2012 年度修士論文

トラヒックデータを対象とした N-gram 確率密度を用いた マルウェア感染検知手法に関する研究

指導： 甲藤　二郎 教授
小松　尚久 教授

2013 年 2 月 8 日

早稲田大学理工学術院 基幹理工学研究科 情報理工学専攻

5111B031-5　　川元　研治

目次

第 1 章

序論

1.1 本研究の背景

情報通信技術が発展するに伴い，インターネットが我々の生活に浸透し世の中に欠かせないものとなっている．具体的には WorldWideWeb や電子メールなど人々のコミュニケーションツールとして利用され，金融や経済などのビジネス面や電力や水道といった生活インフラの基盤の一部も担うようになっている．

ユーザの利便性が向上する一方で，インターネットを悪用するセキュリティの問題が発生している．第三者からの悪意のある活動により，個人や企業の持つ情報資産に様々な被害が及ぶようになった．例えば，他者の PC に侵入することで情報資産の破壊や改ざん，盗難を行う不正アクセス，多大な通信負荷をかけることによりネットワークやサーバを圧迫することで意図した動作をさせない DoS 攻撃 (Denial of Service) などが挙げられる．

同時にこれらの活動をユーザに認識させずに実行させる悪質なプログラムが作成されてきた．これらは悪意のあるソフトウェア (Malicious Software) の略称からマルウェアと呼ばれている [1]．マルウェアに感染すると個人情報の流出や PC の乗っ取りなどの被害を受ける可能性があるため，我々の生活を脅かす存在となっている．

2012 年度の日本国内での被害報告件数はたった 5 種類の不正プログラムのみで 1000 件を超えている [2]．さらに，マルウェア感染の手口は年々複雑化，多様化しており，近年は活動が表面化しにくいボットネットによる被害の増加や Gumblar に代表される Web からの感染も急激に増えている．最近では，スマートフォンの普及に伴い，Android 端末を対象にしたモバイル端末に感染する不正プログラムが急増している．2012 年度における Android 端末に感染する不正アプリ数を図 1.1 に示す．Android 端末の他にも，自動車の車載システムのセキュリティを脅かすマルウェアなども存在する [3]．

また新種のマルウェアは日々増加しているという状況である．年度ごとの新種マルウェアの発生数の動向を図 1.2 に示す [4]．図 1.2 の通り近年は急激な増加の傾向が見られる．2010 年度には年間 200 万種のマルウェアが発生しており、これは 15 秒毎に新種のマルウェアが発生していることになる．新種のマルウェアが爆発的に増加していくと，現在の主流のマルウェア対策であるシグネチャベースによる対策では，対応しきれなくなってしまう．

上記のように，マルウェアは日々進化しており，その被害も大きくなっているため，インターネット上における被害を最小限に食い止めるためにもマルウェアへの対策をより一層強める必要がある．

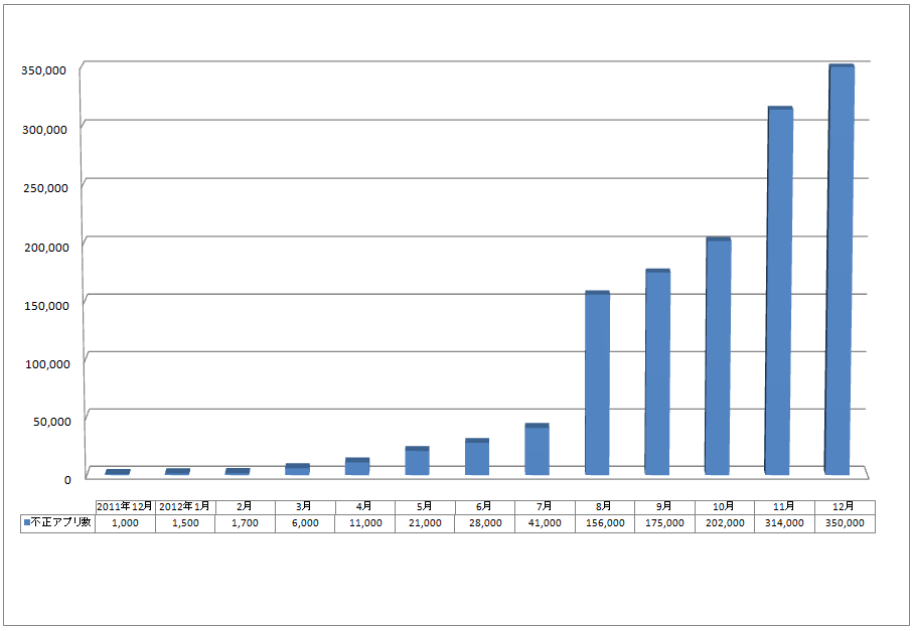

図 1.1　Android 端末に感染する不正アプリ数 (2012 年度累計)

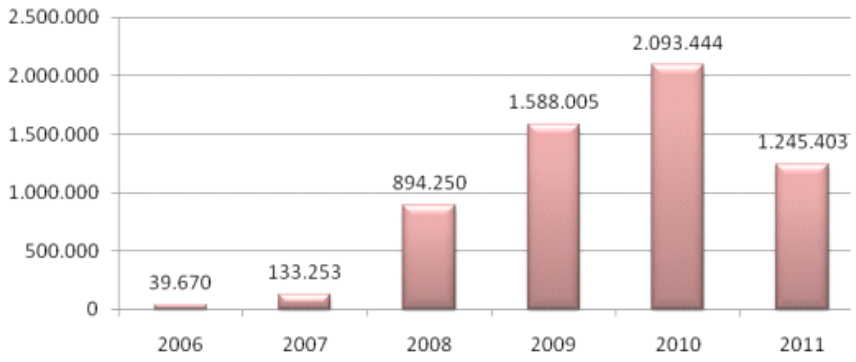

図 1.2　新種マルウェア数の変遷 (2006〜2011)(2011 年は上半期のみ)(横軸: 年度)

1.2　本研究の目的

本研究の最終的な目標は，ルータやサーバを通過する通信を監視し，異常が見られた際にはその通信を棄却したり，マルウェアに感染した PC やサーバのユーザ，管理者にいち早くマルウェアによる感染の通知を行うことで，安心・安全なネットワークを実現することである．

しかしながら，マルウェアは日々複雑な振る舞いをするようになっており，例えばボットなどは感染しても表面化せず潜伏するため，気づきにくいという特徴がある．さらにマルウェアに感染してしまうとネットワーク経由で感染を拡大してしまうという問題がある．

現在のマルウェア対策の主流であるウイルス対策ソフトは，既知のマルウェアの検知が中心であり，マルウェア毎の特徴を示すシグネチャを用意することで検知を行うので，短期間で大量に出現するマルウェアの新種や亜種に対応しきれないという課題がある．過去のマルウェアの挙動から未知のマルウェア*による感染を予測で検知する手法も存在するが，誤検知が起きてしまう場合がある [5]．

そこで，本研究では，インターネットを流れるトラヒックデータに着目して，誤識別なく素早くマルウェアによる感染の有無を判定し (マルウェア感染検知)，マルウェアによる感染の拡大を防ぐことを目指す．感染の拡大を防ぐために，入口対策として侵入検知という分野があるが，攻撃の手口，およびマルウェアの種類の多様化により入口に対策を講じることが難しくなっているため，本研究では一種の出口対策である感染検知に着目する．

トラヒックデータに着目する理由は 2 つある．1 つ目は，トラヒックデータを用いることで PC の外で感染検知が行えるからである．最近のマルウェアはウイルス対策ソフトを無効化してしまうマルウェアも存在するので [6]，PC の外でも感染検知が行えることは重要である．2 つ目は，トラヒックデータには時間的な変化があり識別器に対して連続的な入力が可能だからである．時間的な変化に着目することで，正常な通信とマルウェアによる通信を識別できる可能性があり，さらに，ウイルス対策ソフトでは難しい未知のマルウェアにも対応できる可能性がある．例えばバイオメトリクスでは，発話時の唇動作個人認証において複数のアルゴリズムが提案されているが，時系列を使用しないアルゴリズムと使用するものを比較した際，後者のアルゴリズムのほうが高い精度での認証が可能であることが示されている [7]．

トラヒックデータを用いる際に，時間的な変化に着目することは重要だと考えている．それは，近年のマルウェアによる感染時の通信は正常な通信と区別しにくいという課題を克服するためである [8]．例えば，マルウェア検体のダウンロード通信に着目すると FTP によるものや，ボットに感染した際の指令の通信に着目すると IRC を用いるものなどがあり，一般的なユーザのインターネット利用時の通信と区別がしづらくなっている．つまりマルウェアに感染した時の特徴のみに着目するだけで，感染を正しく判定することは難しくなっている．

* ウィルス対策ソフトの定義ファイル (パターンファイル) に定義されていないマルウェア

そこで，本研究ではまず「感染時通信の特徴との類似性」「正常時通信からの逸脱性」という 2 つのアプローチから感染検知を考える．正常な状態の通信のモデルとマルウェアに感染した際の通信のモデルを用いることで，誤検知の少ない感染検知を行うことを目指す．そして，各通信の時間的な変化をモデル化することで精度の高い識別が可能であると考えている．

また時間的な変化のモデル化を行う際に，どのようなトラヒックデータの特徴に着目するかの検討も重要である．正常なトラヒックデータとマルウェアに感染した後のトラヒックデータにはどこかに違いがあるはずであり，その違いを明確にしておくことは重要である．

すなわち，本研究では，まず正常なトラヒックデータとマルウェアに感染した後のトラヒックデータにおいて，違いが現れるような特徴を探す．その後，正常なトラヒックデータとマルウェアに感染した後のトラヒックデータの時間的な変化に着目し，マルウェア感染検知を行う．

本論文では，まず，パケットのヘッダー情報から得られる特徴量に対してマルウェア感染検知における有効性の評価を行う．その後，マルウェア感染検知に有効な特徴量を用いて，トラヒックデータを対象とした N-gram 確率密度を用いたマルウェア感染検知手法を提案し，その有効性について識別精度とともに報告する．

1.3　本論文の構成

本論文は以下の構成で書かれている．

■第 1 章「序論」
本研究の背景および目的について述べた．

■第 2 章「マルウェアと対策技術」
マルウェアの基本的な性質や特徴，変遷について述べる．また現在行われている対策技術について述べ，異常・感染検知技術についての関連研究および従来手法について説明する．

■第 3 章「マルウェア感染検知のための経年変化を考慮した特徴量評価」
マルウェア感染検知に有効な特徴量について述べる．経年変化を考慮した特徴量評価が必要な理由を述べた後に, 特徴量評価方法, およびマルウェア感染検知に有効な特徴量について述べる．

■第 4 章「N-gram 確率密度を用いたマルウェア感染検知」
本論文の提案手法である，トラヒックデータを対象とした N-gram 確率密度を用いたマルウェア感染検知手法について述べる．提案手法の要素技術である N-gram, 最近傍密度推定法について説明する．

■第 5 章「マルウェア感染トラヒック検知実験」
提案手法の有効性を確認するための検知実験について述べる．実験の概要，提案手法を評価するための比較手法について述べた後に，考察を述べる．

■第 6 章「結論」
「トラヒックデータを対象とした N-gram 確率密度を用いたマルウェア感染検知手法」について今回行った検討のまとめと，それに関する今後の課題について述べる．

第 2 章

マルウェアと対策技術

本章ではマルウェアの基本的性質および対策技術について述べる．まずマルウェアの特徴やそれによる被害，感染後の動作，変遷等を説明し，マルウェア対策の現状について述べる．

2.1 マルウェアの特徴

2.1.1 基本的性質

前述したようにマルウェアとは Malicious(悪意のある) と Software を組み合わせた言葉である．ユーザの望まない不正な動作を行うプログラムの総称として用いられている．

マルウェアという言葉が用いられる以前は，このような不正プログラムをウイルスと呼ぶことが多かった．しかしながら PC やインターネットの普及に伴いウイルスの多様化が進み，感染形態や機能，目的などによって数多くの種類が出現した．またウイルスが現れた当初は愉快犯であったり，技術的興味を満たすことが中心であったが，2000 年前後から金銭搾取へと目的が変移していった．

こういった経緯から多様化・高度化・悪質化する不正なプログラムを，統一してマルウェアと呼ぶことになったのである．

2.1.2 マルウェアの種類 [9]

マルウェアとは悪質なプログラムの総称である．多様化・高度化したマルウェアは非常に多種多様なものとなっている．そこで本項ではいくつかの分類に従って紹介をする．

1. 感染形態に着目した分類
   - ウイルス (Virus)
     ウイルスとは，それ単体では動作せず，自分自身を他のファイルやプログラムに寄生させる感染形態のマルウェアを指す．フロッピーディスクやハードディスクなどのシステム領域を感染対象とするブートセクタ感染型と，実行可能ファイルを主な感染対象とするファイル感染型に大別できる．

- ワーム (Worm)
  ワームとは，単体で動作し自己増殖を行う感染形態のマルウェアを指す，高い感染力を持っており，大規模感染を引き起こす傾向にある．感染手法としては，電子メールやリムーバブルメディアを移動媒体とするもの，Windows のファイル共有やメッセージング機能を利用するもの，OS やアプリケーションの脆弱性に対する攻撃コードを用いるものがある．

- トロイの木馬 (Trojan Horse)
  トロイの木馬とは，有用なプログラムやファイルを装ってユーザ自身によるシステムへの導入・起動を誘い，実際にはユーザの意図しない不正な動作を行うマルウェアを指す．トロイの木馬はユーザの不注意を利用してシステムへの侵入をするため，感染機能を持たないものが多い．

  上記以外にも，フィルタドライバとして実装され，OS のカーネルの深部に潜伏する巧妙な感染形態をもつものもある．マルウェアのファイルやプロセスをアンチウイルスソフトやタスクマネージャに対して隠蔽をするルートキット (Rootkit) や，ユーザのキーボード操作を記録・収集するキーロガー (Keylogger) などは，この感染形態を取ることが多い．

2. 目的に着目した分類
   - スパイウェア (Spyware)
     スパイウェアとは，ユーザの PC 上の個人情報や行動履歴を収集し，特定のサーバなどに送信することを目的としたマルウェアを指す．キーロガーも目的という点ではスパイウェアの一種と考えられる．

   - アドウェア (Adware)
     アドウェアとは，ユーザに企業広告などを提示することを目的としたプログラムである．無害なアドウェアも存在する一方で，ユーザの同意なしに広告を頻繁にポップアップしたり，ユーザの意図しない Web サイトに強制誘導させるものはマルウェアとみなされる．

   - ランサムウェア (Ransomware)
     ランサムウェアとは，ユーザの PC 上のディレクトリやファイルに対して強制的に暗号化やパスワード付き ZIP 圧縮を行うことで，ユーザのデータを「人質」にし，そのデータの復号や解凍の見返りとして，ユーザから身代金 (ransom) を搾取することを目的としたマルウェアである．

- スケアウェア (Scareware)
  スケアウェアとは，ユーザに虚偽の情報を提示し不安 (scare) を煽ることで，無意味なソフトウェアを販売することを目的としたマルウェアである．典型的な例として，偽のマルウェア感染情報をユーザに提示することで Web サイトに誘導し，何も意味の持たないプログラムをアンチウイルスソフトとして販売しようとするものがある．

3. 機能に着目した分類
   - ダウンローダ (Downloader)
     ダウンローダとは，それ自身とは別のマルウェアを特定のサイトからダウンロードし，感染 PC にインストールする機能を持ったマルウェアである．最近のマルウェアの多くは感染後にダウンローダを多段に用いることで解析を困難にしたり，定期的にダウンロードを繰り返したりすることで，新しい機能を持ったマルウェアを容易に拡散させることが可能になっている．

   - ドロッパ (Dropper)
     ドロッパとは，マルウェアを内包した状態で流通し，ユーザの PC 上で実行されると，暗黙のうちにマルウェアをインストール (ドロップ) する機能を持ったマルウェアである．ドロッパの中には Microsoft Word などの文書ファイルになりすまし，実行されると実際の文書を表示すると同時にマルウェアをインストールするという巧妙なものも存在する．

4. その他の分類
   - ボット (Bot)
     ボットとはロボット (robot) の短縮語であり，指令者からの遠隔操作によって，多岐にわたる活動，目的，機能を実現するマルウェアである．ボットに感染した PC はボットネットと呼ばれるネットワークを形成する．ボットネットは小規模なものでは数百，大規模なものでは数十万もの感染 PC 群によって成り立っている．指令者は，指令サーバ【Command & Control サーバ：以下 C&C サーバ】(IRC サーバや HTTP サーバ) 経由でボットネットに制御命令を同報し，その結果，多数のボットが命令に従って一斉動作を行う．スパムメールの大量送信や，DDoS 攻撃，大規模な感染活動など様々なインシデントの原因となっている．ボットネットの概要図を図 2.1 に示す．

マルウェアの種類についてまとめた図を図 2.2 に示す [10].

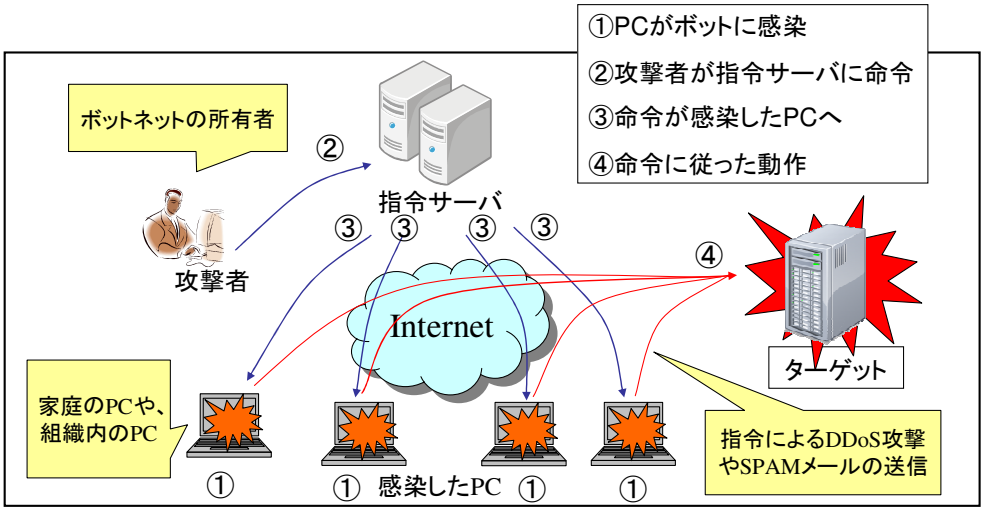

図 2.1　ボットネットの概要

図 2.2　最新のマルウェア事情 (アスキードット PC,2010 年 11 月号,p18 より)

2.2　マルウェアによる被害

マルウェアが登場した当初は，マルウェア作成目的は愉快犯であったり，技術力の誇示が中心であったが，インターネットの普及が進む中で金銭的利益を目的とした個人情報の詐取へと切り替わってきた．そのため感染後の活動も変化してきた．過去のマルウェアではデスクトップ上に画像やメッセージを表示させるなど，被害といえどもさほど大きな問題ではなかった．しかしながら近年では迷惑メールの送信や DoS 攻撃，感染 PC に保存されているクレジットカード番号の

収集などが中心となっている．

2000 年 2 月には，Amazon.com や Yahoo!などの大手 Web サイトが大量の PC からの短時間のアクセスにより，数時間サービスがダウンするという事件が発生した [11]．

2003 年 1 月には，インターネット上で急速に拡散するワームが出現し，ワームによる攻撃コードによって韓国のネットワークが数日間利用不能になった [12]．

2005 年 10 月にオランダで発見されたボットネットは 10 万ものボットから構成されたという報告があった [13]．後の報告で，このボットネットは 150 万以上のボットから構成されていた事実が判明し，ボットネットによる被害の深刻さを物語っていた．この事件の攻撃者は企業への脅迫や，ハッキング行為を行ったとのことである．ボットネットは PC の管理者，ネットワークの管理者の両者における大きな脅威であるといえる．

近年では，標的型攻撃による被害が急増している．株式会社シマンテックによる標的型攻撃の定義は，「金銭や知的財産等の重要情報の不正な取得を目的として特定の標的に対して行われるサイバー攻撃」である [14]．標的型攻撃は Advanced Persistent Threat(APT) 攻撃とも呼ばれている．攻撃方法として最も普及しているのは，フィッシングメールと不正プログラムを組み合わせる方法である．すなわち，攻撃者が偽装したメールを特定の標的に送り，リンクにアクセスさせたり添付ファイルの実行を誘発させることで攻撃を行う方法が広く普及している．独立行政法人情報処理推進機構技術本部セキュリティセンター (IPA) に情報提供のあった標的型攻撃メール件数の推移を図 2.3 に示す [15]．標的型攻撃は攻撃手法の多様化の典型的な例であるといえる．

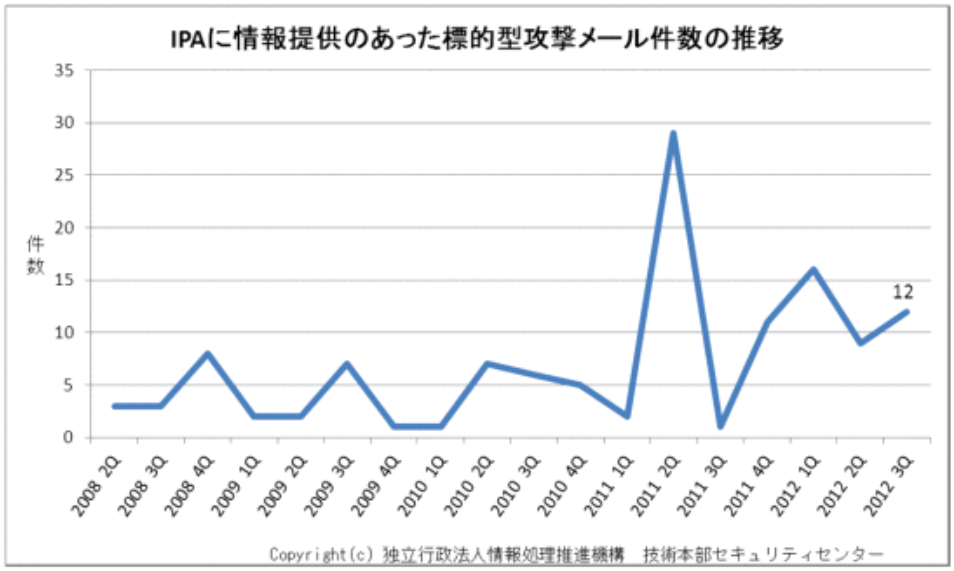

図 2.3　IPA に情報提供のあった標的型攻撃メール件数の推移

2012 年 6 月から 9 月には，他人のパソコンにマルウェアを感染させて乗っ取り，所有者の知らない間に犯行予告を掲示板に書きこむという遠隔操作による事件が発生した [16]．この事件の

問題点は，遠隔操作により男性 4 人が誤認逮捕されてしまったことである．さらに，2012 年 12 月には相次いで Gmail のアカウントをハックされる事件も発生した [17].

上記のように，近年のマルウェアによる被害は，第三者である一般市民へも広がっている．

2.3　マルウェア感染時の動作[18][8]

マルウェア感染時の動作について説明する．マルウェアに感染する際の活動は，感染，接続，転送，攻撃に大別することが出来る．

1. 感染：脆弱性を突いた攻撃や Web 閲覧などによる感染.
2. 接続：インターネット接続の確認や感染の確認を行う．
3. 転送：検体のダウンロードやアップデートを行う．
4. 攻撃：他の PC への攻撃や迷惑メールの送信を行う．

本節では上記の活動について述べる．

2.3.1　感染

マルウェアへの感染は能動感染型と受動感染型に分けられる．以下に特徴を述べる．

- 能動的感染型
  攻撃者が脆弱性の存在する PC を標的として攻撃を行うものである．対象の PC に対してエクスプロイトコード (Exploit Code) という不正なコードが送り込まれる．ネットワーク環境下において，脆弱性のあるサービスを動作させている PC は感染の標的になる．

- 受動的感染型
  PC の管理者が自らの行動で悪意のあるプログラムを実行し，感染するパターンである．現在の感染経路としては，Web からの感染やメールからの感染が主である．Web 経由では Web ページにブラウザの脆弱性を突くコードが含まれており，閲覧と同時にコードが送り込まれる．またメール経由の感染では，メールにマルウェアもしくはマルウェアをダウンロードするためのソフトウェアを添付し，自発的なクリックを促し感染させるものである．

また上記の攻撃には既知の攻撃，未知の攻撃がある．

既知の攻撃とは，過去のマルウェアに用いられた攻撃など，セキュリティベンダに認知されている攻撃を示す．攻撃の特徴として，流れるパケットの特徴や通信の特徴，ソフトウェアの脆弱

性などをシグネチャとして登録しておくことで対処することが出来る．利用者はアンチウイルスソフトのアップデート等の対策を適時行うことで，感染から免れることが出来る．

未知の攻撃とは，ソフトウェアの認知されていない脆弱性を突く攻撃であり，いわゆるゼロデイ攻撃というものである．この攻撃の解析にはある程度の時間を要し，同時にシグネチャの配布を行う必要がある．シグネチャの配布に時間がかかるため，解析完了後も即座に攻撃を検知することは難しい．

2.3.2　接続

感染したホストはいくつかの接続をすることで，活動までの準備を行っている．以下にその活動を述べる．

- インターネット接続の確認
  感染した PC がインターネット接続されているかを確認する．有名なポータルサイト等に接続を試みることで確認を行う．この確認は新たなマルウェアのダウンロードや C&C サーバへの接続のために行われている．

- C&C サーバに接続して指令を待つ
  ボットに感染した際に見られる活動である．ボットに感染した PC はインターネットの接続が確認された場合，指令が配信される C&C サーバに接続する．

2.3.3　転送

攻撃を受けた PC はマルウェアの転送を開始する．検体のダウンロードや検体のアップデートを行うことで，感染した PC に検体を呼び込む．2.1.2 項で紹介したダウンローダーを実行することで，ダウンロードを開始する．この際の通信方式としてプッシュ型，プル型の通信が存在する．概要を図 2.4 に示す．

左図のプッシュ型の通信では，感染したホストは攻撃元ホストや C&C サーバなどからの命令によりポートを開き，ダウンロードホストからマルウェア検体が転送されるのを待つ．プッシュ型ではインターネット上の他のホストから通信が開始されるため，NAPT(Network Address Port Translation) などの外部からの通信が制限された環境においては通信が開始されない．

右図のプル型の通信では，感染したホストからダウンロードホストにマルウェア検体を要求する形でダウンロードが行われる．感染したホストから通信が開始されるため，外部からの通信が制限された環境においてもダウンロードが実行される．

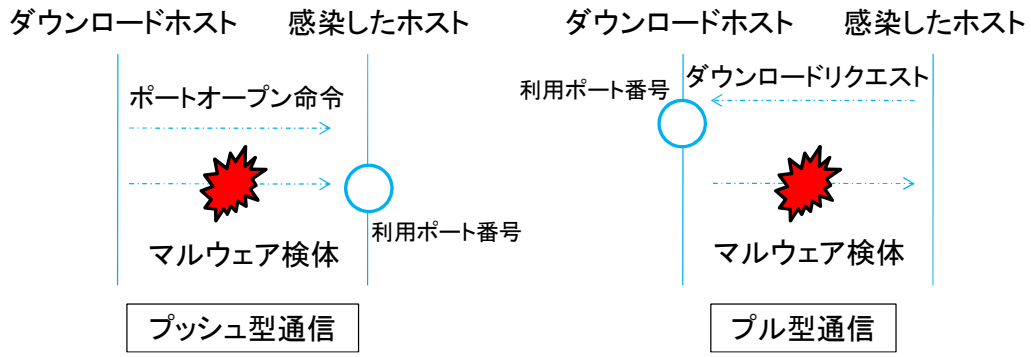

図 2.4　プッシュ型通信とプル型通信

2.3.4　攻撃

感染後の攻撃としては，感染を拡大させるものや，サーバ機能の停止を目的とするものがある．以下それらについて述べる．

- 感染の拡大
  他の PC に対してポートスキャンを行い，脆弱性の調査を行う．同一のネットワーク，インターネット上のホストを対象にスキャンを行う．また脆弱性を発見した際には，感染活動を行いコードやマルウェアを送り込む．

- DoS 攻撃
  Web サーバやアプリケーションサーバ，ルーターなどに対して過度のアクセスを行うことで，対象への負荷をかける攻撃である．サーバは自らのリソースをもとにサービスを提供するため，過度の要求が発生するとサービスの停止や遅延を引き起こして，経済的な打撃を与える．

- スパムメールの送信
  スパムメールの送信とは，感染した PC を介して無差別にインターネットメールを送信するものである．単純に営利目的のものであったり，悪意のある Web ページへの誘導を行うためのメールが主である．営利目的のものでは，自らの利益に結びつくような Web ページのアドレスを記載したメールを送る．悪意のある Web ページへの誘導をするものは，Web ブラウザや Web コンテンツが利用するアプリケーションの脆弱性をつくようなコードが含まれる Web ページのアドレスを含むメールを送信する．

ボットネットが構成されている際，分散された PC からの攻撃が行われる．そのため DoS 攻撃やスパムメールによる脅威はより一層大きなものとなる．2.2 節で挙げた，Amazon.com や Yahoo!

の事件にもボットネットが利用されたという話もあり，脅威のほどが伺える．

2.4　マルウェアの変遷[9]

本節ではマルウェアの変遷について紹介する．歴史の中でも特にマルウェアが悪用された時代にフォーカスを当てる．

2.4.1　1970 年代〜1980 年代中盤

ネットワーク上で自己増殖するマルウェアがはじめて発見されたのは 1971 年である．BBN 社の Bob Thomas がインターネットの原型である ARPANET に放った Creeper が世界初のワームといわれている．

また 1980 年にはゼロックス・パロアルト研究所 (PARC) の John F. Shoch と Jon A. Hupp が，ワームを用いた分散コンピューティングの実験を行った．このワームはネットワークを探索し，アイドル中のマシンを見つけると，ネットワークブートさせ，同時に自分自身を送り込み計算力を借りるという機能であった．PARC 内のイーサネット上にテストのために置かれていたワームが暴走し PARC の相当数のマシンがクラッシュする事態が起こった．

1982 年には当時高校生であった Richard Skrenta が，同級生を驚かせる目的で世界初のウイルス Elk Cloner を作成した．1984 年には，Fred Cohen が論文を発表し，その中で初めて (コンピュータ) ウイルスという用語を定義し，将来的な脅威の予見とともに，対策の困難さを言及した．

1970 年代から 1980 年代中盤にかけては，ウイルスやワームが計算機やネットワーク上で実在し得ることや，それらの脅威が認識され始めた「発見の時代」であったといえる．

2.4.2　1980 年代中盤〜1990 年代中盤

1980 年代中盤から 1990 年代中盤には様々な機能が開発され，実世界で試される「実用の時代」となる．

1986 年，パキスタンの Farooq Alvi 兄弟によって，IBM PC に感染する初のウイルス Brain が作成された．このウイルスは彼らが経営する Brain 社のソフトウェアを違法コピーした PC に感染し，駆除のために Brain 社にコンタクトを求めるメッセージを表示するものであった．同年には初のトロイの木馬 PC-Write も登場している．

1988 年には，sendmail，fingerd，rsh/rexec などの複数の脆弱性とパスワードクラックを悪用した Morris ワームがインターネット上で拡散した．

1990 年以降，Virus Exchange BBS と呼ばれるウイルス情報交換のための掲示板が出現し，MtE(Mutation Engine)，VCL(Virus Creation Labratory)，PS-MPC(Phalcon/Skism Mass Produced Code Generator) に代表されるマルウェア作成ツールが登場した．これによりマル

ウェア作成の技術的なハードルが下がり，実験的なマルウェアが多く登場した．また同時にアンチウイルス無効化機能など，マルウェアの基礎技術が開発された．

2.4.3　1990 年代中盤〜現在

1990 年代中盤以降，マルウェアは「悪用の時代」に突入した．

1990 年後半からウイルスやワームがメールなどインターネット上の通信手段を用いて感染を広めるようになった．1999 年の Bubbleboy や 2000 年の LoveLetter ウイルスは，電子メールクライアントの脆弱性を利用し電子メールを閲覧するだけで感染するため，高速に感染が拡大した．

1995 年には Concept，1999 年には Melissa という MicrosoftWord のマクロ機能を利用したマルウェアも台頭した．これらは感染した Word ファイルを開くことで感染が行われる．Melissa では感染 PC 上の Outlook のアドレス帳に登録された最大 50 個のメールアドレスに向けて，感染した Word ファイルを添付したメールを送信するため，感染が拡大した．

2000 年代前半には，Windows や Windows サーバの脆弱性を狙う攻撃コードを用いたワームの大規模感染が次々と発生した．2001 年 7 月の CodeRed や 2001 年 9 月の Nimda，2003 年 1 月の SQLSlammer，2003 年 8 月の MSBlaster などの代表的なウイルスは攻撃対象のホストが通信をしているサービスに対して攻撃コードを送信し，対象ホストを攻略していた．

P2P ファイル共有ソフト Winny を介して拡散するマルウェア Antinny は，数々の情報漏えいの引き金となった．

2004 年頃から，マルウェアに感染している PC が同期して活動する様子が世界中で観測され始めた．2.1.2 項でも説明した「ボットネット」である．2.2 節でも示したように，ボットネットによる不正行為 (金銭目的) は日々発生するようになった．スパムメールの 9 割近くがボットネットから送信されているという調査もあり，ユーザをフィッシングサイトやマルウェア感染サイトへと導き続けている．

ボットネット登場後，マルウェアはさらに高度化・巧妙化した．傾向として検出を避けるよう静かに潜伏を続けるものが多くなり，フィッシングや SQL インジェクションなどの金銭的価値のある個人情報への攻撃も目立つようになった．

そのような状況下で，2008 年 11 月に Conficker と呼ばれるマルウェアが出現する．この Conficker は過去のワームが備えていた様々な機能の集大成のようなマルウェアであった．USB メモリを介した感染や，感染 PC 間での P2P ネットワークの自律的確立など，今までにはなかった機能も備えていた．2009 年 2 月の段階で約 1200 万台もの感染が報告され，大規模感染の余韻は現在も残っている．

さらに，2009 年 5 月頃から Gumblar 攻撃という新種の攻撃手法も流行し始めた．改ざんされた Web サイトを閲覧することによるマルウェア感染である．大小様々な Web サイトが改ざんされ瞬く間に感染が広がった．マルウェア感染後に感染 PC 上にある FTP アカウント情報を取得

し，FTP アカウントを用いた Web サイトを改ざんするというサイクルを繰り返すもので，今現在もボットネットに並ぶ脅威となっている．

2010 年 3 月には，HTC 製スマートフォン「HTC Magic」にマルウェアが発見されたという事例がある．最近では Android 端末や iPhone といったスマートフォンの普及が進んでいるため，将来的にスマートフォンを標的としたマルウェアが蔓延することも考えられる．

また同月には Mariposa と呼ばれるボットネットを管理していた首謀者が逮捕された．Mariposa のボットネットには 1200 万もの IP アドレスが接続されていた．被害は 190 カ国以上の企業や政府機関，家庭などの PC におよび，銀行講座情報やクレジットカード番号等の情報が盗み出されていた．

2010 年 7 月には Stuxnet と呼ばれるマルウェアが登場した．制御システムや電力会社を狙った初のマルウェアで，工場の操業計画に支障をきたすなど，社会インフラという新たな面への脅威となった．さらに，Stuxnet のような産業制御システムを狙うマルウェアとして，Duqu，Flame などの新種も続々と登場している．

近年の主流な攻撃方法として，2.2 節で挙げた標的型攻撃が挙げられる．その標的型攻撃の際によく使われているのが Poison Ivy である [19]．Poison Ivy は RAT の一種である．RAT とは，遠隔操作機能を適用するツールであり，Remote Access Trojan horse，Remote Access Tool，Remote Administration Tool 等の様々な呼称がある．そして，このマルウェアは現状でも配布用 Web サイトからだれでも入手可能である．すなわちマルウェアが誰でも簡単に作成できる時代に突入したのである．

以上のように，マルウェアの歴史は非常に長く，裏を返すとマルウェアとの戦いとも言え，これからも新たな脅威に対する注意を払い対策を講じていく必要がある．

2.5　マルウェア感染検知

本節では，マルウェア対策の 1 つであるマルウェア感染検知について述べる．まず，マルウェア対策研究の現状について述べた後に，本研究で対象とするマルウェア感染検知について説明する．

現在行われているマルウェア対策研究としては以下のものが挙げられる [20] [21]．

- 感染検知
  PC がマルウェアに感染しているかどうかを検知する手法の研究．

- 検体解析
  感染後の挙動の解析やコードの解析を行う研究である．これは「動的解析」と「静的解析」に分けられる．

動的解析とはマルウェアを実際に動作させて挙動を分析する解析方法であり，静的解析はマルウェアのプログラムコードを分析する解析方法である．

- 広域観測
  おとりシステムであるハニーポットを用いたボットネット等の活動状況の観測を行う研究.

検体解析では，動的解析における動作環境の構築が難しい，静的解析では非常に高度な技術が求められる，などの問題があり，広域観測では，ハニーポットの設置が難しい，および設置にかかるコストの問題などがある．そこで，本研究では 1.2 節で言及した通り感染検知に着目した.

2.5.1　従来手法

文献 [20] [22] を参考にマルウェア感染検知の従来手法について簡単に整理する．従来手法としては以下のものがある.

1. シグネチャベースによる検知
マルウェアごとにシグネチャデータを用意することで，パターンマッチングを行い検知する手法である．新たなマルウェアが現れるごとに分析をする必要があり，未知のものは検知できないといった課題がある.

2. ルールベースによる検知
脆弱性をつく攻撃に関してルールを設定することで，その様なパケットが到着した際に感染を検知する手法である．ゼロデイ攻撃のような，新たな脆弱性をついた攻撃には対応しきれないという課題がある.

3. トラヒックデータを用いた感染検知
ボットネットにおける C&C サーバと感染した PC 間の通信や感染後の DNS クエリの異常に着目することで検知を行う．シグネチャベースによる検知に比べて，過去のマルウェアとの比較が難しいという課題がある.

4. イベント観測
マルウェアの感染後の動作として現れる，DDoS 攻撃やスパムメール発信等の発症活動に着目し感染したホストの検知を行う．動作が起きてからの検知になり，大きな挙動での検知になってしまうという課題がある.

5. ヒューリスティック検知

実行ファイルの挙動などを解析し，ライブラリファイルの書き換えなど，一般的にはあまり見られない特異な挙動を探し未知の検知を行う．しかしながら精度が必ずしも高くなく，誤検知もあるという課題がある．

本研究では，1.2 節で言及した通り，トラヒックデータを用いた感染検知を行う．

2.5.2　トラヒックデータを用いた既存の感染検知

本項では，まずは，本研究で想定している識別器を用いた感染検知について述べる．その後，トラヒックデータを用いた感染検知の既存研究について紹介する．また，マルウェアの感染時を「正常ではない状態」と捉えることもできるため，異常検知手法も参考になると考え，紹介している．

識別器を用いた感染検知

パターン認識とは入力されたパターン (指紋等の画像や) をいくつかのクラスごとに分類することができるとき，あるパターンを複数のクラスに対応させることである [23].

まず入力パターンに対し正規化やノイズ除去といった前処理後，特徴量を数値化して抽出する．抽出できた特徴量をまとめて特徴ベクトルとし識別に用いる．いま d 個の特徴を用いるとき，特徴ベクトルは式 (2.1) で表される．

$$\vec{\mathbf{x}} = (x_1, x_2, \cdots, x_d)^t \tag{2.1}$$

この特徴ベクトルによって構成される空間を特徴空間と呼ぶ．ある 1 つのパターンは特徴空間上の 1 点を示すこととなり，同一クラスに属するパターンは互いに類似しているためまとまったかたまりとして観測される．つまり特徴空間上に存在する複数のパターンを，クラスごとに分類できる識別境界の作成が識別器の設計ということになる．

未知の入力パターンを適切なクラスに分類することが目的であり，そのためにクラスごとのサンプルを用いて学習を行い，識別境界面を作成し，分類することとなる．手順を図 2.5 に示す．

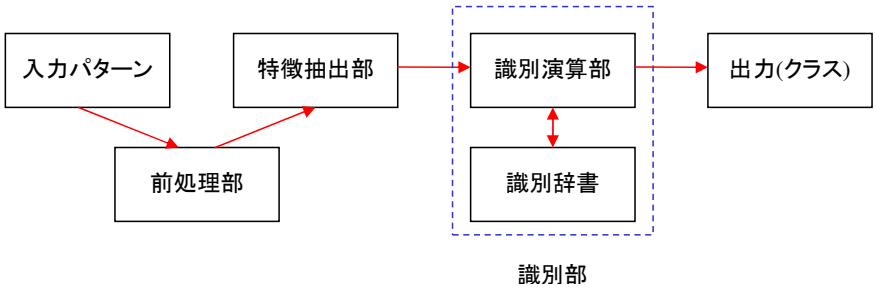

図 2.5　認識系の流れ

本研究ではトラヒックデータを用いた感染検知という観点から，感染の有無を正常なトラヒックデータか否かという 2 つのクラスに対応させて考える．前処理部では，対象となる通信から特定のノードとの通信を切り出して特徴量を抽出しやすい形にする．次に，特徴抽出部でパケットサイズやパケット到着間隔といった特徴量を抽出し，あらかじめ学習をして作成した識別辞書 (コードブック) との比較を行うことで，入力されたトラヒックデータが正常か異常かを決定する．

識別器の設計を行うにあたって重要である点は，各クラスの分布が分類できるような特徴量を用いることである．また識別アルゴリズムに関しても，特徴量の分布に適したものを利用する必要があるため，適切な識別器の選択，妥当性の評価をしなければならない．

検討するマルウェア感染検知の流れを図 2.6 に示す．

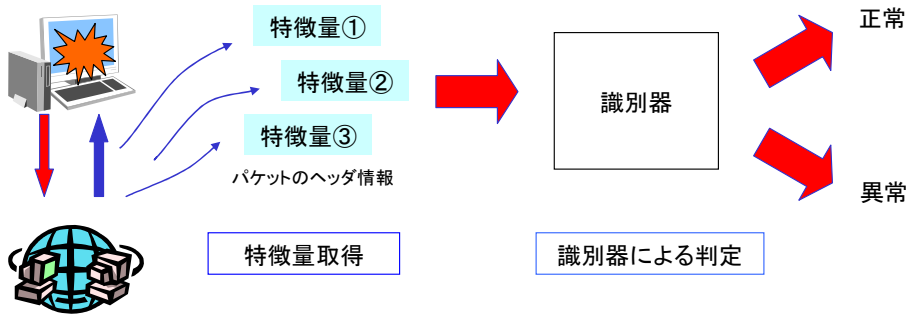

図 2.6　システムの概要

感染の有無を判別したい PC におけるトラヒックデータを取り出し，前処理を行った後にヘッダー情報から複数の特徴量を取得する．これらの特徴量に関して識別器を用いて対象のトラヒックデータが正常か感染しているかの判定を行う．

識別器を用いた感染検知における大切なポイントは 2 つある．正常トラヒックと感染トラヒックで違いが大きくなるような特徴量を選択すること，および，その特徴量に対して適切な識別器を選択するということである．以下では，特徴量選択と，識別器という観点に着目して，既存研究を紹介する．

ペイロード情報を用いた感染検知

ペイロード情報を用いた感染検知や異常検知では，文字列の出現頻度などがよく使われている [24][25].

桑原ら [26] は，ボットの攻撃通信データのペイロード情報から，マルウェアの挙動とそれに対応した特徴的な文字列 (exe,NICK 等) があることを確認し，それらを特徴量として用いている．

Wei Lu ら [27] は，ボットが行う遠隔制御用通信のトラヒックデータに着目する手法を提案し，正常時通信とボットが行う感染 (異常) 時通信のパケットのペイロード情報に着目し，ペイロード内の ASCII 文字コードの出現頻度 (バイト数) を特徴量としている．

山田ら [28] は，侵入検知システムにおける未知攻撃の課題に対する解決策として決定木を用いたアノマリ検知を示し，HTTP リクエスト長，HTTP リクエストの総サイズを特徴量としている．

また，大月ら [29] は，既存研究で用いられているペイロード情報から得られる特徴量に対して，マルウェアの種類 (ワーム，トロイの木馬，ファイル感染型ウイルス) における，感染検知に有効な特徴量の評価を行っている．その結果，ワームでに対しては ASCII 文字コード「i」，ファイル感染型ウイルスに対しては HTTP リクエスト長を用いれば安定的に検知できることを報告している．

ヘッダー情報を用いた感染検知

及川ら [30] は，主成分分析を用いた異常検知を提案している．プロトコルごとのパケット間の相関関係に基づき，主成分軸から距離が離れているものを異常として検出する．正常状態がある軸，または平面に沿って分布し，異常状態は通信プロトコルの分布が崩れ軸から離れる，という仮定に基づいている．

宮本ら [31] は，パケットのヘッダー情報 (TCP フラグ別パケット数，TCP プロトコル別パケット数，パケットサイズ別パケット数) を集約して SVM によりクラスタリングを行い，異常検知を行う手法を提案している．

東角ら [32] は，トラヒックデータからハニーポットの特徴的な挙動として攻撃を受けながらも感染しなかったケース，マルウェア本体のダウンロードにいたらなかったケース，攻撃から活動に至るまでの一通りの動作が行われたケース，複数の感染が確認されたケースが存在することを示している．また，DNS クエリに着目し，Windows2000 では IP アドレスを指定した正引きが行われ，WindowsXP ではリゾルバの逸脱を確認し，感染初期の活動の検知にこれらの特徴が利用できる可能性を示している．

パケット送信間隔に着目した感染検知

S.Kondo ら [22] は，ボットに感染した PC と C&C サーバとのセッションと Web 閲覧等とのパケットサイズの違い，および送受信間隔に着目している．トラヒックデータよりパケットサイズ，送受信間隔によるヒストグラムを特徴量とし，Support Vector Machine(SVM) を用いて C&C サーバのセッションを検知し，SVM が有効であることを示している．

安部ら [33] は，[22] と同様に C&C サーバとの通信の特徴として，パケットサイズと送受信間

隔に着目し，C&C セッションにおける送信パケットが通常の IRC 通信と比較し応答時間が短く，Web 閲覧と比べて送受信パケットサイズの違いが顕著であることを確認している．加えてボットに使用される感染開始時からの通信プロトコルの遷移の仕方を特徴として提案している．

確率を用いた感染検知

M.Bahrololum ら [34] は，KDD CUP 99[35] のデータを用いて，正常，各攻撃に対して GMM を作成し，事後確率が最も高かったクラスを識別結果として異常を検知する手法を提案している．GMM を作成する際の特徴抽出はコネクション毎に行っている．攻撃のクラスとしては，DoS 攻撃，リモートからの不正アクセス，ルート権限への不正アクセス，プローブを用いているが，その後，階層的な GMM を用いてクラスをモデル化することで検知精度を向上させている [36].

Wei Lu ら [37] は，流入するパケット数と流出するパケット数の比を特徴量として，GMM を用いた DDoS 攻撃の検知手法を提案している．DDoS 攻撃の際には，流出するパケット数が非常に少なく，流入するパケット数が多くなると仮定して検知を行っている．実際の検知方法としては，正常な状態の流入するパケット数と流出するパケット数の比を GMM で学習しておき，各 GMM コンポーネントに対する事後確率が非常に小さいか，ほとんど 0 のサンプルを DDoS 攻撃とみなしている．

確率を用いた感染検知・異常検知では正常，異常のモデル作成のために GMM がよく用いられている．

時間的な変化を用いたトラヒック識別

本研究で着目しているトラヒックデータの時間的な変化を用いた既存研究を紹介する．感染検知ではない研究もあるが，感染検知を，正常トラヒックと感染トラヒックの 2 クラスパターン認識問題と考えれば，様々なトラヒック，およびアプリケーション識別研究も参考になると考えられる．

八木ら [38] は，各フロー通信開始直後における数パケットのペイロード長遷移パターンを特徴量にネットワークアプリケーション弁別を行っている．通信の開始時にプロトコル毎の定型的な通信があることに着目している．検討課題として，遷移パターンの長さを長くすると，定型的な制御通信を越えて，データ通信の部分まで踏み込んでしまい，アプリケーションごとに違いが現れなくなってしまうことが挙げられる．

Gautam Thatte ら [39] は，フローではなくタイムスロットを用いた正常トラヒックモデルからの逸脱性による異常検知を行っている．その際の特徴量として，DoS 攻撃，DNS リフレクション攻撃，TCP SYN スキャンなどの一般的な攻撃は一つのパケットサイズを使っている観点からパケットサイズ分布を，攻撃トラヒックの一定比率性と暗号化に強いという観点からパケット割

合を用いている．タイムスロット幅はオーバーラッピングなしの初期値 1 秒に固定し，正常・感染判定ができれば，タイムスロット幅を再計算し小さくするという方法を用いている．

水谷ら [40] は，複数のボットに共通する状態遷移の調査を行っている．多くのボットには共通した挙動の遷移 (攻撃コードの受信→実行ファイルの取得→インターネット接続確認→ C&C サーバへの接続) が存在するため，この挙動の遷移を状態遷移モデルとして捉え，未知のボット検知への利用可能性を指摘している．これは未知のボットの C&C サーバとの通信方法が既知のボットと異なっていても，共通の状態遷移パターンを持ち活動内容が大幅に変更されない限り検知できるということである．また遷移時間 (遷移後の最初の動作時刻－遷移前の最後の動作時刻) の調査結果から，攻撃コードを受けてから実行ファイルを取得するまでの平均遷移時間が約 5 秒であったため，数秒単位で挙動の遷移があることが確認されている．

Chun Yang ら [41] は，トラヒックの時間的な状態遷移に着目したネットワーク異常検知を行っている．この手法では，特徴量抽出→特徴量選択→クラスタリング→状態遷移パターンの抽出→ HMM (隠れマルコフモデル) を用いた識別というアルゴリズムが用いられている．遷移パターンに対してHMMで次の観測系列の確率を算出し，その結果に合わせて異常レベルの警告を発するというものである．任意の半径でクラスタの大きさを表現し，学習データの大部分は正常パターンであり，異常パターンは正常パターンとは統計的に異なるという想定の下，クラスタ内に存在する要素数が閾値以上のものを正常状態のクラスタ，閾値以下のものを異常状態のクラスタとして生成する手法を提案している．

市田ら [42] は，特徴量の時間的な遷移を用いた感染検知を提案している．タイムスロット毎に抽出した特徴量を，ベクトル量子化を用いて作成したコードブックを用いてコードブック番号で表すことで，時間的な変化を表現している．そして，連続する 3 つのコードブック番号を用いて，正常トラヒックと感染トラヒックのモデルを作成し，テストデータがどちらのモデルと似ているかを DP マッチングを用いて計算し，似ているモデルの方のスコアを加算することで感染検知を行っている．

2.5.3　本研究における感染検知

2.5.2 項で挙げた既存の手法には様々な問題が存在する．例えば，時間的な変化を用いない既存手法では，近年似てきているマルウェアによる通信を正常な通信と誤識別してしまう可能性がある．時間的な変化を用いる既存手法も正常，マルウェアに対する柔軟なスコアリングを行うことができていない．すなわち，正常とマルウェアで同じパターンが現れた場合，誤識別をしてしまう場合がある．さらに，確率を用いた既存手法では，特徴量の分布を仮定しているものが多いが，トラヒックはバースト的に発生することがあるので，上手く分布を表現できていないという課題もある．

そこで，本研究では，「ヘッダー情報」からタイムスロット毎に特徴量を抽出し，「時間的な変

化」を用いて正常トラヒックと感染トラヒックの差別化を行い，さらに「確率モデル」を導入することで識別を行う手法を検討する．

ヘッダー情報を用いる理由は 2 つある．1 つ目は暗号化通信への対応であり，2 つ目はプライバシーの保護である．

本研究では，タイムスロット毎に特徴量を抽出している．タイムスロット毎の特徴抽出とは，一定の時間間隔 (タイムスロット) でのトラヒック流量をカウントし，各特徴量を抽出する方式である．タイムスロット毎に特徴量を抽出する理由は，リアルタイムでの検知を行うためである．

既存研究では，フロー毎に特徴量を抽出する方法がよく用いられている．フローとは，プロトコル送信元 IP アドレスとその送信元ポート番号，宛先 IP アドレスとその宛先ポート番号が同じパケット群である．しかし，フロー毎に特徴抽出では，同一フローのパケットを全て収集してからでないと各特徴量を抽出できないため，リアルタイム性に欠けるからである．

確率モデルを用いる理由は，近年のマルウェアによるトラヒックは正常なトラヒックとの区別が難しいからである．すなわち，正常なトラヒックの中にはマルウェアによるトラヒックと同じパターンが存在する可能性があるからである．正常なトラヒックとマルウェアによるトラヒックに同じパターンが存在する場合，シグネチャの様に，マルウェアのパターン，正常のパターンをそのまま登録しておくと，誤検知が起きてしまう可能性がある．そこで，前後のトラヒックを考慮し上手く確率を導入することで，誤検知を減らすことを目指す．

3 章で，マルウェア感染検知において用いるべき特徴量，4 章で，時間的な変化，および確率モデルを用いた提案手法の概要について述べる．

第 3 章

マルウェア感染検知のための経年変化を考慮した特徴量評価

本章ではマルウェア感染検知のための特徴量評価について述べる．まず特徴量評価を行う理由，経年変化について説明し，その後，評価対象となる特徴量，特徴量の評価方法，および評価実験の結果について述べる．

3.1　特徴量評価を行う理由

2.5.2 項で紹介した通り，既存研究ではヘッダー情報から得られる様々な特徴量がマルウェア感染検知に有効であると報告されている．しかし，これらの特徴量の中で特にどの特徴量が有効かということは十分に議論されていない．さらに，新種のアプリケーションやマルウェアが多数誕生しているという現状を考えると，過去にマルウェア感染検知に有効な特徴量が現在も有効かどうかは確かではない．すなわち，識別精度がトラヒックの変化に依存しない特徴量が明らかではない．上記の 2 点を考慮して，識別率だけではなく，経年変化**も考慮した特徴量評価が必要であると考えられる．

3.2　特徴量評価実験概要

特徴量評価実験の概要について説明する．まずは，評価対象となる特徴量を列挙し，次に評価方法，使用したデータ，識別精度評価尺度について説明する．

3.2.1　評価する特徴量

評価する特徴量 36 種類を表 3.1 に示す．既存研究で用いられている，ヘッダー情報，およびそれらの統計値を評価対象とした．

** 学習データを取得した日付とテストデータを取得した日付における特徴量の変化

表 3.1　評価する特徴量 36 種類

番号	特徴量 [単位]
1	パケット数
2	パケットサイズの総数 [byte]
3	パケットサイズの平均 [byte]
4	パケットサイズの最小 [byte]
5	パケットサイズの最大 [byte]
6	パケットサイズの標準偏差 [byte]
7	到着間隔の平均 [秒]
8	到着間隔の最小 [秒]
9	到着間隔の最大 [秒]
10	到着間隔の標準偏差 [秒]
11	SYN パケット数
12	FIN パケット数
13	PSH パケット数
14	ACK パケット数
15	RST パケット数
16	URG パケット数
17	SYN/ACK パケット数
18	FIN/ACK パケット数

番号	特徴量 [単位]
19	PSH/ACK パケット数
20	RST/ACK パケット数
21	TCP パケット中の SYN パケット割合
22	TCP パケット中の FIN パケット割合
23	TCP パケット中の PSH パケット割合
24	TCP パケット中の ACK パケット割合
25	TCP パケット中の RST パケット割合
26	TCP パケット中の URG パケット割合
27	TCP パケット中の SYN/ACK パケット割合
28	TCP パケット中の FIN/ACK パケット割合
29	TCP パケット中の PSH/ACK パケット割合
30	TCP パケット中の RST/ACK パケット割合
31	ICMP 到達不能メッセージ数
32	UDP パケット数
33	送信元ポート番号が 69/UDP のパケット数
34	送信元ポート番号が 80/TCP のパケット数
35	送信元ポート番号が 110/TCP のパケット数
36	送信元ポート番号が 443/UDP のパケット数

3.2.2　評価方法

テストトラヒックを正常なトラヒックかマルウェアに感染した後のトラヒックのどちらかに判別する方法は以下の通りである.

まず，学習用の正常トラヒックと感染トラヒックを別々に用いて正常コードブックと感染コードブックを作成する．コードブックの作成方法は LBG + splitting によるベクトル量子化を用いた．コードブック作成結果が初期値に依らないため，LBG + splitting を採用した．ベクトル量子化の詳細を付録 A に示す．今回の特徴量評価実験では，1 つ 1 つの特徴量を評価することが目的なので，一次元コードブックを作成した.

特徴量を抽出するためのタイムスロット幅は 0.1 秒，1 秒，10 秒，100 秒と変化させ，ベクトル量子化のためのレベル数は 2,4,8,16,32 と変化させて評価実験を行った．ベクトル量子化レベル数は作成されるコードブックの数を表す.

そして，各パラメータ (特徴量，タイムスロット幅，ベクトル量子化レベル数の組) に対して，作成しておいた正常・感染コードブックと，予め正常・感染のラベルが付けられているテストデータとの特徴空間上でのユークリッド距離を基に識別を行う．具体的には，テストデータと正常コー

ドブックの距離がテストデータと感染コードブックとの距離より小さければテストデータを正常と判定し，テストデータと感染コードブックの距離がテストデータと正常コードブックとの距離より小さければテストデータを感染と判定する．

3.2.3　使用したデータ

以下に，特徴量評価実験で使用したトラヒックデータと前処理について述べる．経年変化を調査するため，正常トラヒックデータと感染トラヒックデータのキャプチャ日時を合わせた．

- 正常トラヒックデータ
  1. トラヒックデータについて
     正常トラヒックデータはあるイントラネットにおいてキャプチャした．正常トラヒックのキャプチャには wireshark[43] を使用した．学習データとして，2009 年の 3 月 13 日 (金)，14 日 (土)，15 日 (日) にキャプチャしたデータを用い，テストデータとして，2009 年，2010 年，2011 年にキャプチャしたデータを用いた．
  2. 前処理について
     正常トラヒックと感染トラヒックのキャプチャ環境は同じ環境が望ましいが，感染トラヒックはリソースが限られており，正常トラヒックを感染トラヒックと同じようなハニーポット環境でキャプチャすることは難しい．そこで，正常トラヒックのキャプチャ環境を感染トラヒックのキャプチャ環境に似せるために，以下の要件を満たすように正常トラヒックに対して前処理を行った．
     - 1 つのホストによって発生したトラヒック
       感染トラヒックがハニーポットでキャプチャしたトラヒックであるので，その環境を模倣するために 1 つのホストによって発生したトラヒックに切り出すことは重要である．(正常トラヒックはあるイントラネットの NAT 上でキャプチャされていた．)
     - 正常なユーザが発生させたトラヒック
       あるホストがマルウェアに感染した場合，インターネット経由で新たなマルウェアをダウンロードしたりマルウェアをアップデートしたりする可能性がある．しかし，ダウンロードやアップデートなどの挙動は正常なユーザによっても行われる挙動である．マルウェア感染検知において，マルウェアのダウンロードやアップデートと正常なユーザによるソフトウェアのダウンロードやアップロードを識別できることが重要である．すなわち，正常トラヒックとして正常なユーザが発生させたトラヒックを用いることは重要である．
- 感染トラヒックデータ
  1. トラヒックデータについて
     感染トラヒックデータとして，CCCDATAset2009,2010,2011[44] を用いた (以下

CCC2009,CCC2010,CCC2011)．CCCDATAset の詳細を付録 B に示す．学習データとして CCC2009 を用い，テストデータとして CCC2009,2010,2011 のデータを用いた．

2. 前処理について
   CCCDATAset はハニーポットでキャプチャされるトラヒックで構成されている．例えば，スキャントラヒックや exploit などのトラヒックから構成されている．しかし，CCCDATAset にはマルウェアに感染する前のトラヒックデータも含まれている．感染検知において，感染トラヒックとしてマルウェアに感染した後のトラヒックを用いることは必須である．そこで，マルウェアに感染した後のトラヒックのみを切り出すために前処理を行った．前処理のコマンド等の詳細は付録 B に示しているので，概要だけ以下に示す．
   – ハニーポット環境独自の制御通信を取り除く
   – トラヒックデータを OS のリセット間隔で切り出す
   – CCCDATAset のログと照らし合わせて本当に感染しているか確認し，マルウェアの感染による最初のパケットを探す
   – マルウェアの感染による最初のパケット以降のトラヒックを感染トラヒックとする

- 使用したデータ量について
  学習に用いたサンプル数を表 3.2 に示す．テストに用いたサンプル数は，各年度の各タイムスロット幅で共通で 448 サンプルで合わせた．これは，テストデータの量による識別精度の変化を防ぐためである．

表 3.2　学習データの各タイムスロット幅におけるサンプル数

	0.1 秒	1 秒	10 秒	100 秒
正常	76453 サンプル	26776 サンプル	5838 サンプル	690 サンプル
感染	20380 サンプル	9254 サンプル	2160 サンプル	448 サンプル

3.2.4　識別精度評価尺度

今回の特徴慮評価実験において，特徴量の識別精度の評価尺度として true negative rate(TNR) と true positive rate(TPR) を用いた．TNR は正常トラヒックを正常トラヒックと正しく識別できた割合であり，TPR は感染トラヒックを感染トラヒックと正しく識別できた割合である．マルウェア感染検知の要件としては，TNR と TPR ともに高い特徴量が有効である．なぜなら，TNR が高くなければ，通常利用の際に誤検知が多発してしまい利便性に乏しく，TPR が高くなければ，マルウェアによる感染を正しく検知できずに感染を拡大してしまう可能性があるからである．

3.3　実験結果

本節では，特徴量評価実験の結果について述べる．まずは，TNR，TPR 毎に表 3.1 の特徴量を経年変化が大きいグループと経年変化が小さいグループの 2 つに大別し，それぞれについて感染検知への有効性を考察する．その後，タイムスロット幅，ベクトル量子化レベル数について考察する．

3.3.1　経年変化について

まず，各年の平均の TNR と TPR の 3 年間の推移を表 3.3 に示す．平均の TNR(TPR) とは，各パラメータ (特徴量，タイムスロット幅，ベクトル量子化レベル数の組) における TNR(TPR) の平均値のことである．平均の TNR については 2011 年が一番高くなっている．しかし，平均の TPR については 2011 年が一番低くなっている．以上から，テストデータの経年変化が識別率に大きく影響していることは明らかである．

表 3.3　各年の平均の TNR と TPR の 3 年間の推移

年度	2009	2010	2011
平均 (TNR)	36.1%	35.2%	40.7%
平均 (TPR)	57.0%	54.1%	51.2%

TNR について

図 3.1 に 3 年間を通して平均の TNR が 50%以上であった特徴量を示す．(以下，特徴量番号は表 3.1 に対応している)

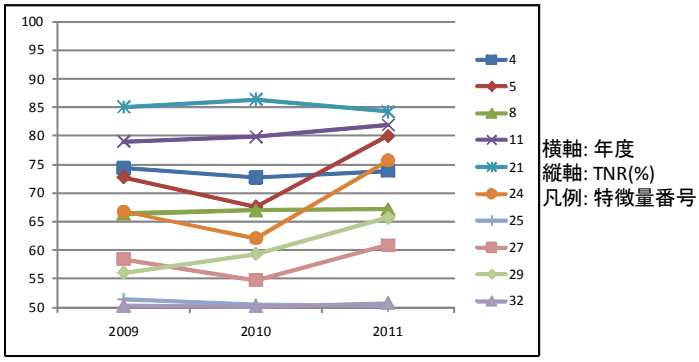

図 3.1　3 年間を通して平均の TNR が 50%以上であった特徴量

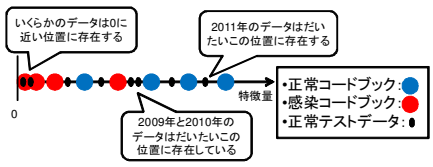

図 3.2　2011 年の平均の TNR が高くなる理由 (イメージ)

- 経年変化が大きい特徴量

  特徴量番号 2,3,9,14,17,18,19,20,34,36 の特徴量は TNR に関して経年変化が大きかった．特徴量番号 9 の特徴量以外は 2011 年の平均の TNR が一番高かった．図 3.2 に 2011 年の平均の TNR が高くなる理由のイメージを示す．上記の特徴量について，2011 年の正常テストデータは特徴量の値が非常に大きくなっており，正常コードブックに非常に近いため正しく識別できている．しかし，2009，2010 年のテストデータは 2011 年のテストデータ程大きい特徴量の値をとっていないので，しばしば感染トラヒックと誤識別されてしまっている．これは，各年度の正常テストデータにおけるパケット数の差が影響している．表 3.4 に各年度の正常テストデータのパケット数を示す．表 3.4 の平均の単位は 1 タイムスロット中のパケット数である．

表 3.4　各年度の正常テストデータ中のパケット数

タイムスロット幅 0.1 秒			
年度	2009	2010	2011
平均	14.3	10.8	5.51
標準偏差	33.8	23.1	9.5
タイムスロット幅 1 秒			
年度	2009	2010	2011
平均	189.9	33.2	30.9
標準偏差	580.4	61.3	43.3
タイムスロット幅 10 秒			
年度	2009	2010	2011
平均	340.8	157.0	1039.7
標準偏差	1561.7	451.1	1176.7
タイムスロット幅 100 秒			
年度	2009	2010	2011
平均	1656.4	6060.0	9225.8
標準偏差	4520.9	1362.3	10603.3

表 3.5　パケットサイズの最小において 3 年間を通して TNR が 90%を超えたパラメータとその TNR

タイムスロット幅	ベクトル量子化レベル数	2009	2010	2011
0.1 秒	4	98.4%	92.6%	90.4%
0.1 秒	8	98.2%	92.6%	93.3%
1 秒	4	99.8%	98.7%	100%
1 秒	8	100%	98.7%	100%
1 秒	16	98.0%	99.1%	100%
10 秒	2	100%	100%	100%
10 秒	4	100%	100%	100%
10 秒	8	100%	100%	100%
10 秒	16	100%	100%	100%
100 秒	2	99.3%	100%	100%
100 秒	4	100%	100%	100%
100 秒	8	100%	100%	100%
100 秒	16	100%	100%	100%

表 3.4 より，2011 年の正常テストデータのパケット数が一番多いことがわかる．各年度のトラヒックデータの通信内容を wireshark で確認したところ, 大きな違いはあまり見られなかった．すなわち，テストデータ中のパケット数は経年変化に大きく影響することがわかった．

- 経年変化が小さい特徴量

  特徴量番号 4,7,8,11,21,25,28,30,31,32 の特徴量は TNR に関して経年変化が小さかった．上記の特徴量には，割合を用いたものが多い (例えば TCP パケット中の SYN パケット割合など)．つまり，割合を用いた特徴量には識別率における経年変化を抑える効果があると考えられる．上記特徴量の内，特徴量番号 4(パケットサイズの最小)，11(SYN パケット数)，21(TCP パケット中の SYN パケット割合) の特徴量は平均の TNR が 3 年間を通して 75%を

超えていた.

– パケットサイズの最小

表 3.5 に 3 年間を通して TNR が 90%を超えたパラメータとその TNR を示す．表 3.6 に正常テストデータと感染テストデータのパケットサイズの最小における平均と標準偏差を示す．平均の単位は 1 タイムスロット中の byte 数である.

表 3.6　正常・感染テストデータのパケットサイズの最小における各年度の平均と標準偏差

タイムスロット幅 0.1 秒						
	正常			感染		
年度	2009	2010	2011	2009	2010	2011
平均	79.5	81.0	93.7	68.6	84.7	114.4
標準偏差	95.6	113.1	134.7	32.4	89.1	174.6
タイムスロット幅 1 秒						
	正常			感染		
年度	2009	2010	2011	2009	2010	2011
平均	60.1	60.8	60.0	70.7	73.3	101.1
標準偏差	1.3	7	0	31.1	34.8	83.1
タイムスロット幅 10 秒						
	正常			感染		
年度	2009	2010	2011	2009	2010	2011
平均	60.9	60.1	60	67.8	71.2	102.2
標準偏差	0.2	3	0	28.2	40.4	113.9
タイムスロット幅 100 秒						
	正常			感染		
年度	2009	2010	2011	2009	2010	2011
平均	60.4	60	60	69.3	66.8	84.2
標準偏差	2.8	0	0	44.9	40.4	60.4

表 3.7　各年度における感染テストデータのパケット数

タイムスロット幅 0.1 秒			
年度	2009	2010	2011
平均	42.3	11.6	3.3
標準偏差	78.6	11.7	4.3
タイムスロット幅 1 秒			
年度	2009	2010	2011
平均	139.6	166.6	6.6
標準偏差	121.2	140.1	13.6
タイムスロット幅 10 秒			
年度	2009	2010	2011
平均	781.6	1439.3	23.4
標準偏差	584.6	1161.6	54.2
タイムスロット幅 100 秒			
年度	2009	2010	2011
平均	3350.4	7578.9	348.7
標準偏差	5212.5	9580.1	1711.7

表 3.6 より，正常トラヒックのパケットサイズの最小はタイムスロット幅を 1 秒以上にとれば常に 60byte に近い値になる．一方，感染トラヒックのパケットサイズの最小は様々な値をとる．すなわち，正常トラヒックにおけるパケットサイズの最小の標準偏差はほとんどの場合 0 だが，感染トラヒックにおけるパケットサイズの最小の標準偏差は正常トラヒックに比べて大きくなっている．この違いは，感染検知において有効だと考えられる．つまり，パケットサイズの最小，およびパケットサイズの最小の標準偏差は感染検知に有効であると考えられる.

– SYN パケット数，TCP パケット中の SYN パケット割合

SYN パケット数，TCP パケット中の SYN パケット割合を特徴量として用いると TNR

は非常に高くなる．これはマルウェア感染時の挙動であるスキャンが大きく影響している．スキャンの影響により，感染コードブックの値は正常コードブックの値よりはるかに大きくなっている．正常時の通信ではあまり多くの SYN パケットが流れない場合が多い．これらの理由により，ほとんどの正常トラヒックは正しく正常と識別されており，TNR が高くなっている．しかし，感染トラヒックにおいて常にスキャンがあるわけではないので，感染トラヒックにスキャンがない場合は正しく識別することは難しい．よって，SYN パケット数，および TCP パケット中の SYN パケット割合は感染検知に有効とは言い難いが，攻撃検知等には利用できる特徴量である．

TPR について

図 3.3 に 3 年間を通して平均の TPR が 70%以上であった特徴量を示す．

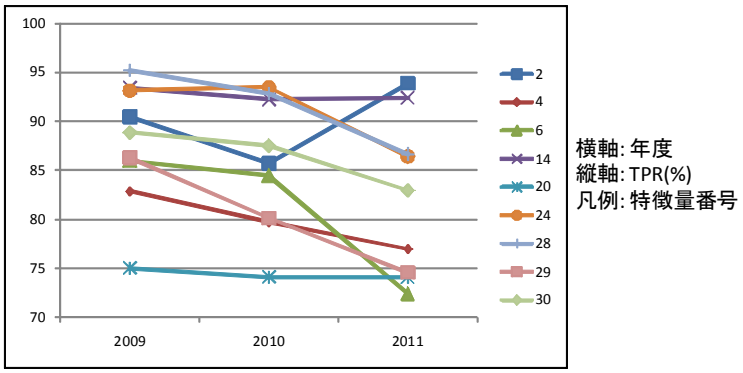

図 3.3　3 年間を通して平均の TPR が 70%以上であった特徴量

- 経年変化が大きい特徴量
  特徴量番号 1,3,5,6,7,9,10,11,21,29 の特徴量は TPR に関して経年変化が大きかった．特徴量番号 1,9 以外の特徴量は，2011 年における平均の TPR が 2009,2010 年より小さくなった．2011 年における平均の TPR が一番低くなった原因は 2 つ考えられる．1 つ目は，2011 年の感染トラヒックは 2009，2010 年の感染トラヒックよりスキャンが少なかったからである．スキャンが少ないとその分通信におけるパケットサイズは大きくなり，正常トラヒックに挙動が似てしまう．よって，特徴量番号 3,11,21 の特徴量の 2011 年における平均の TPR が一番低くなると考えられる．2 つ目は，2011 年の感染トラヒックのパケット数が 2009，2010 年のパケット数より少ないからである．表 3.7 に感染トラヒックにおける各年度のパケット数を示す．平均の単位は 1 タイムスロット中のパケット数である．表 3.7 より，2011 年の感染トラヒックのパケット数が 2009，2010 年のパケット数より少ないことは明らかである．各年度の感染テストデータにおける PSH/ACK パケット数に大きな差はないが，TCP パケット

中の PSH/ACK パケット割合は 2011 年の感染テストデータが一番大きくなっており，正常トラヒックにおけるの TCP パケット中の PSH/ACK パケット割合と近くなっている．よって，特徴量番号 29 の特徴量では 2011 年の平均の TPR が一番低くなっていると考えられる．

- 経年変化が小さい特徴量

  特徴量番号 4,14,17,18,19,20,25,30,31,32,34 の特徴量は TPR に関して経年変化が小さかった．上記特徴量の内，特徴量番号 4(パケットサイズの最小)，14(ACK パケット数)，30(TCP パケット中の RST/ACK パケット割合) の特徴量は平均の TPR が 3 年間を通して 80%を超えていた．

  – パケットサイズの最小

  表 3.8 に 3 年間を通して TPR が 90%を超えているパラメータとその TPR を示す．

表 3.8　パケットサイズの最小において 3 年間を通して TPR が 90%を超えているパラメータとその TPR

タイムスロット幅	ベクトル量子化レベル数	2009	2010	2011
0.1 秒	32	99.8%	94.6%	90.2%
1 秒	32	100%	99.6%	95.8%
10 秒	32	94.0%	92.0%	92.4%

表 3.9　3 年間の正常・感染トラヒックにおける ACK パケットの平均

タイムスロット幅 0.1 秒						
	正常			感染		
年度	2009	2010	2011	2009	2010	2011
平均	10.9	6.9	3.3	2.6	0.6	2.3
タイムスロット幅 1 秒						
	正常			感染		
年度	2009	2010	2011	2009	2010	2011
平均	174.1	20.0	19.0	3.1	1.3	3.4
タイムスロット幅 10 秒						
	正常			感染		
年度	2009	2010	2011	2009	2010	2011
平均	289.1	103.2	787.2	10.0	10.9	11.8
タイムスロット幅 100 秒						
	正常			感染		
年度	2009	2010	2011	2009	2010	2011
兵器	1305.4	432.8	6669.8	40.7	73.5	37.5

TPR に関しても TNR の場合と同様に，パケットサイズの最小，およびパケットサイズの最小の標準偏差は高い識別精度を有している．

– ACK パケット数

表 3.9 に 3 年間の正常・感染トラヒックにおける ACK パケットの平均を示す．表 3.4 と表 3.7 より，正常テストデータと感染テストデータにおけるパケット数に大きな違いは見られない．さらに，感染テストデータにおける ACK パケット数は正常テストデータにおける ACK パケット数に比べて非常に小さい．よって，ACK パケット数は感染検知に有効な特徴量と考えられる．

– TCP パケット中の RST/ACK パケット割合

TCP パケット中の RST/ACK パケット割合に関して，TPR それ自身は高く経年変化も小さいが，正常・感染テストデータともに TCP パケット中の RST/ACK パケット割合が 0 のタイムスロットが多くみられる．そして，感染コードブックの値の方が正常コー

ドブックの値より 0 に近い．すなわち，TCP パケット中の RST/ACK パケット割合は正常トラヒックと感染トラヒックを正しく識別できている訳ではない．このような特徴量は感染検知には有効であるとは言えない．

3.3.2　タイムスロット幅，ベクトル量子化レベル数について

表 3.10 と表 3.11 にそれぞれ各タイムスロット幅，量子化レベル数における 3 年間の平均の TNR と TPR を示す．

表 3.10　各タイムスロット幅における平均の TNR と TPR

タイムスロット幅	0.1 秒	1 秒	10 秒	100 秒
平均の TNR	31.0%	33.6%	36.2%	48.5%
平均の TPR	48.2%	56.3%	57.2%	54.5%

表 3.11　各ベクトル量子化レベル数における平均の TNR と TPR

ベクトル量子化レベル数	2	4	8	16	32
平均の TNR	45.4%	41.6%	40.4%	38.2%	38.0%
平均の TPR	52.3%	52.3%	51.4%	51.5%	48.6%

表 3.10 より，タイムスロット幅 0.1 秒では明らかに短すぎる．そして，実際のリアルタイムでの感染検知を考えるとタイムスロット幅 100 秒では長すぎる．よって，特徴量を抽出するタイムスロット幅は 1 秒か 10 秒が適切と考えられる．

表 3.11 より，1 つの特徴量を用いる場合は，量子化レベル数 32 は高すぎであり，量子化レベル数は 2 か 4 が適切であるとわかる．しかし，本研究における感染検知では，特徴量を組わせて用いることを考えているので，ベクトル量子化レベル数は 8 か 16 が良いと考えている．

3.4　特徴量評価実験まとめ

3.2.4 項で言及した通り，マルウェア感染検知のおいては TNR と TPR ともに高い特徴量を用いることが望ましい．さらに，3.3.1 項の検討結果も考慮することが重要である．よって，マルウェア感染検知の際に，タイムスロット幅を 1 秒，ベクトル量子化レベル数を 2 か 4 として，パケットサイズの最小 (およびパケットサイズの最小の標準偏差)，SYN パケット数，TCP パケット中の SYN パケット割合，ACK パケット数，を特徴量として用いることが有効であると考えられる．

第 4 章

N-gram 確率密度を用いたマルウェア感染検知

本章では，提案手法である N-gram 確率密度を用いたマルウェア感染検知について説明する．具体的には，提案手法の要素技術である N-gram，最近傍密度推定法について説明する．

4.1 N-gram

本研究では，1.2 節，**??**節で言及した通り，特徴量の時間的な変化に着目して感染検知を行うことを考えている．そして，特徴量の時間的な変化を捉えるための手段として N-gram を用いる．

N-gram は主にテキストマイニング等の分野で使われている技術であり，応用として検索エンジンなどにも利用されている [45].

本節では，まず，N-gram について説明する．その後，N-gram を用いる理由について説明し，トラヒックデータへの N-gram の適用方法について説明する．

4.1.1 N-gram について

N-gram の定義は，「あるテキストの総体を前から順に任意の N 個の文字列または単語の組み合わせで分割したもの」である [46]．また，N 個の数 (gram) に応じて，それぞれ「1(uni)-gram,2(bi)-gram,3(tri)-gram···」と呼ばれる．N-gram の具体例を図 4.1 に示す [47].

テキストマイニングの分野では，N-gram の統計をとり，文章の特徴を N-gram 統計量でモデル化することで，Web ページを自動分類したり，作品の著者を推定する既存研究がある [48][49].

4.1.2 N-gram を用いる理由

N-gram は複数の gram を連続的に取り扱う技術であるため，時系列を捉えるための手法ともいえる [50]．すなわち，1 つ 1 つの gram をタイムスロットと考えれば，N-gram の適応がトラヒックデータに対しても可能である．

さらに，N-gram を用いることで，(N-1) 個前からの一連のタイムスロットを連続的に捉えるこ

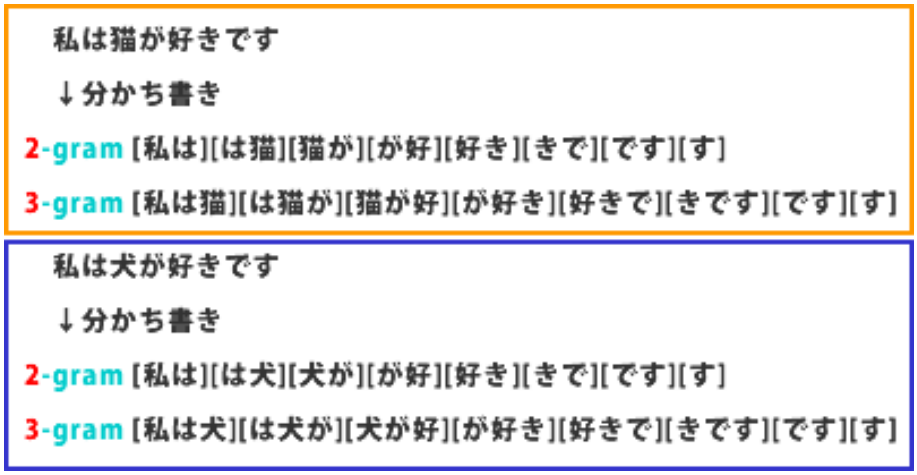

図 4.1　N-gram の具体例 (N=2,3 の場合)

とができるようになるため，時間的な遷移の仕方を大まかに把握しやすいというメリットがある．

加えて，複数のタイムスロットを考慮できるようになるので，正常トラヒックと感染トラヒックの違いが 1 つのタイムスロットを用いた場合よりもより大きくなると考えられるからである．

既存研究では，市田ら [42] も N-gram を用いており，3-gram を用いた際の正常トラヒックと感染トラヒックにおける，最近傍の共通コードブック番号の遷移の仕方の違いを確認している (図 4.2)．

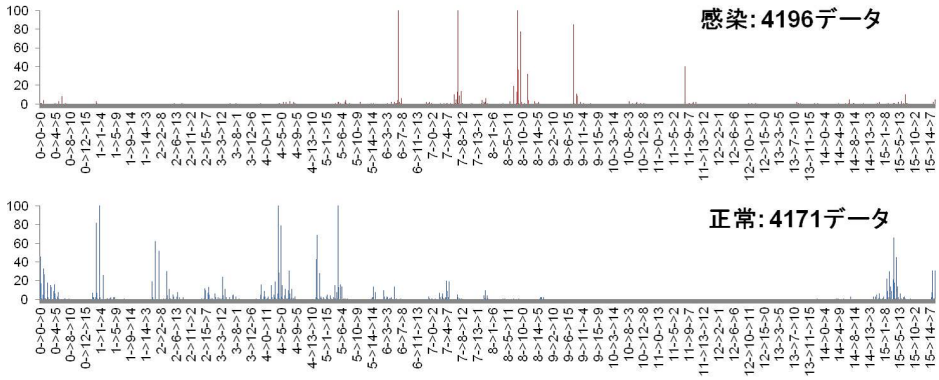

図 4.2　3-gram における正常・感染トラヒックのコードブック番号の遷移の仕方の違い (横軸: 3-gram 遷移パターン，縦軸: 頻度)

上記の理由より，提案手法では時間的な変化の捉え方として N-gram を用いた．

4.1.3　トラヒックデータへの N-gram の適用方法

本項では，提案手法における N-gram のトラヒックデータへの適用方法について説明する．

コードブックを用いて参照されるコードブック番号

既存研究における時間的な変化の捉え方として，特徴量のそのままの値を連続的に取り扱う手法がある [38]．しかし，特徴量のそのままの値を取り扱う場合，特徴量の僅かな変化も遷移と見なされてしまい遷移の解釈の仕方が難しくなってしまう．

そこで，本研究では，ベクトル量子化によって作成した共通コードブックを用いて参照されるコードブック番号を使って大まかに遷移を捉える．すなわち，正常トラヒックでも感染トラヒックでも参照できる共通コードブックを予め用意しておき，あるサンプルに対して，特徴空間上で最近傍のコードブック番号を割り振り，その参照されたコードブック番号を用いて大まかに遷移を捉える．本研究における大まかな遷移の捉え方のイメージを図 4.3 に示す．

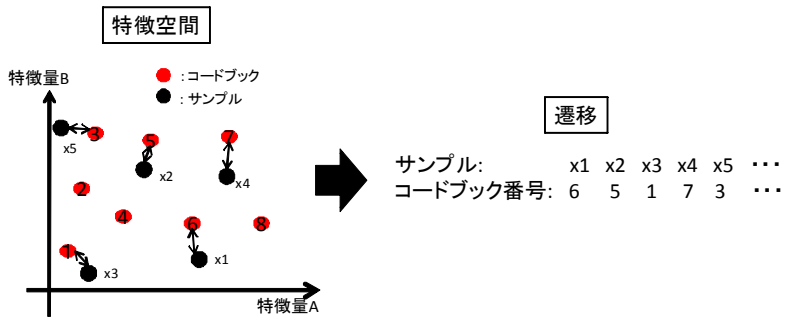

図 4.3　大まかな遷移の捉え方 (イメージ)

コードブックスコア

コードブック番号を用いる場合，大まかな遷移を捉えることができるというメリットがある．しかし，コードブック番号は質的変数かつ，名義尺度であるため，コードブック番号そのものを計算で扱いづらいという問題がある．そこで，コードブック番号のように遷移を大まかに捉えやすく，かつ計算の際にも扱いやすくなるようにコードブック番号を別の値に変換することを考える．提案手法では，コードブック番号をコードブックスコアに変換する．コードブック番号をコードブックスコアに変換する方法を以下に示す．

1. 正常サンプルと感染サンプルを用いてベクトル量子化を行い，共通コードブックを作成する．そして，全ての共通コードブックに対して，各コードブックに含まれる全サンプルに対する感染サンプルの比率 (%) を算出し，その比率をコードブックスコア s_{cdb} とする．
2. 入力サンプル x_i の最近傍のコードブックを探し，最近傍コードブックのコードブックスコア s_{cdb} を入力サンプル x_i に割り当てる．

3. 入力サンプル x_i と最近傍コードブックとの量子化誤差を計算し，その量子化誤差をベクトル量子化の際に計算してある各コードブックの平均 μ と分散 σ^2 を用いて標準化する．標準化した量子化誤差 z_i の算出方法を式 (4.1) に示す．

$$z_i = \frac{x_i - \mu}{\sigma} \tag{4.1}$$

4. z_i を用いて，入力サンプル x_i に割り当てられたコードブックスコア s_{cdb} を補正する．補正後のコードブックスコア t_i を式 (4.2) に示す．

$$t_i = s_i \times \exp(-\frac{z_i^2}{2}) \tag{4.2}$$

上記の方法で，タイムスロット毎にコードブックスコア t_i を割り当て，コードブックスコア列を作成する．そして，コードブックスコア列を N-gram で切り出し，N 次元ベクトルとして扱い，1 つの N 次元ベクトル毎に識別を行う．

4.2　最近傍密度推定法

本節では，提案手法のもう一つの要素技術である最近傍密度推定法について説明する．まずは，最近傍密度推定法の概要について説明し，その後，最近傍密度推定法を用いる理由について説明する．

4.2.1　概要

最近傍密度推定法は，ノンパラメトリックな密度推定方法の 1 つである [51]．最近傍密度推定法において，ある点 x の確率密度は式 (4.3),(4.4) で表される．

$$\hat{p}(x) = \frac{k}{nV} \tag{4.3}$$

$$V = \frac{\pi^{\frac{d}{2}} r^d}{\Gamma(\frac{d}{2}+1)} \tag{4.4}$$

ただし，n は全標本数，d は x の次元数，V はある点 x を中心とし標本を k 個含むような超球 (半径 r) の体積である (Γ はガンマ関数である (式 (4.5))．

$$\Gamma(x) = \int_0^\infty e^{-t} t^{x-1} dt \tag{4.5}$$

最近傍密度推定法では，k を変えることによって，得られる確率密度関数の滑らかさを調整することができる．言い換えれば，k を適切に決めなければ得られる確率密度関数推定量 $\hat{p}(x)$ は真の確率密度関数 $p(x)$ の良い近似にはならない．

4.2.2　最近傍密度推定法を用いる理由

ノンパラメトリックな密度推定方法の別の手法としてカーネル密度推定法がある．カーネル密度推定法，およびノンパラメトリックな密度推定法についての詳細を付録 C に示す．提案手法では，カーネル密度推定ではなく，最近傍密度推定法を用いた．それは，最近傍密度推定法がパターン認識と相性が良いからである．

3 章での検討により，正常なトラヒックとマルウェアに感染した後のトラヒックは特徴空間上である程度は分布が離れることがわかった．すなわち，特徴空間上における，識別対象のデータと正常サンプルとの距離 r_n，および感染サンプルとの距離 r_m は識別に有効な指標である可能性が高い．特徴空間上における距離 r_n，r_m を識別に用いる場合，カーネル密度推定より最近傍密度推定法の方が相性が良い．よって，提案手法では，確率分布作成に最近傍密度推定法を用いた．

また，カーネル密度推定法は高次元データへの対応が難しいという課題がある [52]．最近傍密度推定法では次元数が大きくなっても適応が可能というメリットがある．本研究では，N-gram の確率分布を作成するので，N の値によってデータの次元数が変わってくる．そのため，次元数に柔軟に対応できるような最近傍密度推定法を用いることはメリットがあると考えられる．

上記のように最近傍密度推定法を用いる理由はある．しかし，提案手法にカーネル密度推定法を用いた場合の精度比較は今後の検討課題の 1 つである．

第 5 章

マルウェア感染トラヒック検知実験

本章では，前章で述べた提案手法を用いたマルウェアトラヒック検知実験の結果を報告する．まず，本実験の概要について説明し，実験諸元を述べる．その後，検知実験の結果と考察を示す．

5.1 実験概要

提案手法の有効性を確認するために，評価実験を行った．実験の概要を以下に述べる．

提案手法の評価実験は学習部と識別部に分かれる．

学習部では，正常なトラヒックデータとマルウェアに感染した後のトラヒックデータを用いて，N-gram の確率分布を推定する．まず，学習用の正常トラヒックと感染トラヒックに対してタイムスロット毎に特徴抽出を行う．そして，タイムスロット毎に抽出した特徴量に対して，4.1.3 節で説明した方法を用いてコードブックスコアを割り当てる．コードブックスコアを割り当てた後は，割り当てたコードブックスコア系列を N-gram として切り出し，N 次元ベクトルとして扱う．最後に切り出した N 次元ベクトルを 1 つ 1 つのパターンと見なし，そのパターン群を用いて正常と感染の確率分布 (密度分布) を最近傍密度推定法を用いて推定する．

識別部では，予め正常と感染のラベルが付けられたテストデータに対して，学習部で推定した確率分布を用いて識別を行う．まずは，N 次元ベクトルとして切り出すまでは学習部と同じだが，識別部ではその後，学習部で推定した確率分布を用いて，正常のスコアと感染のスコアを計算する．提案手法においては，スコアとは確率密度のことである．そして，正常のスコアと感染のスコアの大小を計算し，スコアの大きい方を判定結果とする．最後に，正しく識別できているか判定する．識別精度の指標としては, 3.2.4 項で説明した TNR,TPR を用いる．

提案手法のフローチャート，および識別方法のイメージをを図 5.1，図 5.2 に示す．

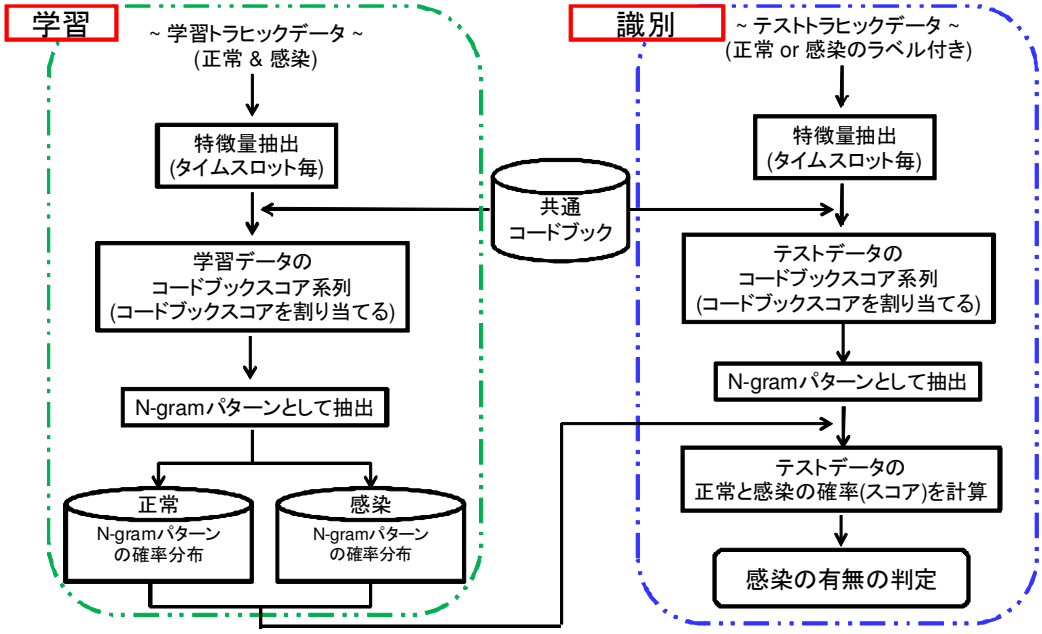

図 5.1　提案手法のフローチャート

5.2　実験諸元

5.2.1　実験データ

マルウェア感染トラヒック検知実験では，正常時の通信トラヒックデータとマルウェアに感染した後のトラヒックデータを用意する必要がある．さらに，提案手法では，共通コードブック作成用トラヒックデータ，確率分布作成用トラヒックデータ，テスト用トラヒックデータの 3 種類のトラヒックデータが必要である．

- 正常トラヒックデータ
  正常トラヒックとしては，前田ら [53] が用いたトラヒックを用いた．具体的には，ある PC で 14 種類のサービスをそれぞれ 10 回分発生させたトラヒックを用いた．キャプチャは時期 A と時期 B に行った．時期 A のトラヒックは 2010 年 6 月に取得したトラヒックであり，時期 B のトラヒックは 2010 年 7 月に取得したトラヒックである．14 種類のサービスの内容を表 5.1 に，トラヒック取得環境，および前処理については付録 D に示す．
- 感染トラヒックデータ
  感染した後のトラヒックデータとして，3.2.3 項で使用した CCC2010,CCC2011 を用いた．

用いる正常トラヒック，感染トラヒックについてまとめた表を表 5.2 に示す．

5.2　実験諸元

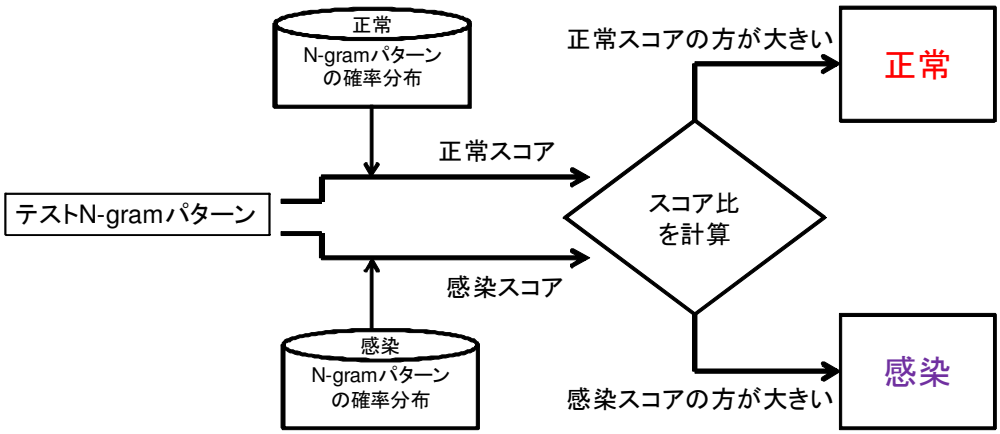

図 5.2　識別方法のイメージ

表 5.1　正常時トラヒックデータ

アプリケーション区分	ネットワークサービス (14 種類)
音声通信	音声ストリーミング 1(TCP と RTP 型)
	音声ストリーミング 2(TCP のみ型)
映像通信	YOUTUBE
	ニコニコ動画
	Peercast
テキスト通信	Skype(chat 機能使用時)
	MSN Messenger
	メール送信
	メール受信
ファイル通信	Bittorrent ファイル転送 (HTTP)Vector 使用
	ファイル転送 (FTP)
オンラインゲーム	FEZ：アクションＲＰＧ
	PANGYA：ゴルフ

表 5.2　評価実験に用いたトラヒックのまとめ

	正常トラヒック	感染トラヒック
共通コードブック作成	時期 A の 14 種類サービスの 5 回分	CCC2010 の honey001
確率分布作成	時期 A の 14 種類サービスの 5 回分，時期 B の 14 種類サービスの 5 回分	CCC2010 の honey002，CCC2011 の honey001
テストデータ	時期 B の 14 種類サービスの 5 回分	CCC2011 の honey002

また，用いた正常トラヒック，感染トラヒックのサンプル数 (タイムスロット数) についてまとめた表を表 5.3 に示す.

表 5.3　各データのサンプル数

	正常トラヒック	感染トラヒック
共通コードブック作成	22365 サンプル	26213 サンプル
確率分布作成	27263 サンプル	52626 サンプル
テストデータ	4996 サンプル	11721 サンプル

5.2.2　特徴量

提案手法では，3 章を参考に，パケットサイズの平均，TCP パケット中の SYN パケット割合，TCP パケット中の ACK パケット割合の 3 種類の特徴量を組み合わせて用いた．さらに，各特徴量に対して正規化を行った．パケットサイズの平均に関しては，正常、感染各トラヒックに対して正常、感染トラヒックの取得環境の MTU(Maximum Transmission Unit) 値で割り，その後 100 倍して正規化した．TCP パケット中の SYN パケット割合，TCP パケット中の ACK パケット割合に関しては，正規化として割合を 100 倍して%の値に変換した.

5.2.3　パラメータ

評価実験におけるパラメータについてまとめた表を表 5.4 に示す.

タイムスロット幅，ベクトル量子化レベル数

市田ら [42] はタイムスロット幅 1 秒でベクトル量子化レベル数 16 として実験を行っていた．さらに，3.3.2 項の結果も踏まえ，タイムスロット幅 1 秒，ベクトル量子化レベル数 16 として評価実験を行った.

最近傍密度推定のパラメータ

4.2.1 項で言及した通り，適切な k の値を決めなければ得られる確率密度関数推定量 $\hat{p}(x)$ は真の確率密度関数 $p(x)$ の良い近似にはならない．k の値が小さすぎるとオーバーフィッティングしてしまい，k の値が大きすぎると密度曲線が滑らかになりすぎてしまう．そこで，本評価実験では，適切な k の値を探すために，$k = 50, 100, 500, 1000$ と値を変えて実験を行った.

表 5.4 実験諸元 (パラメータ)

タイムスロット幅	1 秒
特徴量	パケットサイズの平均 TCP パケット中の SYN パケット割合 TCP パケット中の ACK パケット割合
共通コードブッ作成のためのアルゴリズム	LBG + splitting
ベクトル量子化レベル数	16
N-gram の「N」の値	N=1 ～ 9
最近傍密度推定法の k の値	50,100,500,1000

5.3 比較手法概要

提案手法である，コードブックスコア，N-gram，最近傍密度推定法の有効性を確認するための比較手法について説明する．

- 比較手法 (1)
  提案手法では，タイムスロット毎に特徴量を抽出しているが，既存研究ではフロー毎に特徴量を抽出することが専らである．そこで，侵入検知，異常検知における既存研究でよく用いられているフローと GMM を使った手法と精度比較を行い，提案手法の精度が高いことを示す．具体的には，フロー毎に特徴量を抽出し，GMM で正常フローと感染フローの確率分布を作成し，識別を行う手法である．識別を行う際には，正常モデルと感染モデルの事後確率の大小を用いる．これは，2.5.2 項で挙げた M.Bahrololum ら [34] の手法と同様の手法である．比較手法 (1) の概要を図 5.3 に示す．
  なお，GMM の学習には提案手法における「共通コードブック作成」のためのトラヒックを用い，テストデータには提案手法における「テストデータ」のためのトラヒックを用いた．また，GMM の混合数は 16,32,64,128,256,512,1024 と変化させて比較実験を行った．
- 比較手法 (2)
  コードブックスコアの有効性を示すために，コードブックスコアを用いないで提案手法を用いる手法の精度を調べる．具体的には，抽出した特徴量に何も変換を加えないで最近傍密度推定法を用い，密度を計算し，識別を行う手法である．提案手法との違いは，コードブックスコアへの変換を用いず特徴量をそのまま用いること，密度推定の際の次元数 d が 3 であることだけである．比較手法 (2) の概要を図 5.4 に示す．
- 比較手法 (3)
  提案手法では，正常と感染の確率分布の計算に最近傍密度推定法を用いた．最近傍密度推定法を用いることの有効性を示すために，確率分布を GMM で作成する手法と比較を行う．比

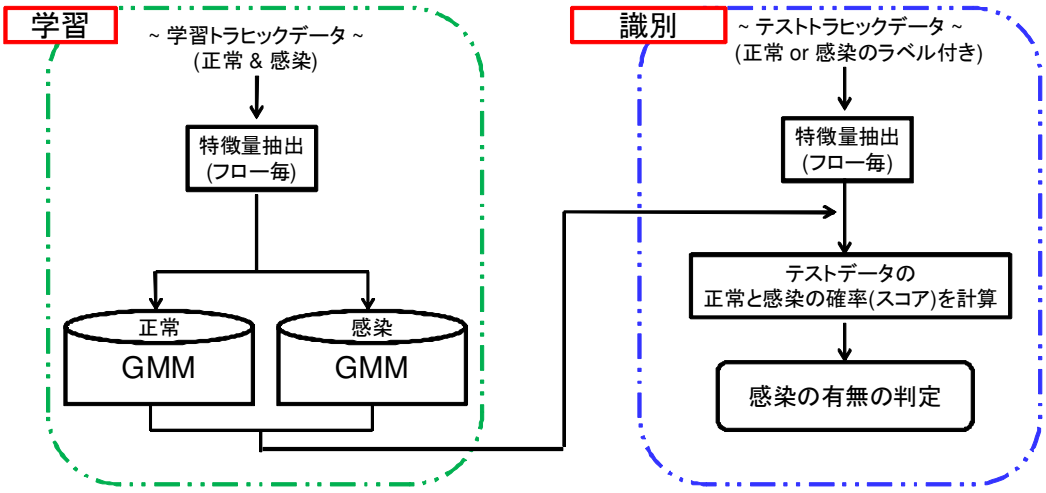

図 5.3　比較手法 (1) のフローチャート

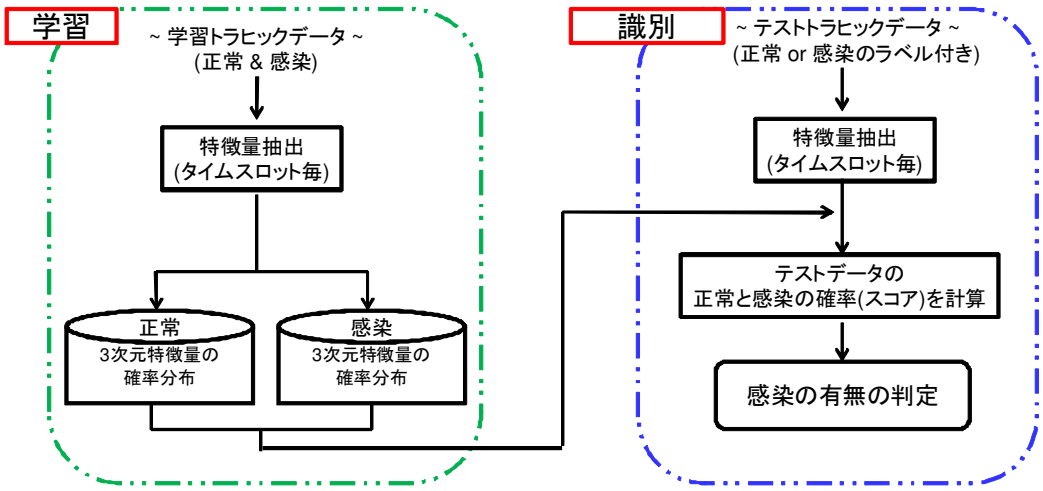

図 5.4　比較手法 (2) のフローチャート

較手法 (3) の概要を図 5.5 に示す.

なお，コードブックの作成には提案手法における「共通コードブック作成」のためのトラヒックを用い，GMM の学習には提案手法における「確率分布作成」のためのトラヒックを用い，テストデータには提案手法における「テストデータ」のためのトラヒックを用いた．また，GMM の混合数は 16,32,64,128,256,512,1024 と変化させ，N-gram の N は N=1 の場合と，提案手法で TNR と TPR の平均が最大になるときの N を用いた.

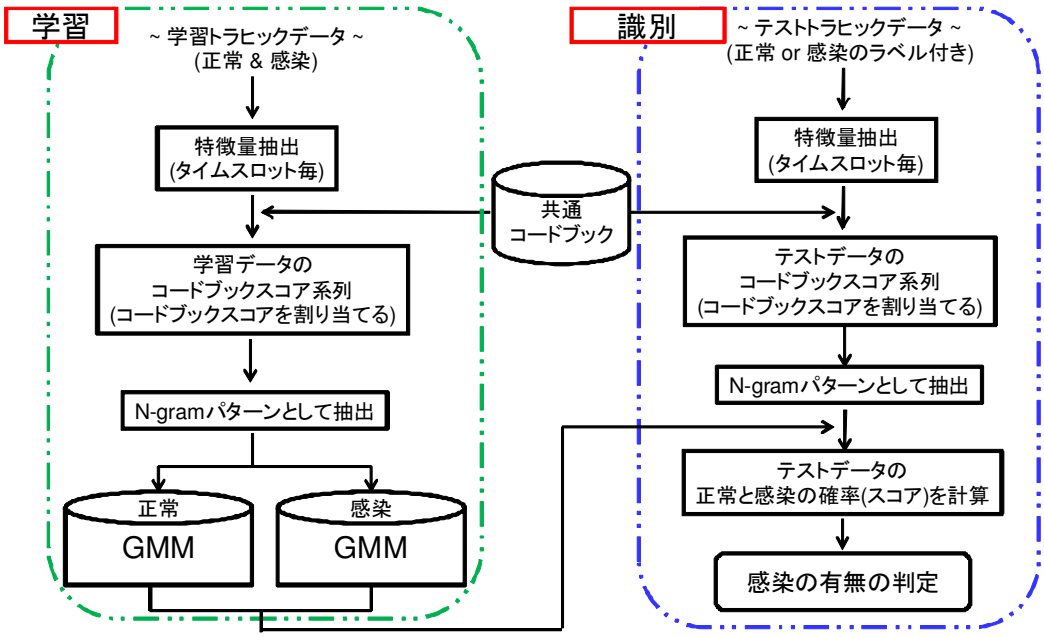

図 5.5　比較手法 (3) のフローチャート

なお，各比較手法において用いる特徴量は提案手法と同じであり，比較手法における GMM はオープンソースの統計解析向けフリーソフトの R の Mclust パッケージを用いて実装した [54].

5.4　実験結果

各手法において，TNR と TPR の平均が最大となる場合の識別率とそのパラメータについてまとめた表を表 5.5 に示す．提案手法において，N=9 のときに TNR と TPR の平均が最大となったので，比較手法 (3) では N=1 の場合と N=9 の場合の結果を表 5.5 に示した．

表 5.5　各手法の識別率のまとめ

	提案手法	比較手法 (1)	比較手法 (2)	比較手法 (3)	比較手法 (3)
パラメータ 混合数 (正常，感染)	k=50,N=9	 (64,64)	k=50	N=1 (59,11)	N=9 (256,32)
TNR(%)	94.43	82.32	88.59	79.06	86.24
TPR(%)	96.07	72.19	90.78	79.78	96.80
TNR と TPR の平均 (%)	95.25	77.26	89.68	79.42	91.52

表 5.5 より，TNR と TPR のバランスが良く，さらに TNR と TPR の平均が最も高かったのは提案手法である．よって，識別率の観点から提案手法の有効性を確認できる．

5.5　考察

5.5.1　識別結果

各手法の識別結果を基に提案手法の有効性を確認する．

- 提案手法
  パラメータを変化させたときの識別率の変化を図 5.6 に示す．

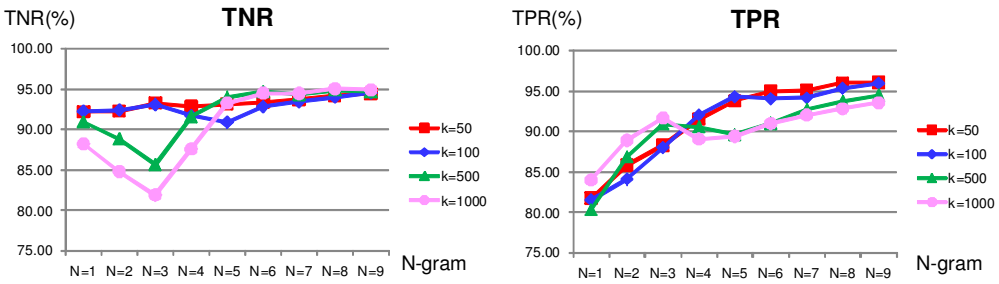

図 5.6　提案手法における識別率の変化

図 5.6 より，TNR に関しては，N の値によらず高い値をとっており，TPR に関しては，N を大きくすると TPR が高くなっており，k=50 において N=1 の場合に比べ N=9 の場合では TPR が約 15%上がっていることがわかる．以上のことから，N-gram として切り出す手法の有効性が確認できる．
それは，N-gram を用いることで，正常サンプルと感染サンプルの分布をより引き離すことができているからである．基本的には，コードブックスコアへの変換により，正常サンプルは小さい値，感染サンプルは大きい値に変換される．しかし，正常サンプルにも感染サンプルにも外れ値があり，正常サンプルでも大きいコードブックスコアの値，感染サンプルでも小さいコードブックスコアの値をとる可能性がある．学習データ，テストデータにおける外れ値の割合を表 5.6 に示す．

表 5.6　正常サンプルと感染サンプルにおけるコードブックスコアの外れ値の割合

	正常	感染
学習データ	4.43%(1208 サンプル)	11.86%(6242 サンプル)
テストデータ	9.13%(456 サンプル)	19.24%(2255 サンプル)

N=1 の場合は，コードブックスコアが大きい正常サンプル，コードブックスコアが小さい感染サンプルの識別を誤っている場合が多い．しかし，N=9 では，複数のタイ

ムスロットを連続的に扱っているため，1 つの外れ値の影響が小さくなっている．すなわち，N-gram として切り出し N 次元ベクトルとして扱うことで，正常サンプルと感染サンプルの特徴の違いを大きくする．そして，正常サンプルと感染サンプルの分布される領域をより引き離し，外れ値の影響を小さくし，誤識別を減らしている．よって, N-gram として切り出し, N 次元ベクトルとして扱うことは有効であると考えられる．

また，最近傍密度推定法におけるパラメータ k については，TNR に関しては $k = 50, 100, 500, 1000$ において識別率に大差はないが，TPR に関しては，k を小さくした方が識別率が高くなっている．これは，k を大きくし過ぎると，超球の半径が大きくなりすぎてしまい，特徴のある密な部分を飛び越えてしまうからだと考えられる．N=2 における正常サンプルと感染サンプルの分布の様子を図 5.7，図 5.8 に示す．図 5.7，図 5.8 中の cdb_score1 の軸が N-gram(bi-gram) の第一成分，cdb_screo2 の軸が N-gram(bi-gram) の第二成分，縦軸 (軸のラベルが何もついていない軸) が頻度を表している．

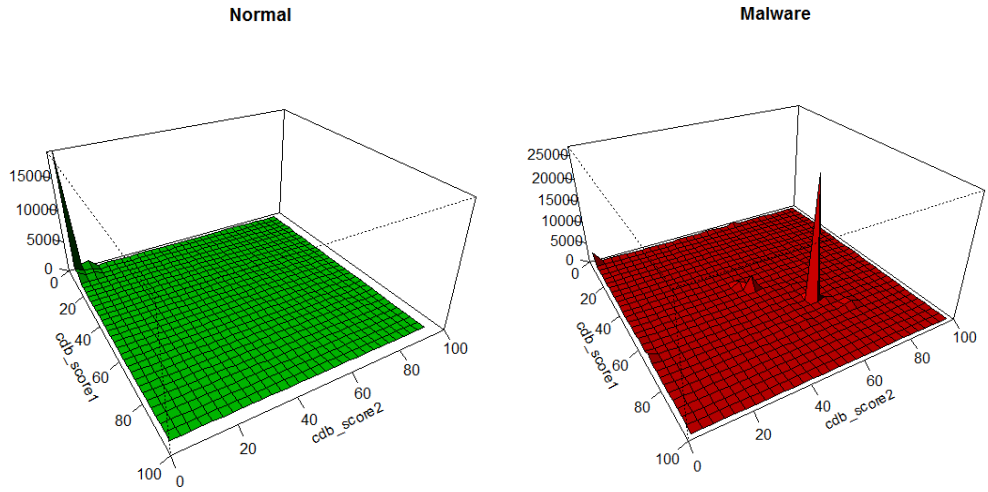

図 5.7　N=2 における学習データの俯瞰図

- 比較手法 (1)
  パラメータを変化させたときの識別率の変化を図 5.9 に示す．
  比較手法 (1) は，提案手法に比べて TPR が低くなった．これは，感染トラヒックに SYN スキャンのようなフローと，パケットサイズの平均が約 5%で TCP パケット中の SYN パケット割合と ACK パケット割合が 0%のフローが大半だったことが原因と考えられる．すなわち，上記 2 種類のフロー以外は正常なフローと判定されてしまっているため，TPR が提案手法より低くなっていると考えられる．

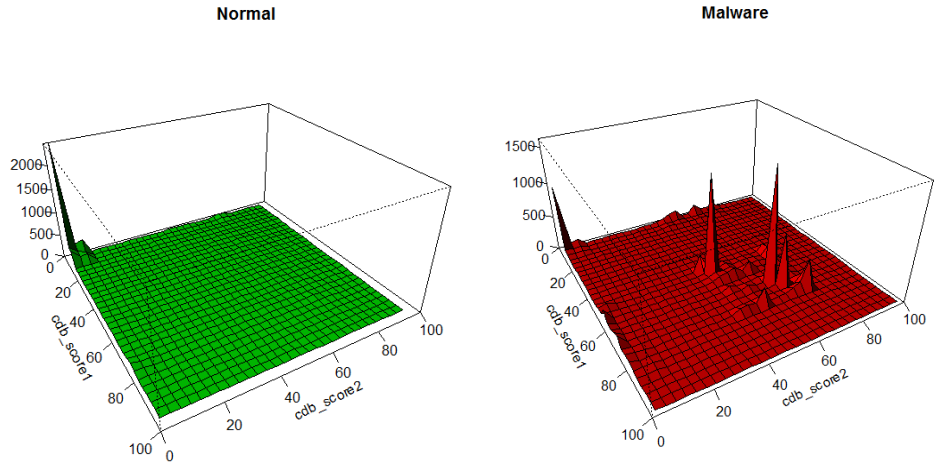

図 5.8　N=2 におけるテストデータの俯瞰図

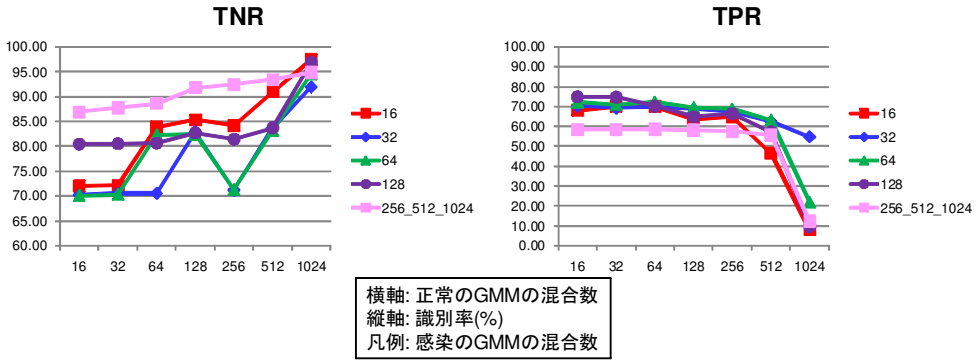

図 5.9　比較手法 (1) における識別率の変化

また，正常の混合数を上げると TNR は高くなり，TPR は低くなっている．さらに，比較手法 (1) においては，TNR と TPR のバランスが提案手法ほど良くない．マルウェア感染検知において，TNR と TPR が両方高いことは重要である．よって，侵入検知，異常検知における既存のよく用いられている比較手法 (1) に比べ，提案手法は有効であるといえる．

- 比較手法 (2)

  パラメータを変化させたときの識別率の変化を図 5.10 に示す．

  TNR に関しては，コードブックスコアを，TPR に関しては，特徴量をそのまま用いる方が良くなっている．これはコードブックスコアへの変換方法による影響によるものであると考

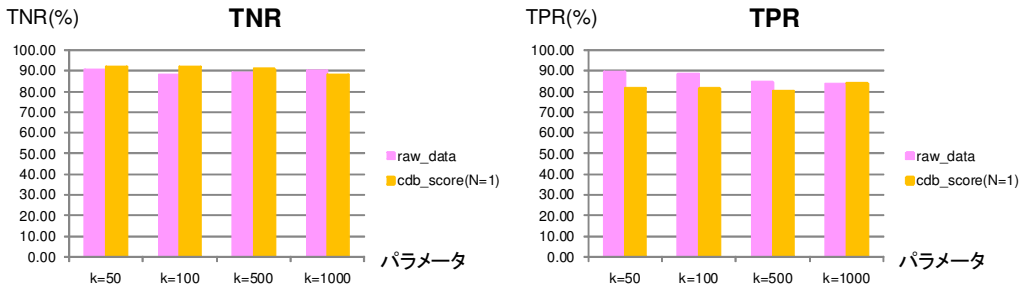

図 5.10　比較手法 (2) における識別率の変化

えられる．コードブックスコアに変換すると，ほとんどの正常サンプルは値の小さい領域に分布されるため誤識別が少ない．しかし，感染サンプルは，大体は値の大きい領域に分布するが，値の小さい領域に分布する場合もしばしばあり，値の小さい領域に分布されたサンプルは誤識別される場合が多いからである．提案手法の N=1 の場合における，確率分布作成用データとテストデータのコードブックスコアのヒストグラムを図 5.11, 図 5.12 に示す．

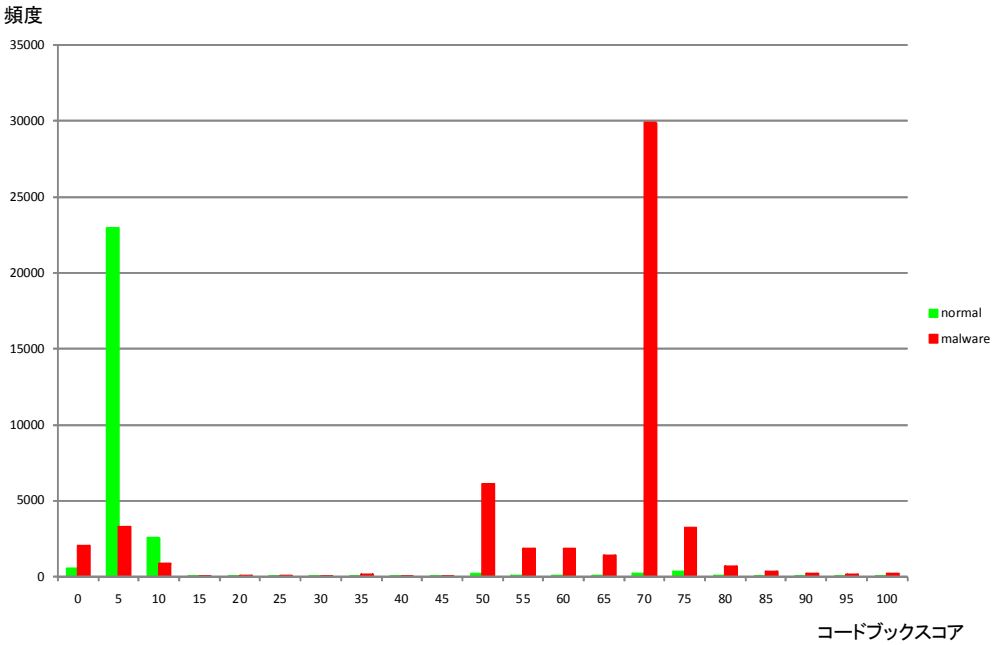

図 5.11　提案手法 (N=1) における確率分布作成用データのコードブックスコアのヒストグラム

比較のため，比較手法 (2) の確率分布作成用データにおける，特徴量を 2 つ組み合わせた際の二次元散布図を図 5.13, 図 5.14, 図 5.15 に示す．
図 5.11, 図 5.12 と図 5.13, 図 5.14, 図 5.15 の比較より，

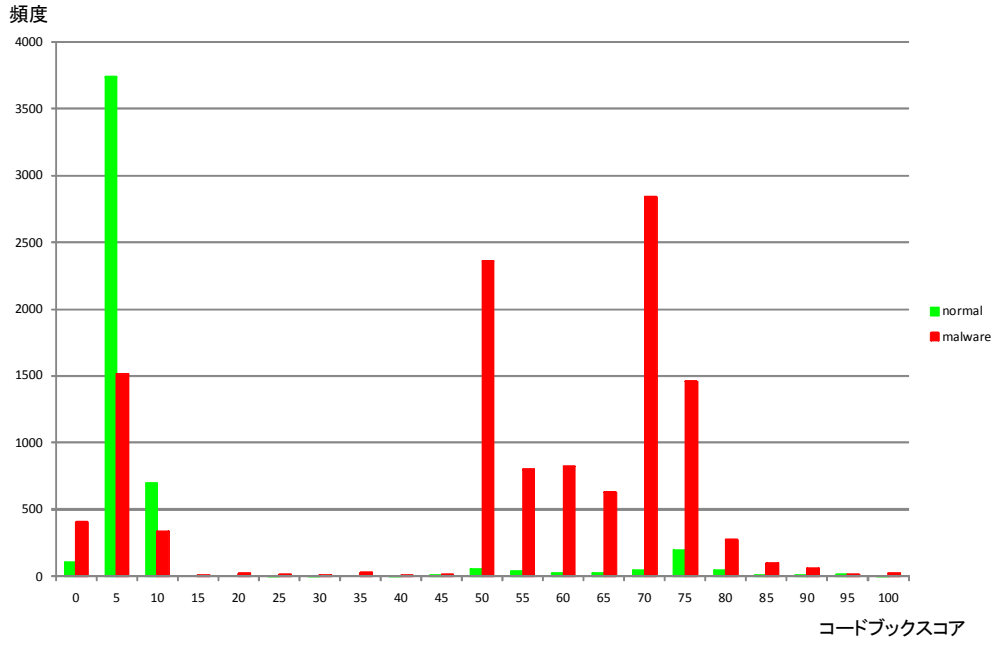

図 5.12　提案手法 (N=1) におけるテストデータのコードブックスコアのヒストグラム

特徴量の値をそのまま用いる方がコードブックスコアを用いるより正常サンプルと感染サンプルで重なる部分が多いことが直観的に確認できる．

コードブックスコアを用いた場合でも，特徴量 3 種類をそのまま用いた場合でも，識別率に大差はない．しかし，特徴量 3 種類をそのまま用いる場合は三次元であり，コードブックスコアは一次元である．三次元データよりも一次元データの方が可視化しやすく，さらにデータの取り扱いが容易である．すなわち，N-gram などのように連続的に値を取り扱う場合には一次元のデータの方が扱いやすい．よって，扱いやすくなる上に，識別率もあまり落とさないコードブックスコアへの変換は有効である可能性がある．

- 比較手法 (3)
  - N=1 の場合

    N=1 におけるパラメータを変化させたときの識別率の変化を図 5.16 に示す．

    図 5.16 より，各混合数の組において，TNR,TPR ともに 80%を超えるものがなく，さらに TNR と TPR にトレードオフの関係がみられる．これは，混合数を上げ過ぎるとオーバーフィッティングしてしまい，混合数を小さくし過ぎるとピーク以外の領域は他のクラスのデータとみなされてしまうからである．TNR が最大のとき，TPR が最大のとき，および TNR と TPR の平均が最大のときの確率分布を図 5.17，図 5.18，図 5.19 に示す．
  - N=9 の場合

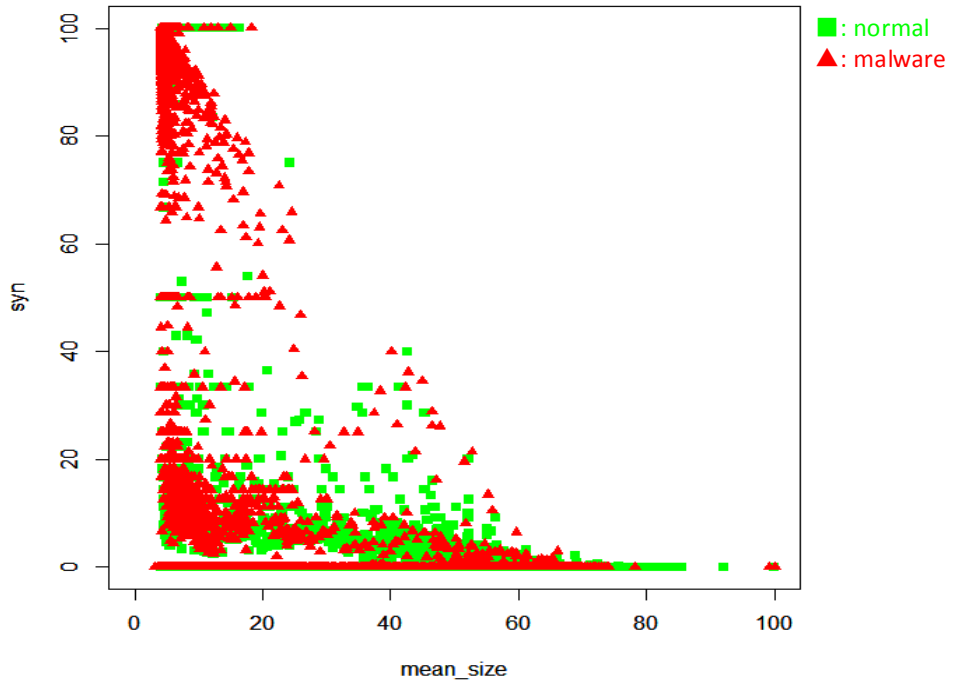

図 5.13　パケットサイズの平均と TCP パケット中の SYN パケット割合を用いた場合の散布図

N=9 におけるパラメータを変化させたときの識別率の変化を図 5.20 に示す．
N=9 の場合は，N=1 の場合に比べ，N-gram の効果により正常サンプルと感染サンプルの分布をある程度引き離すことができているため，識別率は上がっている．しかし，N=1 の場合と同様に TNR と TPR にトレードオフの関係がみられる．N=9 の場合は，TNR と TPR の平均の最大は約 92% であるが，提案手法よりは低い値となっている．また，N-gram を用いると，N の数だけ次元数が高くなり，分布が複雑になる．そのため，分布に仮定を置くパラメトリックな手法より，分布を仮定しないでデータに合わせて密度推定を行うノンパラメトリックな手法の方が有効であり，かつ N-gram と相性が良いと考えられる．さらに，GMM を用いる場合は，正常の確率分布と感染の確率分布では必要な状態数 (混合数) がそれぞれ異なり，チューニングが難しいという課題があるが，最近傍密度推定法の場合は正常，感染で共通の k の値を設定すれば良いだけなのでロバストな識別器の作成に適している．よって，最近傍密度推定法を用いた確率分布作成は有効であると考えられる．

5.5.2　処理時間

本研究では，インターネット経由でのマルウェアによる感染を防止することを 1 つの目標としている．そこで，いち早く感染の有無を判定できることは重要である．すなわち，感染検知

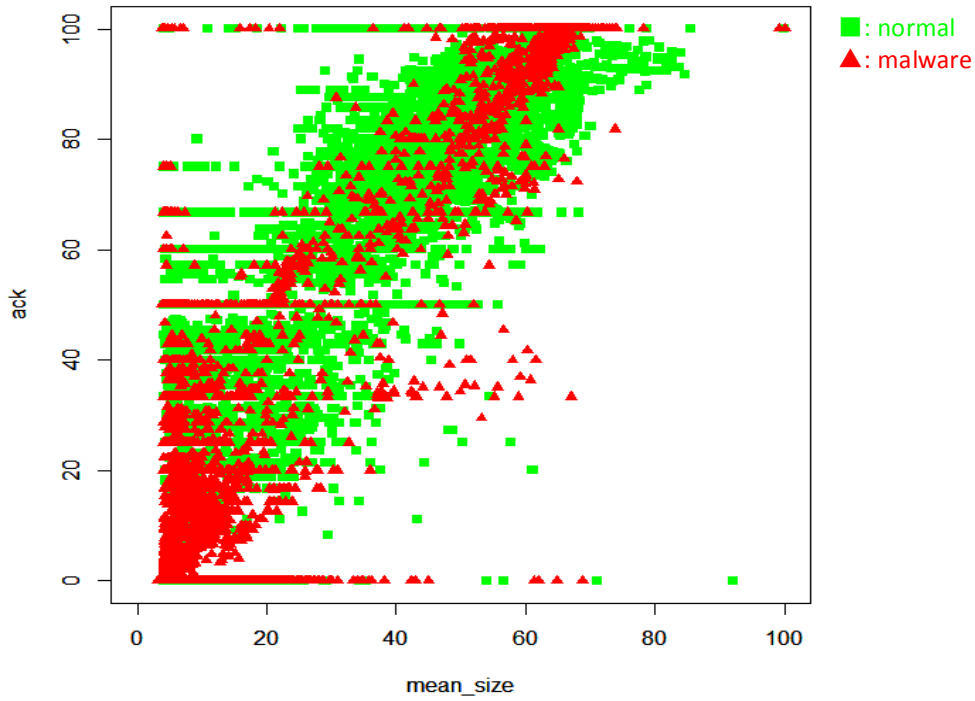

図 5.14　パケットサイズの平均と TCP パケット中の ACK パケット割合を用いた場合の散布図

においてリアルタイム性は 1 つの評価尺度である．本項では, 提案手法の処理時間について考える．

提案手法では，確率分布の作成のために最近傍密度推定法を用いている．最近傍密度推定法では，ある注目点 x と全ての学習データとの距離を計算し，その後に第 k 近傍を探すためにソートを行う必要がある．学習データの量が莫大になればソートにかかる処理時間が莫大になるため，確率分布を作成するのに時間がかかってしまう．よって，提案手法でリアルタイムに確率分布を逐次更新しながら識別を行うのは処理時間の問題で難しい可能性がある．しかし，確率分布をオンラインで逐次更新せず，予めオフラインで作成しておく場合は，処理時間に関する問題は全くない．オフライン検知の処理時間に関わるパラメータは N-gram の N だけである．確率分布を逐次更新しながらリアルタイムで検知を行うスキームを考えることは，今後の課題の 1 つである．

5.5.3　攻撃耐性

近年，正常トラヒックと感染トラヒックの通信挙動が似てきているという問題がある．すなわち，様々なマルウェアに対応できるロバストな感染検知手法を提案することは重要である．本項では，提案手法の攻撃耐性について述べる．

提案手法では，正常トラヒックと感染トラヒックを N-gram として捉えることで，時間的な変

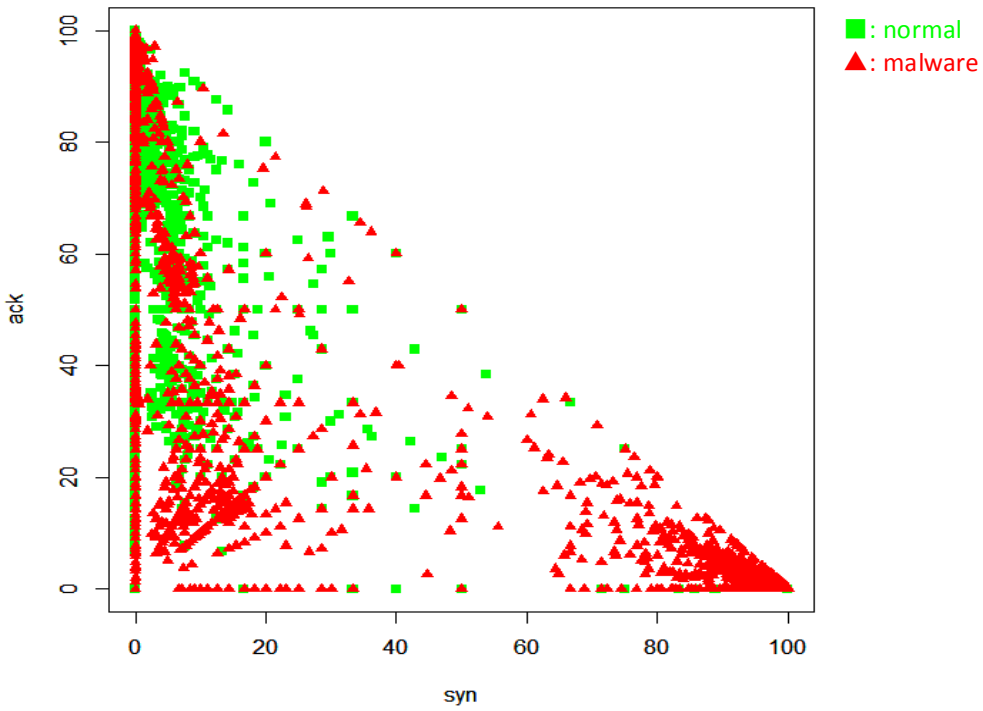

図 5.15　TCP パケット中の SYN パケット割合と ACK パケット割合を用いた場合の散布図

化を考慮し，N-gram の確率分布を用いて検知を行っている．しかし，攻撃者がシステムにおける感染検知方法を知っていて，限りなく正常トラヒックの振る舞いに近い動作をするマルウェアを用いて攻撃した場合，正常トラヒックと誤識別してしまう可能性は十分にある．上記のようななりすまし攻撃への耐性を強めるには，以下の方法が考えられる．

攻撃耐性を強めるために，N-gram の N を大きくし，より長い時間で時間的な変化を観察して識別を行う方法が考えられる．なりすましをしても，マルウェアによる攻撃の際にマルウェア特有の振る舞いが現れる可能性は高い．そこで，長い時間的な変化を観察することで，特徴観察期間にマルウェア特有の挙動が含まれる可能性が高くなるので，N を大きくすることで耐性を強くすることができると考えられる．しかし，5.2.2 項で述べた通り，N を大きくするとリアルタイム性が下がってしまうというトレードオフはある．

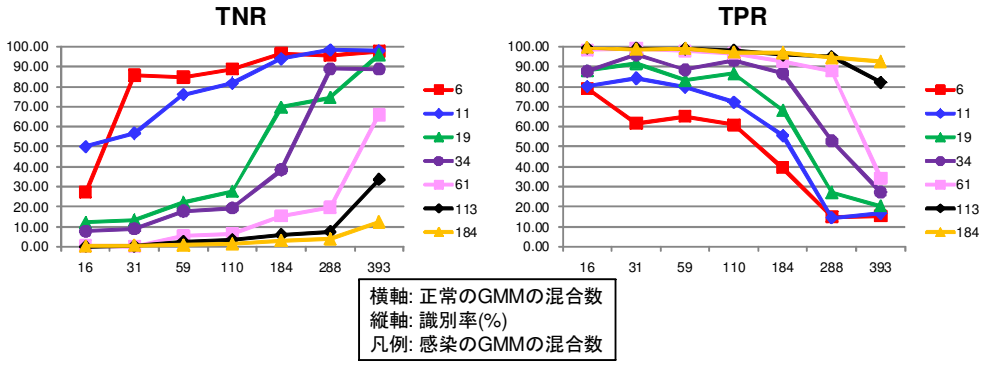

図 5.16　比較手法 (3) の N=1 における識別率の変化

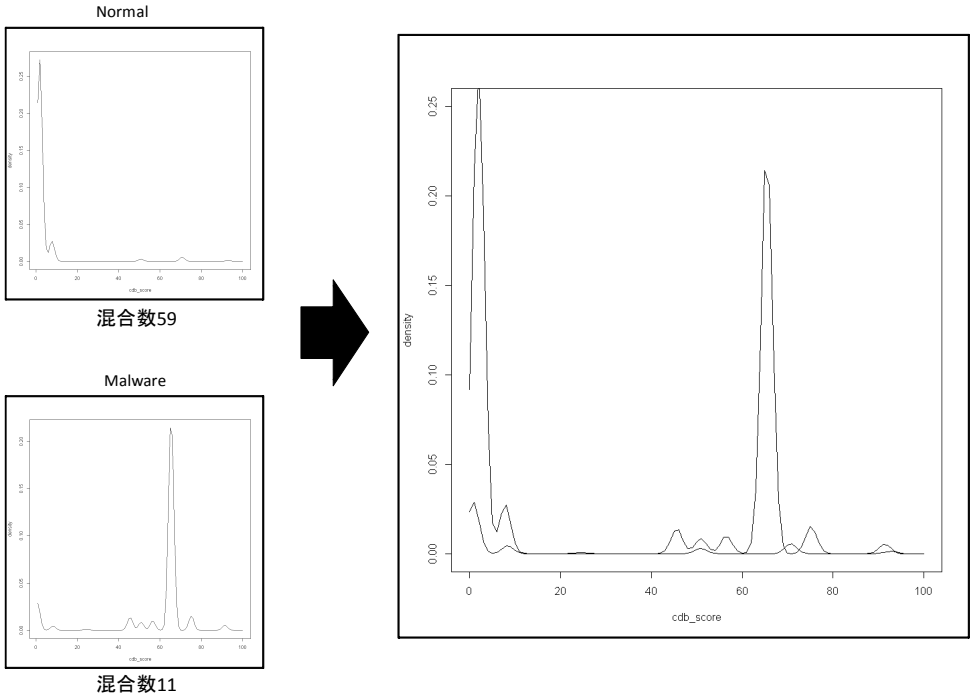

図 5.17　正常混合数 59，感染混合数 11 におけるコードブックスコアの GMM による確率分布

5.5 考察

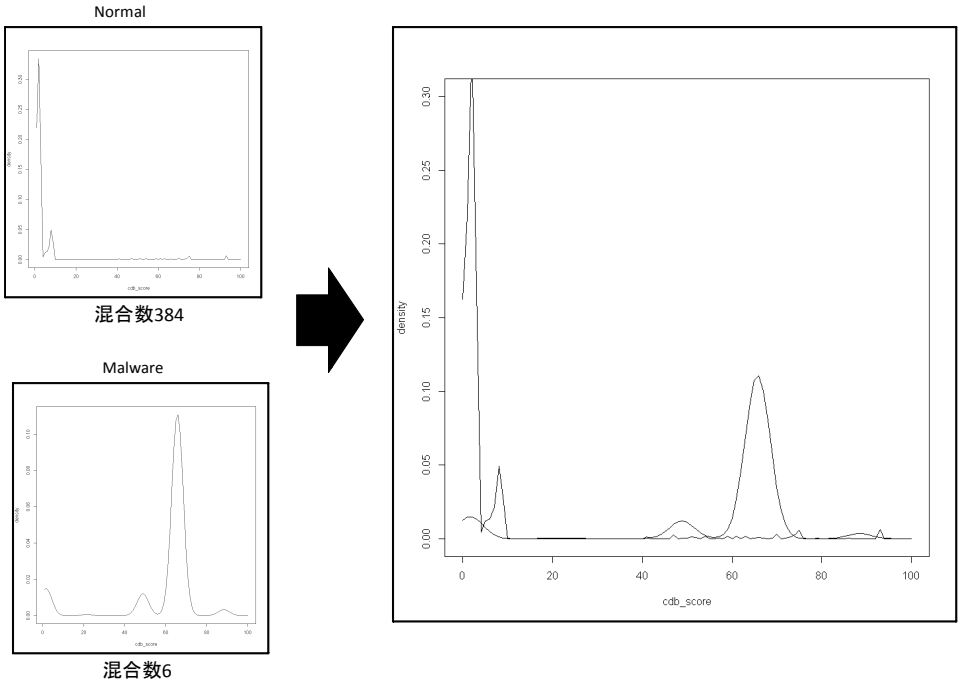

図 5.18　正常混合数 343，感染混合数 6 におけるコードブックスコアの GMM による確率分布

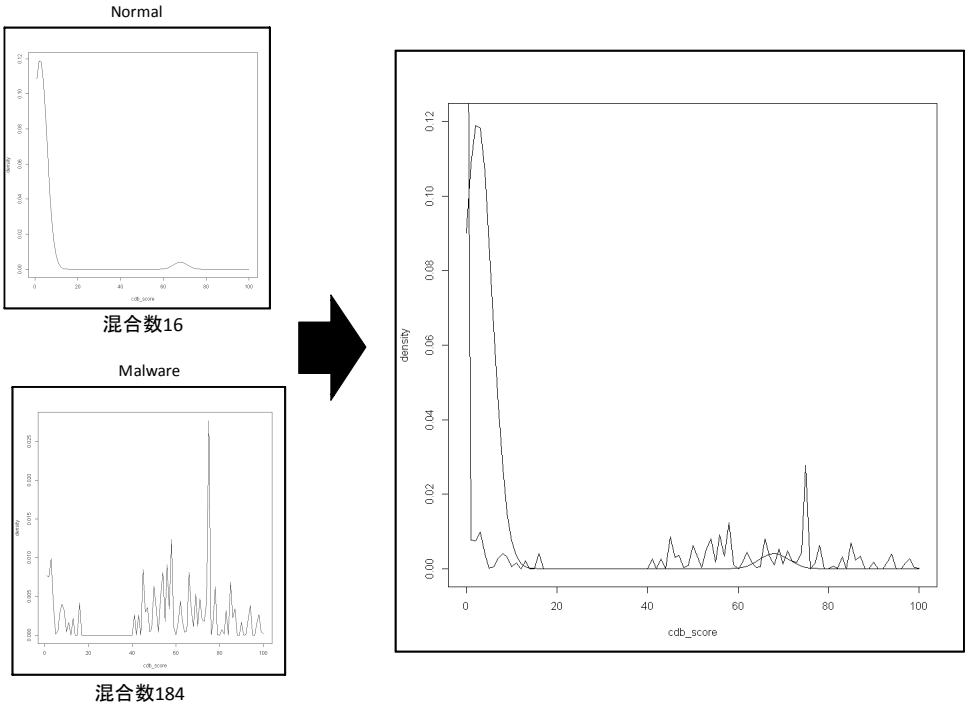

図 5.19　正常混合数 16，感染混合数 184 におけるコードブックスコアの GMM による確率分布

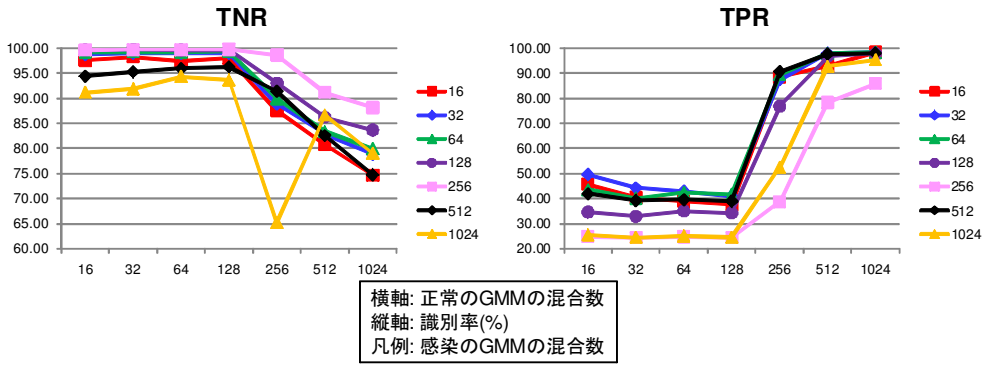

図 5.20　比較手法 (3) の N=9 における識別率の変化

第 6 章

結論

6.1 まとめ

本論文では，まずマルウェアについての基本性質，対策技術などを説明した．その後，既存研究で用いられている特徴量に対して TNR,TPR の観点から評価を行い，特徴量を抽出する時間間隔 (タイムスロット幅) を 1 秒，ベクトル量子化レベル数を 2 か 4 として，パケットサイズの最小 (およびパケットサイズの最小の標準偏差)，SYN パケット数，TCP パケット中の SYN パケット割合，ACK パケット数を特徴量として用いると感染検知に有効であることを確認した．

そして，それらを踏まえた上でマルウェアによる感染を検知するための手段として，N-gram 確率密度を用いた手法を提案した．提案手法の有効性を示すために 3 種類の比較実験も行った．1 つ目の比較手法は，異常検知，侵入検知でよく用いられている，フロー毎に特徴を抽出し，GMM でモデルを用いる手法である．2 つ目の比較手法は，特徴量の値をそのまま用いてノンパラメトリックに確率密度を算出し，識別を行う手法である．3 つ目の比較手法は，N-gram 確率密度を GMM を用いて算出し，識別を行う手法である．

比較手法と精度比較を行った結果，提案手法の有効性を様々な観点から確認できた．具体的には，N-gram により，正常サンプルと感染サンプルの分布を引き離すことができる可能性があること，コードブックスコアを用いることで，識別精度を落とさずに特徴量の次元を削減できたこと，分布を仮定しない最近傍密度推定法を用いた識別は，N-gram とコードブックスコアを用いる場合にはパラメトリックな推定方法より識別率が高くなることがわかった．

6.2 今後の課題

本論文で検討を行った項目に関して，今後の課題を以下に示す．

- バイオメトリクスの分野における技術の応用
  マルウェア感染検知は，正常なトラヒックかマルウェアに感染した後のトラヒックかを識別する 2 クラスパターン分類問題である．代表的な 2 クラスパターン分類問題の分野としてバイオメトリクスがある [55]．バイオメトリクスは生体情報を用いて本人と他人の識別を行う

ものである．マルウェア感染検知において，感染トラヒックを本人，正常トラヒックを他人と考えれば，バイオメトリクスの技術が応用可能である．バイオメトリクスの分野には equal error rate(EER)[56] などの評価尺度があるので，バイオメトリクスにおける既存の評価尺度などをマルウェア感染検知の分野に応用することは今後の課題の 1 つである．

- 正常トラヒックと感染トラヒックの時間的な変化について
  提案手法では，コードブックスコア列を N-gram として扱うことで時間的な変化を捉えた．しかし，評価実験において時間的な変化の捉え方に対する比較を行えていない．そこで，まずは正常トラヒックと感染トラヒックに対して，スペクトル解析などを行い，正常トラヒックと感染トラヒックでは時間的な変化の仕方に違いがあることを確認してから，実際にどのような違いがあるか確認し，その違いを利用した時間的な変化の捉え方を検討する必要があると考えられる．
- HMM との比較
  時間的な変化を捉える手法として音声認識などの分野でよく用いられているのが隠れマルコフモデル (HMM) である [57]．提案手法では，N-gram を用いて時間的な変化を捉えたが，予めモデルを作成しておき，HMM を用いて人間の行動パターンを識別するを行う既存手法などもあるので [58]，HMM は今後の比較対象の 1 つである．
- 共通コードブックの作り方
  提案手法では，正常トラヒックと感染トラヒックの時間的な変化を同じ尺度で扱えるように，共通コードブックを LBG + splitting アルゴリズムで作成した．しかし，既存研究では様々なベクトル量子化アルゴリズムが存在する．一例としては，自己組織化マップ (SOM)[59] や Affinity Propagation(AP)[60] などが挙げられる．これらの手法を用いた場合の精度比較は今後の検討課題である．
- コードブックスコアへの変換方法
  提案手法におけるコードブックへの変換方法では，特徴量の値をそのまま用いる場合とあまり変わらない識別率を保持することができた．しかし，感染サンプルにおいては図 5.11，図 5.12 で示した通り，分布が一か所にまとまっておらず，コードブックスコアの大きい領域と小さい領域にそれぞれピークができてしまった．そこで，正常サンプルはコードブックスコアの小さい領域のみ，感染サンプルはコードブックスコアの大きい領域のみに分布されるようなコードブックスコアへの変換方法を検討することは今後の重要な課題である．
- カーネル密度推定との比較
  提案手法では，最近傍密度推定法によるノンパラメトリックな密度推定を行った．しかし，比較実験として，既存のノンパラメトリックな密度推定法のカーネル密度推定を用いた識別は行っていなかった．最近傍密度推定法による識別精度とカーネル密度推定による識別精度は今後の比較すべきである．
- クロスバリデーションによる最適なパラメータの決定

提案手法では，予めパラメータの値を適当に決めておいて識別を行った．識別精度向上のためには最適なパラメータを知る必要があるので，今後はクロスバリデーションなどを用いて最適なパラメータを知る必要がある．

- ベイズの定理の応用
  提案手法では，単純に 1 つ 1 つの N-gram の確率密度の大小を用いて識別を行っている．そのため，1 つ前の N-gram の情報を考慮した識別方法にはなっていない．これは，正常トラヒックと感染トラヒックの通信挙動が似てきている近年では攻撃耐性の低い手法である．そこで，ベイズの定理を応用し，適切に事前確率を導入し，事後確率を用いて識別を行うことで，正常トラヒックと感染トラヒックでまったく同じパターンが現れた場合でも正しく識別を行える手法を検討することは非常に重要である．

謝辞

本研究は私が早稲田大学理工学術院基幹理工学研究科情報理工学専攻修士課程に在学中の研究成果をまとめたものです.

本研究を進めるにあたり，懇切丁寧な御指導，御助言を賜りました甲藤二郎教授と小松尚久教授に心から深く感謝の意を表します.

また，共同研究者として様々な御意見を賜りました NTT コミュニケーションズ株式会社の畑田充弘様，本研究において様々な御助言を頂きました電気通信大学教授の吉浦裕教授，同大学の市野将嗣氏，同大学修士 1 年の大月優輔氏に深く御礼申し上げます．そして，日々研究において助言を下さいました早稲田大学の大木哲史博士，披田野清良博士，日頃から討論に御参加頂いた甲藤研究室，小松研究室の皆様に深く感謝致します.

2013 年 2 月 8 日

川元　研治

参考文献

[1] 瀬戸洋一他編著. 情報セキュリティ概論. 日本工業出版, 2007. ISBN: 978-4819019170.

[2] トレンドマイクロ株式会社. 2012 年度インターネット脅威年間レポート - 2012 年度, January. 2013. http://jp.trendmicro.com/jp/threat/security_news/monthlyreport/article/20130107041500.html.

[3] Mcafee. 忍び寄るマルウェアの脅威/マカフィーのセキュリティ研究レポート. http://www.mcafee.com/japan/security/rp_automotive_system_security.asp.

[4] G Data Software AG. マルウェアレポート 2011 年上半期, October. 2011. http://sv20.wadax.ne.jp/~gdata-co-jp/press/GData2011H1MalRep.pdf.

[5] Anti-Virus Comparative. Retrospective test, Aug. 2011. http://www.av-comparatives.org/images/stories/test/ondret/avc_report26.pdf.

[6] トレンドマイクロ株式会社. セキュリティソフトを無効にするルートキット型マルウェアとは. http://about-threats.trendmicro.com/RelatedThreats.aspx?language=jp&name=The+Anatomy+of+RTKT_ZACCESS.

[7] 市野将嗣, 坂野鋭, 小松尚久. 核非線形相互部分空間法による話者認識. 信学論 (D-Ⅱ), No. 8, pp. 1331–1338, 2005.

[8] 金井瑛. 通信の類似性に着目したネットワークインシデント検知手法. Master's thesis, 慶應義塾大学大学院, Mar. 2009.

[9] 井上大介, 中尾康二. 1. マルウェアって?(特集 マルウェア). 情報処理, Vol. 51, pp. 237–243, Mar. 2010.

[10] アスキードット PC. 最新のマルウェア事情. p. 18, November 2010.

[11] Inc. Nikkei Business Publications. ヤフーがクラッカの攻撃を受けて 3 時間機能停止、イーベイ、バイ・ドット・コムなども相次いで被害, Feb. 2000. http://www.nikkeibp.co.jp/archives/094/94127.html.

[12] インターネットコム株式会社. 韓国、ワームの影響でインターネットが約 9 時間マヒ, Jan. 2003. http://japan.internet.com/public/news/20030127/20.html.

[13] Tony Carothers. Large botnet in the netherlands taken down, Oct. 2005. http://isc.sans.org/diary.php?storyid=742.

[14] 株式会社シマンテック. 「標的型攻撃」に備える - サイバー攻撃: 標的型攻撃とは, apt とは. http://www.symantec.com/ja/jp/theme.jsp?themeid=apt_insight.

[15] IPA 独立行政法人 情報処理推進機構 技術本部セキュリティセンター. Ipa テクニカルウォッチ フリーメールからの送信が増加傾向に: 最近の標的型攻撃メールの傾向と事例分析. https://www.ipa.go.jp/about/technicalwatch/pdf/121030report.pdf.

[16] YAHOO! JAPAN ニュース. 警察は何も知らなかった... 硬直した捜査，ip アド偏重で誤認逮捕の遠隔操作事件. http://dailynews.yahoo.co.jp/fc/domestic/remote_control_virus/.

[17] 日経 BP 社 ITpro. Gmail の乗っ取りが国内で相次ぐ，パスワードの強化や 2 段階認証の利用を. http://itpro.nikkeibp.co.jp/article/NEWS/20121227/447164/.

[18] Paul Bacher, Thorsten Holz, Markus Kotter, and Georg Wicherski. Know your enemy: Tracking botnets -using honeynets to learn more about bots-, Mar. 2005. http://www.honeynet.org/book/export/html/50.

[19] IIJ Internet Initiative Japan. Iij technical week 2011 セキュリティ動向 2011. http://www.iij.ad.jp/company/development/tech/techweek/pdf/tw2011_10_security.pdf.

[20] 藤原将志, 寺田真敏, 安部哲哉, 菊池浩明. マルウェアの感染方式に基づく分類に関する検討. 情報処理学会 CSEC 研究報告，PRMU97, No. 21, pp. 177–182, Mar. 2008.

[21] NTT 情報流通プラットフォーム研究所. マルウェア対策技術. NTT 技術ジャーナル, Mar. 2010.

[22] S.Kondo and N.Sato. Botnet traffic detection techniques by c&c session classification using svm. *IWSEC2007*, Oct. 2007.

[23] 石井健太郎, 上田修功, 前田英作, 村瀬洋. わかりやすいパターン認識. オーム社, 1998.

[24] 与那原亨, 大谷尚通, 馬場達也, 稲田勉. トラフィック解析によるスパイウェア検知の一考察. 電子情報通信学会技術研究報告, Vol. 2005, No. 70, 2005-CSEC-30, pp. 23–29, 2005.

[25] Marius Kloft et.al. Automatic feature selection for anomaly detection. *Conference on Computer and Communications Security*, 2008.

[26] 桑原和也, 菊池浩明, 寺田真敏, 藤原将志. パケットキャプチャから感染種類を判定する発見的手法について. マルウェア対策研究人材育成ワークショップ 2009(MWS2009), 2009.

[27] Wei Lu, Mahbod Tavallaee, and Ali A. Ghorbani. Automatic discovery of botnet communities on large-scale communication networks. *the 4th International Symposium on Information, Computer, andCommunications Security*, 2009.

[28] 山田明, 三宅優, 田中俊昭, 竹森敬祐. 学習データを自動生成する未知攻撃検知システム. 情報処理学会論文誌, Vol. 46, No. 8, pp. 1947–1958, 2005.

[29] 大月優輔, 市野将嗣, 川元研治, 畑田充弘, 吉浦裕. マルウェア感染検知のためのトラヒックデータにおけるペイロード情報の特徴量評価. マルウェア対策研究人材育成ワークショップ 2012(MWS2012), Nov. 2012.

[30] 及川達也, 和泉勇治, 太田耕平, 加藤寧, 根元義章. 統計的クラスタリング手法によるネットワーク異常状態の検出. 信学技報，NS2002, No. 143, pp. 166–173, Oct. 2002.

[31] 宮本貴朗, 小島篤博, 泉正夫, 福永邦雄. Svm を用いたネットワークトラヒックからの異常検出. 電子情報通信学会論文誌, Vol. B, 通信 J87-B(4), pp. 593–598, Apr. 2004.

[32] 東角芳樹, 鳥居悟. DNS 通信の挙動からみたボット感染検知方式の検討. マルウェア対策研究人材育成ワークショップ 2008(MWS2008), Oct. 2008.

[33] 阿部義徳, 田中英彦. C&C セッション分類によるボットネットの検出手法の一検討. *FIT2007*, Sep. 2007.

[34] M.bahrololum and M.Khaleghi. Anomaly intrusion detection systemu using gaussian mixutre model. *IEEE*, 2009.

[35] Kdd cup 1999 data, October 1999. `http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`.

[36] M.bahrololum and M.Khaleghi. Anomaly intrusion detection systemu using hierarchical gaussian mixutre model. *IJCSNS*, 2008.

[37] Wei Lu and Issa Traore. An unsupervised approach for detecting ddos attacks based on traffic-based metrics. *IEEE*, 2005.

[38] 八木清之介, 和泉勇治, 根元義章. ペイロード長の遷移パタンを用いたネットワークアプリケーション弁別手法. *IPSJ SIJ Technical Reports*, May. 2007.

[39] Gautam Thatte, Urbashi Mitra, and John Heidemann. Parametric method for anomaly detection in aggregate traffic. *IEEE/ACM Transactions on Networking*, Aug. 2009.

[40] 水谷正慶, 武田圭史, 村井純. 通信の状態遷移に着目したボット活動の調査. マルウェア対策研究人材育成ワークショップ 2008(MWS2008), 2008.

[41] Chun Yang, Feiqi Deng, and Haidong Yang. An unsupervised anomaly detection approach using subtractive clustering and hidden markov model. *ICommunications and Networking in China*, Aug. 2007.

[42] 市田達也. 特徴量の時間的な状態遷移を考慮したマルウェア感染検知手法に関する研究. Master's thesis, 早稲田大学理工学術院, Mar. 2012.

[43] Wireshark. `http://www.wireshark.org/`.

[44] 畑田充弘, 中津留勇, 秋山満昭, 三輪信介. マルウェア対策のための研究用データセット ～ mws 2011. マルウェア対策研究人材育成ワークショップ 2011(MWS2011), Oct. 2011.

[45] 森大二郎. 検索エンジンはなぜ見つけるのか-知っておきたいウェブ情報検索の基礎知識. 日経 BP 社, 3 月 2011.

[46] 山田崇仁. N-gram 方式を利用した漢字文献の分析. 立命館白川静記念東洋文字文化研究所紀要, 03 2007.

[47] N-gram ってなんだ. `http://handin.sakura.ne.jp/archives/179`.

[48] 安形輝, 石田栄美, 久野高志, 野末道子, 上田修一. Www ページの自動分類: Ndc の分類体系と yahoo のカテゴリを使った分類. 情報処理学会研究報告，データベース・システム研究報告会, 5 1999.

[49] 井上雅翔, 山名早人. 品詞 n-gram を用いた著者推定手法 -話題に対する頑健性の評価. 日本

データベース学会論文誌, No. 3, 2 2012.

[50] 中川聖一. 音声認識において hmm とトライグラムを超えるもの. 人口知能学会誌, 1 号, 1 2002.

[51] 杉山将. 統計的機械学習 生成モデルに基づくパターン認識. オーム社, 9 月 2009.

[52] 金森敬文, 竹之内高志, 村田昇. R で学ぶデータサイエンス 5 パターン認識. 共立出版, 2009.

[53] 前田浩明, 星健太郎, 市野将嗣, 小松尚久. ロジスティック回帰分析を用いたスコアレベル融合によるトラヒックパターン分類方法に関する一検討. 信学技報, Vol. 111, No. 278, CQ2011-52, pp. 49–54, 11 2011.

[54] 金森敬文, 竹之内高志, 村田昇. R で学ぶデータサイエンス 5 パターン認識. 共立出版, 2009. ISBN: 978-4-320-01925-6.

[55] 小松尚久, 内田薫, 池野修一, 坂野鋭共著. バイオメトリクスのおはなし /あなたの身体情報が鍵になる/. 日本規格協会, 2008. ISBN: 978-4-542-90278-7.

[56] 半谷精一郎編著. バイオメトリクス教科書 -原理からプログラミングまで-. コロナ社, 2012. ISBN: 978-4339008357.

[57] 中川聖一. 確率モデルによる音声認識. 社団法人 電子情報通信学会, 1988. ISBN: 4-88552-072-X.

[58] 本間謙也, 間所浩和, 佐藤和人. 動線解析によるイベント会場での行動パターン分類. 電子情報通信学会, Vol. J95-D, No. 10, pp. 1848–1858, Oct. 2012.

[59] 倉重正義, 飯塚啓太, ターウォンマットラック. 自己組織化マップによるオンラインゲーム内のユーザ移動データのクラスタリング. 情報処理学会研究報告，EC，エンタテイメントコンピューティング, 3 2006.

[60] 大江将悟. 新しいクラスタリング手法の紹介 - affinity propagation -. 計算知能研究室, 10 2009.

付録 A

ベクトル量子化

A.1　ベクトル量子化

本研究で使用したベクトル量子化について以下に示す．ベクトル量子化 (Vector Quantization) とは，K 個の信号をまとめてひとつの K 次元ベクトル，すなわち K 次元信号空間内の一点とし，あらかじめ定められたいくつかの代表点（量子化代表ベクトル）で近似するものである．

まず入力ベクトルを K 次元ベクトルとする．このとき，入力ベクトルが存在する信号空間は K 次元となる．この K 次元空間を $\mathbf{x} = (\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_K})^{\mathbf{T}}$ と書くことにする．ここで K 次元信号空間 R^K を互いに重なり合わない N この領域 $P_1, P_2, \cdots, P_N$ に分割し，各領域 P_i 内に量子化代表ベクトル $\mathbf{y} = (\mathbf{y_{i1}}, \mathbf{y_{i2}}, \cdots, \mathbf{y_{iK}})^{\mathbf{T}}$ を一つ定めておく．ここで，領域 $P_1, P_2, \cdots, P_N$ の集合を P と書き，分割と呼ぶ．こういった記法を用いると，「K 次元 N レベルのベクトル量子化」は K 次元信号空間 R^K からコードブック C への写像 $Q(\cdot)$ として，次のように記述される．

ベクトル量子化の動作：
もし，入力ベクトル x が領域 P_i 内に所属しているならば，そのときベクトル量子化の動作は次式の写像 $Q(\cdot)$ として記述される．

$$\mathbf{Q}(\bullet) = \mathbf{y_i} \tag{A.1}$$

式 (A.1) のベクトル量子化によるサンプル当りの平均ひずみ D は，入力ベクトル $\mathbf{x}$ の確率密度関数 $p(\mathbf{x})$ が既知の場合には次式で与えられる．

$$\begin{aligned} D &= \frac{1}{K} \int_{R^K} d(Q(\mathbf{x})) \cdot p(\mathbf{x}) d\mathbf{x} \\ &= \frac{1}{K} \sum_{i=1}^{N} \int_{P_i} d(\mathbf{x}, \mathbf{y_i}) \cdot p(\mathbf{x}) d\mathbf{x} \end{aligned} \tag{A.2}$$

次元数 K とレベル数 n とが指定されたとき，式 (A.2) の平均ひずみ D を最小とするベクトル量子化器のことを「最適なベクトル量子化器」と呼ぶ．式 (A.2) において，平均ひずみ D の大きさは領域 P_i や量子化代表ベクトル $\mathbf{y_i}$ をどのように設定するかに左右される．従って，ベクトル量子化器を最適化するためには，領域や量子化代表ベクトルを式 (A.2) の D を最小とするように定

めなければならない．このための必要条件は，以下に示すような二つの最適化条件に帰着する．

1. 代表点条件
   K 次元信号空間 R^K の N 個の領域 $P_1, P_2, \cdots, P_N$ への分割 P が与えられたとき，式 (A.2) の平均ひずみを最小とする N 個の量子化代表ベクトル $\mathbf{y_1}, \mathbf{y_2}, \cdots, \mathbf{y_N}$ はそれぞれ対応した領域 $P_1, P_2, \cdots, P_N$ の重心で与えられる．入力ベクトル $\mathbf{x}$ の確率密度関数 $p(\mathbf{x})$ が既知の場合は，量子化代表ベクトル $\mathbf{y_i}$ が次式で計算されることを意味する．

$$\mathbf{y_i} = \frac{\int_{P_i} \mathbf{x} \cdot p(\mathbf{x}) d\mathbf{x}}{\int_{P_i} p(\mathbf{x}) d\mathbf{x}} \tag{A.3}$$

2. 分割条件
   N 個の量子化代表ベクトル $\mathbf{y_1}, \mathbf{y_2}, \cdots, \mathbf{y_N}$ から成るコードブック C が与えられたとき，式 (A.2) の平均ひずみ D を最小とする N 個の領域 $P_1, P_2, \cdots, P_N$ への K 次元信号空間 R^K の分割 P は，次のようにして与えられる．即ち，量子化代表ベクトル $\mathbf{y_1}, \mathbf{y_2}, \cdots, \mathbf{y_N}$ に対応した領域 P_i は，N 個の量子化代表ベクトルの中で $\mathbf{y_i}$ とのひずみが最小となる入力ベクトル $\mathbf{x}$，すなわち $\mathbf{y_i}$ 以外のすべての量子化代表ベクトル $\mathbf{y_i}(j \neq i)$ に対して次式の不等式

$$d(\mathbf{x}, \mathbf{y_i}) \leqq d(\mathbf{x}, \mathbf{y_j}) \tag{A.4}$$

   を満足する入力ベクトル $\mathbf{x}$ の集合として与えられる．
   また，この条件によって式 (A.1) の写像 $Q()$ として定義された量子化動作を次のように書き換えることができる．

   最適なベクトル量子化器の量子化動作：
   入力ベクトル $\mathbf{x}$ が $\mathbf{y_i}$ 以外の全ての量子化代表ベクトル $\mathbf{y_i}(j \neq i)$ に対して式 (A.4) の不等式を満足するならば，その時最適なベクトル量子化器の量子化動作は，入力ベクトル $\mathbf{x}$ から量子化ベクトル $\mathbf{y_i}$ への写像として,$Q(\mathbf{x}) = \mathbf{y_i}$ と記述される．また，本研究ではひずみ測定に，入力ベクトルと量子化代表ベクトルの間のユークリッド距離の 2 乗として定義された 2 乗ひずみ測定を用いている．

A.2　クラスタリングアルゴリズム

本研究では，クラスタリングアルゴリズムとして，LBG+splitting アルゴリズムを用いている．

■ LBG アルゴリズム

LBG アルゴリズムは，適当な初期コードブックから出発し，学習系列に分割条件と代表点条件を繰り返し適用し，良好なコードブックに収束させる設計アルゴリズムである．その処理の流れ

を図 A.1 に示し，手順を以下に示す．

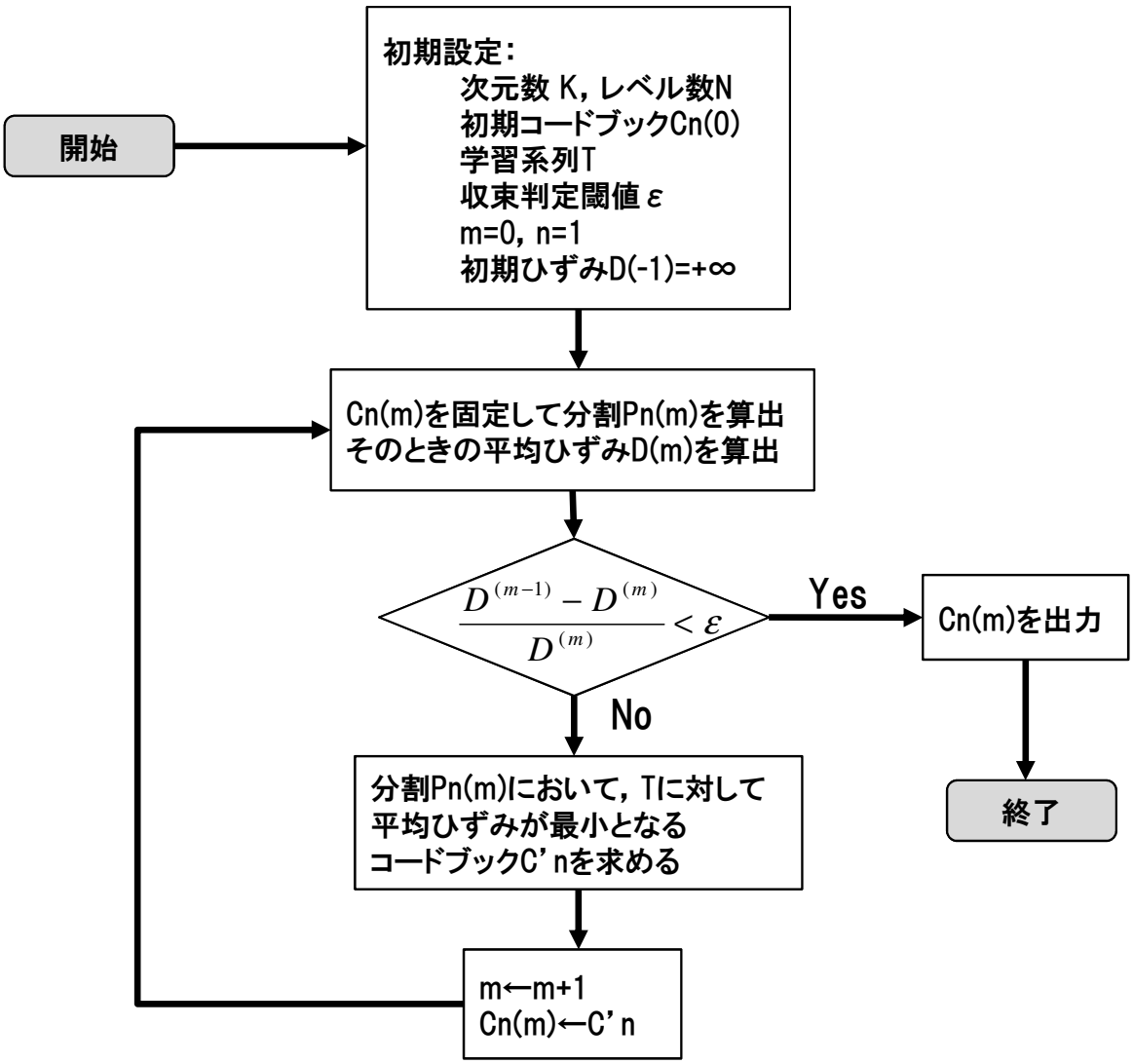

図 A.1　LBG アルゴリズムにおける処理のフローチャート

1. 次元数 K，レベル数 N，N 個の初期量子化代表ベクトル $\mathbf{y_1^{(0)}}, \mathbf{y_2^{(0)}}, \cdots, \mathbf{y_N^{(0)}}$ から成る初期コードブック $C_N^{(0)}$，L 個の K 次元学習ベクトル $\mathbf{x_1^{(0)}}, \mathbf{x_2^{(0)}}, \cdots, \mathbf{x_N^{(0)}}$ からなる学習系列 T，収束判定用しきい値 ϵ が与えられているとする．また，$m = 0$，初期ひずみ $D^{(-1)} = \infty$ と設定する．
2. 量子化代表ベクトル $\mathbf{y_1^{(m)}}, \mathbf{y_2^{(m)}}, \cdots, \mathbf{y_N^{(m)}}$ から成る初期コードブック $C_N^{(m)}$ の下で，平均ひずみを最小とする N 個の領域 $P_1^{(m)}$，$P_2^{(m)}$，・・・，$P_N^{(m)}$ への学習系列 T の分割 $P_N^{(m)}$ を分割条件を適用してから定める．即ち，量子化代表ベクトル $\mathbf{y_i^{(m)}}$ に対応した領域 $P_i^{(m)}$ は N 個の量子化代表ベクトルの中で，$\mathbf{y_i^{(m)}}$ とのひずみが最小となる学習ベクトルの集合で与えられる．こうして，L 個の学習ベクトルが N 個の領域に分割される．また，各領域に所属す

る学習ベクトルを，その領域内の量子化代表ベクトルで置き換えたときに生じる平均ひずみ $D^{(m)}$ を算出する．

3. もし，$(D^{(m-1)} - D^{(m)})/D^{(m)} < \epsilon$ ならば，処理を停止して，$C_N^{(m)}$ を最終的に設計された N レベルのコードブックとして出力する．さもなければ次の手順へ進む．
4. N 個の領域 $P_1^{(m)}$, $P_2^{(m)}$, ・・・, $P_N^{(m)}$ への学習系列 T の分割 $P_N^{(m)}$ の下で，学習系列 T に対して平均ひずみを最小とする N 個の量子化代表ベクトル $\mathbf{y_1^{(m)}, y_2^{(m)}, \cdots, y_N^{(m)}}$ から成るコードブック C'_N を代表点条件を適用して定める．領域 $P_i^{(m)}$ に所属する学習ベクトルの平均ベクトルとして与えられる重心を，量子化代表ベクトル $\mathbf{y'_i}$ とする．さらに，$m \leftarrow m+1$ とし，C'_N をコードブック $C_N^{(m)}$ として，手順 2. へ戻る．

■ splitting アルゴリズム

LBG アルゴリズムにより設計されたコードブックの良否は，初期コードブック $C_N^{(0)}$ と学習系列 T の選定法に強く依存する．初期コードブック $C_N^{(0)}$ は想定される入力ベクトルの分布範囲を被覆していることが望ましい．この条件をある程度満足する初期コードブックの生成法として splitting アルゴリズムが知られている．このアルゴリズムは，N レベルのコードブック C_N の量子化代表ベクトル $\mathbf{y_1, y_2, \cdots, y_N}$ を式 (A.5) のように微小なベクトル δ を用いて接近した二つのベクトルに分割することによって，量子化代表ベクトル $\mathbf{y_1, y_2, \cdots, y_{2N}}$ からなる $2N$ レベルの初期コードブック $C_{2N}^{(0)}$ を生成するものである．この splitting アルゴリズムを LBG アルゴリズムと組み合わせることによって 1 レベルのコードブックから出発して順次 2, 4, 8, ・・・ レベルのコードブックを設計することが出来る．

$$\mathbf{y_i} = \mathbf{y_i} - \delta, \mathbf{y_{i+N}} = \mathbf{y_i} + \delta \tag{A.5}$$

付録 B

CCCDataset2011 について

今回実験で使用した CCCDataset[44] について補足する．

B.1　データの概要

CCCDataset は，Cyber Clean Center(CCC) によって収集されたマルウェア対策研究を対象とした研究用データセットである．マルウェアの対策研究をする上で「共通なデータセットがない」という課題があり，提案手法の評価に用いるマルウェアのサンプルや，感染前後の通信データといった研究用データセットとして提供されている．これによりマルウェア研究の成果の比較が容易になり，新たに研究を始める人にとっても参入が容易になったため，研究の裾野が広がったといえる．

研究成果を共有する場・切磋琢磨する環境として開催された「マルウェア対策研究人材育成ワークショップ (MWS)」において使用された．MWS は 2008 年から現在まで毎年開催され，その都度 CCCDataSet2008，2009，2010，2011 と用意されている．これにより過去のデータとの傾向を比較分析が可能になっている．

B.2　CCCDataset2011 について

本節では今回の実験で用いた CCCDataset 2011 の概要について述べる．

CCC2011 では，マルウェアの解析技術の研究のための「マルウェア検体」，感染手法の検知ならびに解析技術の研究のための「攻撃通信データ」，ボットの活動傾向把握技術の研究のための「攻撃元データ」の 3 つから構成される．以下，それぞれについて概要を述べる．

B.2.1　マルウェア検体

ハニーポットで収集したマルウェア検体のハッシュ値 (MD5，SHA1)50 個をテキスト形式で記載したファイルであり，以下の観点から選定している．

1. 解析結果を照合できる検体：10 検体
2. 未知検体：40 検体

1 は特徴的な機能を有し，技術的に目を通しておきたい検体である．これは事前に静的解析が完了しており，解析精度の評価に活用することを考慮している．具体的には，ユーザの特定の動作をトリガとして動作する検体や，独自かつ高度な通信プロトコルを使用する検体である．2 は 2011 年 1 月の未知検体のうち，収集日が偏らないように任意で選定した検体であり，相当数の検体の自動解析や自動分類を考慮している．なお，対象となるマルウェア検体は，以降の攻撃通信データ及び攻撃元データに一部含まれる検体である．

B.2.2　攻撃通信データ

ハニーポットの通信を tcpdump でパケットキャプチャした libpcap 形式のファイルである．ハニーポットは，ホスト OS 上の 2 台 (honey001，honey002) のゲスト OS がそれぞれインターネット接続されており，パケットキャプチャはホスト OS 上で行っている．ゲスト OS は 2 台とも WindowsXP SP1 であり，これらは定期的にクリーンな状態にリセットされる．データ収集日は 2010 年 8 月 18 日から 8 月 31 日，2010 年 1 月 18 日から 1 月 31 日，総パケット数が 23,009,309 パケット，約 3.8GB のデータサイズである．

B.2.3　攻撃元データ

2010 年 5 月 1 日から 2011 年 1 月 31 日までの 9 ヶ月間にハニーポットで記録したマルウェア取得時のログで，表 B.1 に示す項目を 1 レコードとして記録した csv 形式のファイルである．Windows2000 が稼働するハニーポットも一部含み，国内の複数の ISP にそれぞれ接続された 72 台のハニーポットで記録された約 22MB のデータである．攻撃元データの基本情報を表 B.2 に示す．

表 B.1　攻撃元データのログ項目と例

ログ項目	例（一部を*でマスク）
マルウェア検体の取得時刻	2011-01-14 18:20:01
送信元 IP アドレス	honey016
送信元ポート番号	1029
宛先 IP アドレス	**. 179.100
宛先ポート番号	20000
TCP または UDP	TCP
マルウェア検体のハッシュ値（SHA1）	*****6b8124247f988f96725066d3752ef018549
ウイルス名称	Mal_DLDER
ファイル名	C:/WINNT/system32/fewh.exe

マルウェア検体のダウンロードを開始した時刻がマルウェア検体の取得時刻であり，ゲスト OS の Windows 上でのファイル作成日時となる．送信元 IP アドレスまたは宛先 IP アドレスにおいて，ハニーポットの IP アドレスは各ハニーポットに対応する ID（honey001 等）に置換されて記載されている．ウイルス名称は収集日の翌日午前 3 時の最新パターンファイルを適用したウイルススキャナ（トレンドマイクロ社製）により判定された名称であり，マルウェアとして判定されなかったものは UNKNOWN と表記される．このため，パターンファイルのウイルス名称が更新された場合，同一のハッシュ値であっても，異なるウイルス名称が付与される場合がある．

表 B.2　攻撃元データの基本情報

項目	件数
全レコード数	158,734
TCP によるダウンロードレコード数	136,251
UDP によるダウンロードレコード数	22,483
ダウンロードホスト IP アドレス種類数	89,122
マルウェア検体のハッシュ値種類数	12,591
ウイルス名称種類数 (UNKNOWN 含まない)	316

B.3　感染時データの切り出し

CCC2010 の攻撃通信データには感染するまでのトラヒックデータが含まれている．今回のマルウェアトラヒック検知実験には感染時のみのデータを用いる必要があったため，攻撃通信データから感染時のトラヒックデータを切り出した．

手順と使用した Linux コマンドを以下に示す．

1. 取得環境独自の制御パケットをフィルタリングで除外
   /usr/sbin/tshark -r [入力 pcap ファイル] -R '!(sll.pkttype == 4) && (ip.addr == 10.10.1*.1)' -w [出力 pcap ファイル 1]
   –pcap ファイル名，ハニーポッド番号（honey001 → 10.10.11.1, honey002 → 10.10.12.1）はその都度変更
2. ハニーポットの OS のリセット間隔で切り出す
   /usr/sbin/tshark -r [出力 pcap ファイル 1] -t ad — grep "time.windows.com" > reset-time.log
   –pcap ファイル名，log ファイル名はその都度変更

resettime.log でまともに 20 分ずつリセットされているか確認したうえで，リセット時刻のパケット番号を抽出
echo 1 >> reset_pnum; /usr/sbin/tshark -r [出力 pcap ファイル 1] — grep ”time.windows.com” — cut -d” ” -f1 >> reset_pnum; /usr/sbin/tshark -r [出力 pcap ファイル 1] — tail -1 — cut -d” ” -f1 >> reset_pnum

切り出す開始パケット番号と終了パケット番号を書きだす
START=1; for NUM in ‘cat reset_pnum‘; do END=‘expr $NUM - 1‘; echo $START-$END >> start-end; START=$NUM; done

pcap をスライス
for NUM in ‘seq -w 01 **‘; do /usr/sbin/editcap -r [出力 pcap ファイル 1] sliced/$NUM.pcap ‘head -$NUM start-end — tail -1‘; done
–sliced ディレクトリを作成しておく、スライス後のファイル数は start-end の行数で確認し，**にファイル数指定し sliced に格納する

sliced ディレクトリに移り、スライスした pcap の開始・終了時刻を確認
for NUM in ‘seq -w 01 **‘; do START=‘/usr/sbin/tshark -r $NUM.pcap -t ad — head -1 — awk ’print $2” ”$3’‘; END=‘/usr/sbin/tshark -r $NUM.pcap -t ad — tail -1 — awk ’print $2” ”$3’‘; echo ”$NUM $START $END” >> start-end_time.txt; done

3. 残ったファイルをログと照らし合わせて，感染を確認
4. 感染攻撃の開始パケットを探し，それ以降を感染トラヒックとして抽出する
5. 感染トラヒックの内，ログで単一感染のファイルを抽出する

付録 C

カーネル密度推定法

カーネル密度推定法はノンパラメトリックな密度推定法である．まず，最も単純なノンパラメトリック手法であるヒストグラム法について説明し，その後，カーネル密度推定法について説明する．

C.1　ヒストグラム法

ヒストグラム法は，パターン空間 $\boldsymbol{D}$ を適当に分割し，書く分割内に入る訓練標本数を数え，全体の積分が 1 になるように正規化したものを確率密度関数の推定量とする方法である．分割した領域のことをビンと呼ぶ．便は場所によって形や大きさが異なってもよい．

ヒストグラム法は非常に単純で計算が簡単だが，実用上いくつか問題がある．まず，ビンの形や大きさを決めるのが難しいという問題がある．ビンの幅を適切に決めれなければ真の確率密度関数の良い推定量が得られない．また，推定した確率密度関数がビンの間で不連続になってしまうという問題もある．確率密度関数は滑らかであることが多く，不連続な推定量は必ずしも適切でない．更に，格子分割等の方法で単純にビンを配置するだけでは，入力次元 d の増加に伴いビンの数が指数的に増加してしまう．そうすると，ほとんどのビンに訓練標本数が含まれなくなってしまい，妥当な推定量が得られない．

C.2　ノンパラメトリック法の枠組み

本節では，C.1 節で挙げた問題を解決するノンパラメトリック法の枠組みについて述べる．

ある注目点 x' での確率密度の値 $p(x')$ を推定する問題を考える．注目点 x' を含むパターン空間 $\boldsymbol{D}$ 内のある領域を $\boldsymbol{R}$ で表す．そして，$\boldsymbol{R}$ の体積を V とすると，V は式 (C.1) で表される．

$$V := \int_R dx \tag{C.1}$$

あるパターン x が，領域 $\boldsymbol{R}$ に入る確率 P は式 (C.2) で表される．

$$P := \int_R p(x)dx \tag{C.2}$$

上記の記号を図 C.1 に示す.

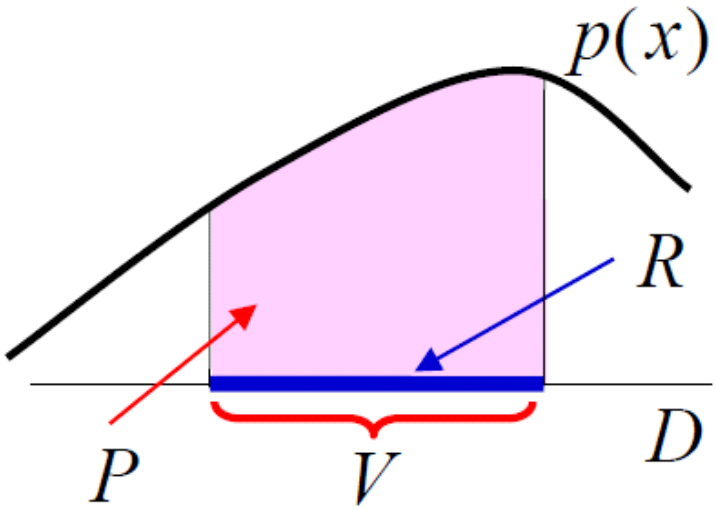

図 C.1　ノンパラメトリック法の表記

確率 P の値は，注目点 x' を用いて，式 (C.3) のように近似できる.

$$P \approx Vp(x') \tag{C.3}$$

近似のイメージを図 C.2 に示す.

一方，n 個の訓練標本 $\{x_i\}_{i=1}^n$ のうち領域 $\boldsymbol{R}$ 内に入っている個数を k で表せば，P の値は式 (C.4) で近似できる.

$$P \approx \frac{k}{n} \tag{C.4}$$

式 (C.3) と式 (C.4) を合わせて P を消去すれば，$p(x')$ の値は式 (C.5) で近似できる.

$$p(x') \approx \frac{k}{nV} \tag{C.5}$$

式 (C.5) の近似の良さは，式 (C.3) と式 (C.4) の近似の良さに依存する．また，式 (C.3) と式 (C.4) の近似の良さは領域 $\boldsymbol{R}$ の選び方に依存する．そこで，式 (C.3) と式 (C.4) の近似の良さと

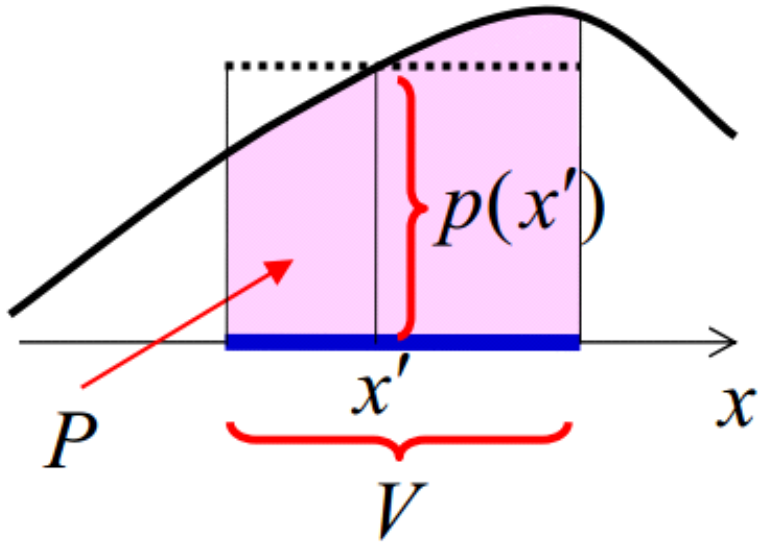

図 C.2　確率 P の長方形近似

領域 $\boldsymbol{R}$ の選び方の関係を評価する．

式 (C.3) は積分の単純な長方形近似 (性格には超直方体近似) であり，領域 $\boldsymbol{R}$ 内で $p(x)$ が定数関数に近い方が近似精度が良い．したがって，領域 $\boldsymbol{R}$ は小さければ小さいほど式 (C.3) の近似精度は向上する．

次に，式 (C.4) の近似の良さを評価する．n 個の訓練標本のうつ k 個が領域 $\boldsymbol{R}$ 内に入る確率は，二項分布を用いて式 (C.6) で与えられる．

$$ {}_n\mathrm{C}_k P^k (1-p)^{n-k} \tag{C.6} $$

${}_n\mathrm{C}_k$ は二項係数であり，式 (C.7) で定義される．

$$ {}_n\mathrm{C}_k = \frac{n!}{(n-k)!k!} \tag{C.7} $$

二項分布の期待値と分散はそれぞれ式 (C.8) で与えられる．

$$ \boldsymbol{E}[k] = nP, \qquad \boldsymbol{V}[k] = nP(1-P) \tag{C.8} $$

したがって，k/n の期待値は真の P と一致する (式 (C.9))．

$$\boldsymbol{E}[\frac{k}{n}] = P \tag{C.9}$$

しかし，期待値が一致するからといって，k/n がいつも P の良い近似になっているとは限らない．分散が大きければ k/n は P の推定量として適切ではない．ここでは，期待値を 1 に正規化した量 z を考える (式 (C.10))．

$$z := \frac{k}{nP} \tag{C.10}$$

z の分散は式 (C.11) で与えられる．

$$\boldsymbol{V}[z] = \frac{1-P}{nP} \tag{C.11}$$

この z の分散が小さければ小さいほど，k/n は P の良い近似になると考えられる．z の分散を P の関数として計算したグラフを図 C.3 に示す．

図 C.3 からわかるように，P が大きければ大きいほど z の分散は小さくなる．P を大きくするためには領域 $\boldsymbol{R}$ を大きくとればよいため，領域 $\boldsymbol{R}$ は大きければ大きいほど式 (C.4) の近似精度が向上する．

前述したように，式 (C.5) の近似の良さは式 (C.3) と式 (C.4) の近似の良さに依存する．式 (C.3) と式 (C.4) の近似の良さは領域 $\boldsymbol{R}$ の選び方に依存しており，式 (C.3) の近似精度を向上させるためには領域 $\boldsymbol{R}$ を小さくした方がよく，式 (C.4) の近似精度を向上させるためには領域 $\boldsymbol{R}$ を大きくした方がよい．したがって，精度良く $p(x)$ を推定するためには，程良い大きさに領域 $\boldsymbol{R}$ を決定しなければならない．そこで, 訓練標本 $\{x_i\}_{i=1}^n$ を用いて領域 $\boldsymbol{R}$ を決めることにする．

領域 $\boldsymbol{R}$ の体積 $\boldsymbol{V}$ を固定して，領域 $\boldsymbol{R}$ に含まれる訓練標本の数 k を訓練標本から決める方法がカーネル密度推定法であり，領域 $\boldsymbol{R}$ に含まれる訓練標本 k を固定して，領域 $\boldsymbol{R}$ の体積 $\boldsymbol{V}$ を訓練標本から決める方法が最近傍密度推定法である．

C.3　パーゼン窓法とカーネル密度推定法

本節では，領域 $\boldsymbol{R}$ の体積 $\boldsymbol{V}$ を固定したもとで，領域 $\boldsymbol{R}$ に含まれる訓練標本の数 k を訓練標本から決める方法を示す．

領域 $\boldsymbol{R}$ として，ある点 x を中心とする一辺の長さが b の超立方体を用いることにする．このとき，領域 $\boldsymbol{R}$ の体積 V は式 (C.12) で表される．

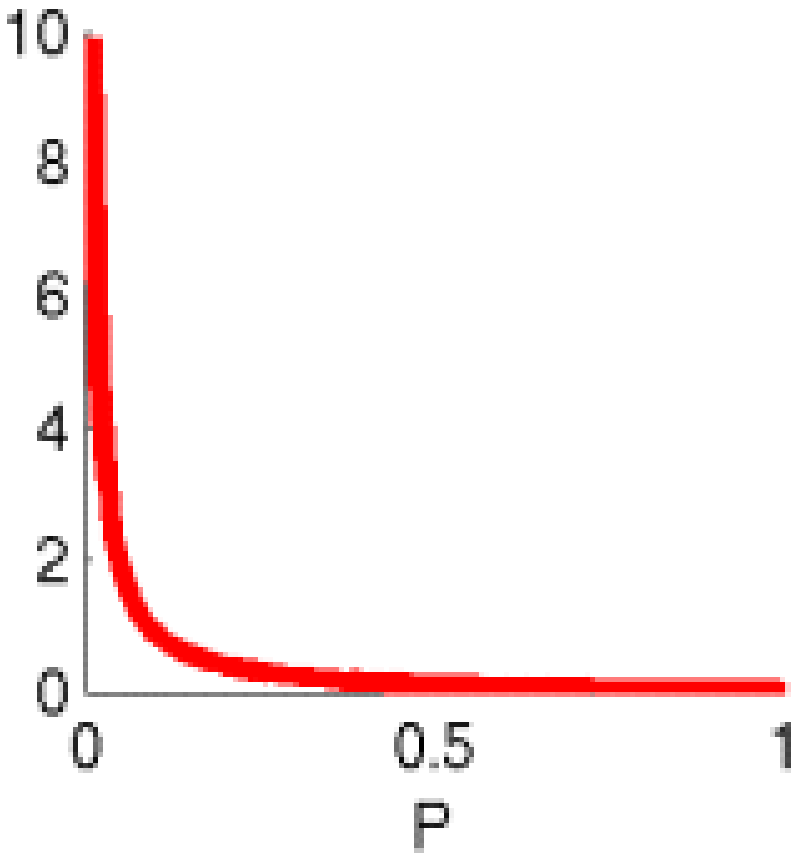

図 C.3　確率 P の長方形近似

$$V = b^d \tag{C.12}$$

d はパターン空間の次元である．一方，領域 $\boldsymbol{R}$ に含まれる訓練標本の数 k は式 (C.13) で表すことができる．

$$k = \sum_{i=1}^{n} W(\frac{x - x_i}{b}) \tag{C.13}$$

ここで，$W(x)$ はパーゼン窓関数とよばれ，式 (C.14) で定義される (図 C.4)．

$$W(x) := \begin{cases} 1 & \max\limits_{i=1,\cdots,x^{(d)}} |x^{(d)}| \leqq \frac{1}{2} \\ 0 & \text{それ以外} \end{cases} \tag{C.14}$$

ただし，$x = (x^1, \cdots, x^d)^T$ である．

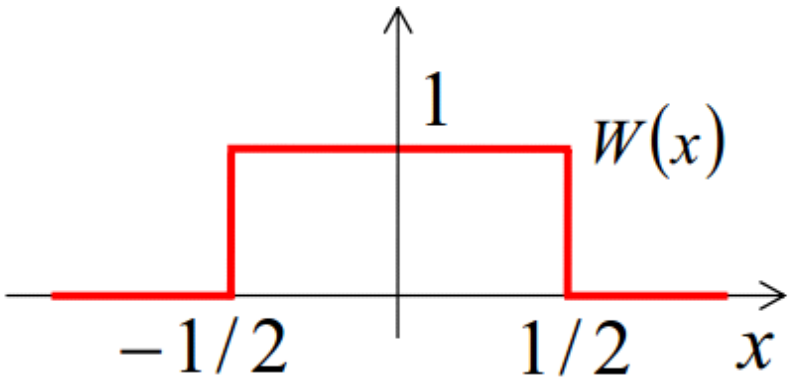

図 C.4　パーゼン窓法

式 (C.12) と式 (C.13) を式 (C.5) に代入すれば, 次の推定量 $\hat{p}_{Parzen}(x)$ が得られる (式 (C.15)).

$$\hat{p}_{Parzen}(x) = \frac{1}{nb^d} \sum_{i=1}^{n} W(\frac{x - x_i}{b}) \tag{C.15}$$

この推定方法をパーゼン窓法とよぶ.

パーゼン窓法による推定結果はヒストグラム法に似ているが，各ビンの幅が訓練標本から適応的に決定されている．パーゼン窓の幅 b の決め方に関しては，尤度交差確認法で決める方法が一般的である.

各ビンの幅が訓練標本から適応的に決定されることから，パーゼン窓法はヒストグラム法よりも適応的であるが，推定結果が領域のつなぎ目で不連続になるという問題は解決していない．この問題は，パーゼン窓法を拡張したカーネル密度推定法によって解決できる．カーネル密度推定法では，パーゼン窓関数の代わりにカーネル関数とよばれる一般の (滑らかな) 関数 $K(x)$ を用いる (式 (C.16)).

$$\hat{p}_{KDE}(x) = \frac{1}{nb^d} \sum_{i=1}^{n} W(\frac{x - x_i}{b}) \tag{C.16}$$

ただし，カーネル関数は式 (C.17) を満たさなければならない.

$$K(x) \geqq 0 \quad for \quad any \quad x \in D, \qquad \int_D K(x)dx = 1 \tag{C.17}$$

カーネル密度推定法では，b をカーネル関数のバンド幅とよぶ．カーネル関数としてよく用いられているのはガウスカーネルである．ガウスカーネルを式 (C.18) に示す．

$$K(x) = \frac{1}{(s\pi)^{\frac{2}{d}}} \exp(-\frac{x^T x}{2}) \tag{C.18}$$

ガウスカーネルでは，バンド幅 b はガウス関数の標準偏差に対応する．ガウスカーネルを用いたカーネル密度推定法の例を図 C.5 に示す．

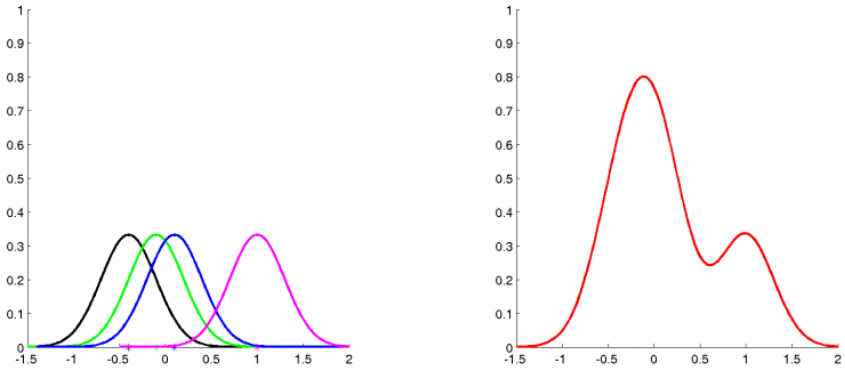

図 C.5　ガウスカーネル密度推定の例 (標本数 4 つ)

図 C.5 からわかるように，滑らかな確率密度関数推定量が得られている．

付録 D

正常サービスのパケットキャプチャリング

D.1　使用した PC スペック

今回使用した PC を以下にまとめる．

表 D.1　使用 PC1 のスペック

PC	Notebook Computer: Sony VAIO VGN-FS33B
CPU	Intel PentiumM 740 1.73GHz
Main Memory	512MB
OS	WindowsXP
ブラウザ	Microsoft Internet Explorer Version 6.0.2900.5512

表 D.2　使用 PC2 のスペック

PC	Notebook Computer: Panasonic Let's note CF-F8
CPU	Intel Core2 Duo 2.26GHz
Main Memory	2GB
OS	WindowsXP
ブラウザ	Microsoft Internet Explorer Version 6.0.2900.5512

D.2　Wireshark

本研究では，パケットのキャプチャリングにおいて Wireshark を用いている．Wireshark とは，Gerald Combs が開発した network protocol analyzer であり，800 以上のプロトコル解析機能や 85000 以上の display filter が特徴となっている．Wireshark におけるパケットキャプチャリングは，UNIX では libpcap，Windows では WinPcap を用いて行っている．

表 D.3　使用 PC3 のスペック

PC	Desktop Computer: Dell DIMENSION 5150C
CPU	Intel Pentium D 820 2.80GHz
Main Memory	1GB
OS	WindowsXP Home edition SP3
ブラウザ	Opera/9.80 Version/10.10

■Wireshark の設定　Wireshark のキャプチャリング設定について表 D.4 に示す．また，キャプチャリング時のスクリーンショットを図 D.1 に示す．

表 D.4　Wireshark のキャプチャリング設定

Capture	Interface: Local Buffer size: 20 megabyte(s)
Capture File(s)	File: 保存ファイル
Display Options	Update list of packets in real time Automatic scrolling in live capture Hide capture info dialog
Name Resolution	Enable MAC name resolution Enable transport name resolution

ここで，Capture における Buffer size は，キャプチャを行っている際にパケットを Drop した場合，より大きな値に変更する必要がある．

D.3　キャプチャリング手順

キャプチャリング手順について，BitTorrent およびストリーミングの場合について示す．

- BitTorrent の場合
  1. Firewall などの常駐ソフトを終了
  2. Wireshark を起動し，キャプチャ開始
  3. オフラインコンテンツや履歴，過去のダウンロードファイルなどの削除
  4. キャプチャ数 0 の状態で，.torrent ファイルを開く
  5. ファイルのダウンロードが完了したら BitTorrent を終了
  6. キャプチャ終了

D.3　キャプチャリング手順

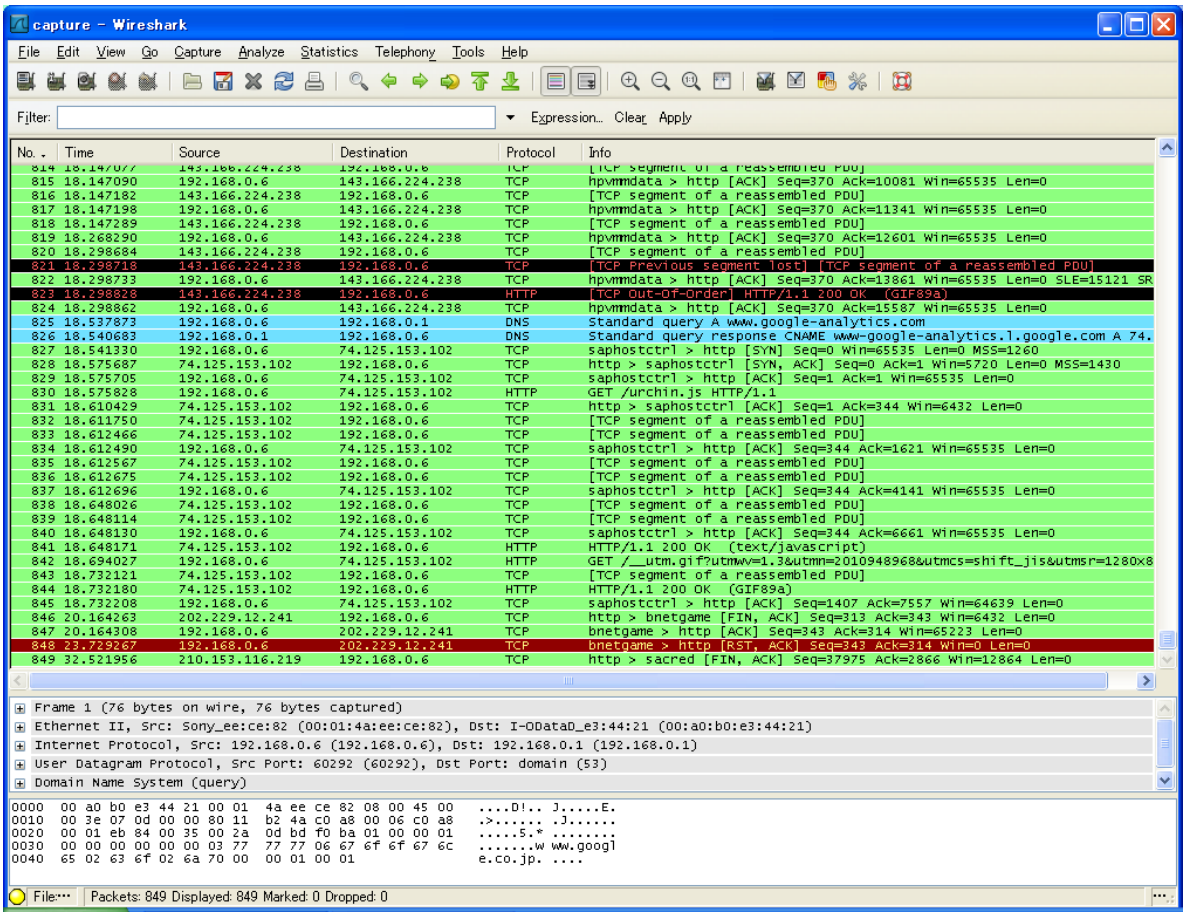

図 D.1　Wireshark によるキャプチャリング時のスクリーンショット

- ストリーミングの場合
  1. Firewall などの常駐ソフトを終了
  2. Wireshark を起動し，キャプチャ開始
  3. オフラインコンテンツや履歴の削除
  4. キャプチャ数 0 の状態で，任意のストリーミングファイル orURL を開く
  5. 任意時間経過後にキャプチャ終了

ストリーミングを Windows Media Player で聴く場合，プレイヤー起動時に不要なパケットが多く流れるため，しばらく放置してパケットが流れないことを確認してからキャプチャを行う．他のサービスについて補助ツールを使用する場合，そのサービスを利用する上で必要な挙動の場合 (ex. Messenger クライアントの起動・終了) はそのサービスのパケットとしてキャプチャする．また，そのサービスに不要な挙動の場合 (ex. PeerCast 補助クライアントのチャンネル更新) はキャプチャしない．

D.4　ノイズフィルタリング

本研究で用いたトラヒックデータにおけるノイズフィルタの一例を以下に示す.

1. 全サービス共通 (自身 IP に関わらないパケットの除去)
   ip.addr == 自身の IP アドレス
2. 各サービスの処理
   - BitTorrent
     (!browser && !tcp.port == 135 && !tcp.port == 445 && !tcp.port == 6000 && !tcp.port == 1433 && !udp.port == 138 && !tcp.port == 139)
   - Online Game1
     (dns || ip.src == 202.213.231.27 || ip.dst == 202.213.231.27 || tcp.port == 5484 || tcp.port == 12000 || tcp.port == 54848 || tcp.port == 19102 || tcp.port == 19101 || ip.src == 211.13.215.57 || ip.dst == 211.13.215.57)
   - FTP
     (ip.src == 204.152.184.73 || ip.dst == 204.152.184.73 || ip.src == 133.243.3.209 || ip.dst == 133.243.3.209)
   - メール受信 (POP3s のポート番号)
     !(!(tcp.port == 995) && !dns)
   - メール送信 (SMTP のポート番号)
     !(!(tcp.port == 587) && !dns)
   - Messenger
     !(!(ip.src == 64.4.0.0/16) && !(ip.dst == 64.4.0.0/16) && !(ip.src == 65.54.0.0/16) && !(ip.dst == 65.54.0.0/16) && !(ip.src == 207.46.0.0/16) && !(ip.dst == 207.46.0.0/16) && !(ip.src == 219.63.0.0/16) && !(ip.dst == 219.63.0.0/16) && !(ip.src == 120.74.243.0/24) && !(ip.dst == 120.74.243.0/24) && !(ip.src == 124.255.194.0/24) && !(ip.dst == 124.255.194.0/24))
     !(!(ip.src == 219.63.65.152) && !(ip.dst == 219.63.65.152) && !(ip.src == 120.74.243.180) && !(ip.dst == 120.74.243.180) && !(ip.src == 124.255.194.26) && !(ip.dst == 124.255.194.26) && !(tcp.port == 1863 || tcp.port == 443 || (udp && !dns)))
   - ニコニコ動画 (全パケットを確認して不要部を除去)
     !(!dns && !(ip.src == 202.248.110.0/24) && !(ip.dst == 202.248.110.0/24) && !(ip.src == 125.63.42.0/24) && !(ip.dst == 125.63.42.0/24) && !(ip.src == 202.219.105.0/24) && !(ip.dst == 202.219.105.0/24) && !(ip.src == 119.110.89.0/24) && !(ip.dst == 119.110.89.0/24) && !(ip.src == 210.135.0.0/16)

&& !(ip.dst == 210.135.0.0/16) && !(ip.src == 66.249.89.0/24) && !(ip.dst == 66.249.89.0/24) && !(ip.src == 125.56.203.0/24) && !(ip.dst == 125.56.203.0/24) && !(ip.src == 192.221.0.0/16) && !(ip.dst == 192.221.0.0/16) && !(ip.src == 61.0.0.0/8) && !(ip.dst == 61.0.0.0/8) && !(ip.src == 74.125.153.0/8) && !(ip.dst == 74.125.153.0/8))

- Online Game2
  (dns || tcp.port == 80 || tcp.port == 7911 || tcp.port == 9018 || tcp.port == 10804 || ip.src == 202.213.231.0/24 || ip.dst == 202.213.231.0/24)
- Peercast
  (tcp.port == 7144 || tcp.port == 7145 || tcp.port == 5158)
- Skype-chat
  !(arp || tcp.port == 1130 || tcp.port == 135 || udp.port == 137 || udp.port == 138 || tcp.port == 139 || tcp.port == 445 || tcp.port == 6000 || tcp.port == 23 || tcp.port == 1433 || (ip.src == 239.255.255.250) || (ip.dst == 239.255.255.250))
- Streaming1
  (media player に関するもの (スタイルシートなど)，またこのサービスのみに出現する HTTP や gif ファイル，関係ない DNS を除去)
  !(!(ip.src == 88.191.102.0/24) && !(ip.dst == 88.191.102.0/24))
- Streaming2
  !(!(ip.src == 64.12.61.3) && !(ip.dst == 64.12.61.3) && !(ip.src == 72.26.204.0/24) && !(ip.dst == 72.26.204.0/24) && !(ip.src == 87.98.169.0/24) && !(ip.dst == 87.98.169.0/24) && !(ip.src == 88.191.69.0/24) && !(ip.dst == 88.191.69.0/24) && !(ip.src == 64.74.207.0/24) && !(ip.dst == 64.74.207.0/24) && !(ip.src == 205.188.234.2) && !(ip.dst == 205.188.234.2))
- HTTP ファイル転送 (imgset.gif も除去)
  !(!(ip.src == 203.191.227.56) && !(ip.dst == 203.191.227.56) && !(ip.src == 211.13.211.4) && !(ip.dst == 211.13.211.4))
- YouTube
  !(!dns && !(ip.src == 74.125.0.0/16) && !(ip.dst == 74.125.0.0/16) && !(ip.src == 66.249.0.0/16) && !(ip.dst == 66.249.0.0/16) && !(ip.src == 72.14.0.0/16) && !(ip.dst == 72.14.0.0/16))

ここに示したノイズフィルタは，そのサービスのサーバのアドレスに依存して変化するものもあるため，適宜変更する必要がある．また，Skype および BitTorrent については，処理が定まらないため，明らかに不要な部分のみ除去している．これらについて，現在判明している不要なパケットを以下に示す．

- casp: tcp.port == 1130
- epmap: tcp.port == 135
- netbios-dgm: udp.port == 138
- netbios-ssn: tcp.port == 139
- microsoft-ds: tcp.port == 445
- x11: tcp.port == 6000
- telnet: tcp.port == 23
- ms-sql-s: tcp.port == 1433

また，BitTorrent について，TCP 以外で現在判明している必要なプロトコルを以下に示す．

- SSDP
- IGMP
- ICMP
- UDP
- NAT-PMP
- NBNS

関連業績

【 受賞 】

MWS2011 **学生論文賞**

マルウェア感染検知のための経年変化を考慮した特徴量評価に関する一考察

2011 年 10 月 マルウェア対策研究人材育成ワークショップ (MWS2011)

川元 研治 (早稲田大学理工学術院)， 市田 達也 (早稲田大学理工学術院)， 市野 将嗣 (電気通信大学)， 畑田 充弘 (NTT コミュニケーションズ株式会社)，小松 尚久 (早稲田大学理工学術院)

【 学会発表 】

マルウェア感染検知のための経年変化を考慮した特徴量評価に関する一考察

2011 年 10 月 マルウェア対策研究人材育成ワークショップ (MWS2011)

川元 研治 (早稲田大学理工学術院)， 市田 達也 (早稲田大学理工学術院)， 市野 将嗣 (電気通信大学)， 畑田 充弘 (NTT コミュニケーションズ株式会社)，小松 尚久 (早稲田大学理工学術院)

スコアレベル融合を用いたマルウェア感染検知手法に関する一検討

2012 年 7 月　マルチメディア，分散，協調とモバイルシンポジウム (DICOMO2012) 市野　将嗣 (電気通信大学)， 川元　研治 (早稲田大学理工学術院)， 大月　優輔 (電気通信大学)， 畑田　充弘 (NTT コミュニケーションズ株式会社)， 吉浦 裕 (電気通信大学)

Evaluation of secular changes in statistical features of traffic for the purpose of malware detection

The 1st International Workshop on ”Data Mining for Info-Communication Service and its Diffusion”(DMICSiD2012) in SNPD2012 / Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing 2012 Studies in Computational Intelligence Volume 443, 2013, pp1-11 Kenji Kawamoto, Masatsugu Ichino, Mitsuhiro Hatada, Yusuke Otsuki, Hiroshi Yoshiura, and Jiro Katto

マルウェア感染検知のためのトラヒックデータにおけるペイロード情報の特徴量評価

2012 年 11 月 マルウェア対策研究人材育成ワークショップ (MWS2012)

大月　優輔 (電気通信大学)，市野　将嗣 (電気通信大学)， 川元　研治 (早稲田大学理工学術院)，畑田　充弘 (NTT コミュニケーションズ株式会社)， 吉浦 裕 (電気通信大学)

マルウェア感染検知のための経年変化を考慮した特徴量評価に関する一考察

川元　研治 †　　市田　達也 †　　市野　将嗣 ‡　　畑田　充弘 †‡　　小松　尚久 †

† 早稲田大学理工学術院基幹理工学研究科
169-8555　東京都新宿区大久保 3-4-1
{kawamoto, ichida, komatsu}@kom.comm.waseda.ac.jp

‡ 電気通信大学大学院情報理工学研究科
182-8585　東京都調布調布ヶ丘 1-5-1
ichino@inf.uec.ac.jp

†‡NTT コミュニケーションズ株式会社
108-8118　東京都港区芝浦 3-4-1　グランパークタワー 17F
m.hatada@ntt.com

あらまし 本研究では，マルウェア感染検知の既存研究でよく用いられている特徴量に対して，マルウェアに感染している感染トラヒックとマルウェアに感染していない正常トラヒックの識別実験により特徴量評価を行った．その際，特徴量毎にベクトル量子化で作成した正常時，感染時のコードブックとテストデータとの特徴空間上での距離を用いて識別を行った．本稿では，感染トラヒックデータとして CCCDATAset，正常トラヒックデータとして同じデータ収集日におけるあるイントラネットのトラヒックデータを使用して，年によらずマルウェア感染検知において有効である特徴量について考察した結果を報告する．

A study of feature evaluation considering effects of year for malware detection

Kenji Kawamoto†　Tatsuya Ichida†　Masatsugu Ichino‡　Mitsuhiro Hatada†‡
Naohisa Komatsu†

†Graduate School of Fundamental Science and Engineering, Waseda UNIversity
3-4-1 Okubo, Shinjuku-ku, Tokyo, 169-8555, JAPAN
{kawamoto, ichida, komatsu}@kom.comm.waseda.ac.jp

‡Graduate School of Informatics and Engineering, The University of Electoro-Communications
1-5-1 Chofugaoka, Chofu-si, Tokyo, 182-8585, JAPAN
ichino@inf.uec.ac.jp

†‡NTT Communications Corporation
Gran Park Tower 17F, 3-4-1 Shibaura, Minato-ku, Tokyo, 108-8118 Japan
m.hatada@ntt.com

Abstract In this paper, we evaluated features used in existing researches based on the experiment that showed how each features could discriminate anomaly traffic that was infected with malware from normal traffic that was not infected with malware. In this evaluation, we made discriminations using the distance between the normal or anomaly codebook made by each features using vector quantization and test data. In this paper, we used CCCDATAset as anomaly traffic data, some traffic data on an intranet which was the same date as anomaly traffic data as normal traffic. Then we report our consideration which features are effective for malware detection with no relation to year effects.

1　本研究の背景と目的

昨今のインターネットの普及により，マルウェアの脅威が広がっている．マルウェアとは悪意のあるソフトウェア (Malicious Software) の略称であり，その被害は個人情報の流出やパソコンの乗っ取りというように我々の生活を脅かす存在となっている．文献 [1] によると 2011 年度上半期の日本国内での被害報告数は約 4 千件にものぼっており，活動が表面化しないボットネットによる被害や Web からの感染の増加，加えて日に日に新種のマルウェアが発生しているという現状で，早急に対策を講じる必要がある．

これまでの対策研究としては，文献 [2] で整理されているが，既知のマルウェアについての対策が中心であり，未知のマルウェアについての対策が不十分という問題がある．そこで本研究では，マルウェアに感染していない状態とマルウェアに感染している状態そのものに違いがあると考え，トラヒックデータを用いた未知のマルウェア感染検知に着目する．さらに，トラヒックデータには時間的な変化があり，時間的な変化に着目することで感染検知の性能が向上する可能性がある．例えばバイオメトリクスでは，発話時の唇動作個人認証において複数のアルゴリズムが提案されているが，時系列を使用しないアルゴリズムと使用するものを比較した際，後者の方が高い精度での認証が可能であることが示されている [3]．

本研究では，ネットワークユーザのプライバシーの問題を考慮し，インターネットとユーザ間のトラヒックデータのパケットペイロードは参照せずに，ヘッダ情報から様々な特徴量を抽出し，それらを識別器に入力することで感染の有無を判定する．しかし，ヘッダ情報から抽出する特徴量について，既存の研究では十分な評価が行われていない．そこで本稿では，感染トラヒックとして CCCDATAset2009, 2010, 2011[4](以下 CCC2009, CCC2010, CCC2011)，正常トラヒックとして同じデータ収集日におけるあるイントラネットのトラヒックデータを用いて，各特徴量の感染トラヒックと正常トラヒックを識別する能力を評価し，マルウェア感染検知に有効と思われる特徴量について検討する．さらに，3 年間の経年変化も考慮することで，年々移り変わるマルウェアや新種のアプリケーションによるトラヒックの変化があっても，大きな影響を受けずに，識別することができる特徴量について考察する．

以下では，既存研究で良く用いられている特徴量を紹介し，本稿で評価する特徴量について述べる．そして，各特徴量の識別能力を評価する実験を行い，識別率の高い特徴量についての考察を述べる．

2　既存の特徴量

既存のマルウェア感染検知やネットワーク異常検知に関する研究で用いられている特徴量を紹介する．

文献 [5] では，タイムスロット型の検出モジュールとフローカウント型の検出モジュールを組み合わせてネットワーク異常検知を行っている．タイムスロット型とは，一定の時間間隔 (タイムスロット) でのトラヒック流量をカウントし，各特徴量を抽出する方式である．タイムスロット型で用いられている特徴量には，TCP の各フラグの出現回数 5 種類や TCP，UDP，ICMP パケット数等がある．一方，フローカウント型とは，フロー毎に特徴を抽出する方式である．フローとは，プロトコル，送信元 IP アドレスとその送信元ポート番号，宛先 IP アドレスとその宛先ポート番号が同じパケット群である．フローカウント型で用いられている特徴量には，パケット数，フラグメントされたパケット数，同一ポート番号のフロー出現回数の逆数等がある．本研究ではタイムスロット型のネットワーク観測方式を採用する．なぜなら，実際の感染検知では，迅速なマルウェア検知が求められるため，同一フローのパケットを全て収集してからでないと各特徴量を抽出できないフローカウント型だと，リアルタイム性に欠けるためである．

文献 [6] では，ネットワークトラヒックから複数の通常状態を定義するクラスタリング手法を提案している．その際に，特徴量として ICMP，SYN パケット数，FIN パケット数，SYN,FIN 以外の TCP パケット数，UDP パケット数を 60 秒毎にカウントし，それらを正規化したものを用いている．

文献 [7] では，ボットと人間の通信挙動には差異があると考え，特定のホスト間におけるデータ送信時間間隔を特徴量として見ることにより，正常なクライアントとボットクライアントの差異を確認している．

既存研究より，特定のパケット数，到着間隔，TCP フラグ，ポート番号に関する特徴量がよく用いられていることがわかる．

表 1: 特徴量 36 種類

番号	特徴量 [単位]
1	パケット数
2	パケットサイズの総数 [byte]
3	パケットサイズの平均 [byte]
4	パケットサイズの最小 [byte]
5	パケットサイズの最大 [byte]
6	パケットサイズの標準偏差 [byte]
7	到着間隔の平均 [秒]
8	到着間隔の最小 [秒]
9	到着間隔の最大 [秒]
10	到着間隔の標準偏差 [秒]
11	SYN パケット数
12	FIN パケット数
13	PSH パケット数
14	ACK パケット数
15	RST パケット数
16	URG パケット数
17	SYN/ACK パケット数
18	FIN/ACK パケット数
19	PSH/ACK パケット数
20	RST/ACK パケット数
21	TCP パケット中の SYN パケット割合
22	TCP パケット中の FIN パケット割合
23	TCP パケット中の PSH パケット割合
24	TCP パケット中の ACK パケット割合
25	TCP パケット中の RST パケット割合
26	TCP パケット中の URG パケット割合
27	TCP パケット中の SYN/ACK パケット割合
28	TCP パケット中の FIN/ACK パケット割合
29	TCP パケット中の PSH/ACK パケット割合
30	TCP パケット中の RST/ACK パケット割合
31	ICMP 到達不能メッセージ数
32	UDP パケット数
33	送信元ポート番号が 69/UDP のパケット数
34	送信元ポート番号が 80/TCP のパケット数
35	送信元ポート番号が 110/TCP のパケット数
36	送信元ポート番号が 443/TCP のパケット数

3 識別実験

3.1 用いる特徴量

本研究では既存研究をもとに，パケットのヘッダ情報から取得できる情報およびその統計値を特徴量として用いる．本研究で検討対象とする特徴量 36 種類を表 1 に示す．

3.2 実験諸元

3.2.1 評価方法

本研究での感染トラヒックと正常トラヒックの識別方法について説明する．はじめに，ベクトル量子化を用いて，感染トラヒックのみを用いて学習を行った感染コードブックと，正常トラヒックのみを用いて学習を行った正常コードブックを予め作成する．今回は，各特徴量を個別に評価することが目的であるので 1 次元コードブックを作成した．トラヒックデータから特徴量を抽出する際のタイムスロット幅は，0.1 秒，1 秒，10 秒，100 秒の 4 種類とし，ベクトル量子化のアルゴリズムには，LBG+Splitting を用い，そのレベル数は 2，4，8，16，32 の 5 種類とした．そして，予め感染トラヒックか正常トラヒックかのラベル付けされた各特徴量毎の 1 次元テストデータを与え，テストデータと感染，正常コードブックとの特徴空間上でのユークリッド距離を計算し，感染コードブックとの距離の方が小さければ感染，正常コードブックとの距離の方が小さければ正常と識別している．

識別結果に対する評価指標として True Positive Rate(以下 TPR) と True Negative Rate(以下 TNR) を用いる．TPR は感染トラヒックを感染トラヒックと正しく識別できた割合である．TNR は正常トラヒックを正常トラヒックと正しく識別できた割合である．各特徴量について，2009 年，2010 年，2011 年のトラヒックデータを用いて，各タイムスロット毎に感染か正常か識別し，TPR，TNR をそれぞれ算出した．

3.2.2 使用したデータについて

本研究では，感染コードブック作成のための学習データに CCC2009，正常コードブック作成のための学習データに 3 月 13 日から 3 月 15 日の 2009 年のトラヒックデータを用いた．テストデータは，感染トラヒックに CCC2009, CCC2010，CCC2011，正常トラヒックに感染トラヒックと同じデータ収集日のトラヒックデータを用いた．

本研究では，感染トラヒックとして CCC2009, 2010, 2011 の攻撃通信データを使用した．しかし，これらの攻撃通信データにはマルウェアに感染するまでのトラヒックが含まれている．今回の特徴量評価実験では感染時のみのデータを用いる必要がある．そこで本研究では，攻撃通信データから感染以降のトラヒックのみを切り出した．その手順としてまずは，取得環境独自の制御パケットをフィルタリングで除外した．次にハニーポットの OS のリセット間隔で切り出す．そして，切り出したファイルを攻撃元データのログファイルの時刻と照らし合わせて，感染を確認し，実際の感染攻撃の開始パケットを探し，それ以降を感染トラヒックとして抽出した．

4　経年変化を考慮した特徴量評価

特徴量全体の傾向としては，年を追うごとに識別率が低下する特徴量が多かった．これは，年々移り変わるマルウェアや新種のアプリケーションの発生にともなう，トラヒックの複雑化，多様化が原因だと思われる．今後も新種のマルウェアやアプリケーションは増加すると思われるので，マルウェア感染検知では，トラヒックの変化の影響をあまり受けずに正しく識別できる特徴量を用いることが必要とされる．その点で経年変化を確認することは重要であると考えられる．

感染検知に有効な特徴量について検討するため，経年変化が小さく TPR,TNR が共に高い特徴量と，経年変化が小さく TPR のみ高い特徴量を抜き出す．なお，経年変化が小さく識別率が高い特徴量とは，あるタイムスロット幅，ベクトル量子化レベル数において，3 年間の TPR, または TNR が 90%以上であった特徴量を指す．マルウェア感染検知の要件は，感染と正常を正しく識別できる特徴量を用いること，加えて，感染のみを正しく識別できる特徴量も合わせて使用することである．

4.1　TPR,TNR 共に高い特徴量

今回の実験では，パケットサイズの最小値が唯一 3 年間の経年変化が小さく，TPR,TNR が共に 90%を超えていた特徴量であった．特徴量としてパケットサイズの最小を用いた際の，経年変化が小さくなるタイムスロット幅と量子化レベル数の組み合わせ，および TPR,TNR をそれぞれ以下の表 2，表 3 に示す．

表 2: パケットサイズの最小の TPR

タイム スロット幅	量子化 レベル数	2009 TPR	2010 TPR	2011 TPR
1 秒	8	99.3%	99.8%	91.5%
1 秒	16	98.9%	100%	91.5%
10 秒	16	99.1%	100%	94.6%
100 秒	8	98.4%	98.7%	93.8%
100 秒	16	98.7%	98.9%	95.5%

次に，タイムスロット幅 1 秒における，パケットサイズの最小値についてのテストデータの平均と標準偏差を以下の表 4 に示す．

表 4 より，感染トラヒックの方が正常トラヒックよりパケットサイズの最小値と，最小値の変動が大きくなることがわかる．すなわち，正常トラヒックでは常にパケットサイズの最小値が 60byte に近い値を取るが，感染トラヒックではパケットサイズの最小値が様々な値を取り得るという性質の違いがある．この性質の違いは，感染検知に有効であると思われる．

表 3: パケットサイズの最小の TNR

タイム スロット幅	量子化 レベル数	2009 TNR	2010 TNR	2011 TNR
1 秒	8	99.3%	100%	98.4%
1 秒	16	99.3%	100%	98.4%
10 秒	16	100%	100%	100%
100 秒	8	100%	99.8%	100%
100 秒	16	100%	99.8%	99.8%

表 4: テストデータの統計値 (パケットサイズの最小)

	2009	2010	2011
平均 (感染)	70.3[byte]	63.8[byte]	97.0[byte]
平均 (正常)	60.0[byte]	60[byte]	60.7[byte]
標準偏差 (感染)	36.0[byte]	2.2[byte]	50[byte]
標準偏差 (正常)	0.2[byte]	0[byte]	4.9[byte]

4.2　TPR のみ高い特徴量

経年変化が小さく，TPR が 3 年間を通して 90%以上の特徴量について，3 年間の TPR の平均が最も高くなるときの TPR の平均値，およびタイムスロット幅とベクトル量子化レベル数を以下の表 5 にまとめる．特徴量番号は表 1 の番号に従う．

表 5 の特徴量を大別すると，パケット数，パケット割合，ポート番号に関するパケット数に分けることができる．

4.2.1　パケット数

表 5 より，タイムスロット幅 0.1 秒，量子化レベル数 2，特徴量として RST/ACK パケット数を用いると 2009 年,2010 年,2011 年の TPR が 100%となっている．しかし，タイムスロット幅を 0.1 秒としたときのテストデータを見ると，感染トラヒック，正常トラヒック共に全てのタイムスロットで RST/ACK パケットが 0 個か 1 個しかないことがわかる．しかし，感染コードブックは正常コードブックより小さい値をとっているため，テストデータとコードブックとの距離を計算すると，感染コードブックとの距

表 5: TPR が高い特徴量の最大 TPR とその条件

特徴量番号	最大 TPR	タイムスロット幅	量子化レベル数
4	98.0%	100 秒	32
14	99.4%	1 秒	8
15	99.8%	10 秒	8
17	99.9%	0.1 秒	2
18	99.9%	1 秒	4
19	96.1%	100 秒	2,8
20	100%	0.1 秒	2
21	98.7%	0.1 秒	16
24	97.9%	0.1 秒	8
25	99.8%	0.1 秒	4
27	99.3%	0.1 秒	8
28	98.1%	1 秒	32
29	98.5%	0.1 秒	8
30	99.4%	1 秒	16
31	99.7%	1 秒	32
32	99.9%	0.1 秒	4
34	100%	0.1 秒	16
35	100%	0.1,1,10,100 秒	2,4,8,16,32
36	100%	0.1,1,100 秒	2,4,8,16,32

離の方が小さくなってしまう．よって，RST/ACK パケット数が 0 か 1 しかないタイムスロットを全て感染トラヒックと判断してしまったため，TPR が 100%になったと考えられる．このように，今回の識別方法では，ほとんどのタイムスロットで特徴が似ているが，特徴の違いが現れる頻度が少ない特徴量を正しく識別できていないことがわかった．

同様の問題が SYN パケット数にも当てはまる．SYN パケット数は既存研究でよく用いられている特徴量である．ポートスキャンや SYN フラッド攻撃の際には，SYN パケットが大量に流れるが，それ以外のときでは感染トラヒック，正常トラヒック共にほとんど流れていない．しかし，学習にポートスキャンや SYN フラッド攻撃などを含んだトラヒックを用いると，感染コードブックの方が正常コードブックより大きな値をとってしまう．そして，SYN パケット数が少ないタイムスロットを全て正常トラヒックと判断してしまう．このような特徴量は，今回の識別方法では感染トラヒックと正常トラヒックを正しく識別できていないが，他の特徴量と組み合わせて用いることで，感染検知に有効な特徴量になり得ると思われる．

さらに，今回の実験においてある特定のパケットの数を特徴量とすることには問題がある．感染トラヒックと正常トラヒックには，1 タイムスロット中のパケット数に大きく差があるという問題が存在するからである．一例として，2011 年のテストデータにおける感染トラヒックと正常トラヒックのパケット数の平均を以下の表 6 に示す．

表 6: 1 タイムスロットにおけるパケット数の平均

タイムスロット幅	感染	正常
0.1 秒	2.4	53.5
1 秒	6.2	252.8
10 秒	21.8	3051.6
100 秒	549.1	8724.6

今回の実験におけるテストデータの 1 つのサンプルは，1 つのタイムスロットと対応している．そのため，1 つのタイムスロットにおいて，感染トラヒックと正常トラヒックにパケット数の差があり過ぎるとそれが特徴となり，感染トラヒックと正常トラヒックに違いが現れてしまい，特徴量を正しく評価できない場合がある．そこで，パケットの数自体に関する特徴量の有効性を補完するための手段として割合を考える．

4.2.2 パケット割合

例えば，ACK パケット数に関して，タイムスロット幅を 10 秒とすると，経年変化が小さく TPR が高かった．次に，TCP パケット中の ACK パケット割合について見てみる．タイムスロット幅を 10 秒としたときの，2011 年の感染トラヒックと正常トラヒックの TCP パケット中の ACK パケット割合のヒストグラムを以下図 1, 図 2 に示す．

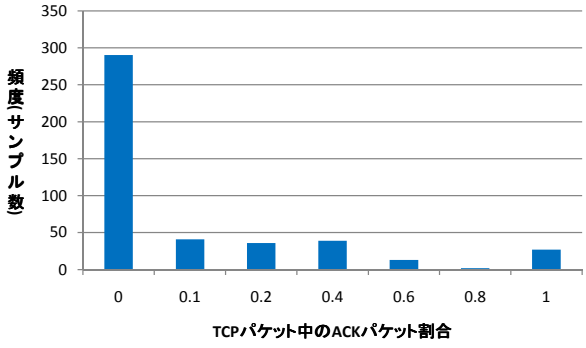

図 1: TCP パケット中の ACK パケット割合 (感染)

図 1, 図 2 より，感染トラヒックでは ACK パケット割合が低く，正常トラヒックでは ACK パケット割合が高いことが確認できた．正常トラヒックではサイズの大きいデータのやり取りが多く，データ通

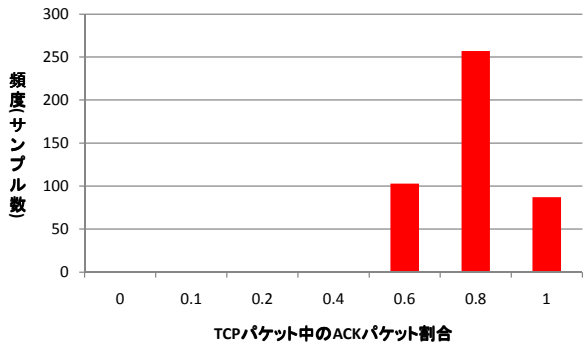

図 2: TCP パケット中の ACK パケット割合 (正常)

信の際の制御パケットが多く流れているため, ACK パケット割合が高くなっていると考えられる.

このように，特徴量としてある特定のパケット数を用いる場合，パケット数だけではなく，その割合も確認することで，感染トラヒックと正常トラヒックの明らかな違いを確認することができた．そして，ACK パケット数，および TCP パケット中の ACK パケット割合は，感染検知に有効な特徴量であることがわかった．しかし，その他の TCP パケット割合は，感染トラヒック，正常トラヒック共に割合が低く，割合がほとんどのタイムスロットで 0.1 より低かった．すなわち，感染トラヒックと正常トラヒックの全パケット数が同じになった場合，パケット数では感染トラヒックと正常トラヒックに違いが現れない可能性があると思われる．

4.2.3　ポート番号に関するパケット数

ポート番号に関するパケット数を特徴量とした場合，TPR が 100%となる場合が多い．しかし，送信元ポート番号が 80/TCP のパケット数は，4.2.1 節で挙げた感染トラヒックと正常トラヒックのパケット数の差の影響を大きく受け，正しく評価できていない可能性がある．さらに，送信元ポート番号が 110/TCP,443/TCP のパケット数にも，4.2.1 節で挙げた RST/ACK パケット数の評価の際の誤識別の問題がある．このような感染トラヒックと正常トラヒックで特徴の違いが現れる頻度の少ない特徴量を正しく評価するためには，別の識別方法を用いる必要があると考えられる．

5　まとめ

本稿ではマルウェア感染検知で用いる特徴量について，その識別能力を調査し，マルウェア感染検知に有効な特徴量について検討した．今回の識別実験から，特徴量全体として，識別率が年を追うごとに低下しているという傾向があり，さらに，今回の識別方法と使用したデータでは，識別率を正しく評価できない特徴量があることもわかった．しかし，特徴量として特定のパケット数を用いる場合，その割合も同時に使用することで感染トラヒックと正常トラヒックの違いが明らかになることを確認した．そして，パケットサイズの最小と ACK パケット数および TCP パケット中の ACK パケット割合が，感染検知に有効である特徴量として使用できる可能性があることを示した．

今後は，識別率の妥当性を検証できなかった特徴量に対して，マルウェア感染検知に対して有効であるか調査し，特徴量を組み合わせたマルウェア感染検知手法について検討していく．

参考文献

[1] インターネット脅威マンスリーレポート　2011 年 5 月度
http://jp.trendmicro.com/jp/threat/security_ news/monthlyreport/article/20110602082147.html

[2] 藤原将志，寺田真敏，安部哲哉，菊池浩明，”マルウェアの感染方式に基づく分類に関する検討”，情報処理学会 CSEC 研究報告，No.21，p177-182，2008 年 3 月

[3] 市野将嗣，坂野鋭，小松尚久，”核非線形相互部分空間法による話者認識”，”信学論 (D-)”,vol.J88,no.8,pp.1331-1338,2005.

[4] 畑田充弘，中津留勇，秋山満昭，”マルウェア対策のための研究用データセット〜MWS 2011 Datasets〜，”マルウェア対策研究人材育成ワークショップ 2011(MWS2011),October 2011.

[5] 佐藤陽平，和泉勇治，根元義章，”複数の検出モジュールの組み合わせによるネットワーク異常検出の高精度化”，電子情報通信学会，信学技報，2004 年

[6] 平松尚利，和泉勇治，角田裕，根元義章，”複数の通常状態を用いたネットワーク異常検出”，電子情報通信学会，信学技報，2006 年

[7] 釘崎祐司，笠原義晃，堀良彰，櫻井幸一，”データ送信間隔に着目した挙動の観測に基づくボット検知手法”，SCIS2009

スコアレベル融合を用いたマルウェア感染検知手法に関する一検討

市野　将嗣[†1]　　川元　研治[†2]　　大月　優輔[†1]
畑田　充弘[†3]　　吉浦　裕[†1]

本研究では，マルウェア感染検知にスコアレベル融合での特徴量融合を用いることを提案する．近年，マルウェアによる被害が多く報告されており，それらの対策として感染検知は不可欠である．そこでマルウェア感染時の通信トラヒックデータを正常時の通信トラヒックデータと比較することで感染の検知を行うシステムを検討する．マルウェア感染検知における特徴量の融合に関して，従来研究の多くが特徴レベル融合を用いている．ただし，各特徴量はヘッダ情報により取りうる値の範囲に差があり，特徴量によって異なる特徴があるのでそれらの特徴を融合すると分布が複雑になる可能性があり適切な識別器の設計が難しいと考えられる．さらに，タイムスロットに基づいた識別の場合，特徴量ごとに識別するのに適切な抽出間隔が異なる可能性があることを確認しており，この場合，特徴レベル融合では，抽出間隔が異なると抽出されるスロット数も異なるため融合が困難である．そこで，本稿ではマルウェア感染検知にスコアレベル融合での特徴量融合を用いることを提案し，研究用データセット CCCDATASet の攻撃通信データを用いた実験結果について報告する．

A study on malware detection method using score level fusion

Masatsugu ICHINO,[†1] Kenji KAWAMOTO,[†2]
Yusuke OHTSUKI,[†1] Mitsuhiro HATADA[†3]
and Hiroshi YOSHIURA[†1]

We propose a method of malware detection using score level fusion. The threat of stealth botnets and infections through web sites is especially increasing. Malware detection has becomes important for the safety of internet usage. We therefore studied the malware detection method by comparing malware traffic with normal traffic. Feature level fusion is often used in feature fusion of malware detection. We think it is difficult to design the classifier because each feature have own range of value and the distribution of fusion of each feature may become complexity. And it is difficult for feature level fusion to fuse the feature extracted from traffic data in which time slot width differs. In this paper, we evaluated the effectiveness of proposed method by using CCCDATASet.

1. まえがき

近年のインターネットの普及により，マルウェアの脅威が広がっている．マルウェアとは悪意のあるソフトウェア (Malicious Software) の略称であり，その被害は個人情報の流出やパソコンの乗っ取りというように我々の生活を脅かす存在となっている．マルウェアによる被害は拡大・深刻化しており，近年では活動が表面化しないボットネットによる被害の増加やガンブラーに代表される Web からの感染が増加しているという現状で，早急に対策を講じる必要がある．

本研究では，トラヒックデータを用いたマルウェアの感染検知に着目する．これは，正常時通信トラヒック，感染時通信トラヒックの特徴をとらえて，パターン認識の技術を用いて感染を検知する手法である．感染したコンピュータ上では感染していることを正しく検知することは難しい．トラヒックデータを用いた感染検知は，対象とする機器での通信トラヒックの入出力のみを用いる手法である．基本的には感染すると何らかのトラヒックが発生するため，トラヒックデータを用いた感染検知は対象とする機器の外部からの感染検知手法として有望であり，さらに，現在，利用が進んでいる家庭の家電製品や会社の機器端末などのネットワーク接続の際のマルウェア感染検知への適用も期待できる．

本研究では，暗号化された通信にも対応でき，かつプライバシーの問題を考慮し，ヘッダ情報の統計量を用いた感染検知手法を検討した．ヘッダ情報の統計量 (パケットサイズの平均値など) は多く考えられるため，どのようにそれらの特徴量を融合して識別するのかを検討する必要があり，これは識別器の統合方法に基づいている． しかしこれまでに特徴量の融合方法についてはほとんど検討されていない．特徴量の融合方法を工夫することによりさらに識別精度向上する可能性がある．そこで，適切な識別器の設計のため，本検討では統計情報を使用したマルウェアによる感染を検知する方法に関して，識別精度に影響のある特徴量の融合方法に関して検討した．

†1 電気通信大学大学院情報理工学研究科総合情報学専攻
†2 早稲田大学大学院基幹理工学研究科情報理工学専攻
†3 NTT コミュニケーションズ株式会社

識別器の統合方法は, 結果レベル融合，特徴レベル融合とスコアレベル融合に分けられる[4)5)]．

結果レベル融合は，各識別器から出力された識別結果を論理演算 (AND,OR) によって融合する手法である．AND,OR による融合は実装が容易であるという利点があるが，AND を用いた場合は FPR(False Positive Rate，正常のデータを識別器が感染と判定した割合) のみが減少し，FNR(False Negative Rate，感染のデータを識別器が正常と判定した割合) は増加する，OR を用いた場合は FNR のみが減少し，FPR は増加する．このように AND,OR による融合は，FPR と FNR のいずれか一方を改善するが，他方は悪化する．

マルウェア感染検知における特徴量の融合に関して，従来研究の多くが特徴レベル融合を用いている[1)2)]．特徴レベル融合は対象データから得られた各特徴量を並べたものを特徴ベクトルとみなし，識別を行う方法である．各ヘッダ情報の統計量の特徴や特徴ベクトルによって張られる特徴空間での分布の様子を踏まえた識別器の設計が必要となる．ただし，各特徴量はヘッダ情報により取りうる値の範囲に差があり，特徴量によって異なる特徴があるのでそれらの特徴を融合すると分布が複雑になる可能性があり適切な識別器の設計が難しいと考えられる．さらに，タイムスロットに基づいた識別の場合，特徴量ごとに識別するのに適切な抽出間隔が異なる可能性があることを確認しており，この場合，特徴レベル融合では，抽出間隔が異なると抽出されるスロット数も異なるため融合が困難であるが，スコアレベル融合では，マッチングスコアから融合を行うため融合が可能である．

スコアレベル融合の識別的手法は，各識別器から出力されるスコアを並べたものを特徴ベクトルとみなしパターン認識技術を用いて認識する方法である．一般に，パターン認識の分野では密度関数の推定に基づく識別法よりも，特徴空間でクラス 1 の分布とクラス 2 の分布を分離する面である識別面を直接的に推定する識別法の方が高性能な識別器を設計できることが知られている．この傾向は，サンプル数が少ないときに顕著である[3)]．スコアレベル融合では各特徴量で識別した際に求まるスコアに対してさらに識別を行うため，スコア空間 (各識別器より出力されるスコアを各軸にとった空間) での分布の様子を考慮した識別を行うことで識別性能が向上する可能性がある．そこで，本稿ではマルウェア感染検知にスコアレベル融合での特徴量融合を用いることを提案する．

本稿では，研究用データセット CCCDATASet の攻撃通信データを用いた実験結果を報告し，マルウェア感染検知に対してスコアレベル融合の適用に関する可能性を実験的に示す．以下，**2.** ではスコアレベル融合を用いた感染検知手法を提案する．**3.** では，研究用データセット CCCDATASet の攻撃通信データを用いた実験結果を示す．**4.** はまとめと今後の課題である．

2. 識 別 手 法

2.1 スコアレベル融合

スコアレベル融合は，図 2 に示すように N 個の識別器から出力されるスコアを並べたものを N 次元の特徴ベクトルとし，正常・感染の 2 クラスの識別問題とみなして，パターン認識技術を適用する手法である．具体的には，図 2 に示すようなスコア空間[⋆1]でスコアを適切に分離するような識別境界面 (識別関数) を作成し，識別を行う．これにより FPR,FNR を同時に改善することができる．

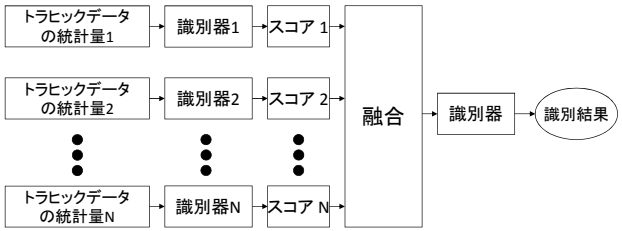

図 1 スコアレベル融合

スコアレベル融合は，特徴空間の次元数分，つまり識別器の数だけの空間的自由度をもつために，より高い認識精度が得られる可能性がある．また，認識精度向上のみを目的とするのではなく，セキュリティレベルに応じた閾値制御 が可能である．一般に，パターン認識の分野では密度関数の推定に基づく識別法よりも，特徴空間でクラス 1 の分布とクラス 2 の分布を分離する面である識別面を直接的に推定する識別法の方が高性能な識別器を設計できることが知られている．この傾向は，サンプル数が少ないときに顕著である[6)]．スコア空間での分布の様子に基づいて，識別面を構成するアルゴリズムを適用すれば識別精度の向上が期待できる．そこで，本研究では, スコアレベル融合を用いることを提案する．

本研究では，入力トラヒックに対して正常のテンプレート[⋆2]と比較した際に求まるスコア

⋆1 各識別器より出力されるスコアを軸にとった空間
⋆2 3 章ではコードブックと表記.

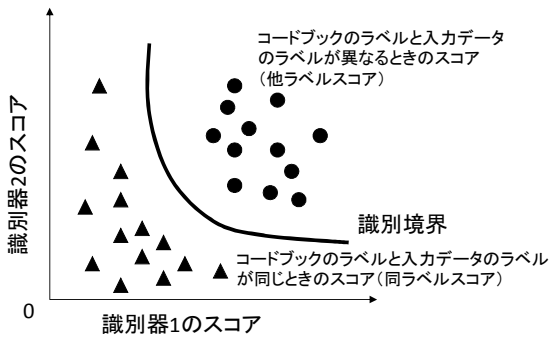

図 2 スコア空間での識別

と，入力トラヒックに対して感染のテンプレートと比較した際に求まるスコアのうち，どちらが同ラベルスコア (学習データとテンプレートのラベルが同じもの) であるかを判定し，正常か感染かを識別する．

2.2 カーネル判別分析

トラヒックデータは，様々な挙動の組み合わせで構成され使用する特徴量により取り得る値の範囲が異なるため，特徴空間での識別境界面は非線形性になる可能性がある．識別面が非線形性を示す場合，線形性を仮定したアルゴリズムではうまく識別できないという問題がある．また，単純に統合により精度向上を狙うのではなく実用的な見地からの検討も重要である．実用シーンを考慮すると，セキュリティレベルに応じて閾値を制御できることが望ましい．たとえばネットワークのセキュリティの場合を考えてみると，個人使用の PC 利用時では，FP(正常のデータを識別器が感染と判定した割合) を下げたいという要件がある．それに対して，重要な情報が保管されている役所等の PC やサーバでは，FN(感染のデータを識別器が正常と判定した割合) を下げたいという要件がある．このように場面に応じた制御が必要である．これらを踏まえ，各識証器から出力されるスコアを特徴ベクトルとし，正常・感染の 2 クラスのパターン識別問題であるとみなし，識別アルゴリズムにカーネル判別分析[7)8)](Kernel Fisher Discriminant Analysis，以下 KFDA) を用いることを提案する．

KFDA は，特徴空間を無限次元もしくは極めて高次元の空間に変換した後でフィッシャーの線形判別分析 (Linear Discriminant Analysis，以下 LDA) を行う手法である．また，LDA[6)] は，特徴空間上の 2 クラスのサンプルの分布からこの 2 クラスを識別するのに最適な 1 次元軸を求める手法である．具体的には，クラス内共分散行列・クラス間共分散行列比を最大にする 1 次元軸を求める方法である．

特徴量の統合に際しては，まず各特徴量の識別器から出力されるスコアを並べた 2 クラスの 2 次元特徴ベクトル $\vec{x}$ に対して，正常，感染のクラス C_1, C_2 に識別することを考える．あらかじめ与えられているクラス $C_i (i = 1, 2)$ の学習用データの集合を χ_i とする．

非線形な関数表現を考えるために関数空間 $\mathcal{F}$ への非線形写像 Ψ を考える．ただし，$\mathcal{F}$ は極めて高次元もしくは無限次元の関数空間である．すると，空間 $\mathcal{F}$ におけるクラス内共分散行列 V^Ψ とクラス間共分散行列 B^Ψ は，n_i をクラス C_i の特徴ベクトルの数，$M = \sum_{i=1,2} n_i$，$\vec{m}_i^\Psi$ をクラス C_i の特徴ベクトルの平均とすると，

$$V^\Psi = \frac{1}{M} \sum_{i=1,2} \sum_{\vec{x} \in \chi_i} \Psi(\vec{x}) \Psi(\vec{x})^T \tag{1}$$

$$B^\Psi = \frac{1}{M} \sum_{i=1,2} n_i \vec{m}_i^\Psi \vec{m}_i^{\Psi T} \tag{2}$$

と書ける．ただし，ここでは簡単のためにデータの重心と座標原点が一致しているものと仮定する．

1 次元空間への変換を表す射影ベクトルを $\vec{w}$ とする．これより，空間 $\mathcal{F}$ 上でのクラス内共分散行列・クラス間共分散行列の比 $J(\vec{w})$ は，

$$J(\vec{w}) = \frac{\vec{w}^T B^\Psi \vec{w}}{\vec{w}^T V^\Psi \vec{w}} \tag{3}$$

のようになる．

LDA との対比から，式 (3) を最大にする $\vec{w} \in \mathcal{F}$ を求める．しかし，これらの行列やベクトルは極めて高次元もしくは無限次元空間に存在するため，直接計算を行うことは極めて困難か不可能である．そこで，核非線形主成分分析[9)](Kernel Principal Component Analysis，以下 KPCA) や SVM の議論と同様に，w はすべての学習パターンを特徴空間 $\mathcal{F}$ に射影したベクトルの線形結合

$$\vec{w} = \sum_{i=1}^{M} \alpha_i \Psi(\vec{x_i}) \tag{4}$$

で表すことができ，結合係数 $\alpha_i (i = 1 \cdots M)$ を要素に持つベクトル α を定義する．そして，Mercer の条件を満たす核関数

$$k(\vec{x},\vec{y}) = (\Psi(\vec{x})\cdot\Psi(\vec{y})) \tag{5}$$

を選択し，$\mathcal{F}$ での内積を核関数で置き換えることによって，式 (3) を

$$J(\alpha) = \frac{\alpha^T KWK\alpha}{\alpha^T KK\alpha} \tag{6}$$

と変形し，式 (6) を最大化する問題に置き換える．ただし，学習データの核関数により求まる行列 K とブロック対角行列 $W(W=(W_i)_{i=1,2}, W_i$ は成分がすべて $1/n_i$ である $n_i \times n_i$ 行列，W は $M\times M$ 行列) である．そして $J(w)$ の最大値は $\lambda V^\Psi w = B^\Psi w$ の最大固有値 λ で与えられることと K の固有ベクトル分解などを用いて，α を求めることができる[⋆1]．

ここでテスト用データの特徴ベクトル $\vec{x}'$ の w による一次元の識別空間への射影は，式 (4)(5) と上記で求められた α を用いて

$$w^T\Psi(\vec{x}') = \sum_{i=1}^{M} \alpha_i k(\vec{x_i},\vec{x}') \tag{7}$$

となり，$\vec{w}$ をあらわに求めないでもその写像 $\vec{w}^T\Psi(\vec{x}')$ を計算できるようになる．

ここで th を定数とし，テスト用データの特徴ベクトル $\vec{x}'$ に対する識別関数を

$$g(\vec{x}') = th + \vec{w}^T\Psi(\vec{x}') \tag{8}$$

と定義すると，$g(\vec{x}')$ の値の正負で正常，感染のクラス C_1, C_2 を識別することができる．さらに，式 (8) の th を閾値とすることで FAR，FRR を制御することができる．

3. 評 価 実 験

本章では，カーネル判別分析を用いたスコアレベル融合の有効性を確認するための実験を行った結果を報告する．

3.1 実験データ

実験データとして，正常時通信データと感染時通信データを用意する必要がある．正常時通信トラヒックデータとしては，あるイントラネットより取得したデータ，一方感染時通信トラヒックデータとして CCC2009,2010,2011[10] を用いた．

正常時トラヒックデータ，感染時トラヒックデータともに，2009 年 3 月 13,14 日と 2010 年 3 月 7，8 日と 2011 年 1 月 21，22 日のデータを使用した．また，感染時データに関しては CCC2009,2010,2011 内のログ情報をもとに明らかにマルウェアに感染していると考えられる通信データを切り出して用いた．学習データとテストデータの組み合わせについて

(1) 学習データとテストデータの取得が同時期の組み合わせ (学習データ:2009 年, テストデータ:2009 年)

(2) 学習データとテストデータの取得が異時期の組み合わせ (学習データ:2009 年，テストデータ:2010 年，学習データ:2009 年，テストデータ:2011 年)

の 2 パターンの実験を行った．

3.2 実験系の概要

実験系の概要を図 3 に示す．トラヒックデータから，一定の時間間隔 (タイムスロット幅) でのトラヒック流量をカウントし，識別に使用する各特徴量を抽出する．

その後，各特徴量での識別を行う．各特徴量の識別では，量子化誤差の算出に，LBG+Splitting アルゴリズムによるベクトル量子化[11] を用いた．トラヒックデータは，様々な挙動の組み合わせで構成され使用する特徴量により取り得る値の範囲が異なり，特徴空間での分布が複雑になる可能性があるためである．ベクトル量子化とは，入力データを任意個の代表パターン（コードブック）の値で近似する処理であり，LBG+Splitting アルゴリズムは，適当な初期コードブックから開始し，学習系列に分割条件と代表点条件を繰り返し適用することで良好なコードブックに収束させるコードブック設計アルゴリズムである．コードブックは，特徴量で張られる空間をベクトル量子化でレベル数分の領域 (クラスタ) に分割し，それぞれの領域にあるサンプル (タイムスロットに該当) の重心になる．スコアの算出については，タイムスロットごとにベクトル量子化のレベル数分のコードブックとタイムスロットとの距離 (二乗誤差) をそれぞれ求め，その最小値をそのタイムスロットのスコアとする．例えば，ベクトル量子化のレベル数を 8 とすると，対象とするタイムスロットとコードブックにある 8 個の重心との二乗誤差をそれぞれ求め，8 個の二乗誤差のうち最小値のものをタイムスロットのスコアとする．そして正常・感染それぞれのトラヒックに含まれるすべてのタイムスロットに対してタイムスロットのスコアを求め，それらの平均値をスコア (平均二乗誤差)) とする．学習データとコードブックのラベルが同じもの (図 3 を参照) から求まるスコアを同ラベルスコア，学習データとコードブックのラベルが異なるもの (図 3 を参照) から求まるスコアを他ラベルスコアとする．

同ラベルスコアと他ラベルスコアを分離するのに最適な部分空間をカーネル判別分析で求める．そして，識別の際にはトラヒックデータを正常・感染コードブックを用いてベクトル量子化し，スコアを求めた後，求めた部分空間へ写像し，正常コードブックから求まったスコアと感染コードブックから求まったスコアのどちらが同ラベルスコアであるかを判定し，

⋆1 導出の詳細は文献[8] を参照していただきたい．

正常か感染かを識別する．

本実験で使用する特徴量は文献[12] の結果に基づき，各タイムスロットにおける最小パケットサイズ，SYN パケット数，ACK パケット数である．

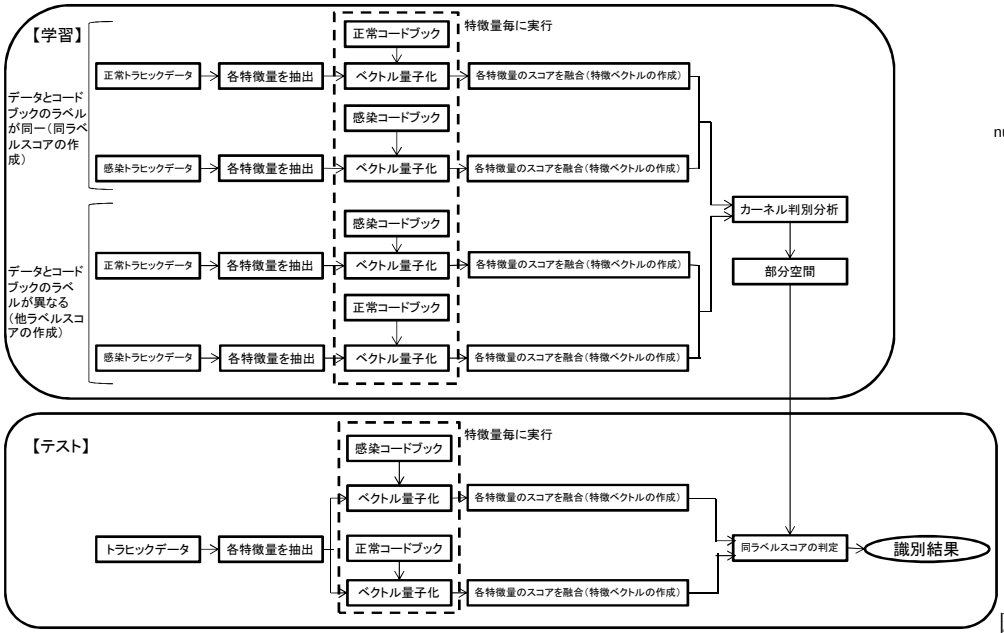

図 3　カーネル判別分析を用いたスコアレベル融合による感染検知

3.3 実 験 結 果

3.3.1 スコア空間の様子

図 3 に示す特徴ベクトル (各特徴量毎にベクトル量子化した際に求まるスコアを並べたベクトル) を 3 次元空間に写像することで分布の様子を調べた．

図 4 は，特徴量を最小パケットサイズ，SYN パケット数，ACK パケット数とし，各スロット毎に求めたスコアをプロットした際の散布図である．各軸は，最小パケットサイズ，SYN パケット数，ACK パケット数であり，同ラベルスコア (図の凡例では genuine) を×，他ラベルスコア (図の凡例では imposter) を□であらわした．図 4 の左下 (原点付近) をさらに拡大したものを図 5 に示す．図 6 は，SYN パケット数，ACK パケット数を用いた 2 次元での散布図である．

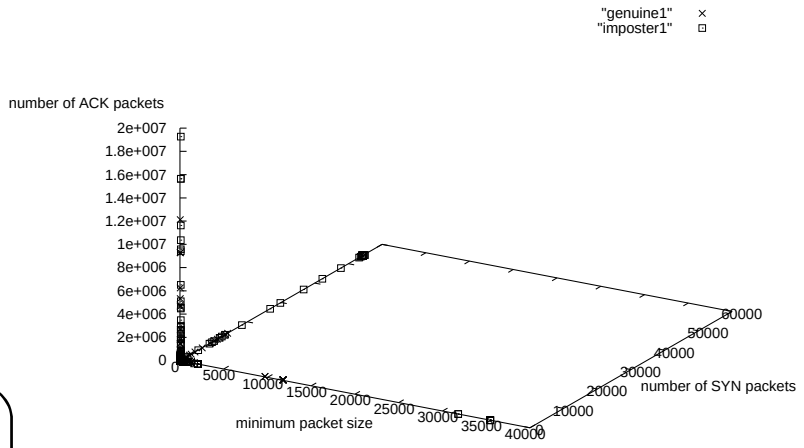

図 4　スコア空間の様子

図 4，5，6 の散布図において，同ラベルスコアと他ラベルスコアは複雑な分布となっている．そのため，線形の識別面では適切に分離することができず，非線形な識別面が必要であることがわかる．

3.3.2 特徴量を融合した際の識別結果

カーネル判別分析を用いたスコアレベル融合の有効性を確認するために，単一の特徴量で識別した場合と線形のアルゴリズムである LDA を用いたスコアレベル融合との比較を行った．

単一の特徴量で識別する場合は，ベクトル量子化により求まるスコアを用いて識別を行った．

核関数として，ガウス型動径基底関数

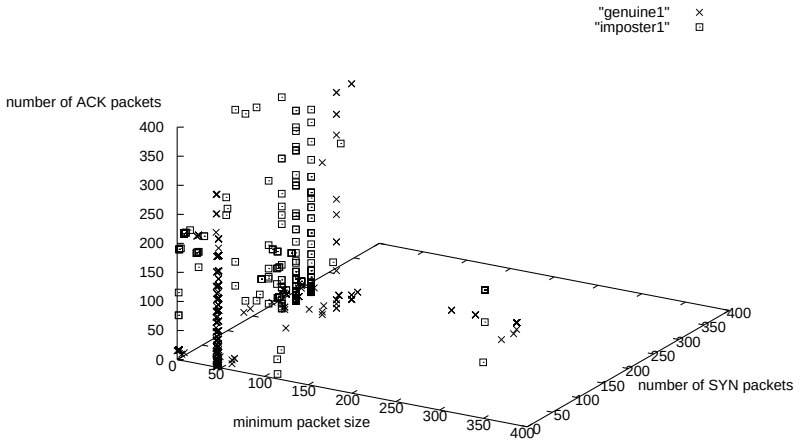

図 5 スコア空間の様子

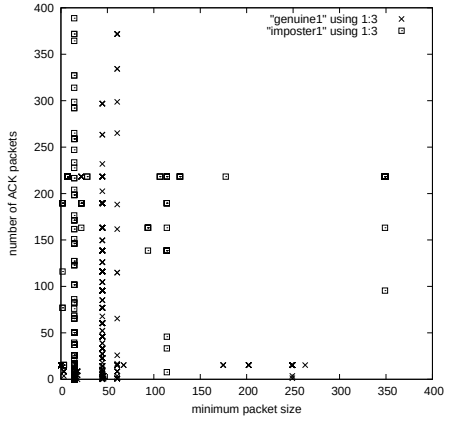

図 6 スコア空間の様子

$$k(\vec{x},\vec{y}) = \exp\left(\frac{-||\vec{x}-\vec{y}||^2}{2\sigma^2}\right) \tag{9}$$

を用い，予備実験により $\sigma = 0.18$ に設定した．また，特徴量抽出の際のタイムスロット幅を 1 秒とした．

識別結果を表 1 に示す[⋆1]．単一の特徴量とスコアレベル融合 (KFDA) を比較すると，スコアレベル融合 (KFDA) の識別率，TPR,TNR ともに精度が良くなっていることがわかる．スコアレベル融合 (LDA) と単一の特徴量の結果を比較すると，単一の場合に比べ精度が悪くなっている．つまり，スコアレベル融合を使うと必ず精度が上がるというわけではなく適切な方法で融合する必要があることがわかる．

3.4 考　　察

3.4.1 KFDA によるスコアレベル融合

スコアレベル融合 (KFDA) が，TPR，TNR ともに高い精度が得られた．その理由について考える．

表 1 を見ると，単一特徴量 (最小パケットサイズ) と単一特徴量 (ACK パケット数) は，TPR が 3 年間を通して高いことがわかる．これは，正常トラヒックは ACK を返す通信などで 60byte の値を取るが，感染トラヒックでは値に変化がみられこれが各スロットの最小パケットサイズとなるため感染を正しく識別できている．それに対して単一特徴量 (SYN パケット数) は，TNR が 3 年間を通して高いことがわかる．これは，正常トラヒックの SYN パケット数が感染トラヒックに比べ少ないため識別しやすくなっている．

単一特徴量 (最小パケットサイズ) と単一特徴量 (ACK パケット数) は TPR が 3 年間を通して高く，単一特徴量 (SYN パケット数) は TNR が 3 年間を通して高い．その特徴を踏まえ，同ラベルスコアと他ラベルスコアを離すような空間をカーネル判別分析で求め，その空間で識別を行っているためにスコアレベル融合 (KFDA) が，TPR，TNR ともに高い精度が得られたと考えられる．

また，表 1 を見ると，スコアレベル融合 (LDA) と単一の特徴量の結果を比較すると，単一の場合に比べ精度が悪くなっている．図 5 より同ラベルスコアの分布と他ラベルスコアの分布の境界が複雑になっているため，LDA による融合では適切に識別できなかったと考えられる．KFDA を実行する際に使用するカーネルパラメータ σ の値が小さいと複雑な境

⋆1 識別率:テストデータを正しく識別 (正常，感染) した割合，TPR:感染時通信のテストデータを識別器が感染と判定した割合，TNR:正常時通信のテストデータを識別器が正常と判定した割合．

表 1 識別結果
Table 1 Identification result

学習データ	2009 年取得データ								
テストデータ	2009 年取得データ			2010 年取得データ			2011 年取得データ		
識別手法	識別率 (%)	TPR(%)	TNR(%)	識別率 (%)	TPR(%)	TNR(%)	識別率 (%)	TPR(%)	TNR(%)
単一特徴量 (最小パケットサイズ)	45.8	91.3	0.2	47.4	94.9	0	39.3	78.6	0
単一特徴量 (SYN パケット数)	74.7	50.4	98.9	74.9	51.1	98.7	49.9	0	99.8
単一特徴量 (ACK パケット数)	65.6	98.9	32.4	64.0	98.2	29.7	74.7	95.1	54.2
スコアレベル融合 (LDA)	28.6	14.3	42.9	28.6	14.3	42.9	28.6	14.3	42.9
スコアレベル融合 (KFDA)	88.2	86.8	89.5	88.2	86.8	89.5	88.2	86.8	89.5

界を表すことができるが，今回の実験では KFDA のカーネルパラメータ σ の値が 0.18 と小さい値で高い識別率が得られたことからも複雑な境界が必要であったと考えられる．

さらに，表 1 より，スコアレベル融合の経年変化 (テストデータを 2009，2010，2011 年取得データに変えていった時の識別精度の変化) が単一の場合に比べ安定していることがわかる．これは，テストデータの取得時期が異なることにより変動が生じ，スコアにも変動が生じることとなるが，KFDA を用いたスコアレベル融合では特徴ベクトル (スコアを並べたベクトル) を高次元空間に写像するため元の空間 (図 5) で見られた分布の重なりが少なくなり，スコアに変動が生じても影響を受けなかったと考えられる．

3.4.2 各タイムスロット幅における識別

今回，実験で使用した最小パケットサイズ，SYN パケット数，ACK パケット数に関して，タイムスロット幅を変えた時の識別を行った．識別結果を表 2 に示す．

この結果を見ると，最小パケットサイズでは，タイムスロット幅 1 秒のときは TPR がよいが，タイムスロット幅 10 秒のときは TNR が良くなっている．SYN パケット数，ACK パケット数に関してもタイムスロット幅を変えることにより識別できるスロットも変わっていることより異なるタイムスロット幅で求めたスコアを融合することによりさらに識別精度を向上させることができる可能性がある．

4. むすび

本稿では，スコアレベル融合を用いたマルウェア感染検知手法を提案した．今回はスコア空間の様子を踏まえカーネル判別分析によるスコアレベル融合を適用し，実験的に有効性を示した．

今後は，他の特徴レベル融合との比較を行い有効性を確認していく．さらに他の研究用データセットを使用した評価実験も行い有効性を確認したい．

参考文献

1) S.Kondo and N.Sato, “Botnet Traffic Detection Techniques by C&C Session Classification Using SVM,” IWSEC2007, October 2007.
2) Livadas C., Walsh B., Lapsley D., Strayer T, “Using Machine Learning Techniques to identify botnet traffic,” In Proceedings of 2nd IEEE LCN Workshop on Network Security, November 2006.
3) 石井健一郎，上田修功，前田英作，村瀬洋, “ わかりやすい　パターン認識，”オーム社，1998.
4) J.Kitter *et al.*,“On Combining Classifiers,” IEEETrans. Pattern Analysis and Machine Intelligence, vol.20, no.3, pp.226-239,1998.
5) Robert P.W.Duin,“The Combining Classifier: to Train or Not to Train?,” In proc. of int. conf. on ICPR2002, August 2002.
6) 石井健一郎，上田修功，前田英作，村瀬　洋，“わかりやすいパターン認識，”オーム社，1998.
7) S.Mika, G.Rätsch, J.Weston, B.Schölkopf and K.R.Müller, “Fisher discriminant analysis with kernels,” Neural Networks for Signal Processing IX.IEEE,pp.41-48,1999.
8) G.Baudat and F.Anouar,“Generalized Discriminat Analysis Using a Kernel Approash,” Neural Computation,Vol.12, pp.2385-2404,2000.
9) B.Schölkopf *et al.*,“Nonlinear component analysis as a Kernel eigenvalue problem,” Neural Computation, vol.10, pp.1299-1319, 1998
10) 畑田充弘，中津留勇，秋山満昭，“マルウェア対策のための研究用データセット 〜MWS 2011 Datasets〜,” マルウェア対策研究人材育成ワークショップ 2011 (MWS2011)，October 2011.

表 2 各タイムスロット幅における識別結果
Table 2 Identification result of each timeslot

学習データ		2009 年取得データ								
テストデータ		2009 年取得データ			2010 年取得データ			2011 年取得データ		
タイムスロット幅 (秒)	識別手法	識別率 (%)	TPR(%)	TNR(%)	識別率 (%)	TPR(%)	TNR(%)	識別率 (%)	TPR(%)	TNR(%)
1	単一特徴量 (最小パケットサイズ)	45.8	91.3	0.2	47.4	94.9	0	39.3	78.6	0
	単一特徴量 (SYN パケット数)	74.7	50.4	98.9	74.9	51.1	98.7	49.9	0	99.8
	単一特徴量 (ACK パケット数)	65.6	98.9	32.4	64.0	98.2	29.7	74.7	95.1	54.2
10	単一特徴量 (最小パケットサイズ)	77.7	55.4	100	30.9	61.8	0	59.9	19.9	100
	単一特徴量 (SYN パケット数)	79.1	61.4	96.9	42.9	62.9	22.8	36.3	0	72.5
	単一特徴量 (ACK パケット数)	53.2	93.9	12.5	49.7	99.3	0	52.7	99.3	6.0

11) Y.Linde, A.Buzo and R.M.Gray, “ An Algorithm for Vector Quantizer Design, ” IEEE Trans. Commun., Vol.COM-28, No.1, pp.88-95, 1980.

12) 川元研治，市田達也，市野将嗣，畑田充弘，小松尚久，“マルウェア感染検知のための経年変化を考慮した特徴量評価に関する一考察,” マルウェア対策研究人材育成ワークショップ 2011 (MWS2011)，October 2011(発表予定)

Evaluation of secular changes in statistical features of traffic for the purpose of malware detection

Kenji Kawamoto, Masatsugu Ichino, Mitsuhiro Hatada, Yusuke Otsuki, Hiroshi Yoshiura, and Jiro Katto

Abstract Applications and malware affecting them are dramatically changing. It isn't certain whether the currently used features can classify normal traffic or malware traffic correctly. In this paper, we evaluated the features used in previous studies while taking into account secular changes to classify normal traffic into the normal category and anomalous traffic into the anomalous category correctly. A secular change in this study is a difference in a feature between the date the training data were caputred and the date the test data were captured in the same circumstance. The evaluation is based on the Euclidean distance between the normal codebook or anomalous codebook made by vector quantization and the test data. We report on what causes these secular changes and which features with little or no secular change are effective for malware detection.

1 Introduction

The threat of malware is increasing. Malware is the word made from " malicious " and " software " and this sort of software compromises the security of or hijacks computers. A certain web site [1] claimed about 4,000 malware incidents occured in the first half of 2011 in Japan. The threat of stealth botnets and infections through web sites is especially increasing. In addition, new kinds of malware are appearing. Malware detection has thus becomes important for the safety of Internet usage.

Fujiwara [2] categorized research on detecting malware and found that it tended to focus on detecting known malware: methods of detecting unknown malware have not been discussed sufficiently. In this paper, we focus on detecting unknown malware by using traffic data because we suppose that normal traffic is quite different from anomalous traffic data. Moreover, we thought that malware might be easier to detect if we treated traffic as a time series signal. For example, there are numerous biometric recognition algorithms that work for lip movements, etc, and Ichino [3] showed that the accuracy of algorithms that use images streams is better than those that use static-image matching.

There are a lot of malware detection methods using packet payload information in previous research. For example, Karamcheti [4] used the inverse distributions of packet contents. However, it is impossible to detect malware in encrypted communication and to maintain privacy. Therefore, we focus on the packet header on the Internet in this research. After extracting the features of these headers, we classified the traffic into normal or anomalous.

Features used in malware detection have not been thoroughly evaluated. In this study, we tried to determine ones that would be effective for classifying normal or anomalous traffic by using CCCDATAset2009, 2010, 2011 [5] (we refer to these sets as CCC2009, CCC2010, CCC2011 later in this paper) as the anomalous traffic data and traffic data captured in an intranet as normal traffic data. We studied secular changes that occur over the course of three years worth of data. A secular change is difference in a feature between the date the training data were caputred and the date the test data were captured in the same circumstance. It is important to take into account secular changes because traffic data may dramatically change in a year. Features for which discrimination rates vary greatly from year to year aren't effective for malware detection. Therefore, secular changes are important factor for the evaluation of features.

This paper is organized as follows. In section 2, we describe the previous research and utilized features. Section 3 explains our experiment, and section 4 discusses accurate features for detecting malware. Section 5 is the conclusion.

2 Related Works

Here, we describe the features used in the previous research on malware detection and network intrusion detection.

Sato [6] discussed a network intrusion detection system that incorporated detection modules based on timeslot and flow count analysis. The timeslot method extracts features at fixed time intervals by refering to the frequency of TCP

Kenji Kawamoto, Jiro Katto
Graduate School of Fundamental Science and Engineering, Waseda University, Tokyo, Japan e-mail: kawamoto@kom.comm.waseda.ac.jp, katto@waseda.jp

Masatsugu Ichino, Yusuke Otsuki, Hiroshi Yoshiura
University of Electoro-Communications, Japan e-mail: ichino@inf.uec.ac.jp, otsuki@uec.ac.jp, yoshiura@hc.uec.ac.jp

Mitsuhiro Hatada
NTT Communications Corporation, Tokyo, Japan e-mail: m.hatada@ntt.com

header flags and the number of TCP, UDP, and ICMP packets. The flow count method, on the other hand, extracts features from every flow. A flow is a group of packets that have the same five-tuple of protocol type, source address, source port, destination address, and destination port. Fragmented packets and the inverse of the same port number frequency are used in flow count methods. In the field of malware detection, it is important to detect malware traffic quickly in order to prevent malware from spreading thourgh the network. However, detecting malware in real time by using flow count method is hard because feature extraction finishes when all the same flow packets are captured. Thus, we shall use the timeslot method in this study.

Hiramatsu [7] studied a clustering method for defining multiple normal states from network traffic data. The normalization numbers of ICMP, SYN, FIN, UDP and TCP except SYN, and FIN packets extracted every 60 minutes are used to define multiple normal states.

Kugisaki [8] focused on the host's transmission intervals as a feature and confirmed that there is a difference in transmission interval between traffic originating from human and botnet.

The above studies show that the number of packets, transmission interval, TCP flags, and port number is often used in the field of malware detection and network anomalous detection.

3 Evaluation Experiment

3.1 Evaluation eature

We use the existing research as a guide to extract features from the packet header and compiled statistics about the header information. Table 1 shows the 36 types of features evaluated in this paper.

Table 1 36 types of features

number	feature [unit]
1	number of packets
2	sum of packet sizes [byte]
3	mean packet size [byte]
4	minimum packet size [byte]
5	maximum packet size [byte]
6	standard deviation of packet size [byte]
7	mean transmission interval [seconds]
8	minimum transmission interval [seconds]
9	maximum transmission interval [seconds]
10	standard deviation of transmission interval [seconds]
11	number of SYN packets
12	number of FIN packets
13	number of PSH packets
14	number of ACK packets
15	number of RST packets
16	number of URG packets
17	number of SYN/ACK packets
18	number of FIN/ACK packets

number	feature [unit]
19	number of PSH/ACK packets
20	number of RST/ACK packets
21	ratio of SYN packets to TCP packets
22	ratio of FIN packets to TCP packets
23	ratio of PSH packets to TCP packets
24	ratio of ACK packets to TCP packets
25	ratio of RST packets to TCP packets
26	ratio of URG packets to TCP packets
27	ratio of SYN/ACK packets to TCP packets
28	ratio of FIN/ACK packets to TCP packets
29	ratio of PSH/ACK packets to TCP packets
30	ratio of RST/ACK packets to TCP packets
31	number of ICMP packets
32	number of UDP packets
33	number of 69/UDP port packets
34	number of 80/TCP port packets
35	number of 110/TCP port packets
36	number of 443/TCP port packets

3.2 Methods used in the experiment

1. Evaluation method
   The method to classify the test traffic into the normal or anomalous is a follows. First, we prepared a normal codebook and an anomalous codebook by separately using normal traffic data and malware traffic data as training data. The codebooks were made by vector quantization. Each codebook have one dimension to evaluate one individual feature. The timeslot interval for extracting features is 0.1, 1, 10, or 100 seconds, the vector quantization algorithm is LBG and splitting and vector quantization level is 2, 4, 8, 16, or 32. Vecotor quantization level means how many codebooks are made by the vector quantization. For the parameters(set of features, timeslot interval and vector quantization level), we discriminated on the basis of the Euclidean distance in the feature space between the labeled test data and the normal or anomalous codebook. If the distance between the test data and the normal codebook is shorter than the distance between the test data and the anomalous codebook, the test data is classified into normal traffic. If not, the test data is classified into malware traffic.

As evaluation indicators, we used the true negative rate (TNR), i.e, the rate at which normal traffic is correctly classified into normal category, and the true positive rate (TPR), i.e, the rate at which malware traffic is correctly classified into anomalous category. For each features and parameters, we calculated TNR and TPR by using traffic data from 2009, 2010 and 2011 in every timeslot.

2. Experimental data

We used CCC2009 for the anomalous codebook and normal traffic data captured on Mar 13, 14, 15, 2009 as the normal codebook. The test data for the malware traffic is CCC2009, CCC2010, and CCC2011 and the test data of the normal traffic is from 2009, 2010, and 2011. The CCCDATAset was captured in a honeypot and the normal traffic was captured in an intranet. The normal traffic and malware traffic data were captured on the same dates.

It would have been desirable to use normal and malware traffic data captured in the same circumstance for the experiment. However, resources on malware traffic are rather limited. In addition, normal traffic data captured in honeypot would not be realistic because nobody generates traffic in a honeypot. To handle this problem, the normal traffic data needs to be preprocessed to imitate the capture circumstances of malware traffic.

- Preprocessing for normal traffic

The normal traffic data was preprocessed to meet the following requirements.

a. Generated from one host.
   It is necessary to imitate the capture circumstances of malware traffic.
b. Generated by normal users.
   If the host is infected with malware, it will download or update new malware or try to connect to the Internet. However, such transmissions are normal in terms of their behavior. In this research area, it is important to be able to distinguish malware transmissions and behavior of human with no malicious intent. Hence, the normal traffic generated by a normal user must be used.

- Preprocessing for malware traffic

In this experiment, we used honeypot traffic data from CCC2009, CCC2010, and CCC2011, which includes scan traffic, exploit traffic, and infected traffic. This means it includes non-infected traffic data. However, it is essential for us to use only infected traffic data in the evaluation experiment. Hence, we did preprocessing to extract the malware traffic from the other attacking traffic data. The procedure for doing so is as follows.

a. Cut out control packets generated only in the honeypot circumstance.
b. Divide the pcap data in the OS reset interval of the honeypot.
c. Check whether traffic is truly infected by refering to the malware collection log provided in the CCCDATAset and look for the first packet of the malware transmission.
d. Extract the traffic data after the first packet of the malware transmission.

4 Experimental Results and Analyses

Here, we summarize the experimental results and analyze which features are effective at detecting malware thourgh secular changes in TNR and TPR, and we classify the features into two categories, one is the case that the secular change is big, the other in which the secular change is small. Then, we determine also which timeslots and vector quantization levels are effective. Finally, we summarize which features overall are the most effective.

First, we looked at the changes in the TNR and TPR over the course of three years. Table 2 shows the discrimination rates of TNR and TPR in 2009, 2010, and 2011. The average TNR or TPR is the mean of the corresponding values calculated for each feature types, timeslot length, and number of vector quantizattion levels.

Table 2 Discrimination rates of TNR and TPR

year	2009	2010	2011
average(TNR)	36.1%	35.2%	40.7%
average(TPR)	57.0%	54.1%	51.2%

The average TNR in 2011 is the highest, while the average TPR in 2011 is the lowest. From this result, it is clear that the secular change in the test data affects the discrimination rate.

4.1 Secular change

1. TNR

Figure 1 shows features for which the average TNR is higher than 50% during the three years.

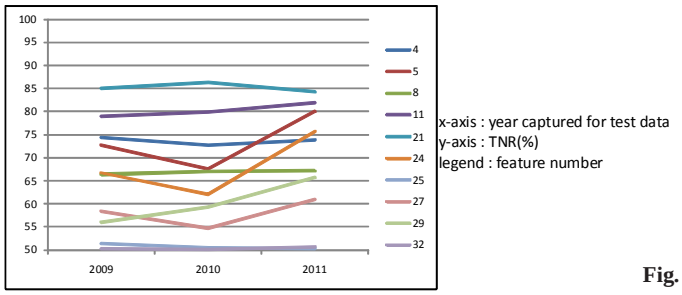

Fig. 1 Change in features for which the average TNR is higher than 50% during three years

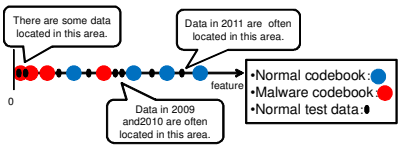

Fig. 2 Why TNR is the highest in 2011

- Features with large secular change

  Features 2, 3, 9, 14, 17,18, 19, 20, 24, 34, and 36(these numbers match those in Table 1) show large secular changes in TNR. Except for feature 9, the average TNR are the highest in 2011. Figure 2 shows why the average TNR of these features is the highest in 2011. In terms of the above features, almost all of the normal test data in 2011 can be classified as normal because the feature values are too high and very close to the normal codebooks. However, the normal test data in 2009 and 2010 are often classified as malware tarffic. This is why the average TNR is the highest in 2011. This situation arises from the differnce in the number of packets in the normal test data. Table 3 shows how many packets there are in each year. The unit of the average is packets per slot.

Table 3 Number of packets of normal test data for each year

timeslot 0.1 seconds			
year	2009	2010	2011
average	14.3	10.8	5.51
standard deviation	33.8	23.1	9.5
timeslot 1 seconds			
year	2009	2010	2011
average	189.9	33.2	30.9
standard deviation	580.4	61.3	43.3
timeslot 10 seconds			
year	2009	2010	2011
average	340.8	157.0	1039.7
standard deviation	1561.7	451.1	1176.7
timeslot 100 seconds			
year	2009	2010	2011
average	1656.4	6060.0	9225.8
standard deviation	4520.9	1362.3	10603.3

Table 4 TNRs over 90% for three years for minimum packet size

timeslot	vector quantization level	2009	2010	2011
0.1 seconds	4	98.4%	92.6%	90.4%
0.1 seconds	8	98.2%	92.6%	93.3%
1 seconds	4	99.8%	98.7%	100%
1 seconds	8	100%	98.7%	100%
1 seconds	16	98.0%	99.1%	100%
10 seconds	2	100%	100%	100%
10 seconds	4	100%	100%	100%
10 seconds	8	100%	100%	100%
10 seconds	16	100%	100%	100%
100 seconds	2	99.3%	100%	100%
100 seconds	4	100%	100%	100%
100 seconds	8	100%	100%	100%
100 seconds	16	100%	100%	100%

Table 3 shows that the number of test data packets in 2011 is the largest. The contents of traffic is similar for each year. Hence, the number of test data packets significantly affects the secular change.

- Features with small secular changes

  Features 4, 7, 8, 11, 21, 25, 28, 30, 31, 32, and 35(these number match those of Table 1) show little secular change in TNR. These features are typically ratios(for example, ratio of SYN packets to TCP packets). Therefore, it is effective to use such features for suppressing drops in discrimination rates caused by secular changes.

  Among these features, features 4 (minimum packet size), 11 (number of SYN packets), and 21 (ratio of SYN packets to TCP packets) have average TNRs higher than 75%.

  – Minimum packet size

    Table 4 shows TNRs over 90% for three years for the minimum packet size.

    Table 5 shows the average and standard deviation of the minimum packet size in the normal and anomalous test traffic data. The unit of the average is byte per slot.

    Table 5 shows that the minimum packet size of normal traffic is almost always 60 bytes if the timeslot interval is larger than 1 seconds. On the other hand, the minimum packet size of anomalous traffic varies. There is an enormous difference between the standard deviation of the minimum packet size of normal traffic and that of anomalous traffic. In normal traffic, the standard deviation is almost always 0, in contrast, it is much larger than zero for anomalous traffic. We suppose that this difference would be effective for malware detection. That is, we think that both of the minimum packet size and its standard deviation are useful and efficient features for detecting malware.

Table 5 Average and standard deviation of minimum packet size in normal and anomalous test traffic

timeslot 0.1 seconds						
	normal			anomalous		
year	2009	2010	2011	2009	2010	2011
average	79.5	81.0	93.7	68.6	84.7	114.4
standard deviation	95.6	113.1	134.7	32.4	89.1	174.6
timeslot 1 seconds						
	normal			anomalous		
year	2009	2010	2011	2009	2010	2011
average	60.1	60.8	60.0	70.7	73.3	101.1
standard deviation	1.3	7	0	31.1	34.8	83.1
timeslot 10 seconds						
	normal			anomalous		
year	2009	2010	2011	2009	2010	2011
average	60.9	60.1	60	67.8	71.2	102.2
standard deviation	0.2	3	0	28.2	40.4	113.9
timeslot 100 seconds						
	normal			anomalous		
year	2009	2010	2011	2009	2010	2011
average	60.4	60	60	69.3	66.8	84.2
standard deviation	2.8	0	0	44.9	40.4	60.4

Table 6 Number of packets of anomalous test data in each year

timeslot 0.1 seconds			
year	2009	2010	2011
average	42.3	11.6	3.3
standard deviation	78.6	11.7	4.3
timeslot 1 seconds			
year	2009	2010	2011
average	139.6	166.6	6.6
standard deviation	121.2	140.1	13.6
timeslot 10 seconds			
year	2009	2010	2011
average	781.6	1439.3	23.4
standard deviation	584.6	1161.6	54.2
timeslot 100 seconds			
year	2009	2010	2011
average	3350.4	7578.9	348.7
standard deviation	5212.5	9580.1	1711.7

– Number of SYN packets, ratio of SYN packets to TCP packets
TNR is very high when the number of SYN packets or ratio of SYN packets to TCP packets is used. This is because anomalous traffic data tends to behave like a SYN scan. Because of this, the values in the anomalous codebook are much larger than those of the normal codebook. Moreover, normal test traffic data doesn't have a lot of SYN packets. Therefore, almost all of the normal traffic data are classified in the normal category. That is why the TNR is very high. However, malware traffic doesn't always have SYN scans. Although it is difficult to use these features for classifying whether traffic is normal or malware, it would be effective for predicting or detecting attack.

2. TPR
Figure 3 shows for which the TPR is higher than 70% over the course of three years.

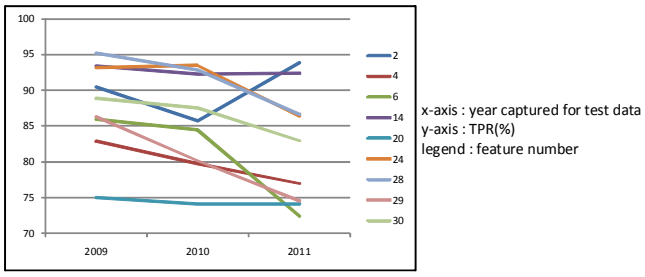

Fig. 3 Changes in TPR higher than 70% over the course of three years

- Features with large secular change
Features 1, 3, 5, 6, 7, 9, 10, 11, 21, and 29 have large secular changes in TPR. Except feature 1 and 9, the average TPR is lower in 2011 than in 2009 and 2010. We consider there are two reasons why TPR is the lowest in 2011. The first reason is that the anomalous traffic data in 2011 has fewer SYN scans than the anomalous traffic data in 2009 and 2010. If traffic data doesn't have a lot of SYN scans, the average packet size is large. The behavior is close to that of normal traffic data. That's why the average TPR is the lowest in 2011 for features 3, 11, and 21. The second reason is that the number of packets in the anomalous test data in 2011 is fewer than in 2009 or 2010. Table 6 shows the number of packets in the anomalous test data. The unit of the average is packets per slot.
It is clear that the number of packets in the anomalous test data is the fewer in 2011 than in 2009 or 2010. There isn't a large year-to-year difference in the anomalous test data as regards the number of PSH/ACK packets. However, the ratio of PSH/ACK packets to TCP packets is the highest in the anomalous test data in 2011 and close to the ratio of the normal test data. That's why the average TPR for feature 29 is the lowest in 2011.
- Features with small secular change

Features 4, 14, 17, 18, 19, 20, 25, 30, 31, 32, 34, and 36 have small changes in TPR. Among these features, those that have average TPRs higher than 80% are 4 (minimum packet size), 14 (number of ACK packets), 21 (ratio of RST/ACK packets to TCP packets).

- Minimum packet size
  Table 7 shows TPRs over 90% over the course of three years for the minimum packet size.

Table 7 TPRs over 90% for three years for the minimum packet size

timeslot	vector quantization level	2009	2010	2011
0.1 seconds	32	99.8%	94.6%	90.2%
1 seconds	32	100%	99.6%	95.8%
10 seconds	32	94.0%	92.0%	92.4%

Table 8 Average number of ACK packets in normal and anomalous test traffic

timeslot 0.1 seconds						
	normal			anomalous		
year	2009	2010	2011	2009	2010	2011
average	10.9	6.9	3.3	2.6	0.6	2.3
timeslot 1 seconds						
	normal			anomalous		
year	2009	2010	2011	2009	2010	2011
average	174.1	20.0	19.0	3.1	1.3	3.4
timeslot 10 seconds						
	normal			anomalous		
year	2009	2010	2011	2009	2010	2011
average	289.1	103.2	787.2	10.0	10.9	11.8
timeslot 100 seconds						
	normal			anomalous		
year	2009	2010	2011	2009	2010	2011
average	1305.4	432.8	6669.8	40.7	73.5	37.5

In terms of TPR, the minimum packet size and its standard deviation are effective for malware detection just as they are for the TNR.

- Number of ACK packets
  Table 8 shows the average number of ACK packets in normal and anomalous traffic for three years.
  From Tables 3 and 6, we can see that there is no large difference in the number of packets between normal and anomalous test traffic. Moreover, the number of ACK packets in the anomalous test traffic is very few in comparison with that in the normal test traffic. Therefore, the number of ACK packets is an effective feature to classify traffic data into normal or anomalous.

- Ratio of RST/ACK packets to TCP packets
  In terms of the ratio of RST/ACK packets to TCP packets, TPR itself is high and the secular change is small. However, both sorts of test traffic have a lot of timeslot intervals in which the ratio of RST/ACK packets to TCP packets is 0. Moreover, the anomalous codebook is closer to 0 than the normal codebook is. Therefore, the ratio of RST/ACK packets to TCP packets can't detect malware correctly. This type of feature is not useful for malware detection.

4.2 Timeslot length

Table 9 shows the average TNR and TPR in each timeslot throughout three years.

Table 9 Average TNR and TPR in each timeslot

timeslot	0.1 seconds	1 seconds	10 seconds	100 seconds
Average TNR	31.0%	33.6%	36.2%	48.5%
Average TPR	48.2%	56.3%	57.2%	54.5%

Table 10 Average TNR and TPR at each vector quantization level

VQ level	2	4	8	16	32
Average TNR	45.4%	41.6%	40.4%	38.2%	38.0%
Average TPR	52.3%	52.3%	51.4%	51.5%	48.6%

It is obvious that 0.1 seconds is too short a period for detecting malware traffic. Moreover, considering actual circumstances and the need for real time detection of malware, 100 seconds interval would be long. We, hence, suppose that it would be better to use 1 or 10 seconds for extracting features.

4.3 Vector quantization level

Table 10 shows the average TNR and TPR at each vector quantization level throughout three years.

In the case of using one feature, level 32 is too high for detecting malware, and level 2 or level 4 is effective in this experiment. We will study a malware detection method combining two or three features in the near future. In such a situation, we think the level 8 or 16 level may be best.

4.4 Effective features for malware detection

The effective features for malware detection are ones with small secular changes and simultaneously high TNR and TPR. Features with either high TNR or high TPR may also be effective. The above analysis shows that the most effective features for malware detection are the minimum packet size(and/or its standard deviation), the number of SYN packets, the ratio of SYN packets to TCP packets, and the number of ACK packets. In addition, 1 or 10 seconds is a good time interval for extracting these features, and level 2 or 4 is an effective for vector quantization level in the case of using one feature.

5 Conclusion

In this paper, we looked at how well the features used in the previous research can classify normal and malware traffic and discussed which of them are actually effective at malware detection. Our analysis showed that secular changes significantly affect the discrimination rate. We guessed that there are two main reasons for secular changes. First, if there are large differences between each test data, the discrimination rate dramatically changes. Second, if some test data have a particular behavior, for example, SYN scan, the features in test data dramatically change.

Considering such secular changes, we concluded that four features are especially effective for malware detection, the minimum packet size(or its standard deviation), the number of SYN packets, the ratio of SYN packets to TCP packets, and the number of ACK packets. The best time interval for extracting features is 1 or 10 seconds and 2 or 4 may be the best level of vector quantization in case of using one feature.

In our research, we have three subjects of future work. First, we should discuss how to combine features so as to improve the discrimination rate.

Second, we should discuss what types of traffic data we should use for training data in order to enhance the discrimination rate. We have found that the number of packets and certain behaviors especially affect it. Therefore, we should look at training data that would emphasise these points.

Third, we should look into the capture circumstances of normal traffic. In this experiment, the normal traffic data was captured in an intranet while the anomalous traffic was captured in a honeypot circumstance. Although it is valid to use normal traffic after it has been preprocessed in the above circumstance, the malware circumstance is much different from the normal traffic circumstance. Therefore, it is important to research normal traffic circumstances in order to perfome a more reliable experiment.

References

1. Internet threat monthly report May 2011. http://ip.trendmicro.com/jp/threat/security_news/monthlyreport/article/20110602082147.html

2. Masashi Fujiwara, Masatoshi Terada, Tetsuya Abe, and Hiroaki Kikuchi, Study for the classification of malware by infection activities, IPSJ CSEC No.21 p177-182, March 2008, in japanese.

3. Masatsugu Ichino, Hitoshi Sakano, and Naohisa Komatsu, Speaker Recognition Using Kernel Mutual Sbuspace Method, The transactions of the Institute of Electronics, Information and Communication Engineers D- vol.J88 no.8 pp.1331-1338, 2005, in japanese.

4. V.Karamcheti, D.Geiger, Z.Kedem, and S.M.Muthukrishan, Detecting malicious network traffic usinginverse distributions of packet contents, the ACM SIGCOMM Workshop on Mining Network Data, pp.165-170, 2005.

5. Mitsuhiro Hatada, Isami Nakatsuru, and Mitsuaki Akiyama, Datasets for Anti-Malware Resarrch ～ MWS 2011 Datases ～, MWS2011, October 2011, in japanese.

6. Yohei Sato, Yuji Waizumi, and Yoshiaki Nemoto, Improving Accuracy of Network-based anomalous Detection Using Multiple Detection Modules, Technical Commitee on Network Systems, 2004, in japanese.

7. Naotoshi Hiramatsu, Yuji Waizumi, Hiroshi Tsunoda, and Yoshiaki Nemoto, Using Multiple Normal States for Network Anomaly Detection, IEICE, 2006, in japanese.

8. Yuji Kugisaki, Yoshiaki kasahara, Yoshiaki Hori, and Koichi Sakurai, Study for botnet detection based on behavior observation of data transmission interval, SCIS, January 2009, in japanese.

マルウェア感染検知のためのトラヒックデータにおけるペイロード情報の特徴量評価

大月 優輔† 市野 将嗣† 川元 研治†† 畑田 充弘††† 吉浦 裕†

†電気通信大学大学院情報理工学研究科
182-8585 東京都調布市調布ヶ丘 1-5-1
otsuki@uec.ac.jp, ichono@inf.uec.ac.jp,
yoshiura@hc.uec.ac.jp

††早稲田大学理工学術院基幹理工学研究科
169-8555 東京都新宿区大久保 3-4-1
kawamoto@kom.comm.waseda.ac.jp

†††NTT コミュニケーションズ株式会社
108-8118 東京都港区芝浦 3-4-1 グランパークタワー16F
m.hatada@ntt.com

あらまし 近年，トラヒックデータを用いたマルウェア感染検知において，複数の特徴量を用い、その組み合わせから正常時通信と感染時通信の識別をしている．しかし，個々の特徴量毎の有効性や，マルウェアの種類毎の有効性について明確に示されていない．そこで本研究では，ペイロードにおける感染検知において，感染時通信として CCCDATAset，D3M2012 を， 正常時通信として 2 種類のイントラネットのトラヒックデータを用い，マルウェアの種類としてワーム，トロイの木馬，ファイル感染型ウイルスを取り上げ，261 種類の特徴量に対して，4 つの量子化レベル数において各々TPR，TNR を求め，個々の有効な特徴量をマルウェアの種類毎に明らかにした．

Evaluating features of payload for malware detection

Yusuke Otsuki† Masatsugu Ichino† Kenji Kawamoto†† Mitsuhiro Hatada††† Hiroshi Yoshiura†

†Graduate School of Informatics and Engineering, The University of Electro-Communications
1-5-1 Chofugaoka, Chofu-si, Tokyo, 182-8585, JAPAN
otsuki@uec.ac.jp, ichono@inf.uec.ac.jp, yoshiura@hc.uec.ac.jp
††Graduate school of Fundamental Science and Engineering, Waseda University
3-4-1 Okubo, Shinjuku-ku, Tokyo, 169-8555, JAPAN
kawamoto@kom.comm.waseda.ac.jp
†††NTT Communications Corporation
Gran Park Tower 16F, 3-4-1 Shibahara, Minato-ku, Tokyo, 108-8118 Japan
m.hatada@ntt.com

Abstract We evaluated features used in related works based on traffic data since effectiveness of using these features in malware detection is not evaluated sufficiently. In the evaluation, CCCDATAset and D3M2012 are used as anomaly traffic data infected with malware, and traffic data captured in some Intranet are used as normal traffic data. We evaluated the features by comparing the distances between normal/anomaly codebooks made by vector quantization and input data. In this paper, we show and discuss the evaluation results of payload features in each type of malware.

1. はじめに

近年，仕事や生活等さまざまな場面でインターネットが必要不可欠な存在となっている．インターネットが普及し，便利さが増す反面，それらを悪用したマルウェアの活動による被害が拡大している．マルウェアとは悪意のあるソフトウェア（Malicious Software）の略称であり，感染したPC のデータ破壊や個人情報の流出等，我々の日常生活を脅かす存在となっている．

2011 年前半に新しく出現したマルウェアの数は，120 万件で，2010 年の後半よりも 15.7%分増加している[1]．その上，近年のマルウェアは亜種が数多く存在し，複数のダウンロードサーバに分散して感染する等，複雑化・高度化が進んでいる．そのため，早急に対策を講じる必要がある．

これに対し，シグネチャ型，イベント監視型などの手法を用いてマルウェアの検知，駆除を行うマルウェア対策ソフトがセキュリティベンダによって開発されている[2]．しかしこれらの検知手法は，マルウェア毎に特徴を示すシグネチャを用意しなければならず，短期間で大量に出現する未知のマルウェアの検知に対応できない．そのため，未知のマルウェアに有効な感染検知，つまり感染してしまうことを前提として，感染後に早期に検知できるようにすることが必要とされている．

そこで近年，マルウェア感染検知の中でもトラヒックデータ検知に着目した手法が注目されている．なかでも，ヘッダ情報よりも情報量が多く，感染時通信と正常時通信を区別する情報が多く含まれていると考えられる，ペイロード情報を用いた検知手法が注目されている．

本研究ではまず，ペイロード情報に基づく感染検知を行う既存研究を分析した．その結果，検知手法の検討がメインで，どの特徴量が正常と感染を区別しやすいかという観点からの検討が十分に行われていないことがわかった．特徴量の有効性を明確にした上で複数の特徴量を適切に組み合わせることで，より高性能な識別ができる可能性がある．

そこで本研究では，マルウェア感染後のトラヒックデータを分析して，感染検知に有効な特徴量の有効性を明らかにした. その目的の方法として，感染時通信に，D3M2012，CCCDATAset2009，2010，2011[3] （以下 CCC2009，CCC2010，CCC2011）の攻撃通信データを，正常時通信に，2 カ所のあるイントラネットに流れる通信データを用いた． また，マルウェアは，種類毎で異なった通信（ワーム：インターネット接続確認等）を行うので，マルウェアの種類毎（ワーム，トロイの木馬，ファイル感染型ウイルス）に分類し，各々の特徴量の有効性を明らかにした．

以下，2.で既存研究においてよく用いられている特徴量を整理し，3.で 2.を踏まえた特徴量評価の目的を述べ，4.で実験方法と本稿において評価する特徴量について述べる．5.で特徴量毎に感染時通信と正常時通信の識別のしやすさを評価する実験を行った結果を示し，6.でマルウェアの種類毎で識別率の高い特徴量について考察する. また，時間的変化を用いた識別の有効性について述べる．

2. 特徴量評価

2.1. 既存研究の特徴量

本章では，既存のマルウェア感染検知やネットワーク異常検知に関する研究で用いられているペイロードの特徴量を整理する．

文献[4]，[5]，[6]等では，特徴量として文字列の出現頻度が使われている．桑原ら[6]は，ボットの攻撃通信データのペイロード情報から，マルウェアの挙動とそれに対応した特徴的な文字列（exe，NICK 等）があることを確認し，それらを特徴量として用いている

また，Wei Lu[7] らは，ボットが行う遠隔制御用通信のトラヒックデータに着目する手法を提案し，正常時通信とボットが行っている感染（異常）時通信パケットのペイロード情報に着目しペイロード内の ASCII 文字コードの出現頻度（バイト数）を特徴量としている．

また，山田ら[8]は，侵入検知システムにおける未知攻撃の課題に対する解決策として，決定木を用いたアノマリ検知を示し，HTTP リクエスト長，HTTP リクエストの総サイズを特徴量としている．

上記より，マルウェア検知用の特徴量として ASCII 文字コードの出現頻度，文字列の出現頻度，HTTP リクエスト長がよく用いられていることがわかる．しかし，いずれの研究においても，特徴量の有効性については評価されていない．

2.2. マルウェアの種類毎の特徴量評価

トレンドマイクロ社のセキュリティデータベース[9]等に示されているように，マルウェアの種類毎で特有の挙動がある．例として，ワームはソフトウェアに存在する脆弱性を利用し，ネットワーク上で感染活動を行う．トロイの木馬は特定のサイトにアクセスし，不正なファイルのダウンロードを要求し，感染した PC がさらなる脅威にさらされるなどの挙動がある．同種類のマルウェアの亜種で共通した挙動の傾向があることが確認できている[9]．このため，マルウェアの種類毎に有効な特徴量が異なる可能性がある．

そこで本研究では，マルウェアの種類毎での特徴量評価を行った．種類毎で有効な特徴量を評価し，有効な特徴量を組み合わせることで，効率的かつ早期の検知に繋がると考える．

3. 特徴量評価の目的

既存研究では，トラヒックデータを用いたマルウェア感染検知において，複数の特徴量を用い，その組み合わせから正常時通信と感染時通信の識別をしている．しかし，個々の特徴量の有効性について明らかではなく，マルウェアの種類毎にどのような特徴量が有効であるかも明らかではない．また，有効な特徴量を感染検知に用いない場合，識別率の低下が考えられる．

そこで本研究では，ペイロードを用いたマルウェア感染検知において，個々の特徴量の有効性をマルウェアの種類毎に明らかにする．マルウェアの種類として，ワーム，トロイの木馬，ファイル感染型ウイルスを定め，有効な特徴量をそれぞれ評価した．

4. 特徴量評価実験概要

4.1. タイムスロット

本研究では，トラヒックデータからの特徴抽出にタイムスロットを用いた．タイムスロットとは，トラヒックデータを特定の秒数で区切った範囲のことを示す．

時間的変化を伴う通信をタイムスロットで分割し，タイムスロットの時間的変化を追うことで，通信全体の時間的変化に着目した識別を行うことができる．また，トラヒックデータの特徴抽出の取得単位としてフローを用いる方法がある．しかしこの手法は，通信が終了するまで特徴量が取得できないため，早期検知に不適切である．一方タイムスロットを用いる方法は，タイムスロット幅を調整することで，早期に検知できる可能性がある．本実験では，タイムスロット幅は 1 秒とし，タイムスロット毎に特徴量を求めた．

4.2. ペイロードの特徴量

本研究では既存研究で多く用いられていた次に示す 261 種類の特徴量を評価対象とする．

- ASCII 文字コードの出現頻度 255 種類
- 特徴的な文字列の出現頻度：5 種類
  (GET，POST ，exe，whatismyip，checkip)
- HTTP リクエスト長

4.3. 評価方法

本研究での感染時通信と正常時通信の識別方法について説明する．

4.3.1. ベクトル量子化によるコードブック作成

ベクトル量子化を用いて，感染時通信のみを用いて学習を行った感染コードブックと，正常時通信のみを用いて学習を行った正常コードブックを予め作成する．今回は，各特徴量を個別に評価することが目的であるため，特徴量毎に 1 次元コードブックを作成した．ベクトル量子化のアルゴリズムには，LBG+Splitting アルゴリズム[10]を用い，レベル数は 2，4，8，16 の 4 種類とした．そして，予め感染時通信か正常時通信かのラベル付けされた各特徴量の 1 次元テストデータ（1 タイムスロット毎から抽出される特徴量）を与え，テストデータと感染，正常コードブックとの距離を計算し，感染(正常)コードブックとの距離の方が小さければ感染（正常）と識別している．

4.3.2. 実験データ

本研究では, 取得環境の違いを評価するために, 正常時通信として，異なる 2 カ所のイントラネットに流れる通信を用いた．これにより，取得環境の違いによる影響を受けにくい特徴量を評価できる．それぞれのデータを，正常コードブック作成のための学習データとテストデータ用に分割して

いる．

また，感染時通信に D3M2012，CCC2009, CCC2010, CCC2011 を用い，感染コードブック作成用の学習データとテストデータ用に分割した．さらにこれらのデータをマルウェアの種類毎でマルウェアの検体数が均等になるように分割した．CCC2009，CCC2010, CCC2011 の攻撃通信データにはマルウェアに感染するまでの通信が含まれている．そこで本研究では，文献[11]と同じ方法を用いて，攻撃通信からマルウェアに感染した後の通信のみを切り出し評価した．

本研究ではマルウェアの種類をワーム，トロイの木馬，ファイル感染型ウイルスとした．ファイル感染型ウイルスとは，拡張子 exe 等の実行型ファイルに感染するウイルスである．また，マルウェア検体名は CCCDATAset において，攻撃元通信データのログファイルに記載されている名称を用いた．D3M2012 においては，マルウェア検体のハッシュ値を用い，G-data 社，トレンドマイクロ社，Kaspersky 社のそれぞれが命名しているマルウェア検体の名称を用いた．表 1 に今回実験で使用した各データセットのマルウェアをまとめた．

表 1:各データセットのマルウェア

種類	CCCDATAset2011	CCCDATAset2010	CCCDATAset2009	D3M2012
ワーム	WORM_DOWNAD.AD	WORM_DOWNAD.AD WORM_MAINBOT.AH WORM_MAINBOT.FY WORM_PALEVO.SMD WORM_PALEVO.BL	WORM_SWTYMLAI.CD	
トロイの木馬			TROJ_BUZUS.AGB	TROJ_GEN.R47C7C2 Trojan.Generic.KD.578000 Trojan.Generic.KD.410743 Trojan-Dropper.Win32.Dapato.aoxn
ファイル感染型		PE_VIRUT.AV PE_VIRUT.XV	PE_BOBAX.AK	

5. 実験結果

マルウェア感染検知の要件は，感染と正常を正しく識別できる特徴量を用いることである．このため，感染・正常のみを正しく識別できる特徴量を併用することも有効である．感染検知に有効な特徴量について検討するため，TPR（感染データを感染と正しく分類した割合），TNR（正常データを正常と正しく分類した割合）が共に高い特徴量，TPR のみが高い特徴量，TNR のみが高い特徴量の観点から検討した．以下に評価実験の結果を示す．

5.1. TPR, TNR 共に高い特徴量

取得環境の影響を受けにくい特徴量を評価する観点から，2 種類の正常時通信を用い，安定的に検知するという観点から，4 つの量子化レベル数の TPR，TNR の平均値を特徴量 261 個に対してそれぞれ求めた．その結果の中から，イントラネット A における平均 TPR・TNR の値が高い上位 15 個と，それに対応するイントラネット B の TPR・TNR を表 2 に示す．表 2（赤字）より，ワームの ASCII 文字コード「i」とファイル感染型ウイルスの「HTTP リクエスト長」の TPR・TNR の値が，2 種類の正常時通信を用いた時の差異が少なく，安定して高い値を示すことがわかった．これらの特徴量を用いたときの詳細を表 3，表 4 に示す．これらの特徴量は，2 種類の正常時通信を用い，量子化レベル数と変化させても，TPR が 95%以上かつ TNR が 80%以上で安定的に検知できることがわかった．

表 2:イントラネット毎の平均 TPR・TNR

マルウェアの種類	特徴量	イントラネットAの平均TPR	イントラネットBの平均TPR	特徴量	イントラネットAの平均TNR	イントラネットBの平均TNR
ワーム	HTTPリクエスト長	100%	99%	ASCII文字コード「>」	89%	79%
	ASCII文字コード「i」	98%	98%	ASCII文字コード「I」	88%	78%
	ASCII文字コード「f」	98%	97%	ASCII文字コード「ETB」	88%	70%
	ASCII文字コード「C」	98%	92%	ASCII文字コード「J」	85%	83%
	ASCII文字コード「E」	97%	94%	ASCII文字コード「H」	84%	74%
	ASCII文字コード「e」	97%	91%	ASCII文字コード「#」	84%	78%
	ASCII文字コード「a」	96%	91%	ASCII文字コード「.」	84%	76%
	ASCII文字コード「0」	95%	97%	ASCII文字コード「B」	84%	74%
	ASCII文字コード「r」	95%	91%	ASCII文字コード「E」	84%	75%
	ASCII文字コード「OR」	95%	91%	ASCII文字コード「P」	84%	75%
	ASCII文字コード「/」	95%	92%	ASCII文字コード「R」	84%	74%
	ASCII文字コード「:」	95%	94%	ASCII文字コード「S」	84%	73%
	ASCII文字コード「c」	95%	93%	ASCII文字コード「i」	83%	83%
	ASCII文字コード「s」	95%	94%	ASCII文字コード「M」	83%	75%
	ASCII文字コード「m」	95%	91%	ASCII文字コード「N」	83%	77%
トロイの木馬	HTTPリクエスト長	100%	100%	ASCII文字コード「<」	85%	72%
	ASCII文字コード「NL*」	100%	100%	ASCII文字コード「US」	85%	76%
	ASCII文字コード「OR」	100%	100%	ASCII文字コード「>」	85%	41%
	ASCII文字コード「0」	100%	100%	ASCII文字コード「¥」	83%	73%
	ASCII文字コード「5」	100%	100%	ASCII文字コード「VT」	82%	56%
	ASCII文字コード「C」	100%	100%	ASCII文字コード「]」	82%	68%
	ASCII文字コード「d」	100%	100%	ASCII文字コード「HT」	81%	65%
	ASCII文字コード「e」	100%	100%	ASCII文字コード「SOH」	80%	56%
	ASCII文字コード「t」	100%	100%	ASCII文字コード「^」	80%	69%
	ASCII文字コード「x」	100%	100%	ASCII文字コード「'」	80%	67%
	ASCII文字コード「A」	100%	100%	ASCII文字コード「FS」	80%	63%
	ASCII文字コード「o」	100%	100%	ASCII文字コード「ESC」	79%	62%
	ASCII文字コード「r」	100%	100%	ASCII文字コード「ACK」	79%	63%
	ASCII文字コード「1」	100%	100%	ASCII文字コード「$」	79%	65%
	ASCII文字コード「2」	100%	100%	ASCII文字コード「EOT」	79%	60%
ファイル感染型ウイルス	HTTPリクエスト長	100%	100%	ASCII文字コード「DC2」	92%	68%
	ASCII文字コード「p」	99%	99%	ASCII文字コード「DC3」	90%	68%
	ASCII文字コード「B」	99%	75%	ASCII文字コード「ETB」	89%	62%
	ASCII文字コード「S」	98%	98%	ASCII文字コード「#」	89%	67%
	ASCII文字コード「e」	98%	98%	ASCII文字コード「S」	89%	69%
	ASCII文字コード「T」	98%	94%	ASCII文字コード「Y」	88%	76%
	ASCII文字コード「i」	98%	98%	ASCII文字コード「'」	87%	79%
	ASCII文字コード「o」	97%	98%	ASCII文字コード「M」	87%	79%
	ASCII文字コード「H」	96%	94%	ASCII文字コード「R」	86%	79%
	ASCII文字コード「X」	95%	74%	ASCII文字コード「US」	86%	69%
	ASCII文字コード「W」	95%	93%	ASCII文字コード「)」	85%	75%
	ASCII文字コード「r」	95%	94%	HTTPリクエスト長	82%	84%
	ASCII文字コード「G」	95%	78%	ASCII文字コード「(」	82%	78%
	ASCII文字コード「N」	95%	80%	ASCII文字コード「f」	82%	73%
	ASCII文字コード「q」	95%	94%	ASCII文字コード「j」	82%	75%

表 3:ワームの ASCII 文字コード「i」の TPR・TNR

	TPR				TNR			
量子化レベル数	2	4	8	16	2	4	8	16
イントラネットA	98%	99%	98%	98%	83%	80%	86%	83%
イントラネットB	97%	98%	98%	97%	83%	80%	80%	82%

表 4:ファイル感染型ウイルスの「HTTP リクエスト長」の TPR・TNR

	TPR				TNR			
量子化レベル数	2	4	8	16	2	4	8	16
イントラネットA	100%	100%	100%	100%	82%	80%	81%	81%
イントラネットB	100%	100%	100%	94%	88%	86%	88%	86%

5.2. TPR のみ高い特徴量

表 2（青字）から TPR のみ高い特徴量として，ワームでは HTTP リクエスト長, ASCII 文字コード「0」，「f」，トロイの木馬では，HTTP リクエスト長, ASCII 文字コード「NL*」，「CR」，「0」，「5」，「A」，「C」，「M」，「d」，「e」，「r」，「t」，「x」，ファイル感染型ウイルスでは ASCII 文字コード「S」，「e」，「i」，「o」，「p」が確認できた．これらは 2 種類の正常時通信を用いても，TPR が 95%以上で安定的に検知できることがわかった．

5.3. TNR のみ高い特徴量

表 2（緑字）から TNR のみ高い特徴量として，ワームでは ASCII 文字コード「J」が確認できた．これらは 2 種類の正常時通信を用いても, TNR が 80%以上で安定的に検知できることがわかった．

6. 考察

5 章で有効だと判断した特徴量について，正常時通信と感染時通信の重なり具合を視覚的に確認するため，出現頻度に着目して考察した．また，マルウェア種類毎の共通した挙動を挙げ，それらの挙動について説明し，マルウェアの種類毎に有効な特徴量の考察した．

6.1. TPR・TNR が共に高い特徴量について

5 章で求められた特徴量に対してヒストグラムを作成した．これらは縦軸を感染時通信（正常時通信）全体のスロット数の割合[%]，横軸を出現頻度としている．5 章で有効だと判断した特徴量の中から例として，ファイル感染型ウイルスの感染時通信とイントラネット A の正常時通信を用いた HTTP リクエスト長のヒストグラムを図 1，図 2 に示す．これらは量子化レベル数を 2 としたときの結果であり，図 1 は HTTP リクエスト長が 200 まで，図 2 は HTTP リクエスト長が 10,000 までを示している．また，図 1 から感染時データの出現頻度が 100 以下であることが確認できる．それに対し，正常時データの多くが 100 以上であることが確認できる．

正常時通信（ユーザの通信）は様々であり，HTTP リクエスト長にばらつきが存在する．感染時データは，HTTP リクエスト長が短いものが多く，感染時コードブックに分類される．これにより，TPR の値が高くなったが，正常時通信の中にも HTTP リクエスト長が短くなるものもあるため，TNR の値は低くなったと考えられる．

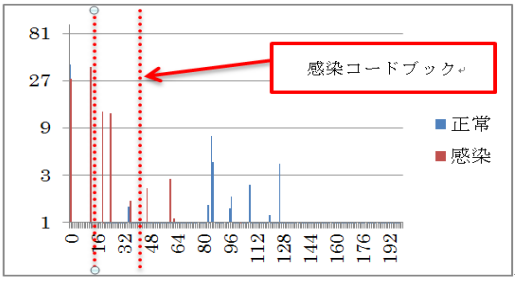

図 1：HTTP リクエスト長のスロット数の割合が 1%以上のヒストグラム
（縦軸：スロット数の割合[%]，横軸：リクエスト長）

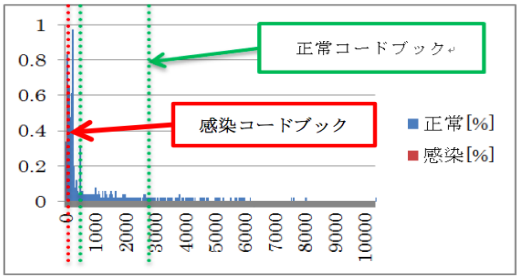

図 2：HTTP リクエスト長のスロット数の割合が 1%以下のヒストグラム
（縦軸：スロット数の割合[%]，横軸：リクエスト長）

ワームの ASCII 文字コード「i」のヒストグラムは図 3，図 4 のようになった．これらの図は縦軸を感染時通信（正常時通信）全体のスロットの割合[%]，横軸を出現頻度[回]としている．また，正常時通信は，上記と同様にイントラネット A の正常時通信を用いている．

ASCII 文字コード「i」の場合も，正常時通信に比べて感染時通信の出現頻度が少ない割合であることが確認でき, 感染コードブックが小さい値で，正常コードブックは比較的大きい値で作成された．これは正常時通信の場合，感染時通信のペイロードの情報よりも多くの情報量が含まれているため，ASCII 文字の出現頻度が多くなるためだと考えられる．また，ASCII 文字コード「i」が有効だと判断された理由として, User-Agent に含まれる情報（Mozilla，Windows 等）や HTTP GET に含まれる情報（image，login 等）が感染時通信に比べて正常時通信で多く出現しているためである．

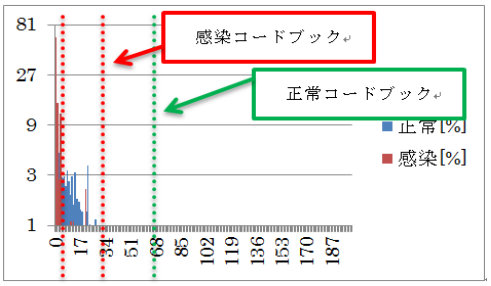

図 3：ASCII 文字コード「i」のスロット数の割合が 1%以上のヒストグラム

（縦軸：スロット数の割合[%]，横軸：出現頻度[回]）

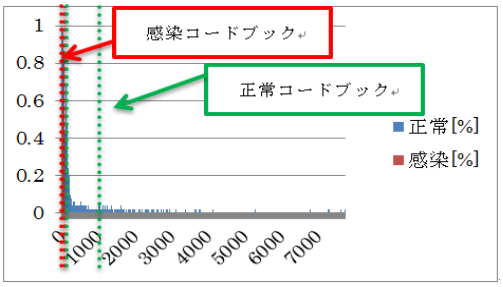

図 4：ASCII 文字コード「i」のスロット数の割合が 1%以下のヒストグラム

（縦軸：スロット数の割合[%]，横軸：出現頻度[回]）

正常の分布と感染の分布の重なり具合に着目すると，主に出現頻度が低いところで正常の分布と感染の分布に重なりがあることが確認できる．

しかし，重なっている部分は正常時通信のごく一部であり（正常時通信全体の 5%未満），正常時通信と感染時通信を分離できるため，5.1 節，5.2 節で評価した特徴量が有効であることが考えられる．また，TNR に関しても，ユーザの通信内容によって出現頻度が異なるが，正常時通信で出現頻度が高いところで分布し，感染時通信の重なっている部分が比較的少ない（正常時通信全体の 20%未満）く，正常時通信と感染時通信を分離できるため，5.1 節，5.3 節で評価した特徴量が有効であることが考えられる．また，イントラネット B の正常時通信を用いた場合でも，同様の結果が得られたことを確認できた．

上記の結果から，正常時通信と感染時通信に違いが見られることが確認でき， HTTP リクエスト長，ASCII 文字コード「i」を用いることで，感染と正常を区別できたと考えられる．

6.2. マルウェア種類毎に有効な特徴量

マルウェア種類毎に有効な特徴量について，ワーム，トロイの木馬，ファイル感染型ウイルスに分類されるマルウェアそれぞれにおいて共通した挙動（ペイロード情報）を挙げ，それらの挙動を説明し，有効な特徴量について考察した．

- **ワーム**

インターネット接続確認

ワームが感染活動を行い，PC を感染させた後に，その PC がインターネットに接続されているかを確認する．その際ワームは、特定のドメインにインターネット接続確認を行う．特定のドメインとは，「www.whatismyipaddress.com」や「checkip.dyndns.org」のような IP アドレスを表示するサイトである．

攻撃通信を行うためのマルウェアのダウンロード

起点となるワームに感染後，各サーバーから他のマルウェアを HTTP GET によりダウンロードする．このコマンドは，「GET /vss.exe HTTP/1.0」や「GET /fdc1.data HTTP/1.0」のようなものである．感染後に行う通信は，HTTP リクエスト長が概ね 100 以下と短くなる．それに対して，正常時通信は概ね 100 以上の通信を行っているものがほとんど（スロット数の割合が 95%以上）である．

よって，ドメイン等に含まれる ASCII 文字コードが正常時通信と比べて出現頻度が低いことや，HTTP リクエスト長が感染時通信で短くなることから，表 2 で示した特徴量はマルウェア感染検知に有効であると判断できる．

- **トロイの木馬**

攻撃通信を行うためのマルウェアのダウンロード

起点となるマルウェアをダウンロード後，各サーバーから他のマルウェアを HTTP GET によりダウンロードする．例としては，「GET /vot.exe HTTP/1.0」や「GET /15Psv3zJ/4ah6NuS.exe HTTP/1.0」のような HTTP GET によるダウンロードを行う．HTTP GET による通信だけを行っているものが多く，ペイロードの情報量が少ない．そのため，正常時通信と比較すると、改行（¥r¥n）の数が少ない傾向がある．また感染後に行う通信は，HTTP リクエスト長が概ね 30 以下と短くなる．それに対し，正常時通信は概ね 100 以上であるものが大半（スロット数の割合が 95%以上）である．

よって HTTP GET 等に含まれる ASCII 文字コードが正常時通信と比べて出現頻度が低いことや，

HTTP リクエスト長が感染時通信で短くなることから，表 2 に示した特徴量はマルウェア感染検知に有効であると判断できる．

- **ファイル感染型ウイルス**

IRC 通信による C&C サーバーに接続（IRC 接続）

ファイル感染型ウイルスは感染活動を行うための準備として，IRC 通信を行い C&C サーバーに接続する．IRC 接続を行った後，攻撃通信を行うためのマルウェアのダウンロードや標的に対して妨害攻撃を行う等の活動が行われる．今回用いた検体では，攻撃通信を行うためのマルウェアのダウンロードを行っていた．IRC 通信を行う際の通信内容は特定の文字列（IRC ドメイン:*norks.org* 001 *tjrrxaek*:等）が同程度の数が繰り返し多く出現する．それらの文字列に含まれる文字は，感染時通信において同程度の数が繰り返し多く出現する．

攻撃通信を行うためのマルウェアのダウンロード

起点となるマルウェアをダウンロード後，マルウェアは各サーバーから他のマルウェアを HTTP GET によりダウンロードする．例としては，「GET /jiri.data HTTP/1.0」や「GET 44.data HTTP/1.0」などのような HTTP GET によるダウンロードを行う．ファイル感染型ウイルスが感染後に行う通信は，HTTP リクエスト長が 10～80 であるのに対し，正常時通信は概ね 100 以上であるものが大半（スロット数の割合が 95%以上）である．

よって，IRC 通信に含まれる ASCII 文字コードが正常時通信と比べて出現頻度が**同程度の数で繰り返し多い**ことや，HTTP リクエスト長が感染時通信で短くなることから，表 2 で示した特徴量はマルウェア感染検知に有効であると判断できる．

以上マルウェア 3 種の考察から，今回用いたデータセットの場合，HTTP 通信を用いたマルウェアの感染活動は，インターネットの接続確認や攻撃通信を行うためのマルウェアのダウンロードを行っている．また，攻撃通信を行うためのマルウェアのダウンロードを行う際に，IRC 通信を行うものや外部サーバーから直接ダウンロードを行うもの等，マルウェアの種類毎で異なる挙動が確認できた．さらに，感染活動を行うときのマルウェアの種類毎でマルウェアのダウンロードを行う時等のペイロード情報の文字列が異なるため，マルウェアの種類毎で有効だと判断された特徴量が異なったと考えられる．

6.3. 時間的変化を用いた識別の有効性

トラヒックデータは時間的な変化を伴うため，時間的変化を考慮した識別の有効性を考察する．

例として，ファイル感染型ウイルスの ASCII 文字コード「o」と正常時通信の比較を図 5，図 6 に示す．これらの図では，縦軸を出現頻度，横軸をスロットとした．出現頻度は 1 スロット毎の 1 パケットの出現頻度である．

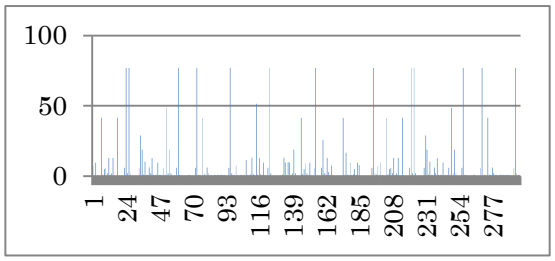

図 5：ファイル感染型ウイルスの ASCII 文字コード「o」の出現頻度の時間的変化
（縦軸：出現頻度[回]，横軸：スロット番号）

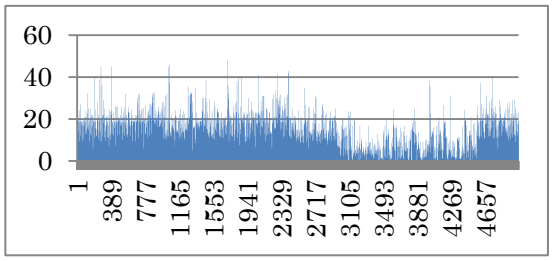

図 6：正常時通信の ASCII 文字コード「o」の出現頻度の時間的変化
（縦軸：出現頻度[回]，横軸：スロット番号）

ファイル感染型ウイルスの出現頻度（図 5）の時間的変化は，正常時通信（図 6）に比べて大きい．また，ファイル感染型ウイルスの出現頻度が高くなっているスロットは，ファイル感染型ウイルスが IRC 通信を行っており，そのペイロード情報に特定の文字列（IRC ドメイン:*norks.org* 001 *tjrrxaek*:など）が連続的に多く出現し，その文字列に含まれる ASCII 文字コードが多く出現するためである．この特徴は，本実験では，ワームやトロイの木馬に出現せず，ファイル感染型ウイルス特有の特徴であった．

時間的な変化を考慮した識別を行うに当たり，今回の実験より以下のような特徴を考慮することが有効であると考える．

i. ワーム
・ 出現頻度の増減を確認する
→増減が小さい場合，感染している可能性有
ii. トロイの木馬
・ 出現頻度が周期的に一定値を示しているかを確認する
→一定の場合，感染している可能性有
・ 出現頻度の増減を確認する
→増減幅が小さい場合，感染している可能性有
iii. ファイル感染型ウイルス
・ 出現頻度の増減を確認する
→増減が大きく，増加したときの出現頻度が一定の場合，感染している可能性有

7. まとめ

本稿では，D3M2012，CCC2009，2010，2011と2種類の正常時通信を用い，マルウェアの種類毎（ワーム，トロイの木馬，ファイル感染型ウイルス）における，感染検知の識別に有効な特徴量を評価した．その結果，マルウェアの種類毎で，特定のASCII文字コードとHTTPリクエスト長が，2つの正常時通信を用い，量子化レベル数を変化させても，TPR・TNRが高く，安定的に検知できることがわかった．具体的には，TPRのみが高い特徴量としてワームで3種類，トロイの木馬で15種類，ファイル感染型ウイルスで5種類を示し，TNRが高い特徴量としてワームで1種類を示した．その中でも，TPR・TNRが共に高い特徴量として，ワームではASCII文字コード「i」とファイル感染型ウイルスではHTTPリクエスト長が特に安定的に検知できることがわかった．

また，ワームでは「インターネット接続確認等」，トロイの木馬では「攻撃通信を行うためのマルウェアのダウンロード等」，ファイル感染型ウイルスでは「IRC接続等」の感染活動行うことが確認でき，挙動と有効な特徴量の間の関連性を明らかにした．さらに，時間的変化においても，正常時通信とマルウェアの種類毎の感染時通信を比較したとき，マルウェアの種類毎の感染時通信で，正常時通信の特徴には表れない時間的な特徴が表れた．

よって，これらの種類毎に有効な特徴量として使用できる可能性があるものとして示した特徴量を適切に組み合わせることで，マルウェア感染検知の検知率の向上につながることが考えられる．

今後は，今回評価できなかった特徴量に対して，マルウェア感染検知に有効であるかを調査し，特徴量を組み合わせたマルウェア感染検知について検討していく．さらに，今回の実験で定義したマルウェアの種類は，ベンダーが定義した種類名を使用したが，マルウェアの挙動（インターネット接続確認やIRC接続等）をクラスタリングし，クラスタごとに有効な特徴量の評価も検討していく．

参考文献

[1] Gdataマルウェアレポート 2011年上半期 http://www.gdata.co.jp/files/GdDataH1MalRep.pdf

[2] 藤原将志，寺田真敏，安部哲哉，菊池浩明，“マルウェアの感染方式に基づく分類に関する検討，”情報処理学会 CSEC研究報告，No.21，p177-182，2008年3月

[3] MWS2012 実行委員会，研究用データセット MWS 2012 Datasetsについて，http://www.iwsec.org/mws/2012/about.html#datasets

[4] 与那原亨　大谷尚通　馬場達也　稲田勉，“トラフィック解析によるスパイウェア検知の一考察，”電子情報通信学会技術研究報告，Vol.2005, No.70, 2005-CSEC-30, pp.23-29

[5] Marius Kloft et.al，“Automatic feature selection for anomaly detection，” Conference on Computer and Communications Security 2008

[6] 桑原和也，菊池浩明，寺田真敏，藤原将志，“パケットキャプチャから感染種類を判定する発見的手法について，”マルウェア対策研究人材育成ワークショップ2009(MWS2009)，2009年

[7] Wei Lu et.al.，“Automatic Discovery of Botnet Communities on Large-Scale Communication Networks,” the 4th International Symposium on Information, Computer, and Communications Security, 2009

[8] 山田明，三宅優，田中俊昭，竹森敬祐，“学習データを自動生成する未知攻撃検知システム，”情報処理学会論文誌，Vol.46，No.8，pp.1947-1958，2005

[9] トレンドマイクロ　セキュリティデータベース，http://jp.trendmicro.com/jp/home/index.html

[10] Linde Y, Buzo A. and Gray R，“An Algorithm for Vector Quantization，” IEEE Trans，Commun，Vol.28 No,1 pp84-95,1980

[11] 川元研治，市田達也，市野将嗣，畑田充弘，小松尚久，“マルウェア感染検知のための経年変化を考慮した特徴量評価に関する一考察，”マルウェア対策研究人材育成ワークショップ2011(MWS2011)，2011年