターンにディリクレ分布を仮定

識別対象検体のAPIコール列

Kullback-Leibler情報量（KL情報量）を計算

N-gramの抽出



検体	KL情報量
△	23.6
◇	52.6
×	204.9
○	516.0
◎	1365.2
☆	5007.8

1検体ずつ処理

KL情報量が小さい（=類似度が高い）検体から解析を行うことで、効率よく作業を進められる

提案手法に基づく実験（1/4）

検体の識別結果、所要時間、最大メモリ消費量を評価

- 検体の検知率が最良となるウィンドウサイズNも併せて探索
  - $3 \leq N \leq 7$を満たす全てのNについて実験
- 検体（のAPIコール列）には、FFRI Datasets 2013を利用
  - APIコール列、及びアンチウイルスソフトウェアによる検査結果が正しく記録されていない検体は除外
  - 残りの検体の中から基準検体を1つ選択し、他を識別対象検体とした

提案手法に基づく実験（2/4）

出力フォーマット

- 基準検体の「亜種」か「非亜種」を判定し、その結果を出力
  - 実験では、KL情報量をそのまま出力するわけではない点に注意
    - KL情報量の一覧を定量的に評価することは難しいため
  - KL情報量に「閾値」を設定し、KL情報量がそれより小さい検体を「亜種」、大きい検体を「非亜種」と判定

提案手法に基づく実験（3/4）

出力の評価方法

- 4社のアンチウイルスソフトウェアによる検査結果を利用
  - 3社以上のベンダーによる検査結果（科名）が基準検体と一致した識別対象検体を「亜種」と見なす

		識別器による判定結果	
		亜種	非亜種
各識別対象検体の科名が基準検体と一致したベンダー数	4 or 3	True Positive （T-P / 正常判定）	False Negative （T-N / 検知漏れ）
	2	評価から除外	
	1 or 0	False Positive （F-P / 誤判定）	True Negative （T-N / 正常判定）

提案手法に基づく実験（4/4）

実験環境

CPU	Intel Core i7-860 2.8GHz（4core / 8thread）
Memory	DDR3-1600 8GB（4GB * 2 / Dual Channel）
OS	Microsoft Windows 8 Pro（x86_64）
Runtime	.NET Framework 4.5（x86_64）
Programming Language	C#

提案手法の実験結果（1/2）

① 識別結果とウィンドウサイズ N

	ウィンドウサイズ N					入力した検体数
	3	4	5	6	7	
T-P（亜種を正しく検出）	220	225	225	226	**228**	233
F-N（亜種の検知漏れ）	13	8	8	7	5	
T-N（非亜種を正しく検出）	940	953	943	942	933	1,023
F-P（非亜種の誤検知）	83	70	80	81	90	

検知率は $N=7$ の時に最大となり、
その値は 228/233 = 97.85% である

提案手法の実験結果（2/2）

② 識別所要時間

③ 最大メモリ消費量

	ウィンドウサイズ N				
	3	4	5	6	7
識別所要時間（s）	4.57	10.33	16.98	23.44	30.02
最大メモリ消費量（MB）	129.13	290.98	434.70	570.54	688.28

SVMに基づく実験

Support Vector Machine（SVM）に基づく識別器を開発し、提案手法と同様に実験

- 本実験におけるSVMの仕様・設定
  - LIBSVMというライブラリを利用
  - ソフトマージンを設けた上で識別
    - LIBSVM付属のPythonスクリプトでチューニング済
  - カーネルは不使用（線形分離による識別を実施）
    - APIコール列のディリクレ分布は識別に十分な次元数を持つため、より高次元にマッピングする必要はないと判断
- 実験環境は、提案手法に基づく実験と同一

SVMの実験結果（1/3）

① 識別結果とウィンドウサイズN

	ウィンドウサイズN					
	3	4	5	6		
T-P （亜種を正しく検出）	228	228	225	227	**228**	233
F-N （亜種の検知漏れ）	5	5	8	6	5	
T-N （非亜種を正しく検出）	912	946	948	945	9	1,023
F-P （非亜種の誤検知）	111	77	75	78	**81**	

提案手法：**228**

提案手法：**90**

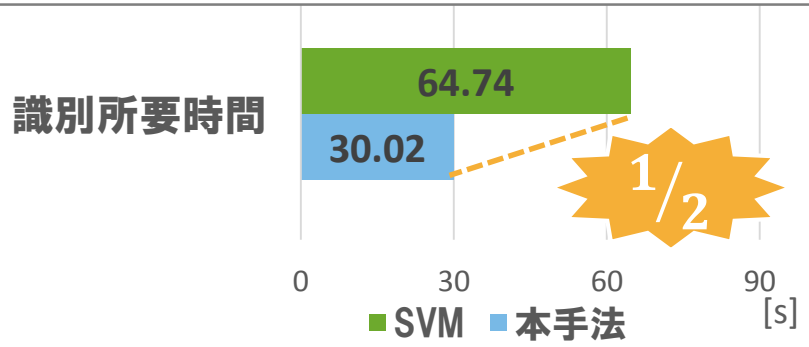
提案手法の検知率はSVMと同一だが、F-P（誤検知）の割合はSVMの方が1割ほど低い

SVMの実験結果（2/3）

② 識別所要時間

		ウィンドウサイズN				
		3	4	5	6	7
所要時間（s）	SVM	9.98	21.82	36.73	50.11	64.74
	提案手法（再掲）	4.57	10.33	16.98	23.44	30.02

Ex) $N = 7$における識別所要時間のグラフ



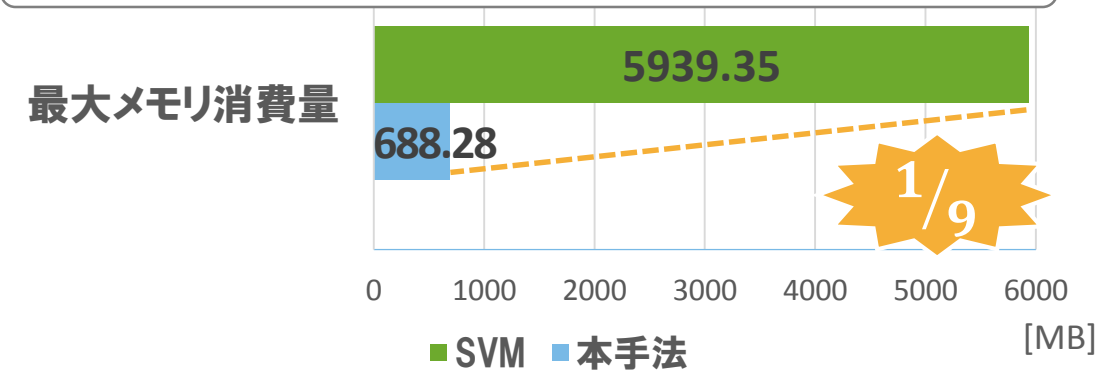
提案手法はSVMの約1/2の時間で識別を行える！

SVMの実験結果（3/3）

③ 最大メモリ消費量

		ウィンドウサイズ N				
		3	4	5	6	7
メモリ消費量（MB）	SVM	1003.26	2702.56	4115.26	5381.70	5939.35
	提案手法（再掲）	129.13	290.98	434.70	570.54	688.28

Ex) $N = 7$における最大メモリ消費量のグラフ



提案手法はSVMの約1/9のメモリ消費量で識別を行える！

関連研究

- システムコール列に着目した異常検知
  - A Sense of Self for Unix Processes（1996）
  - コールスタック検査による Windows 版異常検知システム（2006）
- N-gramを利用したマルウェア検知
  - N-grams-based File Signatures for Malware Detection（2009）

最後に

- まとめ
  - KL情報量を用いた亜種同定システムの考案、及びそれに基づく識別器の開発・評価実験
    - 実験より、97.85%の亜種を識別出来ることが分かった
- 課題
  - 利用機会が必然的に限定される
    - 基準検体を予め決めなくてはならない
  - ⇨ 検体の全自動でクラスタリングする手法を開発