2012年度 修士論文

悪性Webサイト探索のための効率的な巡回順序の決定法

提出日：2013年2月1日

指導：後藤滋樹教授

早稲田大学 基幹理工学研究科 情報理工学専攻
学籍番号：5111B073-1

千葉 大紀

目次

図一覧

表一覧

第 1 章

序論

1.1 研究の背景

Web 経由のマルウェア[1]感染事例が増加している．マルウェアに感染すると，端末上の個人情報が攻撃者に送付されたり，攻撃者によって端末が不正に制御され，Distributed Denial of Service (DDoS) 攻撃や迷惑メール送信に利用される．2009 年からは，Web ブラウザやそのプラグインの脆弱性を攻撃対象とする悪性 Web サイトが急増しており，マルウェアに感染する端末の増加や被害の深刻化につながっている [2]．この脅威に対し，ユーザが悪性 Web サイトへアクセスすることを防ぐための対策が進められている．例えば，あらかじめ悪性 Web サイトを発見しておくことで，ユーザを保護する対策が講じられている [2, 3, 4]．悪性 Web サイトを発見するためには，Web クライアント型ハニーポットが用いられる．Web クライアント型ハニーポットとは，Web 空間を巡回し，悪性 Web サイトの情報を収集・分析するシステムである．このシステムで収集した Internet Protocol (IP) アドレスや Uniform Resource Locator (URL) の情報はブラックリストの生成に利用され，ブラックリストをセキュリティ機器に適用することで，ユーザを保護することができる．

しかし，この対策手段には Web クライアント型ハニーポットの性能に起因する二つの課題が存在する．一つは，巡回すべき悪性 Web サイトの選定である．Web 空間は日々拡大する一方，Web クライアント型ハニーポットが単位時間あたりに巡回できる Web サイトの数には限界がある．また，攻撃者は悪性 Web サイトを数多く展開しており，その URL は短い期間で遷移している [2]．この状況の中で，すべての悪性 Web サイトを Web クライアント型ハニーポッ

[1]マルウェア (malware) とは英語の malicious (悪意のある) と software を組み合わせた混成語であり [1]，悪意のあるソフトウェアの総称である．

トによって発見することは難しい．もう一つの課題は，Web サイトの再巡回である．Web サイトは Web クライアント型ハニーポットによって一度巡回・検査されるだけでは不十分であり [5]，再巡回をおこない継続的に検査する必要がある．なぜなら，巡回された時点では悪性 Web サイトではなくとも，その後悪性 Web サイトと変化する可能性があるためである．上記の二つの課題を解決するためには，より悪性の可能性が高い Web サイトを適切に選定し，Web クライアント型ハニーポットの巡回を効率化する必要がある．

1.2　研究の目的

1.1 節で示した二つの課題を解決するため，本研究では Web クライアント型ハニーポットがより効率的に悪性 Web サイトを発見するための最適な巡回順序を決定する手法を提案する．具体的には，まず Web サイトの IP アドレス，WHOIS 情報，Fully Qualified Domain Name (FQDN) 文字列の分析により，良性 Web サイトと悪性 Web サイトを識別し得る特徴を抽出する．次に，抽出した特徴を用いて教師あり機械学習を適用し，未知の Web サイトの悪性度を推定する．悪性度が高いと推定される Web サイトから順番に巡回することで，より効率的に未知の悪性 Web サイトを発見することができる．本研究は，悪性 Web サイトに対する既存の対策手段を置き換えるものではなく，既存手法をより効果的に利用できるように拡張をおこなうものである．本研究の手法を導入すると，未知の悪性 Web サイトをより早期に発見することができる．したがって，本研究はより安心・安全な Web 空間の実現に貢献できる．

1.3　論文の構成

本論文は以下の章により構成される．

第 1 章 序論

本論文の概要について述べる．

第 2 章 悪性 Web サイトと対策技術

悪性 Web サイトの脅威とその対策技術や関連研究について解説する．

第 3 章 提案手法

本論文の提案手法を説明する．

第 4 章 特徴量の有効性評価

提案手法で利用する特徴量の有効性を実データを用いて示す.

第 5 章 提案手法の性能評価

提案手法にしたがった性能評価と考察をおこなう.

第 6 章 結論

本論文の結論を述べるとともに，残された課題を示す.

第 2 章

悪性 Web サイトと対策技術

本章では，はじめに本研究の対象となる悪性 Web サイトとその攻撃の仕組みを説明する．次に，悪性 Web サイト対策における既存技術や関連研究について解説する．本論文では既存技術や関連研究を三つの分類 (ブラックリスト，Web レピュテーション，Web クライアント型ハニーポット) に分け，それぞれの概要と課題を整理する．

2.1 悪性 Web サイトの脅威

2009 年から Web ブラウザやそのプラグインの脆弱性を攻撃対象とする悪性 Web サイトが急増している [3, 6]．このような悪性 Web サイトで使われる攻撃手法を Drive-by-download 攻撃という．Drive-by-download 攻撃の仕組みを図 2.1 を用いて説明する．ユーザが脆弱な Web ブラウザを用いて入口サイトにアクセスすると，Hypertext Transfer Protocol (HTTP) リダイレクトにより中継サイトへ自動転送される．入口サイトとは攻撃者によって用意される Web サイトであり，次の二種類に大別できる．一つは，正規の Web サイトが改ざんされる場合である．アクセス数が多い有名な Web サイトが改ざんされると，攻撃されるユーザ数が増加する．もう一つは，ユーザが興味を持つ Web サイトが新たに用意される場合である．この場合，ソーシャルネットワーキングサイトやスパム (迷惑メール) を用いて入口サイトが拡散される [7]．中継サイトとは，次の中継サイトや攻撃サイトに接続させるための Web サイトであり，HTTP リダイレクトによる自動転送がおこなわれるように設定される．ユーザは複数の中継サイトを経由し，最終的に攻撃サイトへ自動転送される．攻撃サイトとは，実際に脆弱性を突いた攻撃をおこなう Web サイトであり，攻撃サイトに到達したユーザの Web ブラウザ環境に脆弱性がある場合に攻撃が成立する．攻撃が成立すると，別途攻撃者によって用意されるマルウェア配

布サイトよりマルウェアがユーザ端末に強制的にダウンロードされ，ユーザの端末がマルウェアに感染する．

ユーザを Drive-by-download 攻撃によるマルウェア感染から保護するのは難しい．この攻撃では，攻撃が成立するまでに複数の悪性 Web サイトが関連し，その構造は複雑である [8]．また，各悪性 Web サイトで用いられる自動転送コードや攻撃コードは，検知されにくいように難読化されている [2]．さらに，各悪性 Web サイトは非常に短命であり，対策手段による検知を防ぐため攻撃者によって頻繁に変更される．そのうえ，攻撃に利用される脆弱性の数は多く，新たな脆弱性も次々と利用されるため根本的な対策手段は存在しない．

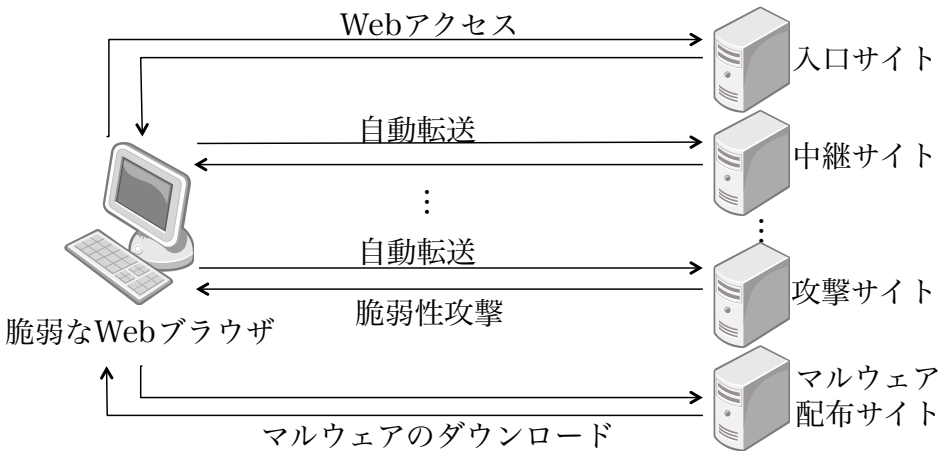

図 2.1: Drive-by-download 攻撃の仕組み

2.2　悪性 Web サイト対策の既存技術・関連研究

2.2.1　ブラックリスト

ブラックリストは悪性 Web サイトの URL や IP アドレスで構成され，通信のフィルタとしてユーザを保護するために利用される．ブラックリストによるフィルタリングはネットワーク上およびクライアント端末上でおこなうことができる．ネットワーク上のフィルタリングの例としては，Domain Name System Blacklist/Blocklist (DNSBL) [9, 10, 11] の利用や商用のセキュリティ機器における適用がある．DNSBL は DNS のプロトコルを用いて提供されるブラックリストであり，ユーザは DNSBL を提供している DNS サーバへ問合せを行うことで，ブラックリストを参照することができる．クライアント端末上の例としては，Web ブラウザに含まれるフィ

ルタリング機能がある．例えば，Microsoft 社の Internet Explorer [12] には SmartScreen フィルタ [13] が実装されている．また，Apple 社の Safari [14]，Google 社の Google Chrome [15]，そして Mozilla の Firefox [16] には，Google 社の SafeBrowsing API [17] を用いたフィルタリング機能が実装されている．上記の Web ブラウザでは，ユーザが閲覧する URL がブラックリストに含まれる場合にその URL へのアクセスを遮断する．

ブラックリストは上記のように広い範囲で利用されており，悪性 Web サイト対策として一定の成果を上げている [18]．しかし，攻撃者は悪性 Web サイトを頻繁に変更するため，ブラックリストに掲載されていない未知の悪性 Web サイトが常に存在することになる [2, 19, 20]．したがって，ブラックリストだけでは悪性 Web サイトの対策としては不十分であり，未知の悪性 Web サイトを検知できる手法との併用が必要となる．

2.2.2　Web レピュテーション

Web レピュテーションとは，既知の悪性 Web サイトから得られる情報を用いて，未知の Web サイトの悪性度を評価する仕組みである．悪性度が高いと推定される Web サイトへのアクセスをフィルタリングすることにより，ユーザを保護することができる．Web レピュテーションは，商用のセキュリティ機器に採用されているだけでなく，学術分野でも以下の研究例が存在する．

- Antonakakis ら [21] は，ドメイン名の評価システム Notos を提案している．Notos は，DNS のゾーン情報，Border Gateway Protocol (BGP) 経路情報，Autonomous System (AS) の情報を用いて，良性なドメイン名と悪性なドメイン名のそれぞれのネットワーク的な特徴をモデル化する．そのモデルを用いることで，未知のドメイン名に対する悪性度の評価を試みている．

- Felegyhazi ら [22] は，未知の悪性なドメイン名を事前にブラックリストに追加する仕組みを提案している．具体的には，既知の悪性なドメイン名から得られる WHOIS 情報や DNS の情報を用いることで，未知の悪性ドメインの予測をおこなっている．

- Ma ら [23] は，教師あり機械学習を用いて未知の URL を良性か悪性の二値に分類する手法を提案している．この手法では，URL に含まれる文字の特徴や，WHOIS 登録情報や Web サーバの地理的な情報を利用しており，これらの特徴が悪性な URL を識別する上で有効であることを示している．

- Yadav ら [24] は，ドメイン名に利用される文字列の特徴に着目した悪性なドメイン名の検出法を提案している．特に，アルゴリズム的に生成された悪性ドメイン名の文字列はランダム性が高いことを示しており，その特徴を利用した統計的機械学習手法を開発している．

- 秋山ら [2] は，検索エンジンを利用して既知の悪性 Web サイトの URL の近隣に存在する URL を効率的に抽出することにより，ブラックリストに掲載されていない未知の悪性 Web サイトを発見する手法を提案し，その有効性を示している．

- Invernizzi ら [25] は，検索エンジンに特定の検索クエリを与えることで未知の悪性 Web サイトを発見する手法を提案している．具体的には，既知の悪性 Web サイトのコンテンツやドメインの登録情報の分析を行い，悪性 Web サイトが得られる可能性の高い検索クエリを作成することで，未知の悪性 Web サイトが発見できることを示している．

上記のような Web レピュテーションは，2.2.1 節で説明したブラックリストとは異なり，未知の悪性 Web サイトにも対応できる可能性がある．その一方で，Web レピュテーションでは良性 Web サイトを誤って悪性と判断してしまう誤検知が発生し，ユーザの利便性を損なう場合がある．また，攻撃者はこのような Web レピュテーション技術に対抗するためにドメイン名や URL を頻繁に変更することが知られている [18]．したがって，Web レピュテーションでは悪性 Web サイトを正しく特定し続けるのは難しい．

2.2.3　Web クライアント型ハニーポット

Web クライアント型ハニーポットを利用した悪性 Web サイトへの対策が研究・開発されている [2, 3]．Web クライアント型ハニーポットとは，Web 空間を巡回し，悪性 Web サイトを発見するためのシステムであり，エミュレータを用いて構成される低対話型と，実際のオペレーティングシステムやブラウザによって構成される高対話型の 2 種類に大別できる [2]．低対話型の Web クライアント型ハニーポットは，Web ブラウザの既知の脆弱性を再現するエミュレータを用いて構成される．エミュレータは実際の Web ブラウザではないため，ハニーポット自体がマルウェアに感染するリスクがない．しかし，未知の脆弱性には対応できないため，悪性 Web サイトの情報収集能力に限界がある．代表的な低対話型の Web クライアント型ハニーポットには HoneyC [26] がある．一方，高対話型の Web クライアント型ハニーポットは，実際のオペレーティングシステムやブラウザを利用してシステムが構成され，低対話型よりも多くの

攻撃情報を収集可能であり，未知の悪性 Web サイトの調査に利用される．代表的な高対話型の Web クライアント型ハニーポットには，BLADE [27]，Capture-HPC [28]，Marionette [2] がある．

高対話型の Web クライアント型ハニーポットを利用して，未知の悪性 Web サイトの攻撃情報を収集・分析し，その分析結果を用いてブラックリストを生成するシステムが提案されている [3]．このシステムは，Web クライアント型ハニーポットによって新たに発見した未知の悪性 Web サイトをブラックリストに追加可能であり，2.2.1 節で説明した通常のブラックリストに比べて優位性がある．さらに，このシステムでは Web クライアント型ハニーポットによる調査で確実に悪性と判明したものをブラックリストに追加するため，2.2.2 節で説明した Web レピュテーションと比べて誤検知が起こる可能性は低い．

このような高対話型の Web クライアント型ハニーポットを用いた対策手段は，未知の悪性 Web サイトの脅威に対しても有効である．しかし，この対策手段にはコストやスケーラビリティに起因する二つの課題が存在する．一つは，巡回すべき悪性 Web サイトの選定である．攻撃者は悪性 Web サイトを数多く展開しており，さらにその URL は短い期間で遷移している [2]．一方，Web クライアント型ハニーポットが単位時間あたりに巡回できる Web サイトの数には限界がある．この状況の中で，すべての悪性 Web サイトを Web クライアント型ハニーポットを用いて発見することは難しい．もう一つの課題は，Web サイトの再巡回である．Web サイトは Web クライアント型ハニーポットによって一度巡回・検査されるだけでは不十分である [5]．なぜなら，巡回された時点では悪性 Web サイトではなくとも，その後悪性 Web サイトに変化する可能性があるためである．上記の二つの課題を解決するためには，より悪性の可能性が高い Web サイトを適切に選定できる手段を導入し，Web クライアント型ハニーポットの巡回効率を高める必要がある．

第 3 章

提案手法

本章では，提案手法である悪性 Web サイト探索のための巡回順序の決定法を説明する．はじめに 3.1 節で提案手法の概要を説明し，その後 3.2 節と 3.3 節で提案手法の仕組みを詳しく解説する．

3.1　巡回順序決定システム

提案手法を実現する巡回順序決定システムの概要図を図 3.1 に示す．本システムは，Web クライアント型ハニーポットの一部として動作し，本システムが出力する巡回順序付きの巡回 URL リストを利用してハニーポットが Web 空間を巡回する．

図 3.1 の右側に Web クライアント型ハニーポットが Web 空間を巡回する仕組みを示す．なお，Web クライアント型ハニーポットのアーキテクチャとしては，秋山ら [8, 29] が提案している Marionette を想定する．Marionette は 2.2.3 節で説明した高対話型の Web クライアント型ハニーポットであり，マネージャと複数のエージェントによって構成される．マネージャは，巡回 URL リストと巡回ログを管理し，各エージェントの動作を制御する．一方，各エージェントは，複数の Web ブラウザのプロセスを用いて巡回 URL リストに含まれる URL を巡回する．

図 3.1 の左側に，巡回順序決定システムの仕組みを示す．本手法は，以下の四つの手順で巡回 URL リストの巡回順序を決定する．

手順 1　特徴抽出エンジンは，既知の良性・悪性 Web サイト (訓練データ) から IP アドレス，WHOIS 情報，FQDN 文字列を用いて特徴ベクトルを作成する．なお，特徴抽出エンジンについては 3.2 節でその詳細を説明する．

手順 2　機械学習エンジンは，手順 1 で作成した特徴ベクトルをもとに教師あり機械学習を適用し，訓練モデルを作成する．なお，機械学習エンジンについては 3.3 節で解説する．

手順 3　巡回 URL リスト (テストデータ) を本システムの入力として与え，手順 1 と同様の手順で巡回 URL リストのそれぞれの URL から特徴抽出エンジンを用いて特徴ベクトルを作成する．

手順 4　機械学習エンジンは，手順 2 で作成した訓練モデルをもとに，手順 3 で作成した巡回 URL リストの特徴ベクトルから，それぞれの URL の悪性度を算出する．より悪性度が高いと算出された URL から優先的に巡回できるように巡回順序を決定し，その結果を巡回順序付き巡回 URL リストとして出力する．Web クライアント型ハニーポットは本システムによって出力された巡回順序付き巡回 URL リストを利用して Web 空間の巡回をおこなう．

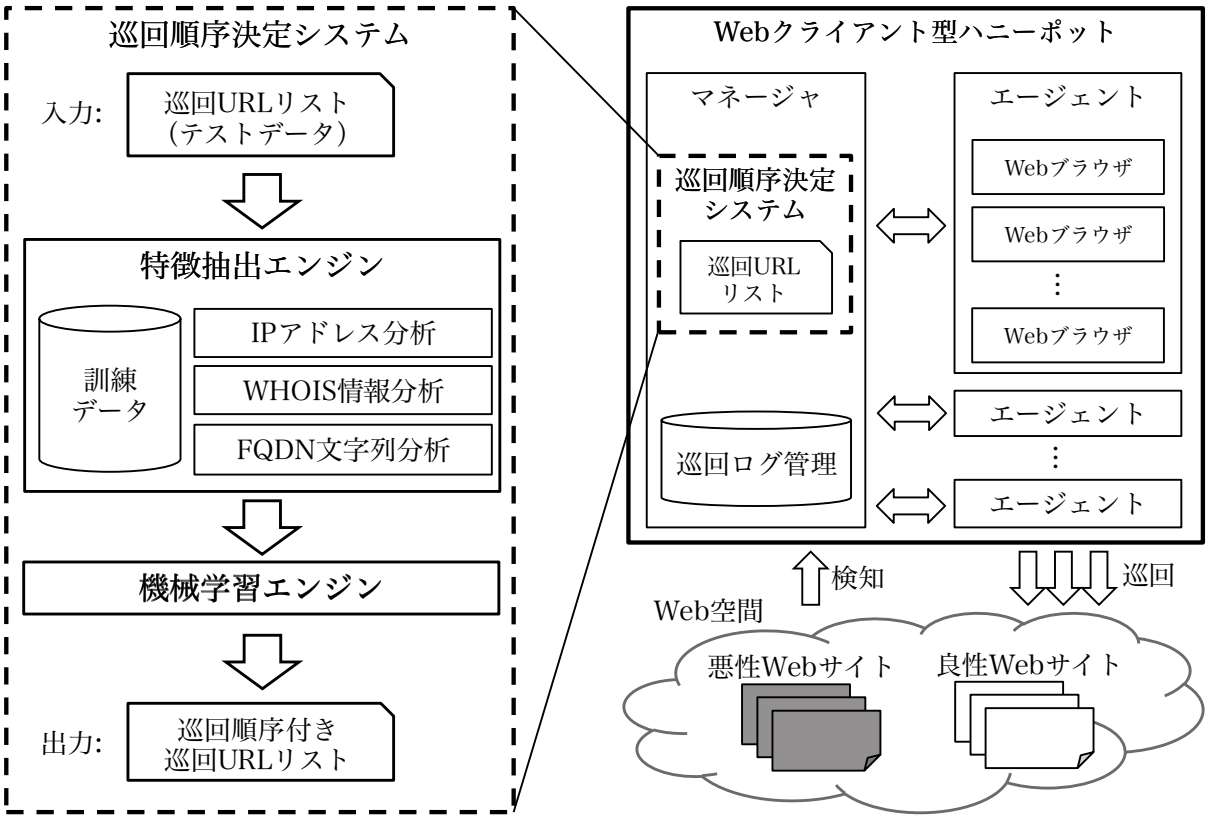

図 3.1: 巡回順序決定システム

3.2　特徴抽出エンジン

巡回順序決定システムの中の特徴抽出エンジンについて説明する．本エンジンは，3.1 節で示した手順 1 および手順 3 で利用する．手順 1 では，既知の良性・悪性 Web サイト (訓練データ) から特徴を抽出し，特徴ベクトルを生成する．手順 1 で利用する訓練データの例を表 3.1 に示す．一方，手順 3 では未知の Web サイト (テストデータ) から，手順 1 と同様に特徴ベクトルを生成する．手順 3 で利用するテストデータの例を表 3.2 に示す．

本研究で提案する特徴抽出エンジンでは，以下の三つの分析手法 (IP アドレス分析，WHOIS 情報分析，FQDN 文字列分析) を利用した特徴抽出をおこなう．ここでのポイントは，良性 Web サイトと悪性 Web サイトを識別し得る特徴を効率的に抽出することである．

表 3.1: 訓練データの例

ラベル	IP アドレス	WHOIS 情報 (ドメイン登録日)	FQDN 文字列
良性	192.0.2.1	1995/08/14	hoge1.example.com
悪性	198.51.100.88	2005/09/14	hoge0123.example4.jp
...	...	...	...

表 3.2: テストデータの例

ラベル	IP アドレス	WHOIS 情報 (ドメイン登録日)	FQDN 文字列
未知	192.0.2.2	1995/08/31	hoge2.example.net
未知	203.0.113.24	2005/09/14	hoge5678.example9.jp
...	...	...	...

3.2.1　IP アドレス分析

Web サイトの IP アドレスからの特徴抽出手法を説明する．中心となるアイデアは，IP アドレスの時間的および空間的な特徴を利用することである．悪性 Web サイトの URL やドメイン名，そして利用される攻撃コードが頻繁に変更されるのに対し，IP アドレスは時間的な変動が小さい [30]．また，悪性な活動 (ボットネット，迷惑メール送信元，悪性 Web サイト) に利用される IP アドレスは，あるネットワークブロックに空間的に偏ることが明らかになってい

る [30, 31, 32, 33]．なお，本研究で用いるデータにも上記の特徴があることは，後の 4.2 節で検証する．

われわれは以前の研究 [30] で，Internet Protocol version 4 (IPv4) アドレスから上記の特徴を効率的に抽出する手法を提案した．本研究では，上記研究で提案した手法のうち，良性 Web サイトと悪性 Web サイトの識別精度が最も良い手法を利用する．具体的には，特徴抽出をおこなう Web サイトに対応する IPv4 アドレスの第 1〜3 オクテットの各数値および，IPv4 アドレスの第 1・2 オクテットの組合せ，そして第 1・2・3 オクテットの組合せから特徴ベクトルのビット列 $\{b_0, \cdots, b_{1279}\}$ を定義する．すなわち，それぞれの要素 b_k $(0 \leq k \leq 1279)$ を，以下の式で定義する．

$$\begin{cases} b_k = 1 & (k \text{ in } \bigcup_{n=1}^{3} \{2^8 \cdot (n-1) + X_n\}) \\ b_k = 1 & (k \text{ in } \bigcup_{m=3}^{4} \{2^8 \cdot m + (\sum_{n=1}^{m-1} X_n) \bmod 2^8\}) \\ b_k = 0 & (\text{otherwise}) \end{cases}$$

ここで，X_n は特徴抽出をおこなう IP アドレスの第 n $(1 \leq n \leq 3)$ オクテットの 10 進表記である．例えば，IP アドレス 198.51.100.88 から上記の式にしたがって特徴抽出をおこなうと特徴ベクトルの要素 b_k $(0 \leq k \leq 1279)$ は下記のようになる．

$$\begin{cases} b_k = 1 & (k = 198\ (= X_1)) \\ b_k = 1 & (k = 307\ (= 2^8 + X_2)) \\ b_k = 1 & (k = 612\ (= 2^8 \cdot 2 + X_3)) \\ b_k = 1 & (k = 1017\ (= 2^8 \cdot 3 + (X_1 + X_2) \bmod 2^8)) \\ b_k = 1 & (k = 1117\ (= 2^8 \cdot 4 + (X_1 + X_2 + X_3) \bmod 2^8) \\ b_k = 0 & (\text{otherwise}) \end{cases}$$

ここで，$k = 198, 307, 612$ はそれぞれ IPv4 アドレスの第 1〜3 オクテットの各数値から定義され，$k = 1017, 1117$ はそれぞれ第 1・2 オクテットの組合せおよび第 1・2・3 オクテットの組合せから定義されたものである．

3.2.2　WHOIS 情報分析

Web サイトのドメイン名から得られる WHOIS 情報を利用した特徴抽出手法を説明する．本手法では，WHOIS 情報のうちドメイン登録日を利用して，登録日の新しさに着目した特徴抽出をおこなう．具体的には，ドメインが登録されてからの経過日数 (登録期間) を特徴ベクトルの要素 W として以下の式で定義し，W が小さいほど，悪性度が高くなるよう特徴抽出をおこなう．

$$W = d_n - d$$

ここで，d_n は特徴抽出をおこなう日付，d は特徴抽出をおこなう Web サイトのドメイン登録日とする．

この手法は，新しいドメイン登録日を持つ Web サイトの方がより悪性度が高いという既存研究による報告に基づいている．例えば，Ma ら [23] は，良性・悪性 Web サイトを識別するための特徴として，ドメイン登録日，更新日，有効期限日，レジストラ名，登録者情報を利用している．その中でドメイン登録日は特に有効な特徴であることが指摘されている．また，Felegyhazi ら [22] は，既知の悪性 Web サイトと同一のドメイン登録日を持つサイトを悪性 Web サイトの候補と仮定する手法を提案しており，ドメイン登録日が比較的新しいものに悪性 Web サイトが多く含まれることを示している．なお，本研究で用いるデータにおけるドメイン登録日と良性・悪性 Web サイトの関係は，後の 4.3 節で検証する．

3.2.3　FQDN 文字列分析

Web サイトの FQDN 文字列の長さ，エントロピー，n-gram から得られる情報を利用した特徴抽出手法を以下順番に説明する．

FQDN 文字列の長さ　FQDN 文字列の長さを特徴として利用する．実データを用いて，良性・悪性 Web サイトの FQDN 文字列の長さをそれぞれ分析した結果，その分布に大きな差があることが判明した．この分析結果は以下の 4.4.1 節に記載する．この特徴を利用するために，特徴ベクトルの要素 L を以下の式で定義する．

$$L = (\text{FQDN 文字列の長さ})$$

FQDN 文字列のエントロピー　FQDN 文字列のエントロピー (平均情報量) を特徴として利用する．実データを用いた分析の結果，悪性 Web サイトに使われる FQDN 文字列は文字列としてのランダム性が高く，エントロピーが高くなるという性質があることがわかった．なお，この分析結果は以下の 4.4.2 節に記載する．ここで，FQDN 文字列のエントロピーを利用した特徴ベクトルの要素 E を，n 文字の FQDN 文字列 $X = \{x_1, x_2, \cdots, x_n\}$ のエントロピーを用いて，以下の式で定義する．

$$E = -\sum_{i=1}^{n} p(x_i) \log p(x_i)$$

ここで，$p(x_i)$ は，FQDN 文字列 X において文字 x_i が出現した経験確率である．

FQDN 文字列の n-gram　FQDN 文字列から n-gram ($n = 2$) を抽出し，特徴として利用する．n-gram とは，ある文字列の中から切り出した n 文字の連続した部分文字列の集合のことである．基本的なアイデアは上記のエントロピーと同様で，悪性 Web サイトにおける FQDN 文字列の高いランダム性を利用することである．具体的には，FQDN 文字列から 2 文字の連続した文字列を切り出す．ここで，FQDN に存在する文字として，アルファベット (26 文字)，数字 (10 文字)，記号 (ドット，ハイフン) がある．これらの 2 文字の組合せのうち，数字または記号が少なくとも 1 文字含まれるもの (例 '1a', 'e-') のみを特徴として抽出する．以下の 4.4.3 節では，数字または記号を含む文字列の出現頻度を分析することで，本手法の有効性を実証する．ここで，各 FQDN 文字列の n-gram 文字列を利用した特徴ベクトルをビット列 $\{g_{'-0'}, \cdots, g_k, \cdots, g_{'z9'}\}$ として定義する．ここで -0，$z9$ は数字または記号が含まれる 2 文字の組合せの意味である．それぞれの要素 g_k は以下の式で定義する．

$$g_k = N_k$$

ここで N_k は，n-gram 文字列 k の各 FQDN 文字列における出現頻度である．

FQDN 文字列の n-gram における例外処理　上記の n-gram を用いた特徴抽出手法における例外処理の必要性を述べる．悪性 Web サイトの FQDN 文字列にはランダム性が高い文字列が含まれる可能性が高い．一方で，良性 Web サイトの FQDN 文字列の中にも，商品名・会社名・コンテンツ識別子の都合により，ランダム性の高い文字列が使われることがある．このままの状態で，n-gram 文字列の特徴ベクトルを利用すると，適切に評価できない良性 Web サイトが存在する．そこで，特徴抽出エンジンでは，n-gram の特徴を利用しないドメイン名 (例外ド

メイン名) をあらかじめ定義することでこの問題を回避する．この例外ドメイン名には，悪性 Web サイトの可能性が極めて小さいドメイン名を登録する．本研究では，例外ドメイン名として 20 個のドメイン名 (e.g. 大手検索業者のサービス，大手ソーシャルアプリケーション) を選択した．以下の 5 章では，例外ドメイン名を用いた例外処理をおこなう場合とおこなわない場合の性能を比較する．

3.3　機械学習エンジン

3.2 節の特徴抽出エンジンが抽出した各特徴ベクトルを統合し，教師あり機械学習手法を適用する．適用可能な機械学習手法としては，k 近傍法 (k-Nearest Neighbor，k-NN)，フィードフォワードニューラルネットワーク (Feedforward Neural Network)，そしてサポートベクターマシン (Support Vector Machine，SVM) がある．

k 近傍法はテストデータと訓練データとの間の距離を計算し，その近さによってテストデータを分類する基本的な手法である．k 近傍法はテストデータを与えるたびに毎回距離の計算をする必要があるため，計算量が高いという欠点がある [34]. また，通常の k 近傍法では本研究で提案している特徴ベクトル空間を適切に扱うことはできない．なぜなら，本研究で提案している特徴ベクトルはスパースかつ高次元なベクトルであり，ベクトル間の距離を適切に定義するのが難しいためである．

フィードフォワードニューラルネットワークは，データの分類にも適用可能な計算モデルである．フィードフォワードニューラルネットワークの実行には多くのパラメータを決定する必要があるだけでなく，大域的最適解を得ることはできないという問題がある [34].

サポートベクターマシン (SVM) は本論文の執筆時点で最も精度が高い分類手法のうちの 1 つである [35]．SVM は k 近傍法とは異なり高次元の特徴ベクトルを適切に扱うことができる．また，SVM はフィードフォワードニューラルネットワークとは異なり動作に必要なパラメータも少なく，大域的最適解を得ることができる [34, 35]．さらに，SVM は関連研究 [23, 30] において高い精度で悪性 Web サイトを検知した報告があるだけでなく，これまでに多種多様な課題に対して適用され [35]，優れた識別能力があることが知られている．

上記の各機械学習手法の比較の結果，本研究の機械学習エンジンでは，高い精度かつ決定すべきパラメータが少ない SVM を選択する．なお，本研究では SVM の実装として LIBSVM [35] を利用する．

3.3.1　訓練モデルの作成

3.1 節で示した手順 2 では，訓練データをもとに SVM を用いて訓練モデルを生成する．SVM は訓練モデルの生成のため，訓練データの特徴ベクトルを入力し，そのデータを高次元に写像した上で，データを識別するための境界超平面を構築する．SVM により境界超平面を構築する際には，境界超平面とそれぞれの訓練データとの距離 (マージン) を最大化するアプローチ (マージン最大化) をとることで，最適な境界超平面を構築する．図 3.2 に，SVM の境界超平面の概念図を示す．白い丸印は良性訓練データ，黒い丸印は悪性訓練データの特徴ベクトルの配置例を示しており，その間に境界超平面が構築された状況を示している．

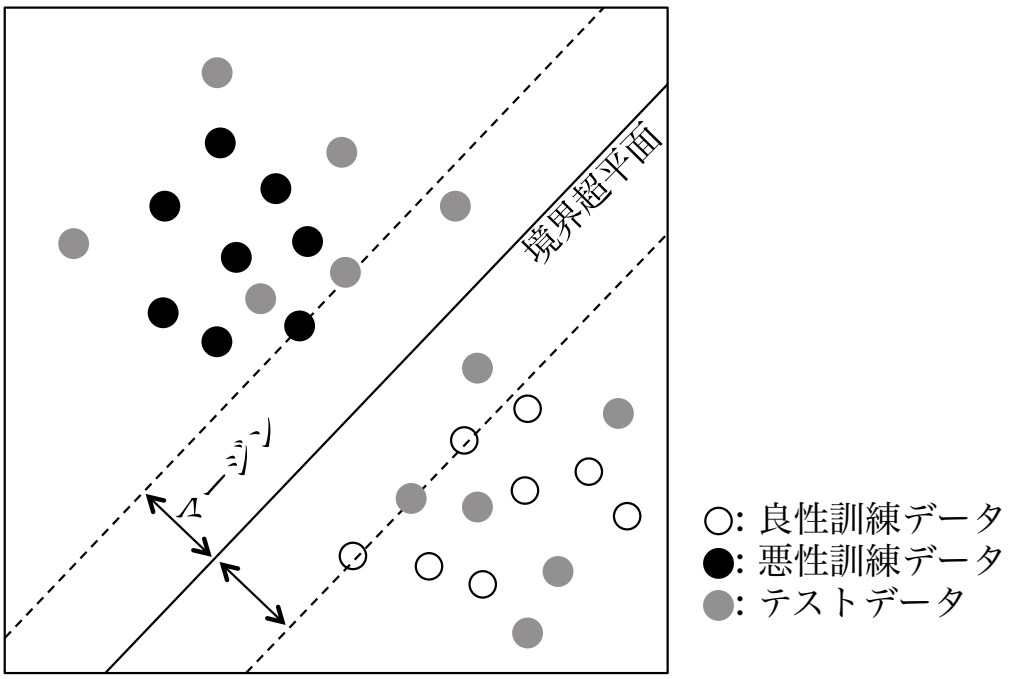

図 3.2: SVM の境界超平面の概念図

境界超平面の構築を定式化する．SVM の識別関数は

$$y(\mathbf{x}) = \mathbf{w}^{\mathbf{T}}\phi(\mathbf{x}) + \beta$$

と表される．ここで，$\mathbf{x}$ は特徴ベクトル，$\mathbf{w}$ は境界超平面の位置を移動するパラメータ，$\mathbf{w}^{\mathbf{T}}$ は $\mathbf{w}$ の転置行列，$\phi(\mathbf{x})$ は特徴ベクトル $\mathbf{x}$ の高次元空間への写像関数，そして β はバイアス項である．なお，訓練データに含まれる N 個の特徴ベクトルはそれぞれ $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N$ と表され，それぞれに対応するラベル $t_1, \cdots, t_N$ ($t_n \in \{-1$ (良性), 1 (悪性)$\}$) が付与されている．また，入

力データを高次元空間に写像するためのカーネル関数としては，ガウスカーネル [35] を採用する．ここで，最適な境界超平面を構築するためには，パラメータ $\mathbf{w}$ と β を調整し，マージンを最大化する必要がある．これは次の最適化問題

$$\mathrm{argmax}_{\mathbf{w},\beta}\left\{\frac{1}{|\mathbf{w}|}\min_{n}\left[t_n(\mathbf{w}^T\phi(\mathbf{x}_n)+\beta)\right]\right\}$$

を解くことに帰着する．本研究ではこの最適化問題の数値計算のために，Sequential Minimal Optimization (SMO) アルゴリズム [36] を利用した．

3.3.2　悪性度の算出

3.1 節で示した手順 4 では，巡回 URL リスト (テストデータ) に含まれる URL の最適な巡回順序を決定する．なお，テストデータは良性か悪性かが判明していない未知の Web サイトの URL で構成される．テストデータの巡回順序を決定するには，次の三つの操作をおこなう．まず，手順 3 によりテストデータのそれぞれの URL に対応する特徴ベクトルを作成する．テストデータの特徴ベクトルの配置例を，図 3.2 の灰色の丸印として示す．次に，特徴ベクトルから手順 2 で作成した訓練モデル (境界超平面) をもとに悪性度を算出する．最後に，算出した悪性度をもとに URL を悪性度が高い順序に並び替えることで，巡回順序付きの巡回 URL リストを得る．このリストは悪性度が高いと推定される URL から順番に並んだ URL リストのため，Web クライアント型ハニーポットがこのリストに基づき巡回をおこなうことで，従来よりも効率的に悪性 Web サイトを発見することが可能となる．

悪性度の算出を定式化する．一般的な SVM を用いると良性と悪性の二値分類の結果しか得られない．しかし，今回は文献 [37] で提案されている手法を応用し，境界超平面からの距離を近似することにより，悪性度を連続値で算出する．具体的には，ある特徴ベクトル $\mathbf{x}$ の悪性度 $p(\mathbf{x})$ を，

$$p(\mathbf{x}) = P(t=1|\mathbf{x}) = \sigma(Ay(\mathbf{x})+B)$$

と定義する．なお，$p(\mathbf{x})$ の範囲は $0.0 < p(\mathbf{x}) < 1.0$ であり，0.0 が最も良性，1.0 が最も悪性とする．ここで $\sigma(a)$ はロジスティックシグモイド関数 $\sigma(a) = 1/(1+\exp(-a))$ を表す．また，パラメータ A と B は訓練データの値の組合せ $(y(\mathbf{x}_n), t_n)$ によって定まる交差エントロピー誤差関数を最小化することで訓練データ毎に一意に定まる．

第 4 章

特徴量の有効性評価

本章では， 3.2 節の特徴抽出エンジンで抽出した各特徴量が良性 Web サイトと悪性 Web サイトを識別するのに有効であることを実データを用いて検証する．まず，本研究で利用した実データを 4.1 節に示し，次に利用したそれぞれの特徴量 (IP アドレス，WHOIS 情報，FQDN 文字列) についての評価結果を 4.2 節から 4.4 節まで順番に示す．

4.1 訓練データセット

評価のために用意した訓練データセットについて説明する．提案手法では 3.3 節で示したとおり，事前に既知の良性・悪性 Web サイトの情報 (訓練データ) を用いた教師あり機械学習をおこなう．本研究で利用した訓練データセットの内訳を表 4.1 に示す．良性訓練データとして，Alexa 社が提供している Alexa Top sites [38] に掲載されている Web サイトのうち上位 10,000 件 (重複なし) を利用した．Alexa Top sites は，Web サイトの平均日別訪問者数および過去 1ヶ月間のページビュー数によって算出されている Web サイトランキングであり，上位に掲載されている Web サイトは良性である可能性が高い．一方，悪性訓練データとして，公開の悪性 Web サイトブラックリストである Malware Domain List (MDL) [39] のうち，35,438 件 (重複なし) を利用する．なお，良性と悪性の両訓練データは，2011 年 4 月 30 日現在までに収集したものを利用している．

表 4.1: 訓練データセット

データ	収集期間	Web サイト数
良性訓練データ	2011/4/30	10,000
悪性訓練データ	2009/1/1〜2011/4/30	35,438
合計		45,438

4.2　IP アドレスの特徴

本節では良性 Web サイトと悪性 Web サイトを識別する上での，IP アドレスの有効性について検証する．ここでは，悪性 Web サイトに用いられる IP アドレスの時間的安定性および空間的局所性に着目した分析をおこなう．

4.2.1　時間的安定性

悪性 Web サイトに用いられる IP アドレスの時間経過にともなう変化を分析する．ここでは，表 4.1 に示した悪性訓練データに含まれる IP アドレスを分析対象とする．悪性 Web サイトの IP アドレスを Autonomous System (AS) 単位で集計し，IP アドレス数が多い上位 5 件の AS の分析をおこなった．分析対象の AS の内訳を表 4.2 に示す．なお，表 4.2 中の AS 番号はセキュリティ上の理由によりマスク処理を施している．ここで，それぞれの悪性 Web サイトの IP アドレスがはじめて観測された日の基準日 (2009 年 1 月 1 日) からの経過日数を計算する．その結果を各 AS ごとに累積分布 (Cummulative Distribution Function，CDF) として図 4.1 に示す．図 4.1 の累積分布のグラフが横ばいではなく，継続的に右肩上がりとなっているのは，時間が経過しても当該 AS で新たな悪性 Web サイトの IP アドレスが観測され続けていることを表している．表 4.2 および図 4.1 に示した結果より (1) 特定の AS に多くの悪性 Web サイトの IP アドレスが含まれる，(2) 悪性 Web サイトの IP アドレスを多く含む AS は 2 年以上継続的に利用されている，という事実が明らかになった．

上記のように悪性 Web サイトの IP アドレスは AS 単位で継続的に同じものが利用される一方，悪性 Web サイトの URL や FQDN は非常に短い期間で変化することが知られている [20]．例えば，本研究で悪性訓練データとして利用している Malware Domain List (MDL) [39] に含まれる URL の 60%以上は，1ヶ月未満しか有効ではないという調査結果 [2] がある．さらに，攻撃者は用意した URL がブラックリスト化されるのを防ぐために，大量に新しい URL を生成

することが明らかになっている [2, 18]．本研究における分析と既存研究の報告から，悪性 Web サイトの IP アドレスは URL や FQDN に比べて時間的に変化しにくい性質 (時間的安定性) があることがわかった．この IP アドレスの時間的安定性は，3.2.1 節で示した特徴抽出エンジン (IP アドレス分析) で IP アドレスを特徴量として採用する根拠となる．

表 4.2: 分析対象の AS

AS 番号	FQDN 数	IP アドレス数
AS #1	1,389	482
AS #2	2,422	400
AS #3	1,061	355
AS #4	761	280
AS #5	1,047	275

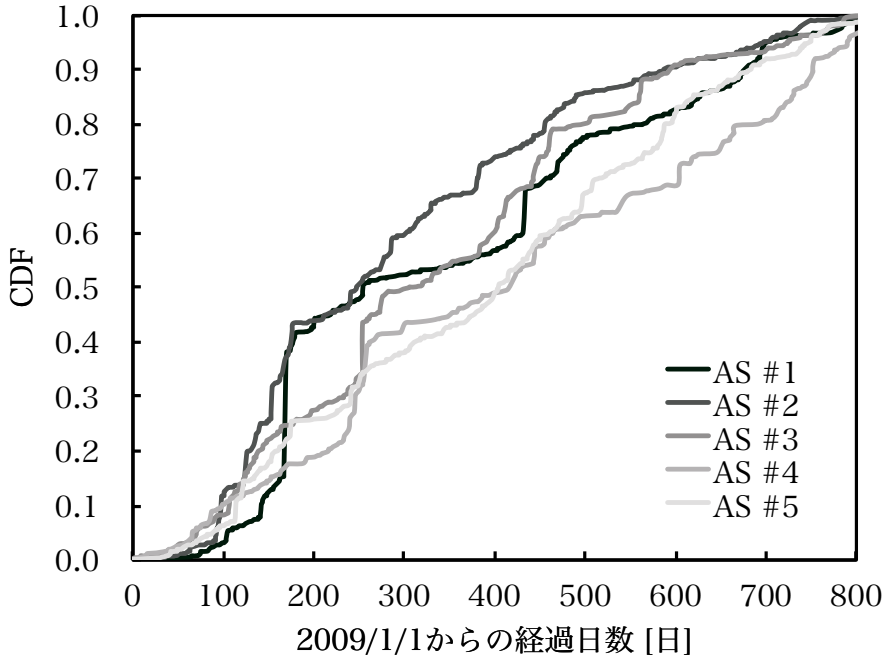

図 4.1: IP アドレス数の累積分布

4.2.2　空間的局所性

悪性 Web サイトに用いられる IP アドレスの空間的な特徴を分析する．ここでは，表 4.1 に示した訓練データセットに含まれる IP アドレスを分析対象する．今回は良性と悪性 Web サイトのそれぞれの IP アドレスをヒルベルト曲線に基づく 2 次元グラフ上に配置することで，悪性 Web サイトに使われる IP アドレスの空間的な局所性を示す．ヒルベルト曲線とは，再帰的に定義される空間充填曲線のうちの一つである [41]．この曲線を用いることで，IP アドレスの近接性を維持したまま IPv4 アドレス空間を 2 次元グラフとして視覚化できる [30, 31, 40, 42]．IP アドレスをヒルベルト曲線上に配置していく例を図 4.2 に示す．図 4.2 は，IP アドレス空間の 0.0.0.0/24 から 0.0.16.0/24 までをヒルベルト曲線にしたがって配置する例を示しており，一つの四角は一つの/24 のネットワークブロックを表している．

図 4.3 は，表 4.1 の訓練データセットの IP アドレスのうちあるネットワークアドレスブロック A.B.0.0/16 に含まれるものをヒルベルト曲線を用いて視覚化したものである．なお，第 1・第 2 オクテットはセキュリティ上の理由によりそれぞれ A・B とマスク処理をおこなっている．ここで，灰色の四角は良性 Web サイトの IP アドレスブロック，黒色の四角は悪性 Web サイトの IP アドレスブロックを表す．図 4.3 より，悪性 Web サイトに使われている IP アドレスが，特定のネットワークアドレスブロックに偏っていることが視覚的に確認できる．3.2.1 節で示した特徴抽出エンジン (IP アドレス分析) では，この特徴を活用した特徴ベクトル抽出をおこなう．

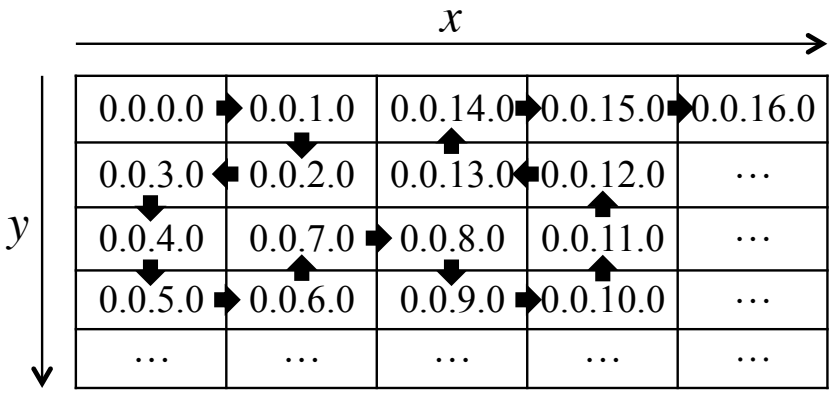

図 4.2: IP アドレスのヒルベルト曲線上への配置例

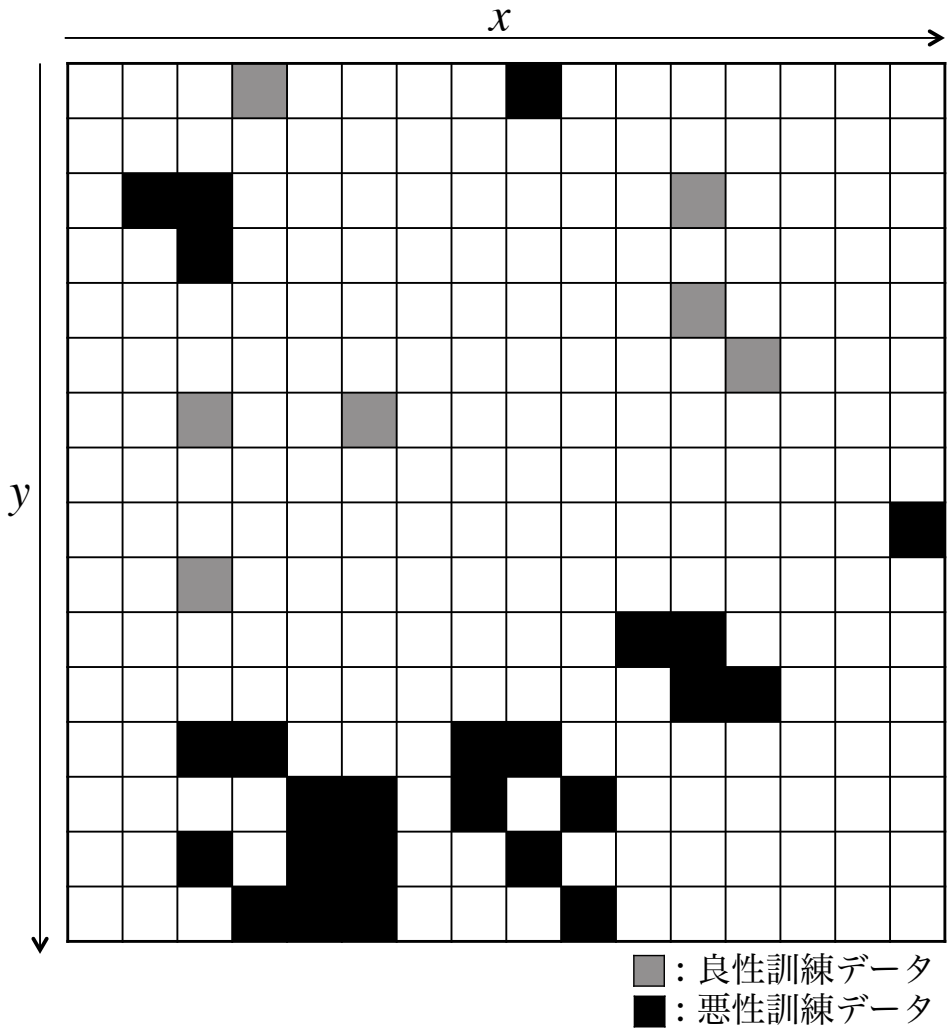

図 4.3: IP アドレス空間 (A.B.0.0/16) の可視化

4.3　WHOIS 情報の特徴

表 4.1 に示した訓練データセットに含まれる良性と悪性 Web サイトのドメイン名から WHOIS 情報を取得し，そのドメイン登録日を調査した．図 4.4 は良性と悪性ドメイン名それぞれの登録期間の累積分布 (CDF) である．ここで，図 4.4 の横軸は，3.2.2 節で定義した各ドメイン名が登録されてからの登録期間 W であり，短いほど新しいドメインである．関連研究 [22, 23, 25] でも指摘されているとおり，悪性 Web サイトのドメイン登録日は，良性 Web サイトに比べて新しいものが多いことが確認できた．この特徴を用いて，3.2.2 節で示した特徴抽出エンジン (WHOIS 情報分析) では，ドメイン登録日が新しいほど高い悪性度が付与されるよう特徴抽出をおこなう．ただし，古いドメイン登録日のものがすべて良性であるとは限らない．例えば，co.cc ドメインは 1997 年 10 月に取得されたドメインであるが，無料のサブドメインや URL 転送のサービスに利用されており，悪性 Web サイトの温床となっている [43]．

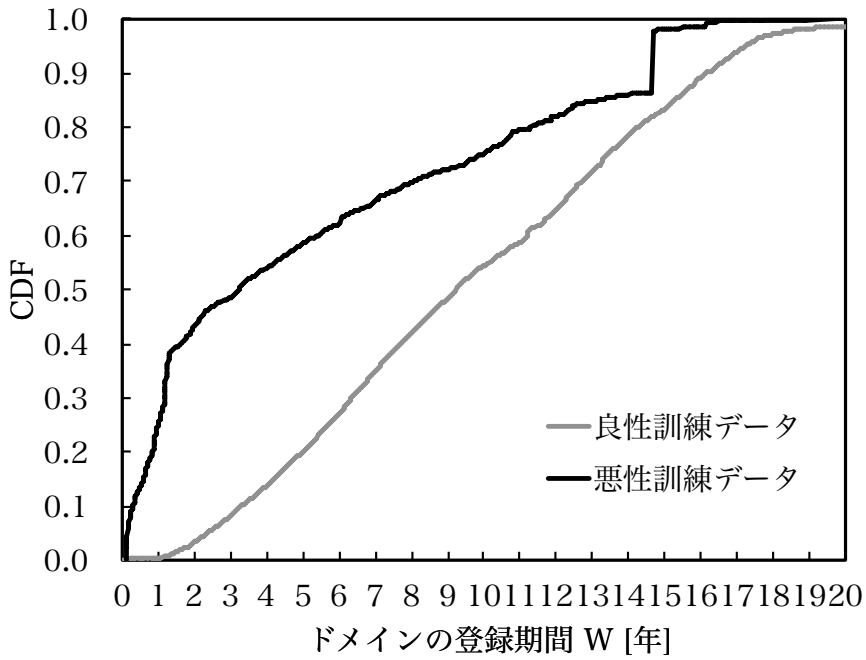

図 4.4: ドメイン登録日の累積分布

4.4　FQDN 文字列の特徴

4.4.1　文字列の長さの比較

表 4.1 に示した訓練データセットに含まれる良性と悪性 Web サイトの FQDN 文字列の長さを比較し，表 4.3 にその基本統計量を示す．また，図 4.5 にその累積補分布 (Complementary Cumulative Distribution Function，CCDF) を示す．表 4.3 と図 4.5 より，FQDN 文字列の長さに関して (1) 悪性 Web サイトの FQDN 文字列の長さは良性 Web サイトよりも広い範囲に分散している，(2) 極めて長い FQDN 文字列の場合には悪性 Web サイトの可能性が高い，ということがわかる．3.2.3 節に示した特徴抽出エンジン (FQDN 文字列分析) では，本節で示した FQDN 文字列の長さの性質を利用した特徴抽出をおこなう．

表 4.3: 文字列の長さの基本統計量

	良性訓練データ	悪性訓練データ
最小	8.0	5.0
最大	35.0	122.0
平均	16.0	16.9
標準偏差	3.5	7.2
分散	12.5	51.7

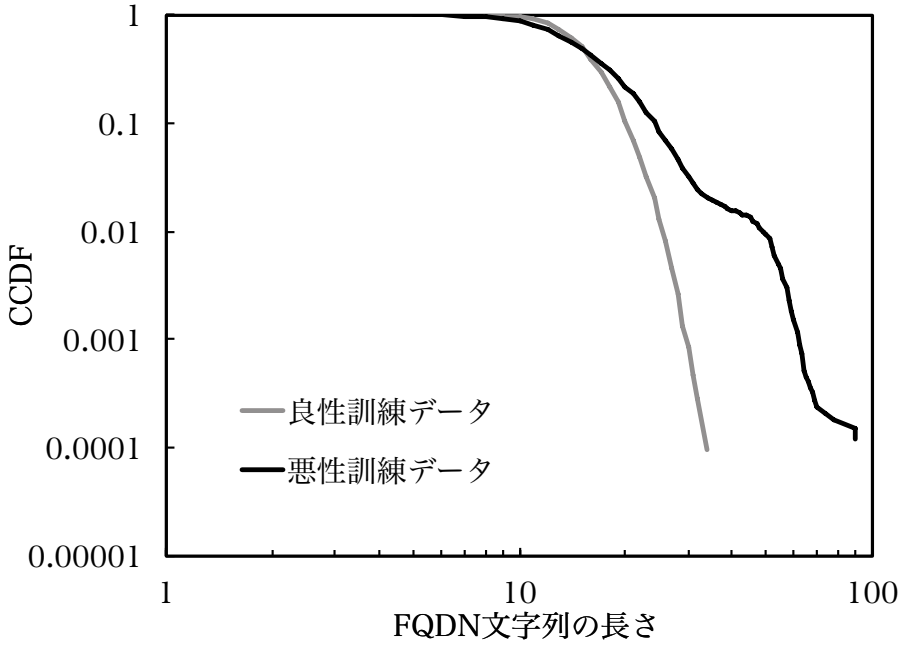

図 4.5: FQDN 文字列の長さの累積補分布

4.4.2　文字列のエントロピーの比較

表 4.1 に示した訓練データセットに含まれる良性と悪性 Web サイトの FQDN 文字列のエントロピーを比較し，表 4.4 に基本統計量，図 4.6 にその累積補分布 (CCDF) を示す．なお，本研究における FQDN 文字列のエントロピーは 3.2.3 において定義している．表 4.4 と図 4.6 より，FQDN 文字列のエントロピーに関して，(1) 悪性 Web サイトの FQDN 文字列のエントロピーは良性 Web サイトよりも広い範囲に分散している，(2) 悪性 Web サイトの FQDN 文字列のエントロピーの最大値は良性 Web サイトよりも大きい，ということがわかる．3.2.3 節に示した特徴抽出エンジン (FQDN 文字列分析) では，この FQDN 文字列のエントロピーの特徴を考慮した特徴抽出をおこなう．

表 4.4: 文字列のエントロピーの基本統計量

	良性訓練データ	悪性訓練データ
最小	1.34	1.05
最大	4.18	4.56
平均	3.18	3.37
標準偏差	0.32	0.36
分散	0.10	0.13

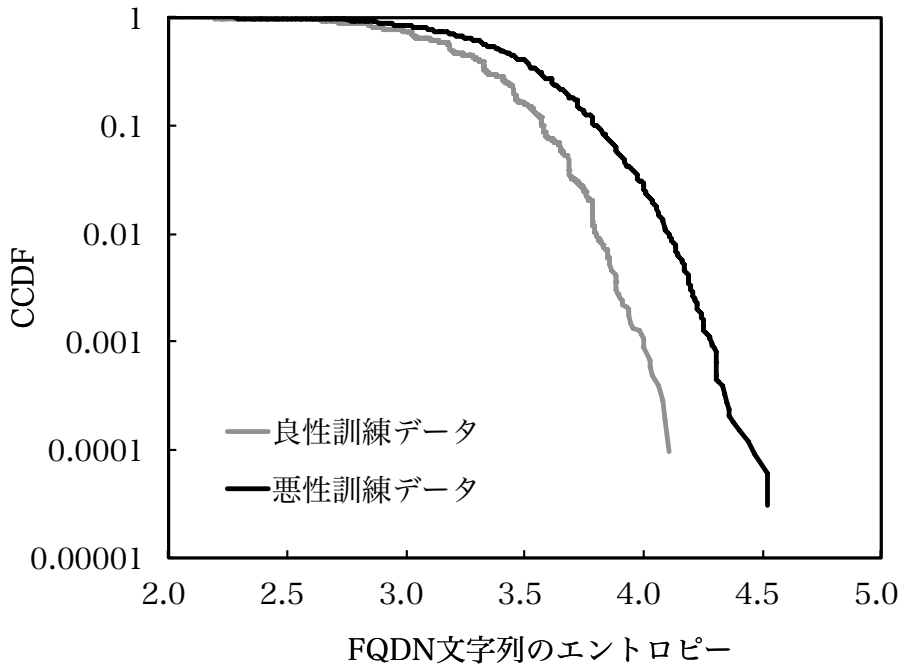

図 4.6: FQDN 文字列のエントロピーの累積補分布

4.4.3　文字列の n-gram の比較

表 4.1 に示した訓練データセットに含まれる良性と悪性 Web サイトの FQDN 文字列の n-gram ($n = 2$) を調査した．その結果，n-gram 文字列のうち，少なくとも 1 文字が数字あるいは記号で構成されているものに，良性と悪性を識別し得る特徴があることがわかった．図 4.7 に出現頻度が上位 30 位までの n-gram の頻度分布を示す．図 4.7 より，良性と悪性のそれぞれで特徴的な n-gram 文字列が存在し，それらの出現頻度に差があることがわかる．これは，良性と悪性では FQDN に使用される文字列が異なるということを示しており，3.2.3 節で説明した特徴抽出エンジン (FQDN 文字列分析) では，この特徴を利用した特徴ベクトル抽出をおこなう．

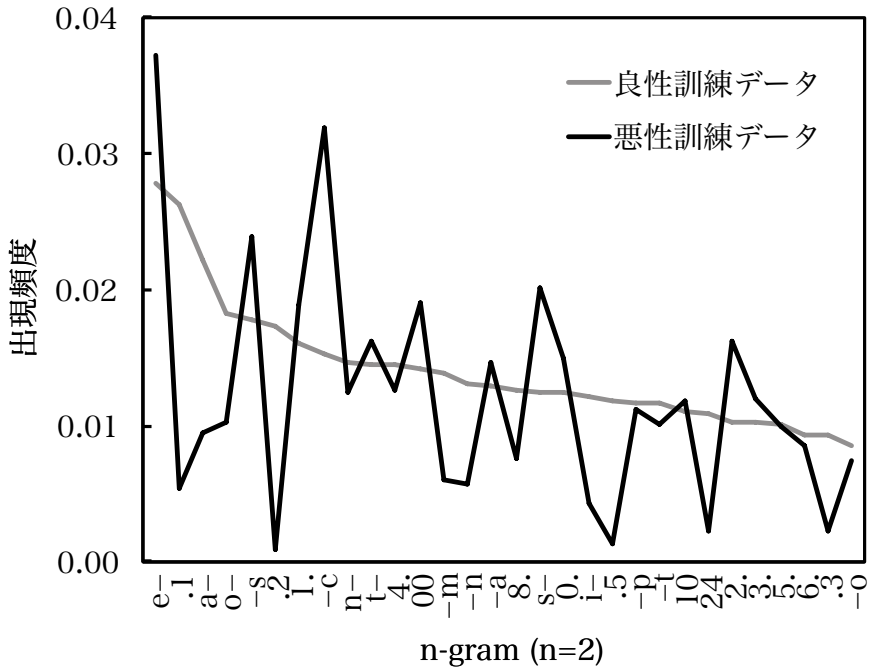

図 4.7: FQDN 文字列の n-gram の頻度分布

第 5 章

提案手法の性能評価

本章では，3 章で説明した提案手法 (巡回順序決定システム) の性能を実データを用いて評価し，提案手法の既存手法に対する優位性を示す．まず，5.1 節で評価のために用意したテストデータセットについて説明する．次に，5.2 節で提案手法による悪性 Web サイトのヒット率，5.3 節で提案手法を用いた Web サイトの総巡回時間を評価する．また，5.4 節では本手法によって発生するエラーの原因を考察する．そして，5.5 節では特徴選択を行う際の提案手法の性能を評価し，5.6 節では提案手法の時間経過にともなう性能変化を分析する．

5.1 テストデータセット

提案手法を評価するために作成したテストデータセットについて説明する．テストデータセットは，3.1 節で示した手順 3 と手順 4 において巡回 URL リストとして利用する．今回は提案手法の有効性を示すために，実際にユーザに閲覧されている良性 Web サイトと，評価時点で未知であった悪性 Web サイトが混在するテストデータセットを用意する．テストデータセットの内訳を表 5.1 に示す．良性テストデータは，あるネットワークの 2 週間のトラヒックデータを用いて収集された 96,597 件 (重複なし) である．この良性テストデータに悪性 Web サイトが混入している可能性を排除するため，Google Safe Browsing API [17] を用いたチェックをおこない，悪性 Web サイトの可能性がある 2,515 件を除外した．一方，悪性テストデータとしては，訓練データにも利用した Malware Domain List (MDL) [39] から 10,561 件 (重複なし) を利用する．ただし，悪性テストデータには，2011 年 5 月 1 日〜2012 月 4 月 18 日の 353 日間に新たに登場した悪性 Web サイトを利用している．さらに，悪性テストデータからは既存のブラックリストで防御可能な悪性 Web サイトをすべて除去している．したがって，このテスト

データセットを用いることで Web 空間の中に存在する未知の悪性 Web サイトに対する評価をおこなうことができる．

表 5.1: テストデータセット

データ	収集期間	Web サイト数
良性テストデータ	2011/5/1〜2011/5/14	96,567
悪性テストデータ	2011/5/1〜2012/4/18	10,561
合計		107,128

5.2　ヒット率

提案手法によって巡回順序を決定する際の，悪性 Web サイトのヒット率を計測する．ここで，提案手法 (巡回順序決定システム) の特徴抽出エンジンで利用する特徴量の組合せを表 5.2 に示す．特徴 A では，特徴抽出エンジンから得られる特徴量のうち IP アドレスのみを利用する．特徴 B では，特徴抽出エンジンで得られるすべての特徴量 (IP アドレス，WHOIS ドメイン登録日，FQDN 文字列の長さ，FQDN 文字列のエントロピー，FQDN 文字列の n-gram) を利用する．特徴 C では，3.2.3 節で解説した FQDN 文字列の n-gram における例外処理を適用する．すなわち，例外ドメイン名に一致した FQDN からは，n-gram の特徴を抽出しない．特徴 D では，IP アドレス以外のすべての特徴量を利用する．

表 5.2: 特徴抽出エンジンの組合せ

特徴抽出エンジン	特徴 A	特徴 B	特徴 C	特徴 D
IP アドレス	✓	✓	✓	-
WHOIS ドメイン登録日	-	✓	✓	✓
FQDN 文字列の長さ	-	✓	✓	✓
FQDN 文字列のエントロピー	-	✓	✓	✓
FQDN 文字列の n-gram	-	✓	✓	✓
FQDN 文字列の n-gram 例外処理	-	-	✓	✓

既存手法を用いてランダムに巡回する場合 (既存) と，提案手法を用いて巡回順序を決定してから巡回する場合 (特徴 A〜特徴 D) の悪性 Web サイトのヒット率の比較をおこない，その結果を表 5.3 に示す．ここで，ある長さの巡回 URL リストを選択した際に，そのリスト中に実際に含まれていた悪性 Web サイト数の割合をヒット率と定義する．既存手法のヒット率が約 10%となるのは，表 5.1 に示したテストデータセットに含まれる悪性 Web サイトの割合が約 10%となっているからである．提案手法 (特徴 A〜特徴 D) を用いる場合は， 3.3.2 節で算出した悪性度をもとに悪性度が高い URL から巡回する．表 5.3 より，巡回 URL リスト長が 5,000 までは特徴 A のヒット率が最も大きい．巡回 URL リスト長が 10,000 を超えたあとは，特徴 B と特徴 C のヒット率が特徴 A よりも大きくなる．これは，巡回 URL リスト長が小さいときには，特徴 B と特徴 C で良性 Web サイトに誤って高い悪性度を付与するエラーの影響が無視できないためである．このエラーについては，5.4 節で詳しく解説する．特徴 D の場合は，特徴 A〜特徴 C よりヒット率が低い．特徴 D は，唯一 IP アドレスの特徴を利用していない手法であり，この結果より IP アドレスがほかの特徴に比べて有効な特徴量であることがわかる．この実験結果より，提案手法によって巡回順序を決定することで，悪性 Web サイトへのヒット率を高めることができることが確認できた．

表 5.3: 悪性 Web サイトのヒット率

巡回 URL リスト長	既存	特徴 A	特徴 B	特徴 C	特徴 D
1,000	10%	100%	69%	94%	54%
5,000	10%	83%	71%	82%	32%
10,000	10%	56%	58%	63%	33%
20,000	10%	40%	41%	43%	32%
30,000	10%	30%	31%	31%	26%
40,000	10%	24%	24%	24%	21%
50,000	10%	20%	20%	20%	18%
60,000	10%	17%	17%	17%	16%
70,000	10%	15%	15%	15%	14%
80,000	10%	13%	13%	13%	13%
90,000	10%	12%	12%	12%	11%
100,000	10%	10%	11%	11%	10%

5.3　総巡回時間

既存手法を用いてランダムに巡回する場合 (既存) と，提案手法を用いて巡回順序を決定してから巡回する場合 (特徴 A〜特徴 D) の総巡回時間を比較する．総巡回時間とは，ある特定数の悪性 Web サイトを発見するまでにかかるすべての所要時間のことであり，短い方がより性能が良い．既存手法における総巡回時間は，Web クライアント型ハニーポットによる巡回時間のみとなる．一方，提案手法における総巡回時間は，巡回順序決定システムにおける所要時間と Web クライアント型ハニーポットによる巡回時間の和となる．

巡回順序決定システムにおける所要時間を表 5.4 に示す．ここで，特徴次元数とは表 5.2 で示した特徴抽出エンジンで利用する特徴量の組合せ (特徴 A〜特徴 D) で利用する特徴量の数である．また，訓練モデル生成時間とは 3.1 節で説明した手順 1 と手順 2 にかかる時間，巡回順序決定時間は手順 3 と手順 4 にかかる時間，合計所要時間は手順 1 から手順 4 までにかかる時間のことである．表 5.4 より特徴 A〜特徴 D のすべての場合において，巡回順序決定時間は，訓練モデル生成時間よりも短いことがわかる．また，合計所要時間は特徴次元数に比例し，特徴次元数が最も多い特徴 B・特徴 C の合計所要時間が最も長いことが確認できた．なお，特徴次元数と所要時間の関係については， 5.5 節でさらに詳しく分析する．なお，巡回順序決定システムを動作させるマシンは，CPU: Intel Xeon X3430 (2.40GHz)，メモリ: 4GB の Linux サーバである．

Web クライアント型ハニーポットによる巡回時間としては，文献 [29] で提案されているマルチエージェント・マルチプロセス化された Web クライアント型ハニーポットの実験結果を参照する．具体的には，図 3.1 の右側に示したエージェントが 10 個用意され，各エージェント中で 20 個の Web ブラウザのプロセスが並列に動作している環境を想定する．この場合，Web クライアント型ハニーポットは 1,000 個の URL を 600 秒で巡回できると仮定できる．

表 5.4: 巡回順序決定システムの所要時間

	特徴 A	特徴 B	特徴 C	特徴 D
特徴次元数	1,280	1,995	1,995	715
訓練モデル生成時間 (手順 1, 手順 2)	281 s	451 s	451 s	293 s
巡回順序決定時間 (手順 3, 手順 4)	99 s	111 s	111 s	60 s
合計所要時間 (手順 1〜手順 4)	380 s	562 s	562 s	353 s

既存手法および提案手法の総巡回時間を表 5.5 に示す．表 5.5 より，提案手法 (特徴 A〜特徴 D) の方が既存手法よりも総巡回時間が短く，より効率的に悪性 Web サイトを発見することができることが確認できた．特に，悪性 Web サイト発見数が 3,000 の場合，特徴 A の総巡回時間は既存手法の 0.12 倍 ($= 2,180/18,217$) と大幅に時間を短縮できることがわかる．ただし，悪性 Web サイト発見数が 100 の時は，既存手法の方が特徴 B・特徴 C よりも総巡回時間が短い．これは，悪性 Web サイト発見数が少ない時は巡回順序決定システムにおける所要時間 (表 5.4) が総巡回時間に占める割合が大きくなるためである．

今回の評価では，悪性 Web サイト発見数が増えるほど，既存手法と提案手法の差が小さくなる．なぜなら，今回のテストデータに含まれる悪性 Web サイトは表 5.1 に示すとおり，合計 10,561 件であり，悪性 Web サイト発見数が 10,000 に近づくほど，見つけるべき悪性 Web サイトが減少するためである．

表 5.5: 総巡回時間

悪性 Web サイト発見数	既存	特徴 A	特徴 B	特徴 C	特徴 D
100	583 s	440 s	756 s	624 s	413 s
500	3,139 s	680 s	1,032 s	871 s	653 s
1,000	6,097 s	980 s	1,403 s	1,205 s	2,249 s
2,000	12,193 s	1,580 s	2,164 s	1,899 s	4,082 s
3,000	18,217 s	2,180 s	2,961 s	2,570 s	5,857 s
4,000	24,391 s	3,226 s	3,972 s	3,435 s	7,620 s
5,000	30,306 s	4,761 s	5,253 s	4,555 s	9,483 s
6,000	36,470 s	7,141 s	6,916 s	5,976 s	11,639 s
7,000	42,519 s	8,599 s	9,204 s	7,875 s	14,365 s
8,000	48,479 s	13,413 s	11,709 s	10,185 s	18,850 s
9,000	54,634 s	17,651 s	16,546 s	15,404 s	30,375 s
10,000	60,883 s	33,568 s	30,115 s	29,153 s	45,515 s

5.4　エラーの分析

提案手法によって良性 Web サイトの悪性度が誤って高く算出されるエラーが発生する場合がある．本節では，このエラーの原因を表 5.2 で示した特徴抽出エンジンで利用する特徴量の組合せ (特徴 A〜特徴 D) ごとにそれぞれ分析する．

特徴 A　特徴 A は IP アドレスのみを特徴量として用いる．したがって，良性と悪性 Web サイトが同一または近い IP アドレスを持つ場合 (e.g. ホスティングサイト) に良性 Web サイトに誤って高い悪性度を付与するエラーが発生した．

特徴 B　特徴 B では，特徴 A とは異なり IP アドレス以外に，WHOIS 情報や FQDN 文字列からも特徴を抽出するため，特徴 A で発生したエラーの多くを除去できる．一方，ランダム性の高い FQDN 文字列を持つ良性 Web サイトの悪性度を高く付与する新たなエラーが発生した．エラーとなった良性 Web サイトを分類した結果を表 5.6 に示す．例えば，ソーシャルアプリケーションやブラウザの拡張機能におけるコンテンツの識別子として利用されるものや国際化ドメイン名 [44] に利用される Punnycode にランダム性の高い文字列が存在し，いずれも悪性 Web サイトの FQDN に含まれる文字列と類似していることから，エラーが発生するということがわかった．

表 5.6: 特徴 B でエラーとなる良性 Web サイト

Web サイトの種類	件数
ソーシャルアプリケーション	33
一般 Web サイト	26
ホスティングサイト	20
国際化ドメイン名を持つ Web サイト	10
Web メール	4
ファイル共有サイト	2
その他	5
合計	100

特徴 C　特徴 C では， 3.2.3 節で説明した例外ドメイン名による例外処理の追加をおこなうことにより，特徴 B で新たに発生したエラーが減少した．この結果，5.2 節のヒット率，5.3 節の

総巡回時間の両方で，特徴 C は特徴 B に比べて性能が向上することが確認できた．

特徴 D　特徴 D は IP アドレスの特徴量を利用しない．そのため，ドメイン登録日が比較的新しく，長い FQDN 文字列を持つ良性 Web サイトの悪性度を高く付与するエラーが発生した．また，5.2 節および 5.3 節で特徴 C と特徴 D の性能を比較した結果，IP アドレスの特徴量が提案手法の性能を大きく向上させることがわかった．

5.5　特徴選択による性能変化

提案手法で利用する各特徴量の識別能力を分析し，それをもとに特徴選択をおこなう．特徴選択とは，特徴抽出エンジンで得られた特徴量の中でより有用なものを選択して利用することである．本節では，特徴選択により提案手法の性能がどのように変化するのかを調査する．ここでは， 5.2 節で示した悪性 Web サイトのヒット率および 5.3 節で測定した総巡回時間において最も良い結果が得られた特徴 C を選択して評価を進める．なお，この分析ではこれまでと同様に訓練データセットとして表 4.1，テストデータセットとして表 5.1 に示したデータをそれぞれ利用する．

5.5.1　F-score に基づく特徴量の順位

提案手法で利用するすべての特徴量に対して，それぞれの識別能力を F-score (Fisher score) を用いて算出する．F-score とは，特徴量の識別能力を表す統計的な評価基準 [45, 46] であり，k 個の訓練データ x_i $(i = 1, \cdots, k)$ があるとき，l 個の特徴量の中の j 番目の特徴量 $(j = 1, \cdots, l)$ の F-score は次の式で定義される．

$$F(j) \equiv \frac{(\overline{b}_j - \overline{x}_j)^2 + (\overline{m}_j - \overline{x}_j)^2}{\frac{1}{n_b - 1}\sum_{i=1}^{n_b}(b_{i,j} - \overline{b}_j)^2 + \frac{1}{n_m - 1}\sum_{i=1}^{n_m}(m_{i,j} - \overline{m}_j)^2}$$

ここで，n_b と n_m はそれぞれ良性訓練データと悪性の訓練データの個数，$\overline{x}_j$，$\overline{b}_j$，$\overline{m}_j$ はそれぞれ全訓練データ，良性訓練データ，悪性訓練データの j 番目の特徴量の平均値，$b_{i,j}$ と $m_{i,j}$ はそれぞれ i 個目の良性と悪性訓練データの j 番目の特徴量を意味する．$F(j)$ の分子は良性と悪性の群間の平均平方を表し，$F(j)$ の分母は良性と悪性それぞれの群内の平均平方を表している．F-score の数値が大きいほど，その特徴量による識別能力が高いことを示す．本研究では F-score の数値が大きい順に 1 から l までの順位をつける．

提案手法 (特徴 C) で利用するすべての特徴量 (特徴次元数: 1,995 次元) に対して F-score を算出し，順位をつけた結果を表 5.7 に示す．ただし，表 5.7 では IP アドレスの各オクテットに関する数値と FQDN 文字列の内容はセキュリティ上の理由によりマスク処理を施している．また，今回は紙面の都合により上位 25 件のみを表示している．

表 5.7 より，WHOIS 情報のドメイン登録日の識別能力が高いことがわかる．これは 4.3 節で示したとおり，悪性 Web サイトのドメイン登録日が良性 Web サイトに比べて新しい日付に偏っているためである．また，IP アドレスの特徴量は上位 25 件中 22 件を占め非常に有効な特徴であることがわかる．特に上位オクテット (第 1〜第 2 オクテット) の特徴量は， 4.2.2 節で示した空間的局所性に大きく関係するため，より大きな F-score となっている．一方，FQDN 文字列の特徴量のうち上位の順位のものは WHOIS 情報と IP アドレスに比べて少ない．これは FQDN 文字列の特徴量の絶対数が少なく，相対的に順位が低くなったためである．

5.5.2　特徴選択によるヒット率の変化

5.5.1 節で算出した特徴量の順位をもとに特徴選択をおこない，選択した特徴量の数 (特徴次元数) に応じた悪性 Web サイトのヒット率の変化を調査する．悪性 Web サイトのヒット率とは 5.2 節で定義したとおり，巡回 URL リストに実際に含まれる悪性 Web サイトの数の割合のことである．ヒット率が高いほど，その手法の性能が良いことを意味する．

選択する特徴量の数ごとに悪性 Web サイトのヒット率の計測をおこない，その結果を表 5.8 に示す．提案手法 (400 位〜1,995 位) では，特徴量の順位 1 位からそれぞれの順位までの特徴量を選択してヒット率を計測する．例えば，400 位の場合には上位 1〜400 位までの特徴量を選択する．なお，1,995 位の場合は提案手法 (特徴 C) で抽出するすべての特徴量を利用するため， 5.2 節の表 5.3 における特徴 C の結果と一致する．

表 5.8 より既存手法を用いてランダムに巡回する場合 (既存) のヒット率は 5.2 節で示したとおり約 10%となる．一方，提案手法 (400 位〜1,995 位) の場合はいずれも既存手法よりもヒット率が高い．また，表 5.8 より巡回 URL リスト長が 1,000 から 20,000 までは利用する特徴量が多いほどヒット率が増加し，1,995 位までのすべての特徴量を使う際に最もヒット率が高いことがわかる．しかし，巡回 URL リスト長が 30,000 より大きい場合には 1,200 位までの特徴量を使う際にヒット率が最も高くなることがわかった．これは，提案手法 (特徴 C) で抽出する特徴量のうち上位順位の特徴量がより有用であることを示している．

表 5.7: 各特徴量の F-score に基づく順位

順位	特徴量	F-score
1	WHOIS 情報 (ドメイン登録日)	0.112
2	IP アドレス (第 1 オクテット: 1)	0.023
3	IP アドレス (第 1 オクテット: 2)	0.021
4	IP アドレス (第 1 オクテット: 3)	0.010
5	IP アドレス (第 1 オクテット: 4)	0.007
6	IP アドレス (第 1 オクテット: 5)	0.006
7	IP アドレス (第 1 オクテット: 6)	0.006
8	IP アドレス (第 1・2 オクテット: 1)	0.006
9	IP アドレス (第 2 オクテット: 1)	0.005
10	IP アドレス (第 2 オクテット: 2)	0.005
11	FQDN 文字列 (n-gram: aa)	0.005
12	IP アドレス (第 2 オクテット: 3)	0.004
13	IP アドレス (第 1 オクテット: 7)	0.004
14	IP アドレス (第 1 オクテット: 8)	0.004
15	IP アドレス (第 1・2 オクテット: 2)	0.004
16	FQDN 文字列 (n-gram: ab)	0.004
17	IP アドレス (第 2 オクテット: 4)	0.004
18	IP アドレス (第 1・2・3 オクテット: 1)	0.004
19	IP アドレス (第 2 オクテット: 5)	0.004
20	IP アドレス (第 2 オクテット: 6)	0.003
21	IP アドレス (第 1 オクテット: 9)	0.003
22	IP アドレス (第 1・2 オクテット: 3)	0.003
23	IP アドレス (第 2 オクテット: 7)	0.003
24	IP アドレス (第 1・2 オクテット: 4)	0.003
25	IP アドレス (第 1・2 オクテット: 5)	0.003

表 5.8: 悪性 Web サイトのヒット率 (特徴選択)

巡回 URL リスト長	既存	400 位	800 位	1,200 位	1,600 位	1,995 位
1,000	10%	67%	79%	86%	86%	94%
5,000	10%	66%	70%	74%	78%	82%
10,000	10%	54%	58%	62%	61%	63%
20,000	10%	38%	41%	42%	43%	43%
30,000	10%	30%	31%	32%	31%	31%
40,000	10%	24%	25%	25%	24%	24%
50,000	10%	20%	20%	20%	20%	20%
60,000	10%	17%	17%	17%	17%	17%
70,000	10%	15%	15%	15%	15%	15%
80,000	10%	13%	13%	13%	13%	13%
90,000	10%	12%	12%	12%	12%	12%
100,000	10%	11%	11%	11%	11%	11%

5.5.3　特徴選択による総巡回時間の変化

5.5.1 節で算出した特徴量の順位をもとに特徴選択をおこない，選択した特徴量の数 (特徴次元数) に応じて総巡回時間を比較する．総巡回時間とは 5.5 節と同様に，ある特定数の悪性 Web サイトを発見するまでにかかるすべての所要時間のことである．総巡回時間が短いほど，その手法の性能が良いことを意味する．既存手法における総巡回時間は，Web クライアント型ハニーポットによる巡回時間のみとなる．一方，提案手法における総巡回時間は，巡回順序決定システムにおける所要時間とハニーポットによる巡回時間の和となる．なお，本節の実験環境は 5.3 節と同様である．

まず，巡回順序決定システムにおける所要時間を測定した結果を表 5.9 に示す．今回は特徴順位を 400 位から 1,995 位まで変更し，それぞれの場合の所要時間を測定する．なお，1,995 位の場合は提案手法 (特徴 C) で抽出するすべての特徴量を利用するため， 5.3 節の表 5.4 における特徴 C の結果と一致する．表 5.9 より，所要時間は選択する特徴量の数に比例することがわかる．また，特徴順位に基づく特徴選択により巡回順序決定システムのコストは調整可能であることが示された．

表 5.9: 巡回順序決定システムの所要時間 (特徴選択)

特徴順位	400 位	800 位	1,200 位	1,600 位	1,995 位
所要時間	318 s	344 s	386 s	513 s	562 s

次に，既存手法を用いてランダムに巡回する場合 (既存) および提案手法 (400 位〜1,995 位) における総巡回時間を表 5.10 に示す．悪性 Web サイト発見数が 100 の場合以外は，提案手法 (400 位〜1,995 位) の総巡回時間が既存手法に比べて大幅に短いことがわかる．提案手法の間で総巡回時間を比較すると，悪性 Web サイト発見数が 500 までは特徴順位が 400 位までの特徴量を使った場合が最も総巡回時間が短い．これは表 5.9 に示した巡回順序決定システムにおける所要時間の影響が大きいためである．悪性 Web サイト発見数が 1,000 の場合は，1,200 位の総巡回時間が最も短くなり，その後悪性 Web サイト発見数が 2,000 から 8,000 の間は 1,995 位の総巡回時間が最も短い．一方，悪性 Web サイト発見数が 9,000 と 10,000 の時は，1,200 位の総巡回時間が最も短くなる．これは悪性 Web サイト発見数が 1,000 以降は，巡回順序決定システムにおける所要時間の影響は小さくなり，その代わりに 5.5.2 節で示した悪性 Web サイトのヒット率が強く影響するためである．

表 5.10: 総巡回時間 (特徴選択)

悪性 Web サイト発見数	既存	400 位	800 位	1,200 位	1,600 位	1,995 位
100	583 s	387 s	411 s	452 s	576 s	624 s
500	3,139 s	710 s	724 s	712 s	842 s	871 s
1,000	6,097 s	1,274 s	1,137 s	1,105 s	1,209 s	1,205 s
2,000	12,193 s	2,144 s	1,951 s	1,927 s	1,963 s	1,899 s
3,000	18,217 s	2,993 s	2,810 s	2,773 s	2,697 s	2,570 s
4,000	24,391 s	4,239 s	3,913 s	3,612 s	3,586 s	3,435 s
5,000	30,306 s	5,599 s	4,842 s	4,635 s	4,708 s	4,555 s
6,000	36,470 s	7,470 s	6,723 s	6,117 s	6,370 s	5,976 s
7,000	42,519 s	10,512 s	8,639 s	8,125 s	8,663 s	7,875 s
8,000	48,479 s	13,831 s	11,340 s	10,896 s	11,020 s	10,185 s
9,000	54,634 s	18,173 s	16,427 s	14,703 s	14,959 s	15,404 s
10,000	60,883 s	31,567 s	29,184 s	28,335 s	29,793 s	29,153 s

5.6　時間経過による性能変化

本節では，提案手法の時間経過にともなう性能変化を評価する．ここでは，5.2 節で示した悪性 Web サイトのヒット率および 5.3 節で測定した総巡回時間において最も良い結果が得られた特徴 C を選択して評価を進める．

5.6.1　時間経過評価用のデータセット

時間経過に応じた性能評価をおこなうため，本節では新たに訓練データセットとテストデータセットを用意する．本節で利用する訓練データセットを表 5.11 に示す．今回の訓練データセットの収集は 4.1 節と同様におこなうが，収集期間を変化させて 5 種類 (訓練 1〜訓練 5) 用意する．また，テストデータセットの内訳を表 5.12 に示す．このテストデータセットは 5.1 節と同様に作成する．ただし，悪性テストデータとしては，訓練データセット (訓練 1〜訓練 5) の収集期間より後の 2012 年 1 月 1 日〜2012 年 4 月 18 日の 108 日間に新たに登場した悪性 Web サイトを選択する．すなわち，訓練データセット (訓練 1〜訓練 5) とこのテストデータセットを利用することで，未知の悪性 Web サイトに対する評価をおこなうことができる．

表 5.11: 訓練データセット (時間経過)

データ	収集期間	Web サイト数
訓練 1	2009/1/1〜2011/4/30 (28ヶ月)	45,810
訓練 2	2009/1/1〜2011/6/30 (30ヶ月)	47,444
訓練 3	2009/1/1〜2011/8/31 (32ヶ月)	50,888
訓練 4	2009/1/1〜2011/10/31 (34ヶ月)	52,675
訓練 5	2009/1/1〜2011/12/31 (36ヶ月)	54,386

表 5.12: テストデータセット (時間経過)

データ	収集期間	Web サイト数
良性テストデータ	2011/5/1〜2011/5/14	96,567
悪性テストデータ	2012/1/1〜2012/4/18	1,973
合計		98,540

5.6.2　時間経過によるヒット率の変化

時間経過にともなう悪性 Web サイトのヒット率の変化を調査する．悪性 Web サイトのヒット率とは 5.2 節で説明したとおり，巡回 URL リストに実際に含まれる悪性 Web サイトの数の割合のことである．ヒット率が高いほど，その手法の性能が良いことを意味する．まず，表 5.11 に示した訓練 1〜訓練 5 を用いて，それぞれの訓練データに対応する訓練モデルを作成する．次に，各訓練モデルごとに表 5.12 のテストデータセット (巡回 URL リスト) における悪性 Web サイトのヒット率の計測をおこない，その結果を表 5.13 に示す．

既存手法を用いてランダムに巡回する場合 (既存) のヒット率は約 2%となり，提案手法 (訓練 1〜訓練 5) のいずれの場合よりも低い．ここで，既存手法におけるヒット率が約 2%となるのは，今回のテストデータセットに含まれる悪性テストデータの割合が約 2%となるからである．

提案手法 (訓練 1〜訓練 5) のヒット率を比較する．表 5.11 より訓練 1 から訓練 5 にかけて時間が経過し，収集期間が増加する．表 5.13 のどの巡回 URL リスト長の場合でも，訓練 1 から訓練 5 へと収集期間が増加すればするほど，ヒット率が上昇することが確認できる．これは提案手法の性能が時間経過に伴い向上することを意味しており，提案手法は日々変化し続ける悪性 Web サイトへ対応できる可能性が高い．

表 5.13: 悪性 Web サイトのヒット率 (時間経過)

巡回 URL リスト長	既存	訓練 1	訓練 2	訓練 3	訓練 4	訓練 5
1,000	2%	49%	53%	58%	59%	67%
2,000	2%	36%	39%	44%	46%	49%
3,000	2%	29%	33%	36%	38%	42%
4,000	2%	26%	28%	31%	32%	34%
5,000	2%	23%	25%	27%	28%	29%
6,000	2%	21%	23%	24%	25%	26%
7,000	2%	19%	20%	22%	22%	23%
8,000	2%	17%	19%	20%	20%	21%
9,000	2%	16%	17%	18%	18%	19%
10,000	2%	15%	16%	17%	17%	17%

5.6.3　時間経過による総巡回時間の変化

時間経過にともなう総巡回時間の変化を調査する．総巡回時間とは 5.5 節で定義したとおり，ある特定数の悪性 Web サイトを発見するまでにかかるすべての所要時間のことである．総巡回時間が短いほど，その手法の性能が良いことを意味する． 5.6.2 節と同様に，表 5.11 に示した訓練 1〜訓練 5 に対応する訓練モデルを作成し，それぞれの訓練モデルごとに表 5.12 のテストデータセット (巡回 URL リスト) を用いた評価をおこなう．

総巡回時間を計測した結果を表 5.14 に示す．ここで，既存手法の総巡回時間はテストデータセット (巡回 URL リスト) を用いて Web クライアント型ハニーポットがランダムに巡回する時間である．一方，提案手法 (訓練 1〜訓練 5) の総巡回時間は，巡回順序決定システムにおける所要時間と Web クライアント型ハニーポットによる巡回時間の和である．

表 5.14 より提案手法 (訓練 1〜訓練 5) の総巡回時間は既存手法よりも大幅に短いことがわかる．また，表 5.14 より提案手法 (訓練 1〜訓練 5) では，訓練 1 から訓練 5 にかけて収集期間が増加するほど総巡回時間が短くなることが確認できた．これは 5.6.2 節に示したとおり，巡回順序決定システムによって悪性 Web サイトのヒット率が高まり効率的に巡回できるためである．

表 5.14: 総巡回時間 (時間経過)

悪性 Web サイト発見数	既存	訓練 1	訓練 2	訓練 3	訓練 4	訓練 5
100	3,332 s	637 s	631 s	630 s	630 s	627 s
300	8,933 s	874 s	848 s	813 s	809 s	780 s
500	14,843 s	1,180 s	1,119 s	1,059 s	1,031 s	985 s
700	20,651 s	1,707 s	1,503 s	1,346 s	1,305 s	1,201 s
900	26,966 s	2,411 s	2,106 s	1,821 s	1,723 s	1,537 s
1,100	32,781 s	3,238 s	2,823 s	2,454 s	2,260 s	2,004 s
1,300	38,423 s	4,653 s	3,810 s	3,238 s	3,028 s	2,579 s
1,500	44,723 s	6,458 s	5,273 s	4,500 s	4,272 s	3,781 s
1,700	51,508 s	10,988 s	8,259 s	7,132 s	6,766 s	6,177 s
1,900	57,021 s	25,399 s	21,183 s	19,136 s	18,177 s	16,096 s

第 6 章

結論

6.1 まとめ

本研究では，限られたリソースしか持たない Web クライアント型ハニーポットがより効率的に悪性 Web サイトを発見するための手法を提案した．具体的には，まず Web サイトの IP アドレス，WHOIS 情報，FQDN 文字列から良性と悪性 Web サイトを識別し得る特徴を抽出する．次に抽出した特徴を用いて機械学習を適用し，未知の Web サイトの悪性度を推定する．悪性度が高いと推定される Web サイトから順番に巡回することで，より効率的に未知の悪性 Web サイトを発見することができる．

実データを用いた性能評価の結果，提案手法が未知の悪性 Web サイトへのヒット率を高め，Web クライアント型ハニーポットの総巡回時間を大幅に削減できることがわかった．また，利用する特徴量をその識別能力に応じて選択することで，提案手法における所要時間を調整することができることを示した．さらに，提案手法で作成した訓練モデルは時間が経過しても未知の悪性 Web サイトを発見するのに有効であることがわかった．

6.2 今後の課題

本研究で残された課題は，以下に示す特徴抽出エンジンの拡張，機械学習エンジンの改善，そして実運用における評価である．

6.2.1　特徴抽出エンジンの拡張

本研究で提案した特徴抽出エンジンでは，Web サイトの IP アドレス，WHOIS 情報，FQDN 文字列のみを用いた．特徴抽出エンジンは機械学習エンジンや Web クライアント型ハニーポットとは独立しているため，任意の情報を新たな特徴として追加可能である．新たな特徴を利用することで，提案手法の性能がさらに向上する可能性がある．

例えば，Internet Protocol version 6 (IPv6) アドレス情報は新たな特徴の候補のうちの一つである．本論文の執筆時点では，IPv6 アドレスを用いた悪性 Web サイトの情報が少ないために十分な分析をおこなうことができなかった．今後 IPv6 アドレスを用いる悪性 Web サイトが多く登場する可能性があるため，IPv6 アドレスに特化した特徴抽出手法を提案する予定である．

6.2.2　機械学習エンジンの改善

本研究では機械学習エンジンとして SVM を採用した．機械学習エンジンは特徴抽出エンジンや Web クライアント型ハニーポットとは処理が独立しており，SVM を別の機械学習手法に容易に置き換えることができる．学習における精度やコストの観点でほかの機械学習手法との比較をおこない，最適な手法を選択することで，機械学習エンジンのさらなる性能改善が期待できる．

6.2.3　実運用における評価

本研究で提案した巡回順序決定システムを，実運用されている Web クライアント型ハニーポットへ導入して評価をおこなう必要がある．実運用をおこなうことでさらなる課題を明らかにし，その課題を解決することで，より実用的なシステムとして仕上げたい．

謝辞

本修士論文の作成にあたり，日ごろよりご指導をいただいた早稲田大学基幹理工学研究科の後藤滋樹教授に深く感謝いたします．研究活動を進めるにあたり，多くのご助言をいただいた日本電信電話株式会社 NTT ネットワーク基盤技術研究所の森達哉氏に心より感謝いたします．また，セキュリティ分野の研究やマルウェア解析競技大会への参加を通じて日ごろから議論を進めた高田雄太氏に感謝いたします．

最後に，ともに研究をおこなった後藤滋樹研究室の諸氏に感謝いたします．

参考文献

[1] 井上 大介, 中尾 康二, “マルウェアって? (特集マルウェア) ,” 情報処理, vol.51, no.3, pp.237–243, March 2010.

[2] M. Akiyama, T. Yagi, and M. Itoh, “Searching Structural Neighborhood of Malicious URLs to Improve Blacklisting,” Proc. the 11th IEEE/IPSJ International Symposium on Applications and the Internet (SAINT 2011), pp.1–10, Munich, Germany, July 2011.

[3] 八木 毅, “マルウェア感染を検知・制御するブラックリストシステムの設計,” 電子情報通信学会技術研究報告 信学技報, vol.112, no.29, pp.49–54, May 2012.

[4] 針生 剛男, 秋山 満昭, 青木 一史, 八木 毅, 岩村 誠, 倉上 弘, “進化するマルウェア等によるサイバー攻撃の検知・解析・対策技術 (特集 進化する脅威とこれからのサイバーセキュリティ),” NTT 技術ジャーナル, vol.24, no.8, pp.13–17, August 2012.

[5] 笠間 貴弘, 井上 大介, 衛藤 将史, 中里 純二, 中尾 康二, “ドライブ・バイ・ダウンロード攻撃対策フレームワークの提案,” 情報処理学会コンピュータセキュリティシンポジウム (CSS 2011), vol.2011, no.3, pp.780–785, October 2011.

[6] 秋山 満昭, 佐藤 一道, 岩村 誠, 伊藤 光恭, “Gumblar の長期観測による分析,” 電子情報通信学会技術研究報告 情報通信システムセキュリティ (ICSS), vol.110, no.79, pp.69–74, June 2010.

[7] S. Vaknin, “How to avoid, remove Facebook malware,” `http://howto.cnet.com/8301-11310_39-20070931-285/how-to-avoid-remove-facebook-malware/`

[8] M. Akiyama, M. Iwamura, Y. Kawakoya, K. Aoki, and M. Itoh, “Design and Implementation of High Interaction Client Honeypot for Drive-by-download Attacks,” IEICE Transactions on Communications, vol.E93-B, no.5, pp.1131–1139, May 2010.

[9] J. Levine, "Request for Comments 5782: DNS Blacklists and Whitelists," IETF, http://www.ietf.org/rfc/rfc5782.txt, February 2010.

[10] URIBL, http://www.uribl.com/

[11] OpenDNS, http://www.opendns.com/

[12] Internet Explorer - Microsoft Windows, http://windows.microsoft.com/en-US/internet-explorer/download-ie

[13] SmartScreen Filter, http://windows.microsoft.com/en-US/internet-explorer/products/ie-9/features/smartscreen-filter/

[14] Apple - Safari, http://www.apple.com/safari/

[15] Chrome Browser - Google, http://www.google.com/chrome/

[16] Mozilla Firefox Web Browser, http://www.mozilla.org/firefox/

[17] Google Safe Browsing API, http://code.google.com/intl/en/apis/safebrowsing/

[18] M. A. Rajab, L. Ballard, N. Jagpal, P. Mavrommatis, D. Nojiri, N. Provos, and L. Schmidt, "Trends in Circumventing Web-malware Detection," Google, Google Technical Report, July 2011.

[19] K. Sato, K. Ishibashi, T. Toyono, and N. Miyake, "Extending Black Domain Name List by Using Co-occurrence Relation between DNS Queries," Proc. the 3rd USENIX Conference on Large-scale Exploits and Emergent Threats (LEET 2010), Pages: 8–8, San Jose, California, USA, April 2010.

[20] C. Curtsinger, B. Livshits, B. Zorn, and C. Seifert, "Zozzle: Fast and Precise In-browser Javascript Malware Detection," Proc. the 20th USENIX Security Symposium, Pages: 3–3, San Francisco, California, USA, August 2011.

[21] M. Antonakakis, R. Perdisci, D. Dagon, W. Lee and N. Feamster, "Building a Dynamic Reputation System for DNS," Proc. the 19th USENIX Security Symposium, Pages: 18–18, Washington, D.C., USA, August 2010.

[22] M. Felegyhazi, C. Kreibich, and V. Paxson, “On the Potential of Proactive Domain Blacklisting,” Proc. the 3rd USENIX Conference on Large-scale Exploits and Emergent Threats (LEET 2010), Pages: 6–6, San Jose, California, USA, April 2010.

[23] J. Ma, L. K. Saul, S. Savage and G. M. Voelker, “Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs,” Proc. the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2009), pp.1245–1254, Paris, France, June 2009.

[24] S. Yadav, A. K. K. Reddy, A. L. N. Reddy, and S. Ranjan, “Detecting Algorithmically Generated Malicious Domain Names,” Proc. the 10th ACM SIGCOMM Conference on Internet Measurement (IMC 2010), pp.48–61, Melbourne, Australia, November 2010.

[25] L. Invernizzi, S. Benvenuti, P. M. Comparetti, M. Cova, C. Kruegel, and G. Vigna, “EvilSeed: A Guided Approach to Finding Malicious Web Pages,” Proc. IEEE Symposium on Security and Privacy (SP 2012), pp.428–442, San Francisco, California, USA, May 2012.

[26] C. Seifert, I. Welch, P. Komisarczuk, “HoneyC the Low-interaction Client Honeypot,” Proc. the 2007 NZCSRCS, Waikato University, Hamilton, New Zealand, 2007.

[27] L. Lu, V. Yegneswaran, P. Porras, and W. Lee, “BLADE: An Attack-agnostic Approach for Preventing Drive-by Malware Infections,” Proc. the 17th ACM Conference on Computer and Communications Security (CCS 2010), pp.440–450, Chicago, Illinois, USA, October 2010.

[28] Capture-HPC, `https://projects.honeynet.org/capture-hpc/`

[29] M. Akiyama, Y. Kawakoya, and T. Hariu, “Scalable and Performance-efficient Client Honeypot on High Interaction System,” Proc. the 12th IEEE/IPSJ International Symposium on Applications and the Internet (SAINT 2012), pp.40–50, Izmir, Turkey, July 2012.

[30] D. Chiba, K. Tobe, T. Mori, and S. Goto, “Detecting Malicious Websites by Learning IP Address Features,” Proc. the 12th IEEE/IPSJ International Symposium on Applications and the Internet (SAINT 2012), pp.29–39, Izmir, Turkey, July 2012.

[31] S. Hao, N. A. Syed, N. Feamster, A. G. Gray, and S. Krassser, “Detecting Spammers with SNARE: Spatio-temporal Network-level Automatic Reputation Engine,” Proc. the 18th USENIX Security Symposium, pp.101–118, Montreal, Canada, August 2009.

[32] M. P. Collins, T. J. Shimeall, S. Faber, J. Janies, R. Weaver, M. De Shon, and J. Kadane, “Using Uncleanliness to Predict Future Botnet Addresses,” Proc. the 7th ACM SIGCOMM Conference on Internet Measurement (IMC 2007), pp.93–104, San Diego, California, USA, October 2007.

[33] A. Ramachandran and N. Feamster, “Understanding the Network-level Behavior of Spammers,” Proc. ACM SIGCOMM 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pp.291–302, Pisa, Italy, September 2006.

[34] C. M. Bishop, “Pattern Recognition and Machine Learning (Information Science and Statistics),” Springer, 2006.

[35] C. C. Chang, and C. J. Lin, “LIBSVM : A Library for Support Vector Machines,” ACM Transactions on Intelligent Systems and Technology, vol.2, pp.27:1–27:27, April 2011.

[36] J. Platt, “Fast Training of Support Vector Machines Using Sequential Minimal Optimization,” Advances in Kernel Methods, MIT Press, pp.185–208, 1999.

[37] J. Platt, “Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods,” Advances in Large Margin Classifiers, MIT Press, pp.61–74, Mar. 1999.

[38] Alexa Top Sites, `http://www.alexa.com/topsites/`

[39] Malware Domain List, `http://malwaredomainlist.com/`

[40] 千葉 大紀, 八木 毅, 秋山 満昭, 森 達哉, 後藤 滋樹, “多種多様な攻撃に用いられる IP アドレス間の相関解析,” 情報処理学会コンピュータセキュリティシンポジウム (CSS 2011), vol.2011, no.3, pp.185–190, October 2011.

[41] T. Asano, D. Ranjan, T. Roos, E. Welzl, and P. Widmayer, “Space-filling Curves and Their Use in the Design of Geometric Data Structures,” Theoretical Computer Science, vol.181, no.1, pp.3–15, 1997.

[42] X. Cai, and J. Heidemann, “Understanding Block-level Address Usage in the Visible Internet,” Proc. the ACM SIGCOMM 2010 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pp.99–110, New Delhi, India, August 2010.

[43] P. Coogan, “12 Million Exploit Attacks Originating from the CO.CC Domain,” Symantec Security Response Blog, http://www.symantec.com/connect/blogs/12-million-exploit-attacks-originating-cocc-domain

[44] “国際化ドメイン名,” 社団法人日本ネットワークインフォメーションセンター (JPNIC), http://www.nic.ad.jp/ja/dom/idn.html

[45] Y. W. Chen, and C. J. Lin, “Combining SVMs with Various Feature Selection Strategies,” Studies in Fuzziness and Soft Computing, Springer, vol.207, pp.315–324, 2006.

[46] Y. W. Chang, and C. J. Lin, “Feature Ranking Using Linear SVM,” Journal of Machine Learning Research - Proceedings Track, vol.3, pp.53–64, 2008.