ネットワークトラフィックフローにおける シーケンスパターンに基づくマルウェア分類手法

林　孝英† 山口　由紀子‡ 嶋田　創‡ 高倉　弘喜‡

†名古屋大学大学院 情報科学研究科
464-8601 名古屋市千種区不老町
lim@net.itc.nagoya-u.ac.jp

‡名古屋大学 情報基盤センター
464-8601 名古屋市千種区不老町
{yamaguchi,shimada,takakura}@itc.nagoya-u.ac.jp

あらまし インターネットの発展とともに，マルウェアを用いたサイバー攻撃は重大な社会問題となっている．しかし，C&C サーバとの通信に HTTP などの汎用性の高いプロトコルを使うことが多くなっているため，マルウェア感染を全て検知することが難しくなりつつある．そこで本研究では，マルウェア動作時のネットワークトラフィックについてフローデータのシーケンスパターンを作成し，シーケンスパターンの類似度計算によりマルウェア分類を行う手法を提案する．また，実ネットワークで収集したマルウェアに関するトラフィックフローに対し提案手法を適用した結果と考察を行う．

Network Behavior-Based Malware Classification Method Based on Sequence Pattern of Traffic Flow

Hyoyoung Lim† Yukiko Yamaguchi‡ Hajime Shimada‡ Hiroki Takakura‡

†Graduate School of Information Science, Nagoya University
Furo-cho, Chikusa-ku, Nagoya, 464-8601, Japan
lim@net.itc.nagoya-u.ac.jp

‡Information Technology Center, Nagoya University
Furo-cho, Chikusa-ku, Nagoya, 464-8601, Japan
{yamaguchi,shimada,takakura}@itc.nagoya-u.ac.jp

Abstract With the wide spread of the Internet, cyber attacks which utilizes malwares have become serious social problem. Since the malwares frequently adopt general protocols, e.g., HTTP, to communicate with C&C servers, it becomes impossible to detect all types of malware infection. In this paper, therefore, we propose a malware classification method that focuses on the network behavior of malwares. The behavior is translated into alphabetical sequence pattern. By modifying two dynamic programming algorithms, the behavior is analyzed to find out the most similar malware. We also performed evaluation by using malware traffic collected from the real environment.

1　はじめに

2013 年に McAfee が発表したマルウェア報告書によると，前年一年間に収集された新しいマルウェアは，一日平均 100,000 個以上であった．検知できない高度なマルウェアの存在を考えると，その実数を把握することは不可能である．また，従来のマルウェアは，P2P サービスにより入手した違法コピープログラムを使用したり，電子メールの添付ファイルまたはメッセンジャーのファイルを開いた際に感染していたが，最近では，攻撃者がマルウェアを仕掛けた Web サービスに接続すると感染する水飲み場型攻撃など，その手口が多様化している．さらに，マルウェア作成ツールも高機能化し，難読化・暗号化を行なえるものが簡単に入手できる．このため，マルウェアの感染を完全防ぐことは困難となった．

日々増加するマルウェアと比べて，分析者の人数の少なさを考えると，作業負担を軽減するマルウェア自動分類システムが必須となる．特に，既存マルウェア対策ソフトウェアが検知できない，新種や亜種のマルウェアを早期に発見する手法が求められるようになった．

本研究では，マルウェアを実際に実行させて収集したトラフィックフローから，その特徴を示すシーケンスパターンを生成し，既存マルウェアのパターンと比較を行う分類手法を提案する．提案手法の有効性を示すために，実環境で収集したマルウェアのトラフィックデータを用いて分類精度の評価を行った．その結果，同種ファミリ内では，トラフィックフローのシーケンスに高い類似性が存在することが判明した．一般に，ネットワーク上の挙動が似ているマルウェアは，そのプログラム構造や攻撃者の目的が類似している可能性が高く，高精度なマルウェア分類はサイバー攻撃対策の一つとして高い有効性を持つと考えられる．

2　関連研究

マルウェア分類手法には大きく分けて 2 種類ある．ファイル自体を解析して内部の要素間の関係やプログラム構造を把握する静的解析と，マルウェアを一定の環境下で実行させた際の挙動を観察して解析する動的解析である．

近年のマルウェアでは暗号化や難読化が施されているため，静的解析ではマルウェアとして検知することが困難になりつつあり，マルウェアの挙動に基づいて検知を行う動的解析が有効である [1]．マルウェアの挙動解析は，システム上とネットワーク上に大別される．前者については，マルウェアの動作時に，ファイルやプロセスの変化を記録した挙動のプロファイルを使った研究 [2]，パフォーマンスモニタやシステムコール，システムコールのシーケンスを活用した分類手法 [3] がある．

最近では，後者の挙動解析の重要性が高まり，それに基づいたマルウェア分類手法の研究も進みつつある．トラフィックフローを抽出して，プロトコルごとにグラフを作成し，グラフの類似度でマルウェアを分類する研究 [4]，HTTP のリクエスト情報によりマルウェアを解析する研究 [5, 6] がある．しかし，１つのマルウェアをとっても，ネットワーク上の挙動は多種多様になるため，HTTP のような特定のプロトコルに限定した解析には限界がある．また，パケット一つ一つを解析する断片的な手法では，悪意ある行為全般を俯瞰した類似度分析を行えない．一方で，セッション中の文字列やバイナリ列を逐次解析する手法では，高速な分類が行なえず，実用性に問題がある．

従って，１つのマルウェアが起こすトラフィック全体を特徴化し，それに基づいて，マルウェアの分類を行う新たな手法が必要である．

3　提案するマルウェアの分類手法

提案手法では，動的解析を介して得られたマルウェアのネットワークトラフィックフローからシーケンスデータを抽出し，シーケンスデータの類似度を計算することで，ネットワーク活動から同種ファミリを算出する．

提案手法は以下の 4 段階で構成される．

1. 特徴抽出
2. クラスタリング
3. シーケンスデータの作成
4. 類似度計算および分類

表 2: フローデータの例

Dur	Seq	Proto	SrcAddr	DstAddr	Sport	Dport	Dir	State	TotPk
2.99995	1	RARP	00:0c:29:89:7d:fa	00:0c:29:89:7d:fa	-	-	who	INT	2
0.00000	2	IGMP	10.0.0.0	224.0.0.1	-	-	→	INT	1
0.00033	3	ARP	192.168.1.1	192.168.1.2	-	-	who	CON	2
0.75628	4	UDP	192.168.1.2	10.0.0.1	1037	53	↔	CON	2
0.00350	5	TCP	192.168.1.2	*.*.*.158	1035	80	→	CON	5
0.00071	5	TCP	192.168.1.2	*.*.*.158	1035	80	→	RST	5
0.00005	6	UDP	192.168.1.2	192.168.1.255	138	138	→	INT	3
0.00013	6	UDP	192.168.1.2	192.168.1.255	138	138	→	REQ	3

表 1: 提案手法で用いる特徴量

特徴	説明
Dur	record total duration
Seq	argus sequence number
Proto	transaction protocol
SrcAddr	source IP address
DstAddr	destination IP address
Sport	source port number
Dport	destination port number
Dir	direction of transaction
State	transaction state
TotPkts	total transaction packet count
SrcPkts	src → dst packet count
DstPkts	dst → src packet count
SrcLoad	source bits per second
DstLoad	destination bits per second

3.1 特性抽出

マルウェアの実行時にキャプチャしたトラフィックデータからフローデータを抽出する．

一般に，フローは5-tuple情報(SrcIP，SrcPort，DstIP，DstPort，Transport Layer Protocol)が同じで一方向のパケットのセットと定義される．フローベースのトラフィック分析手法より，多くの特徴を対象とすることで様々な分析が可能であるという長所がある．

同じツールによって生成された同一の目的を持つマルウェアは，元のプログラムコードが異なっていても，部分的に似たような通信を行うことが推測される．したがって，マルウェアが関係するフローデータを元にマルウェアを分類することで有益な情報が得られると期待される．

本手法ではArgus[1]を使い，ネットワークトラフィック中の各フローの特徴量を表すフローデータを生成する．表1に，提案手法で用いるフローデータの特徴量を示す．フローデータは，前述の5-tuple情報に加え，フローの継続時間(Dur)，シーケンス番号(Seq)，転送方向を表すDir情報，フローの状態(State)，パケット数(TotPk，SrcPk，DstPk)，秒あたりのビット数(SrcLoad，DstLoad)の14種の特徴量から構成される．

表2にフローデータの例を示す．SeqはArgusがフローデータの生成順に割り当る通し番号である．Dirは Src，Dst，両方，MACアドレス解決(who)等を表す．Stateは初期化状態(INT)，接続状態(CON)，リセット(RST)，リクエスト(REQ)等を表す．なお，文字属性で表される特徴量はクラスタリングに適するように，数値属性に変換した．

本手法では，マルウェアの活動と関連性が高いと思われているTCPとUDPプロトコルを抽出したフローデータを用いた．

3.2 クラスタリング

特徴抽出によって得られたフローデータをクラスタリングし，フローを分類する．本稿では，分類手法として，幅広く使われている分割クラスタリングアルゴリズムのK-meansを用いる．K-meansアルゴリズムは，与えられたデータをk個のクラスタに分類し，各クラスタに割り当てられたデータと当該クラスタの中心との距離の分散を最小化するように動作する．i番目のクラスタの中心をμ_i，各クラスタに属する点の集合をS_iとするとき，全体の分散Vは式(1)のように計算される．

$$V = \sum_{i=1}^{k} \sum_{j \in S_i} |x_j - \mu_i|^2 \qquad (1)$$

[1]http://qosient.com/argus/

表 3: クラスタリングの結果 (k=8)

クラスタ	A	B	C	D	E	F	G	H	合計
フロー数	3,810	15,545	3,265	3,827	2,541	4,665	12,615	8,342	54,610
割合 (%)	6.9	28.5	6.0	7.0	4.6	8.5	23.1	15.2	100

表 4: シーケンスデータの例 (k=8)

マルウェア名	シーケンス
Backdoor.Win32.Androm.bndf	BHBGGBGGBHDBHBHBHBHBHDBHBHBHBHB...
Backdoor.Win32.Androm.bnlt	BHBHBHBHDBHBHBHBHBHCBHBHBHBHBHB...
Backdoor.Win32.Androm.bwvx	DBHHHHHDHHHHHHHDHHHCHHHHCBAADBH...
Backdoor.Win32.DarkKomet.bggv	DCDBHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH...
Backdoor.Win32.DarkKomet.bvxu	DDBHCCDCBAABHDAAAADDDCDBC
Backdoor.Win32.DarkKomet.cfcd	DDBHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH...

以下に，提案手法で用いたアルゴリムの概要を示す．まず，k 個のクラスタを任意に選定する．次に，分類対象のデータ毎に，各クラスタの中心までの距離を求め，中心に最も近いクラスタへ割り当てる．全てのデータの割り当てを完了したら，各クラスタに割り当てられたデータの平均値を求め，それぞれの中心 μ_i として定義し直し，新たな μ_i に対して再び分類対象のデータの割り当てを行う．その結果，クラスタに割り当てられるデータに変化が無い，あるいは，変化が閾値以下であればアルゴリズムを終了させる．

クラスタリングの例として，515 個のマルウェアに関するネットワークトラフィックから抽出した 54,610 個のフローデータを，K-means アルゴリズムによって 8 個のクラスタに分類した結果を表 3 に示す．なお，この例では，各クラスタに $A \sim H$ のラベルを割当てている．

3.3 シーケンスデータの作成

クラスタリング結果に基づき，マルウェアのフローデータに対応するシーケンスデータを作成する．

表 4 に，k=8 で作成されたシーケンスの例を示す．表から読み取れるように，同じマルウェアファミリに属するマルウェアのフローでは類似したシーケンスが得られる可能性がある．すなわち，シーケンスパターンの類似度に基づくマルウェア分類が期待できる．

3.4 類似度計算および分類

バイオインフォマティクス分野では，DNA 配列 (シーケンス) が類似すれば，その化学的特徴により類似した構造を持つことになり，これによって類似する生物学的機能を有することが知られている．このため，シーケンスの分析において，2 つのシーケンスを比較し，それらの共通部分の配置を求めるシーケンスアライメントアルゴリズムが用いられている．

既存のシーケンスアライメント手法として，動的計画法の一種である Needleman-Wunsch[7] や Smith-Waterman[8] アルゴリズムが提案されている．このアルゴリズムは一致部と不一致部 (ギャップ) にそれぞれ重みを与える形で求めたスコアの最大値を利用し，シーケンスの共通部分列を判断するもので，複数のギャップが含まれていても，類似する共通部分列を検出できる．

Needleman-Wunsch アルゴリズムはシーケンス全体の類似性を調べることが目的である．2 つのシーケンスを $X = \{x_1, x_2, \ldots, x_{n-1}, x_n\}$ と $Y = \{y_1, y_2, \ldots, y_{m-1}, y_m\}$ とすると X_i と Y_j のスコア $N_{i,j}$ は式 (2) によって求められる．

$$N_{i,j} = max \begin{cases} N_{i-1,j-1} + S\left(X_i, Y_j\right), \\ N_{i-1,j} + gap, \\ N_{i,j-1} + gap. \end{cases} \tag{2}$$

$$S\left(X_i, Y_j\right) = \begin{cases} match & \left(X_i = Y_j\right), \\ mismatch & \left(X_i \neq Y_j\right). \end{cases} \tag{3}$$

表 5: シーケンス生成アルゴリズムによる類似度の違い (k=8)

マルウェアのファミリ名	サンプル数	W 法 (%)	N 法 (%)	A 法 (%)
Backdoor.Win32.Androm	4	64.2	58.3	61.3
Backdoor.Win32.DarkKomet	5	67.6	23.7	45.7
Backdoor.Win32.Simda	4	72.7	42.4	57.5
Trojan.Win32.Agent	5	59.4	32.0	45.7
Trojan.Win32.Inject	7	59.0	34.2	46.6
Trojan.Win32.VB	3	85.1	37.7	61.4
Trojan-Downloader.Win32.Agent	2	60.9	43.7	52.3
Trojan-PSW.Win32.Tepfer	2	54.2	13.1	33.7
Trojan-Spy.Win32.Zbot	9	70.3	43.2	56.7
Virus.Win32.Sality	2	91.2	91.0	91.1

Smith-Waterman アルゴリズムは 2 つのシーケンスの長さが異なる場合でも，最も高いスコアを出す共通部分列を見つけることができる．Smith-Waterman でのスコア $W_{i,j}$ は式 (4) で求められる．$S(X_i, Y_j)$ は Needleman-Wunsch アルゴリズムと同じである．

なお，式 (4) に示した通り，Smith-Waterman では共通部分列の探索途中でスコアがマイナスになった場合，スコアを初期値 0 にリセットした後，その位置から探索を再開するアルゴリズムである．従って，2 つのシーケンス内で最も類似するシーケンス区間を探すローカルアライメントに適している．

$$W_{i,j} = max \begin{cases} W_{i-1,j-1} + S(X_i, Y_j), \\ W_{i-1,j} + gap, \\ W_{i,j-1} + gap, \\ 0. \end{cases} \quad (4)$$

スコアパラメータ $match$，$mismatch$，gap は定数であり，本提案手法ではそれぞれ，10，-5，-5 に設定した．

提案手法では，これらのシーケンスアライメントアルゴリズムを応用し，シーケンスの類似度から，マルウェアファミリの分類を行う．類似度の計算は，異なる視点から判断するために Needleman-Wunsch と Smith-Waterman の両方と，その平均値を用いて行った．

Needleman-Wunsch の類似度を 類似度$_N$ と，Smith-Waterman の類似度を 類似度$_W$ とすると，式 (5) より，平均の類似度 類似度$_A$ を求める．

$$\text{類似度}_A = \frac{\text{類似度}_N + \text{類似度}_W}{2} \quad (5)$$

4 検証実験

本節では，提案手法の有効性を検証するため，実環境において採取されたマルウェアのトラフィックデータを用いて評価を行った．

4.1 実験データ

実験は，NTT のマルウェア動的解析システム Botnetwatcher[9] において，2013 年 10 月から 2014 年 3 月にかけて収集されたマルウェアの動的解析結果を用いて行った．Botnetwatcher は開放されたネットワーク環境でマルウェアを実行させ，その通信監視により，送受信パケットを収集するシステムである．本研究では，収集されたパケットの内，Botnetwatcher 自身が関係する通信を除去したものを実験データとした．

実験では，市販のマルウェア対策ソフトウェア (Kaspersky[2]) で分類されたファミリを比較対象として評価する．Botnetwatcher から収集した 515 個のトラフィックキャプチャファイルのうち，Kaspersky でいずれかのファミリに分類された 450 検体を対象とした．

450 個のマルウェアは Kaspersky で 38 種のファミリに分類された．このデータから，各シーケンスデータを作成，シーケンスの類似度を算出し，比較を行った．

4.2 マルウェアファミリ間の類似性

表 5 にファミリ名とサンプル数，同種ファミリ内でのシーケンスデータの類似度を比較した

[2]http://www.kaspersky.com/

表 6: BA と TA の類似度計算結果 (%，k=8)

	BA 1	BA 2	BA 3	TA 1	TA 2	TA 3
BA 2	81.2	-	-	-	-	-
BA 3	52.5	50.1	-	-	-	-
TA 1	42.7	42.7	41.7	-	-	-
TA 2	36.5	30.9	34.3	31.1	-	-
TA 3	42.1	37.8	38.6	53.3	37.8	-
TA 4	40.6	41.4	43.6	53.8	30.4	57.4

BA:Backdoor.Win32.Androm
TA:Trojan.Win32.Agent

表 7: BA と TR の類似度計算結果 (%，k=8)

	BA 1	BA 2	BA 3	TR 1	TR 2	TR 3
BA 2	81.2	-	-	-	-	-
BA 3	52.5	50.1	-	-	-	-
TR 1	45.2	36.5	40.3	-	-	-
TR 2	41.2	38.8	38.7	73.5	-	-
TR 3	40.1	38.8	46.0	83.7	73.4	-
TR 4	39.9	37.6	40.5	86.8	78.7	85.2

BA:Backdoor.Win32.Androm
TR:Trojan-Ransom.Win32.Agent

結果の一部を示す．Smith-Waterman(W 法) と Needleman-Wunsch アルゴリズム (N 法)，および，2 つの平均 (A 法) を用いて同種マルウェア間の類似度を求めた．その結果，多くのファミリにおいて，シーケンスに高い類似性が存在することを確認した．

表 6，7 にファミリ Backdoor.Win32.Androm (BA) と Trojan.Win32.Agent(TA)，および，BA と Trojan-Ransom.Win32.Agent(TR) の A 法による類似度を示す．同種ファミリ内の類似度は表中の背景をグレーとしてある．

BA はトロイの木馬型のマルウェアであり，リモートサイトから他のマルウェア，グレイウェアまたはスパイウェアをダウンロードし，コンピュータにインストールするものである．また，TA はシステムの特定のファイルを削除したり，新しいファイルを作成する機能を有する．

表 6 の TA をみると同種ファミリ内での類似度が低いことが分かる．しかし，表 8 に示すように，他のマルウェア対策ソフトウェア (AVG[3]) の診断結果では，TA の 4 つのマルウェアは全て異なるファミリと判定されており，実際には挙動の違いが大きめである可能性が高い．また，マルウェアの命名規則に基づけば，それぞれの機能も異なる事が予想される．

一方，表 7 の結果では，同種ファミリ内での類似度は非常に高く，異種ファミリ間での類似度は低いことが分かる．同様に AVG で TR ファミリを調べたところ，Trojan.Generic35 という 1 つのファミリとして判定されている．よって，提案手法は複数のマルウェア対策ソフトウェアが同種

表 8: TA ファミリの AVG による分類

TA 1	Trojan.Inject2.MDE
TA 2	Trojan.CoinMiner.AKQ
TA 3	Trojan.Dropper.Generic_r.AF
TA 4	Win32/DH{IBMDBmcJDOQ}

ファミリと判定をしている場合は類似度が高くなり，それぞれが異なる判定をしている場合は類似度が低くなると言える．

4.3 マルウェア分類の有効性検証

同じマルウェアであっても，毎回同じトラフィックシーケンスを生成する訳ではない．そこで，本手法のマルウェア分類に関する有効性を検証するため，以下の実験を行った．

まず，実験データ 450 個を時間順に並べる．そのうち最新のものから 100 個，すなわち，351 から 450 番目までのマルウェアを未分類のものとし，順番に分類を行う．なお，n 番目のマルウェアを分類する際には，1 から $n-1$ 番目までのマルウェアとの類似度計算を行う．その中から類似度の高い順に 1 から 50 位までのマルウェアを求め，それぞれのマルウェアのファミリの類似度順位とした．

表 9 に実験結果の一部を示す．表 9 中のファミリ番号と類似度順位は 4.1 節で述べた Kaspersky によって分類された 38 種のファミリを表す．表より，3 個のマルウェアが正しく分類できていないことが分かる．Backdoor.Win32.Androm.A はそれ以前 ($n-1$ 番目) までに出現していたにもかかわらず正しく分類ができなかった．一方，Backdoor.Win32.Simda と Packed.Win 32.Katusha.o

[3]http://www.avg.co.jp/

表 9: 類似度における順位の例 (%，k=8)

		類似度における順位							
マルウェア名前	ファミリ番号	1 位	2 位	3 位	4 位	5 位	6 位	7 位	8 位
Backdoor.Win32.Androm.A	12	8	7	23	36	4	29	16	16
AdWare.NSIS.Agent	7	7	29	12	16	8	4	9	23
Backdoor.Win32.Androm.B	12	12	27	30	16	24	24	8	19
Backdoor.Win32.Simda	6	35	35	24	35	24	24	11	24
Backdoor.Win32.Androm.C	12	12	8	7	7	23	29	4	16
Packed.Win32.Katusha.o	15	6	24	30	30	24	24	30	30
Trojan-Spy.Win32.Zbot	36	36	23	12	8	12	16	24	24

はそれ以前 ($n-1$ 番目) までには存在しなかった初出のマルウェアである．100 個中に，このような初出マルウェアは 28 個存在した．以降の評価では，このような初出マルウェアは分類精度の計算から除外する．

次に，初出マルウェアを除いた，72 マルウェアに対し，式 (6) で定義する分類精度を求めた．その結果を表 10 に示す．さらに，上位 i 位 (Top$i$) までに正解が含まれる場合の分類精度を図 1 に示す．以上の結果から，上位 5 位までの分類であれば，提案手法 (A 法) が最も高い分類精度となることが分かった．

$$\text{分類精度} = \frac{\text{正しく分類された既知マルウェアの数}}{\text{既知マルウェアの総数}} \tag{6}$$

表 10: アルゴリズム別分類精度 (%，k=8)

	W 法	N 法	A 法
分類精度	28.9	42.5	46.8

W 法と N 法の分類精度差は，それぞれのシーケンスアライメントの特徴の違いによるものと考えられる．W 法のアルゴリズムは長さに大差があっても，2 つのシーケンスから最も類似した共通部分列を見つけることができる．そのため，違う長さを持つシーケンスに適用すると分類性能が高くなる．一方，N 法のアルゴリズムはシーケンス全体を対象に共通部分列を探索する，長さが同程度のシーケンスに適用すると分類性能が高くなる．提案手法では，2 つアルゴリズムの長所を生かし，それらの平均を求めることで，分類精度を向上させることができたと言える．

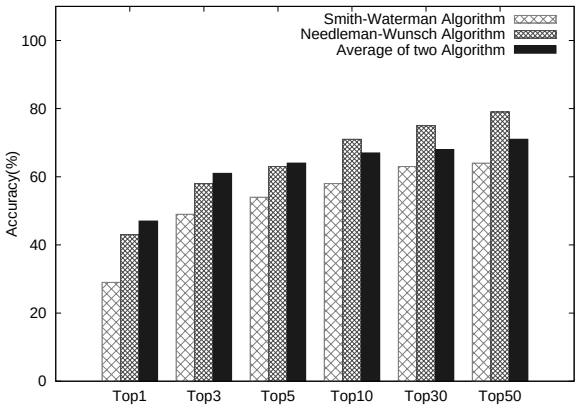

図 1: アルゴリズムにおける分類精度の変化

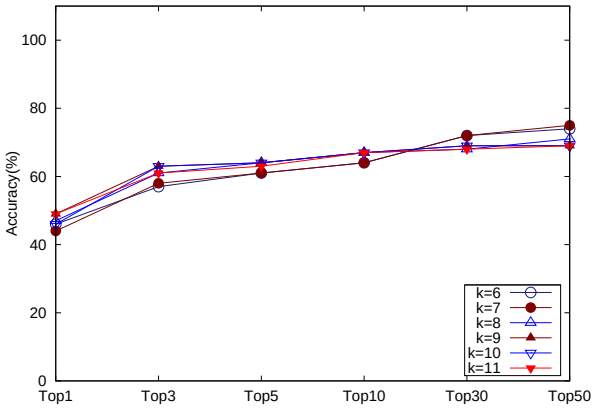

図 2: クラスタ数 (k) における分類精度の変化

4.4 考察

実験により，マルウェアのトラフィックフローからマルウェア分類が可能であることが分かった．ここではクラスタリングのパラメータ k による安定性と実験結果について考察する．

そこで，クラスタ数による影響を見極めるため，k を 6 から 11 まで変化させた結果を図 2 に示す．図の横軸は順位結果を基にして 1 位から 50 位までを，縦軸は分類精度をそれぞれ表している．図から，実験した範囲内であれば，k による影響は軽微であると判断した．以上より，本稿が提案するクラスタリング手法によるマルウェアの分類は，非常に安定しておりかつ効果が高いと考えられる．

一方，4.3 節より，提案手法でも分類精度が 46.8% と決して高いものではない．これは，実験に用いたマルウェアのサンプル数が十分ではなく，クラスタリングがやや大雑把になってしまった点，および，フローデータの特徴のうち文字属性を数値属性に置き換えたが，その大小に意味が存在しない点が原因であると考えている．

このため，さらに大規模なマルウェアサンプルを用いた性能評価と文字属性の特徴量を適切に扱うクラスタリング手法の開発が必要になる．

5 おわりに

本稿では，ネットワークトラフィックのフローシーケンスパターンを用いたマルウェアの分類手法を提案した．本手法は，既存マルウェアファミリのネットワーク上で挙動 (シーケンス) と未分類のマルウェアの挙動を比較することにより，最も類似したマルウェアファミリを推定することが可能となる．

また，提案手法では，DNA シーケンスの共通部分列判定を行う Smith-Waterman と Needleman-Wunsch の両アルゴリズムを応用し，その両者の利点を活用する類似度計算法を採用した．これにより，前者で 28.9%，後者で 42.5%だった分類精度を 46.8%まで向上させることができた．

今後の課題として，より大規模なデータセットによる精度の検証と処理時間の向上が挙げられる．また，未知のマルウェアのシーケンスパターンに対する分類において，既存のマルウェアのネットワーク上の挙動との差異や類似点を明らかにしたいと考えている．

謝辞

本研究は平成 25 年度総務省情報通信技術の研究開発「サイバー攻撃の解析・検知に関する研究開発」の支援を受けている．また，助言とデータセットを提供して頂いた NTT セキュアプラットホーム研究所の八木氏に深く感謝する．

参考文献

[1] K. Rieck, et al, “Learning and classification of malware behavior”, Detection of Intrusions and Malware, and Vulnerability Assessment, pp. 108-125, 2008.

[2] U. Bayer, et al, “Scalable, Behavior-Based Malware Clustering”, NDSS, Vol. 9, 2009.

[3] R. Canzanese, et al, “Toward an Automatic, Online Behavioral Malware Classification System”, SASO 2013, 2013.

[4] S. Nari and A. A. Ghorbani, “Automated malware classification based on network behavior”, ICNC 2013, pp. 642–647, 2013.

[5] R. Perdisci, W. Lee, et al, “Behavioral Clustering of HTTP-Based Malware and Signature Generation Using Malicious Network Traces”, NSDI, pp. 391–404, 2010.

[6] M. Z. Rafique, et al, “Evolutionary algorithms for classification of malware families through different network behaviors”, GECCO 2014, pp. 1167–1174, 2014.

[7] S. B. Needleman and C. D. Wunsch, “A general method applicable to the search for similarities in the amino acid sequence of two proteins”, Journal of molecular biology, vol. 48, no. 3, pp. 443–453, 1970.

[8] T. F. Smith and M. S. Waterman, “Identification of common molecular subsequences”, Journal of molecular biology, vol. 147, no. 1, pp. 195–197, 1981.

[9] 青木一史ら, “半透性仮想インターネットによるマルウェアの動的解析”, CSS 2009, 2009.