情報処理学会論文誌 Vol.61 No.2 385–396 (Feb. 2020)

機械学習を用いた自律進化型悪性サイトアクセス抑制手法の設計と実装

藤井 翔太[1,a)] 鬼頭 哲郎[1] 重本 倫宏[1] 仲小路 博史[1] 藤井 康広[1]

受付日 2019年4月19日, 採録日 2019年11月7日

概要：サイバー攻撃の激化にともない，悪性サイトへのアクセスを防止する技術が求められている．悪性サイトへのアクセス防止を図る既存手法としては，ブラックリストやホワイトリストを用いたものがあるが，ブラックリストは新たな悪性サイトへの即座の追従が困難であるという課題がある．ホワイトリストに関しても，良性サイトを網羅するのが困難なために，そこから漏れた良性サイトへのアクセスも防止してしまい，結果として通常業務への悪影響を生じかねないという課題がある．また，機械学習を用いた判定手法もあるが，100%の精度は困難なため，同様に良性サイトへのアクセスを防止してしまう可能性がある．そこで，これらの課題を解決する手法を提案する．提案手法は，機械学習の方式をベースにしつつ，悪性と判断したサイトへのアクセスを即時遮断するのではなく，機械には突破困難な追加認証を課し，突破できなかった場合のみアクセスを遮断する．これにより，機械学習方式の利点を享受しつつ，業務遂行に必要なサイトを誤って悪性と判定した際の業務阻害を緩和する．また，人間によるアクセスを非悪性サイト，マルウェア等による機械的なアクセスを悪性サイトとして機械学習モデルの判定結果と結果が正しかったか否か判定することが可能となる．この判定結果を機械学習モデルにフィードバックすることにより，運用の中での自律的な判定精度向上を図る．本稿では，提案手法の設計を述べるとともに，提案手法を実装・評価し，悪性サイトへのアクセスを遮断しつつ，追加認証結果を用いることにより機械学習モデルの精度向上や精度低下抑制が可能なこと，処理性能が実用の範囲内であることを確認した．

キーワード：悪性 Web サイト，追加認証，機械学習

Design and Implementation of Autonomous Evolution System for Preventing Malicious Web Access with Machine Learning

Shota Fujii[1,a)] Tetsuro Kito[1] Tomohiro Shigemoto[1] Hirofumi Nakakoji[1] Yasuhiro Fujii[1]

Received: April 19, 2019, Accepted: November 7, 2019

Abstract: Along with the increasing of cyber attacks, method for preventing access to malicious web sites is required. Blacklist and whitelist are famous conventional approach to prevent access to malicious web sites. However, these approaches have some limitation. Blacklist approach is hard to prevent access to new malicious web sites. Whitelist approach is hard to enumerate all benign websites; thus, there is a possibility to prevent access to benign website in error. Machine Learning (ML) also can be used to prevent access to malicious websites, however, 100% accuracy cannot be guaranteed; thus, there is a possibility to prevent access to benign website as with whitelist approach. We propose method for preventing access to malicious website with solving above problems. Our proposed method adopts ML approach. In case ML model detects suspicious access, proposed method does not block it immediately but requests additional authentication that program such as a malware cannot pass thorough. To do so, negative impact of preventing access to benign websites can be reduced. In addition, by feedback the result of additional authentication, updating ML model can be carried out like online training. In this paper, we implement and evaluate the prototype of proposed method. The evaluation shows the feedback from additional authentication can improve accuracy of ML model and processing time has no problem for practical use.

Keywords: malicious web site, additional authentication, machine learning

1. はじめに

近年，標的型攻撃に見られるように，攻撃が高度化しており，企業や国家にとって重大な脅威となっている．ここで，マルウェアのダウンロード [1] に加えて，マルウェアとの通信，フィッシングサイトの表示，スパムの発信 [2] 等，悪意を持ったサイト（以降，悪性サイト）がサイバー攻撃において重要な役割を有している．このことから，被害を抑制するためには，悪性サイトとの通信を遮断することが重要であるといえる．

悪性サイトへの通信を抑制する方法の 1 つに，インテリジェンス（ブラックリスト等）を用いるものがある．しかし，インテリジェンスには，潜在的に偽陽性が含まれており，仮に業務遂行に必要な非悪性サイトがブラックリストに誤って含まれていた場合，当該サイトにアクセスできず，業務阻害の要因となってしまう．業務阻害を抑制する方法として，インテリジェンスを事前に精査し，非悪性サイトを人手で除外する方法も考えられるが，Web サイトの精査という別のコストが生じてしまう．また，未知のドメインを用いた攻撃に対して脆弱であるという欠点もある．他の方法として既知の悪性サイトから機械学習等によって特徴を学習し，悪性サイトへの通信を抑制する手法もある．本手法は，ブラックリスト方式と比較すると未知の悪性サイトにもある程度効力があるものの，ブラックリストと同様に，潜在的な誤りを含んでおり，かつ判定結果が正しいか否か判断するのが困難である．このため，業務遂行に必要なサイトを誤って悪性と判定し，アクセスを防止した場合，業務阻害につながってしまう．

そこで，本稿では機械学習を用いた自律進化型の悪性サイトアクセス抑制手法を提案する．提案手法は機械学習による方式をベースにアクセス先の不審度を算出し，不審度が一定値以上のものに対しては，アクセスの抑制を図る．この際，即座にアクセスを防止するのではなく，いったん追加認証を課すことにより，人間による業務上必要なアクセスは許可しつつ，マルウェア等による機械的なアクセスは遮断する．これにより，未知の悪性サイトにも有効性がある，管理コストが比較的小さいという利点を享受しつつ，業務遂行に必要なサイトを誤って悪性と判定した際の業務阻害を緩和する．また，人間によるアクセスを非悪性サイト，マルウェア等による機械的なアクセスを悪性サイトとして機械学習モデルの判定結果と結果が正しかったか否か判定することが可能となる．この判定結果を機械学習モデルにフィードバックすることにより，運用の中での自律的な判定精度向上を図る．

[1] 株式会社日立製作所研究開発グループ
Research & Development Group, Hitachi, Ltd., Yokohama, Kanagawa 244–0817, Japan
[a)] shota.fujii.xh@hitachi.com

本稿では，上記の基本アイデアを基にした提案手法の設計を述べる．また，プロトタイプを実装し，提案手法の有効性について評価した結果を示す．本稿の貢献は，以下のとおりである．

- 機械学習を用いた自律進化型の悪性サイトアクセス抑制手法を提案した．提案手法は，機械学習によりアクセス先サイトの不審度を判定する．この際，不審度が一定値以上のものに対しては，即座にアクセスを防止するのではなく，いったん追加認証を課すことにより，人間による業務上必要なアクセスは許可しつつ，マルウェア等による機械的なアクセスは遮断する．また，アクセスを許可したか否かの結果を機械学習モデルにフィードバックすることにより，運用の中での自律的なモデルの精度向上を図る．
- 提案手法のプロトタイプを実装し，追加認証の結果をフィードバックすることにより，アクセス先サイトが悪性か否かの判定精度をフィードバックしない場合よりも向上させることができることを示した．また，性能評価を行い，その実行速度が実用の範囲内であることを実証した．

本稿の構成は次のとおりである．まず，2 章で既存の悪性サイトへのアクセス防止手法とその課題について述べる．3 章で同課題を解決する手法を提案し，4 章で実装・評価を行う．その後，5 章で提案手法の制限事項について議論する．また，6 章で関連研究について述べ，最後に 7 章でまとめを述べる．

2. 背景と課題

先述したように，悪性サイトへのアクセスを遮断することにより，情報漏えいや感染拡大を抑制可能である．こうした状況から，悪性サイトへのアクセスの遮断を目的とした手法が提案されており，それらはホワイトリスト方式，ブラックリスト方式，および機械学習方式の 3 つに大別できる．

ホワイトリスト方式では，業務に必要なサイトをホワイトリストに登録し，リスト内のサイトへのアクセスのみ許可する．これにより，ホワイトリストにない不審なサイトへのアクセスを遮断し，悪性サイトにアクセスすることによる被害を未然に防止する．一方で，すべての良性サイトを網羅するのは現実的に困難であり，リストから漏れた良性サイトへのアクセスについては，誤って遮断してしまうという課題がある．この過検知を抑制するにはつねにリストを更新し続けなければならず，リストの管理コストが大きい．

ブラックリスト方式では，マルウェアがアクセスするサイト等をブラックリストに登録し，リスト内のサイトへのアクセスを遮断する．これにより，実績のある悪性サイトへのアクセスを明示的に遮断可能である．一方で，ブラッ

クリストは潜在的に偽陽性を孕んでおり，誤って良性サイトへのアクセスを遮断してしまう可能性がある．これを抑制するには，良性サイトがブラックリストに誤って含まれていないか精査する必要があり，リストの管理コストが大きい．また，未知の悪性サイトについては，アクセスを遮断できないという課題もある．

機械学習方式では，既知の悪性・良性サイトからその特徴を学習し，新たにアクセスするサイトが悪性・良性のどちらであるかを判定する．その後，悪性と判断したサイトへのアクセスを遮断する．これにより，ホワイトリスト方式やブラックリスト方式のリスト管理コストが大きい，未知サイトに脆弱であるといった課題を緩和しつつ，悪性サイトへのアクセスを遮断する．一方で，100%の精度確保が困難なことから，良性サイトを悪性であると過検知してしまう可能性を孕んでおり，結果としてはホワイトリスト方式と同様に過検知を起こしてしまい，業務に必要な良性サイトへのアクセスを遮断してしまう可能性がある．たとえば，文献 [3] の機械学習方式では，70%の未知の悪性ドメインを検出できる一方で，0.35%の良性ドメインを誤って悪性と判断してしまうと報告されている．また，良性サイトを悪性と判定する等，モデルが誤った結果を出した際には再学習する必要があるが，モデルの算出した結果が正しいか否かを機械的に判断する術がなく，専門家の継続的な管理が必要であるという課題がある．特に，悪性サイトは，時間とともに変化する Concept Drift [4] の性質を有しており，1 度モデルを作成したら完了ではないため，この課題は顕著に影響する．

また，著者らはこのような状況に鑑み，マルウェアの動的解析結果の情報や，共有されたインテリジェンスを活用することでサイバー攻撃に対して集団防御を実現する自律進化型防御システム（AED：Autonomous Evolution of Defense）の研究を進めてきた [5], [6], [7]．マルウェアの中には，自身がインターネットと通信可能か否かを判断するために，実行初期に正規のサーバに対して疎通確認を行うものが存在する．このため，マルウェアを動的解析した結果得られたマルウェアの通信先を遮断すると，業務へ悪影響（可用性の低下）を与える可能性がある．我々の研究グループが提案する AED は，このような不確実性の高い脅威情報を用いて対策を実現するシステムである．具体的には，マルウェアの動的解析や共有されたインテリジェンスから得られた不審サイト情報をグレーリストとして管理し，クライアントがその不審サイトへアクセスしようとした場合に，プロキシで追加認証を要求する．これにより，たとえ誤った情報による認証追加であったとしても人間による業務上必要なアクセスは許可しつつ，マルウェア等の機械によるアクセスを遮断することを可能とする．ただし，いずれかの形で追加認証を要求するか否か判断するためのリストを管理しなければならず，リスト管理コストが大きいという課題は残存している．

まとめると，上記 4 手法は以下に示すいずれかの課題を有している．

（課題 1）リストの管理コストが大きい．
（課題 2）未知の悪性サイトに対応できない．
（課題 3）過検知による悪影響が大きい．
（課題 4）判定結果が正しいか不明である．

本稿では，上記の課題を抑制しつつ，悪性サイトへのアクセスを遮断する手法を提案する．

3. 提案手法

3.1 基本アイデア

まず，ホワイトリスト方式やブラックリスト方式は，有用な一方で，リストが完璧でなければ悪影響が出てしまう．そこで，両方式を採用しつつ，リスト外のサイトに関しては別途判定機構を設けることにより，リストが完璧でなければならないという制限やそこに起因するリストの管理コストが大きいという課題を緩和する（対課題 1）．リスト外のサイトに対する判定機構については，2 章で言及した未知の悪性サイトへの対応可能性から，機械学習手法方式を用いる（対課題 2）．この際，機械学習方式の欠点である過検知による悪影響が大きい点を緩和するために，機械学習モデルによって悪性と判断したものを即遮断するのではなく，マルウェアは突破困難な追加認証を課し，追加認証を突破した際はアクセスを許可する（対課題 3）．また，この追加認証の結果とモデルの判定結果を突合するによって，判定結果が正しいか判断でき，モデルへのフィードバック等も可能になる．この際，複数ユーザの判断を集合知として活用してアクセス先サイトの性質を判定することによる高精度化を図る（対課題 4）．

上記の基本アイデアをベースに，システムの設計を行う．

3.2 全体像

提案システムの全体像を**図 1** に示す．まずアクセス先を各種リストと照合し，ホワイトリストに一致する場合はアクセスを許可，ブラックリストに一致する場合はアクセスを遮断する（リスト含有判定機構）．アクセス先が両リストに含まれない場合は，機械学習による悪性度判定に必要な情報を取得し（外部情報取得機構），同情報を用いて判定を行う（悪性度推定機構）．アクセス先を悪性と判断した場合，追加認証をユーザに課し，追加認証を突破しなかった場合はアクセスを遮断し，突破した場合はアクセスを許可する（追加認証実施機構）．また，追加認証結果と機械学習モデルの判定結果を突合し，学習モデルを更新する（認証結果反映機構）．最後に，ここまでの処理結果を活用してホワイトリストやブラックリストを更新する（リスト更新機構）．

以降では，それぞれの機構についての詳細を述べる．

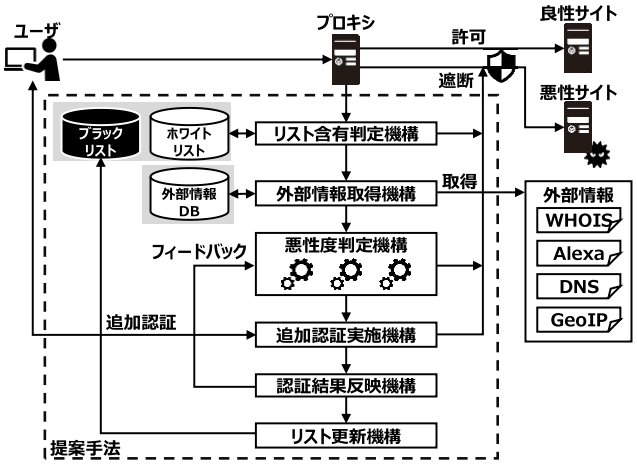


図 1 提案手法の全体像
Fig. 1 Overview of the proposed method.

3.3 各機構の詳細

3.3.1 リスト含有判定機構

本機構では，アクセス先がホワイトリストやブラックリストに含まれないか確認する．アクセス先がリストに含まれ，かつホワイトリストに一致する場合はアクセスを許可し，ブラックリストに一致する場合はアクセスを遮断する．アクセス先が両リストに含まれない場合は，これ以降の処理へと進み，アクセス可否の判定を行う．

3.3.2 外部情報取得機構

本機構は，後段の悪性度推定に必要な情報を取得する．具体的には，推定対象 Web サイトの URL をベースとして外部のサーバに問合せを行い，WHOIS 情報，DNS 情報，Alexa ランク，および地理情報を取得する．

また，1 度取得した情報は，データベース（以降，DB）に保管することにより，以降外部アクセスなしに参照する．このため，同じドメインに対する 2 回目以降のアクセスでは，DB を参照して外部情報を取得することにより，外部アクセスをともなうことによる処理時間の長大化や同一情報提供元に対するアクセス過多を抑制できる．

3.3.3 悪性度推定機構

本機構は，前述のとおり機械学習によってアクセス先の悪性度を推定する．ブラックリストを悪性サイトの，ホワイトリストを良性サイトの教師データとして学習し，推定モデルを構築する．特徴量としては URL 文字列から取得できる情報に加えて，外部情報取得機構を用いて取得した情報を利用する．なお，各特徴量は，良性サイトと悪性サイトの間に違いとして現れやすい点に着目して選出しており，各値は，0〜1 の間に正規化して利用する．具体的には，最小値を 0，最大値を 1 とし，その間を比例配分する．また，後述の URL 文字列のように上限のない特徴量に関しては，特徴量化時点での最大値が 1 になるように正規化を行う．特徴量一覧については，値域とともに**表 1** に示し，以降で詳述する．

表 1 各推定器における特徴量と値域
Table 1 Feature values and their range of classifiers.

推測器	カテゴリ	通番	特徴量	値
1	URL 文字列	1	URL 文字列長	0-
		2	ドメイン文字列長	0-
		3	パス文字列長	0-
		4	URL 文字列に含まれる数字の数	0-
		5	ドメイン文字列に含まれる数字の数	0-
		6	パス文字列に含まれる数字の数	0-
		7	パス文字列に含まれるトークン数	0-
		8	パス文字列に含まれる平均トークン数	0-
		9	パス文字列に含まれる最長トークン数	0-
		10	ドメイン文字列全体に対する最長英単語の長さが占める割合	0-1
		11	FQDN のジニ係数	0-1
		12	FQDN が “.” を含むか	0/1
		13	FQDN が “-” を含むか	0/1
		14	URL が拡張子で終わるか否か	0/1
		15	URL 文字列が'exe'を含むか否か	0/1
		16	URL 文字列が'php'を含むか否か	0/1
		17	ドメインが IP アドレスか否か	0/1
2	WHOIS 情報	18	ドメイン登録期間（年数）	0-10
		19	ドメイン性質（初登録 or 更新有）	0/1
		20	ドメイン初登録からの年数	0-
		21	ドメイン登録時間（0-23）	0/1
		22	ドメイン登録曜日（月火水木金土日）	0/1
		23	レジストラ（当該レジストラのうち，悪性データ分が占める割合）	0-1
3	DNS 情報	24	A レコード数	0-
		25	AAAA レコード数	0-
		26	CNAME レコード数	0-
		27	MX レコード数	0-
		28	NS レコード数	0-
		29	PTR レコード数	0-
		30	TXT レコード数	0-
		31	逆引きが設定されているか否か	0/1
		32	ネガティブ TTL（SOA レコードの minimum 値）	0-
	Alexa ランク	33	Alexa ランク（訪問者数ランク）	0-
		34	Alexa ランクの差分（アクセス数ランク−訪問者数ランク）	0-
	地理情報	35	国（IP アドレスに対応する国のうち悪性データ分が占める割合）	0-1

URL 文字列では，悪性 URL は，良性 URL よりも，複雑である [3], [8], [9]（通番 1〜11），特定の文字・拡張子の出現頻度が高い [3], [9], [10]（通番 12〜16），およびドメインと紐付けられていない [8], [9]（通番 17）という傾向をつかむために特徴量を選定した．なお，通番 11 のジニ係数とは，集合の複雑性を測る指標の 1 つであり，集合の複雑性が低いほど 0 に，高いほど 1 に近づく．このため，FQDN をアルファベットや記号 1 文字ごとの集合と見なし，ジニ係数を算出することにより，その複雑性を測ることができる．

WHOIS 情報では，攻撃に利用されるドメインは使い捨てであり，生存期間が短い場合がある [11], [12], [13], [14]（通番 18〜20），攻撃者の登録作業コストや登録の金額面でのコストを抑制するために，まとめて登録される場合がある [3], [13]（通番 21，22），および，攻撃者がドメインを取得する際のレジストラには偏りがある [11], [12], [15], [16] という傾向をつかむために，通番 23 を選定した．

DNS 情報では，各種レコード（通番 24〜31）に加え，悪性サイトのネガティブ TTL は短い傾向にある [17], [18] という特徴をつかむために，特徴量（通番 32）を選定した．

また，悪性サイトは，アクセス数が正規サイトより少ないという仮定の下，アクセス数ランキングである Alexa ランク（通番 33，34）を利用した．さらに．攻撃に利用されるドメインは特定の国に偏っているとの報告 [10] から，通番 35 を選出した．

ここで，アンサンブル学習によって複数の推定器を組み合わせることによって，汎化能力が向上することや Concept Drift への追従性が向上することが知られている [19]．そこ

で，ここまでに述べてきた特徴量をまとめて単一の推定器に利用するのではなく，3 つの推定器を用意し，推定器 1 で URL 文字列，推定器 2 で WHOIS 情報，および推定器 3 でその他の情報（DNS 情報，地理情報，および Alexa ランク）を利用して，各推定器の加重平均をとることによりアクセス先サイトの脅威度を判定する．各推定器に与える重みの初期値には，学習時に教師データの分類精度を記録しておき，その分類精度の推定器間での比率を正規化した値を利用する．たとえば，同様の教師データを推定器 1 が 90%，推定器 2 が 85%，推定器 3 が 75%の精度で分類できた場合，その比率は，90 : 85 : 75 であり，この比率の合計が 1 になるよう正規化する．つまり，それぞれの推定器に対して，0.36，0.34，0.30 の重みを与える．

また，本機構は，対象 Web サイトの脅威度を 0（良性寄り）～100（悪性寄り）の連続値で算出する．この際，対象 Web サイトが悪性か否かを分類する閾値を定めておき，脅威度が閾値未満の場合は良性，閾値以上の場合は悪性と判断する．

3.3.4 追加認証実施機構

脅威度推定機構によって，アクセス先を悪性と判断した場合，本機構を用いて追加認証をユーザに課する．具体的には，人間と機械を判別するチューリングテストである Completely Automated Public Turing test to tell Computers and Humans Apart（以降，CAPTCHA）認証を課する．この追加認証を突破しなかった場合はアクセスを遮断することにより，マルウェアによる機械的なアクセスを遮断する．公開ブラックリスト化されている悪性サイトは 90%以上がマルウェア由来であるという報告 [20] もあり，悪性度推定機構によって閾値以上の悪性度が算出されたサイトに対して追加認証を課することにより，多くの悪性サイトへの通信を遮断することが期待できる．また，人手でのアクセスに関しても，追加認証を提示することにより，アクセス先の悪性度が一定以上あるという情報とともにユーザへアクセス可否を問うため，事前情報がない場合よりもアクセス先が悪性か否かの判断しやすいと推察される

また，推定器が業務に必要な良性サイトを悪性と誤認した場合でも，追加認証を突破すればアクセスを許可することにより，業務への悪影響を最小限に抑制する．

3.3.5 認証結果反映機構

悪性度推定機構で採用したアンサンブル学習においては，複数存在する推定器の加重平均をとる際，精度が高い推定器に大きい重みを与えることで，全体としての推定精度が向上する．そこで，本機構は，追加認証結果と脅威度推定機能を構成する個々の推定器の推定結果を突合し，その両者が一致したものは精度が高いものとして精度が高い推定器の重みを増やし，そうでない推定器の重みを減らす．本処理により，高精度な推定器の重用とそれによる推定精度の向上を図る．各推定器の更新式は，Adaboost [20] のものを利用し，推定器 m の重み α を誤り率 ϵ に応じて更新する（式 (1)）．なお，今回は実装の容易性を考慮して式 (1) を採用したが，精度の高い推定器により大きな重みを与えることのできる式であれば，他の式を用いてもよい．

$$\alpha_m = \frac{1}{2}\log\frac{1-\epsilon_m}{\epsilon_m} \tag{1}$$

ここで，フィードバックを実施する際，特に人手によるアクセスの際の追加認証結果とサイトの性質は，必ずしも一致するとは限らない．たとえば，セキュリティ意識の低いユーザが不用意に追加認証を突破してアクセスする場合等が考えられる．この場合，誤った結果を推測器群へフィードバックすることになってしまい，サイトの性質判定を誤った推測器に重みが加わるといった悪影響が生じる可能性がある．そこで，単一ユーザの判断結果のみを用いるのではなく，複数ユーザの判断を集約し，集合知として活用することによりアクセス先サイトの性質を推定し，同課題の緩和を図る．

3.3.6 リスト更新機構

最後に，本機構によって，脅威度推定機構の推定結果や追加認証結果を利用して良性サイトをホワイトリストに，悪性サイトをブラックリストに追加する．

このように，提案手法は，各機構によって悪性サイトへのアクセスを抑制しつつ，追加認証の結果を用いて脅威度推定機構の重みを自律的に調整し，精度向上を図るとともに，ホワイト/ブラックリストを構築する．

4. 評価

4.1 実装

提案手法は，前章で述べたとおり，機械学習によってサイトの脅威度を推定するが，この部分の実装には機械学習ライブラリである scikit-learn [21] を利用した．また，WHOIS，DNS，Alexa ランク，および地理情報といった各種外部情報の取得には，それぞれ Python-whois [22]，dnspython [23]，Alexa API [24]，および GeoIP [25] を利用した．

4.2 データセット

良性サイトのサンプルとして，オープン Web ディレクトリである DMOZ [26] から収集したサイト群，悪性サイトのサンプルとして hpHosts [27]，spamhaus [28]，Malware Domain List [29]，および aguse [30] から収集したサイト群を利用した．この際，良性/悪性サイト情報それぞれ 50,000 件ずつ，合計 100,000 件をランダムに取得した．これらの情報を以降の節での評価に利用する．

4.3 評価項目

評価項目は，以下のとおりである．

(1) 脅威度推定精度

Web サイト脅威度推定機能は，複数の推定器を利用し

表 2 性能測定環境
Table 2 Experimental environment.

CPU	Intel Core i7-2600 （4 コア，8 スレッド）
メモリ	2,048 MB
OS	Ubuntu 14.04 LTS

て，アクセス先サイトの脅威度を推定する．良性データ/悪性データを用いてこの推定精度を検証する．

(2) 精度変動

提案手法は，追加認証の結果を各推定器の重みにフィードバックすることにより，精度向上を図る．本評価では，想定のとおり精度が向上するか検証する．

(3) 処理性能

提案手法はフォワードプロキシに実現し，ユーザのアクセスごとにアクセス先の脅威度を測定し，脅威度が一定値以上の場合は追加認証を要求する．この挙動は，Web アクセスのオーバヘッドとなるため，その値が実用範囲内か検証する．なお，処理性能は，**表 2** に示す環境で測定した

4.4 評価結果

4.4.1 脅威度推定精度

本評価では，4.2 節で言及したデータセットのうち，良性サンプル 25,000 件と悪性サンプル 25,000 件の合計 50,000 件を用いて，提案手法の脅威度推定精度を検証する．まず，URL 文字列のみを用いた場合，WHOIS 情報のみを用いた場合，および DNS 情報+ Alexa ランク+地理情報を用いた場合の 3 パターンにおいて，各種アルゴリズム（線形 SVM，ロジスティック回帰，決定木，K 近傍法，ランダムフォレスト，3 層ニューラルネット，およびアダブースト）を用いて評価を実施した．その後，全情報をまとめて 1 つの推定器に利用した場合と各パターンで最も高精度であったアルゴリズムのものを組み合わせた場合の提案手法で比較評価を行った．各アルゴリズムでは，3.3.3 項で述べたとおり，対象 Web サイトの脅威度を 0（良性寄り）～100（悪性寄り）の連続値で算出する．また，悪性サイト/良性サイトの分類を行う閾値には，50 を用いた（脅威度が 50 未満の場合：良性と判断，脅威度が 50 以上の場合：悪性と判断）．なお，提案手法において，各推定器の重みには，同データセットに対する各推定器の正解率の比率を利用した．さらに，提案手法に関しては，閾値をどの値に設定すればよいか，過検知率（False Positive Rate，以降 FPR），見逃し率（False Negative Rate，以降 FNR），および正解率の観点から評価した．なお，これらの評価には，10-分割交差検証を用いた．

まず，各パターンにおける脅威度推定精度の測定結果を**表 3** に示す．表 3 から，URL 文字列のみ，WHOIS 情報のみを用いた場合はランダムフォレスト，DNS 情報+ Alexa ランク+地理情報を用いた場合は K 近傍法がそれぞれ 68.13%，82.22%，および 85.92%と最高精度であることが確認できる．さらに，上述のアルゴリズムを用いた 3 つの推定器を組み合わせて利用したところ，各推定器単体のいずれ（68.13%，82.22%，および 85.92%）よりも高く，かつすべての情報を単純にまとめて利用した場合（89.43%）よりも高い 91.05%の精度で Web サイトを分類できた．また，**図 2** は，各パターンでの最高精度を出した分類器の Receiver Operating Characteristic（以降，ROC）曲線である．

表 3 各パターンにおける脅威度推定精度の測定結果（%）
Table 3 Accuracies of threat estimation of each method.

アルゴリズム	パターン				
	URL	WHOIS	DNS	全て利用	提案手法
線形 SVM	59.59	81.60	72.36	78.71	
ロジスティック回帰	65.80	81.83	72.36	85.85	
決定木	68.06	79.18	81.55	86.49	
K 近傍法	67.14	78.48	85.92	80.00	
ランダムフォレスト	68.13	80.86	82.58	89.43	
ニューラルネット	66.05	82.02	79.85	87.40	
アダブースト	66.86	82.22	81.30	89.28	
最高精度	68.13	82.22	85.92	89.43	91.05

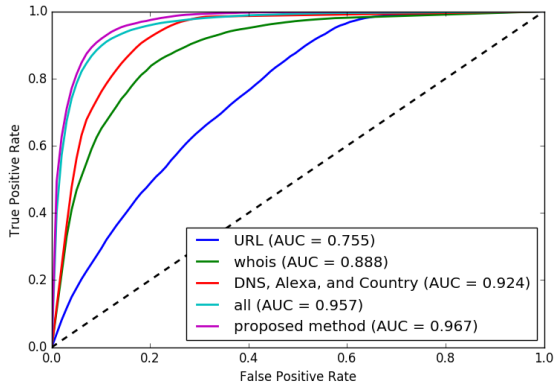


図 2 各パターンにおける ROC 曲線
Fig. 2 ROC curve of each method.

ROC 曲線において，曲線下の面積を Area Under the Curve（以降，AUC）と呼び，AUC が 1 に近いほど，識別性能が高いことを示す．図 2 から，提案手法の AUC が 0.967 と正解率と同様に他のいずれのパターンにおける値（0.755，0.888，0.924，および 0.957）よりも高いことが分かる．以上のように，提案手法が各推定器単体の場合や全情報をまとめて 1 つの推定器で利用した場合のいずれよりも高精度で Web サイトの脅威を推定できた．また，複数の文献において，URL 文字列のみで推定することの問題点が示唆されている（精度の確保が難しい [8]，短縮 URL を誤判定してしまう [9] 等）が，本実験においても，URL 文字列のみを使ったものは，最高でも 68.13%と他に比べて精度が低く，各文献での示唆内容を裏付けるものとなった．

次に，提案手法において，閾値を 0～100 の間で変動させ，それぞれの値における FPR，FNR，および正解率を算出した．この結果を**図 3** に示す．図 3 から分かるように，閾値を 48 に設定することで，92.03%と最も高い正解率を

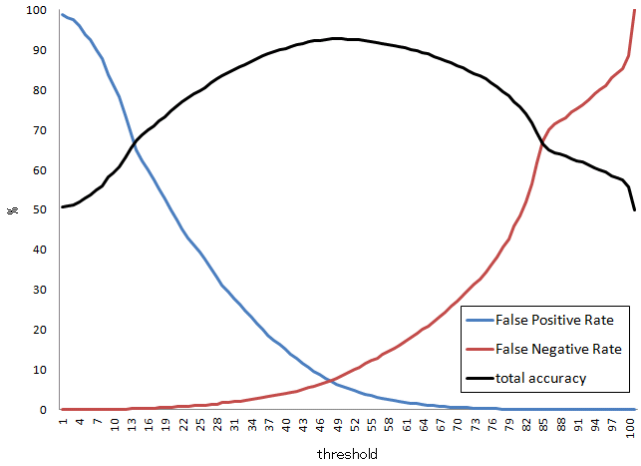


図 3 閾値ごとの FPR，FNR，および正解率（%）
Fig. 3 FPR, FNR, and accuracy by threshold.

得られている．また，閾値を下げると良性なサイトであっても不審サイトとして過検知してしまう可能性（FPR）が高まり，反対に閾値を上げると悪性サイトを見逃してしまう可能性（FNR）が高まる．ここで，悪性サイトを見逃してしまうと，同サイトへのアクセスが発生して被害が生じうることや良性サイトを不審サイトとして過検知してしまった場合でも，AED であれば追加認証さえ突破すればアクセスは続行でき，業務効率への影響は最小限に抑制できることから，FPR の増大はある程度許容できるとともに，FNR を抑制することが望ましい．最高精度（92.03%）を出せる閾値 48 の場合は，FNR が 7.18%だが，これを半分以下に抑えようとした場合，閾値を 37 にすることにより，正解率を 89.34%にとどめつつ達成できる（3.433%）．

以上の評価結果から，複数の推定器を用いることによって，単体の場合よりも精度が出せること，正解率の視点からは閾値を 48 にすればよいことが分かった．

4.4.2 精度変動

本評価では，4.3.1 項で，50,000 件のデータを用いて訓練したモデルを用いて，残り 50,000 件のデータが良性・悪性のいずれかを推定する．この際，フィードバックを行わない版をベースラインとし，フィードバックを行った場合，そのベースラインと比較してどの程度精度が向上するかを評価する．この際の閾値は，4.4.1 項の評価で最も高精度を出した 48 とした．なお，文献 [7] での評価結果より，悪性サイトへのアクセスはすべてマルウェアによるもので追加認証を突破できず，良性サイトへのアクセスに対する追加認証はすべて正しく突破するものとして評価した．

評価用データ 50,000 件に対するベースライン版とフィードバック版の正解率の変動を**図 4** に示す．破線がベースライン版，実線がフィードバック版を示している．25,000 件までは，わずかではあるもの，フィードバック版がベースライン版よりも高い正解率である．また，それ以降はともに正解率が下がっているが，フィードバック版は，ベースライン版に比べて，正解率の低下が緩やかである．また，フィードバック版における各推定器の重みの変動を**図 5** に示す．図から，25,000 件周辺で URL 文字列を用いる分類器の重みが急激に低下していることが分かる．これは，評価用データの 25,000 件目以降については，URL 文字列の分類器の正解率が低かったことに起因する．このように，正解率の低い分類器の重みを低下させたことがベースライン版と比較した際の精度低下の抑制に寄与したと推察される．

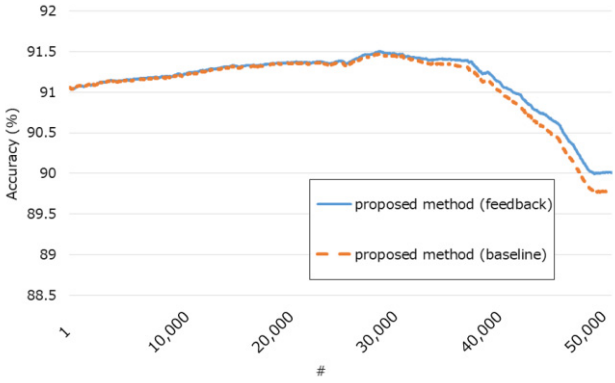


図 4 評価用データに対する正解率の変動の比較
Fig. 4 Accuracy comparison between baseline version and feedback version.

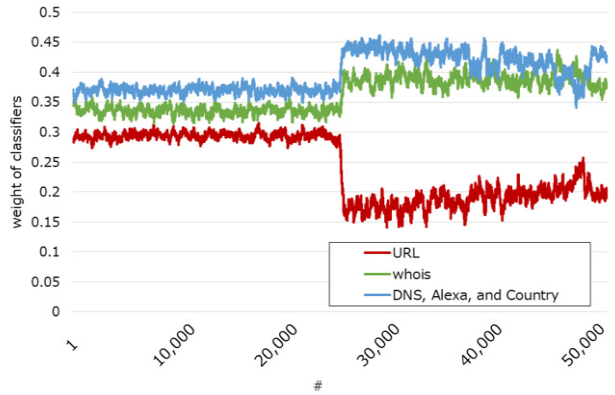


図 5 各推定器の重みの変動
Fig. 5 Weight fluctuations of each classifier.

以上の結果から，追加認証結果をフィードバックすることにより，自律的な精度向上や精度低下の抑制が可能であることが確認できた．

4.4.3 処理性能

本評価では，提案システムにおいて，グレーリストに含まれるサイトにアクセスした際，利用者に追加認証画面が表示されるまでに要する時間，すなわち情報取得と脅威度判定に要する時間の合計を測定した．なお，測定は，マルチユーザモード下において time コマンドを用いて行った．

測定結果の処理時間ごとの度数と累積比率を**図 6** に示す．図から，93.8%の処理時間が 3 秒以内に収まっていることが分かる．多くのユーザがページのロードが 3 秒以内なことを期待しているという調査結果 [31] があるが，提

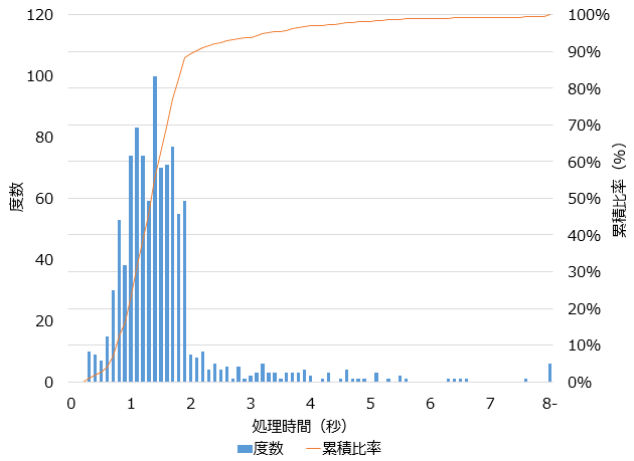


図 6 処理時間ごとの度数と累積比率

Fig. 6 Frequency and cumulative ratio by processing time.

案システムのページロード時間は 93.8%がその範囲内に収まっていることから，処理性能の面でも実用範囲内であると考える．

5. 議論と制限

5.1 追加認証突破

提案手法は，悪性度推定機構によって悪性サイトだと判定した場合，追加認証突破の可否に応じてアクセス可否を決定する．このため，良性サイトであっても，人間がブラウザの戻るボタンを押下する等によって追加認証を突破しない場合が考えられる．これは，業務での利用には必要ないという判断の下アクセスしなかったものであると考えられため，業務阻害を抑制しつつ悪性サイトへのアクセスを遮断するという観点からは問題ないと考える．また，マルウェア以外の良性サイトへの機械的なアクセスについては，文献 [6] の実験において CRL（Certificate Revocation List）や OCSP（Online Certificate Status Protocol）といった電子証明書関連情報を提供するサイトやソフトウェアの更新確認サイトが含まれることを確認している．これらのサイトは，多くの端末から定常的にアクセスされるものであり，アクセスログ等を基にホワイトリストへ追加し，追加認証の対象から除外できると推察される．

また，悪性サイトであっても人間が誤って追加認証を突破してアクセスしてしまう場合も考えられる．これに関しては，利便性とリスクのバランスをとり，許可することとしている．ただし，前述のとおり，マルウェアによる機械的なアクセスは抑制可能である．

なお，今回は CAPTCHA 認証を追加認証として利用したが，CAPTCHA 認証を突破する研究 [32], [33] も行われている．追加認証の方式は可換であるため，追加認証実施機構においては，機械的な突破に対してロバストな手法を逐次採用するのが望ましいと推察される．

5.2 本技術で対応できない攻撃

提案手法は，前述のように，悪性と判定したサイトでも利用者が追加認証を突破すればアクセスを許可する．このため，フィッシングサイトのように，利用者を騙して攻撃を達成する攻撃への対応は困難である．ただし，フィッシングサイトの検出手法は多数提案 [34], [35], [36], [37], [38] されており，これらの手法を別途組み合わせることにより，各種攻撃に対応可能であると考える．

6. 関連研究

6.1 Web サイトの性質を判別する研究

本節では，提案手法と同様に，Web サイトの性質を判別する研究について述べる．また，同研究は，以下の 3 種類に大別できる．

(1) Web サイトへのアクセスが不要なもの

文献 [8] は，既知の悪性 URL 群と近い性質を持った URL を未知の URL 群から抽出し，Bayesian sets と呼ばれる類似要素探索アルゴリズムを用いてブラックリストを構築することを目指したものである．文献 [9] は，URL 文字列ベースで決定木によって良性/悪性サイト分類を行っている．文献 [10] は，URL のみからフィッシングサイトか否かを判定するものである．Page Rank やドメインにフィッシング特有の文字列が出現しているか否かを特徴量とし，ロジスティック回帰により判定を行っている．

各研究の利点として，Web サイトへアクセスしないため，比較的判定が高速なことや攻撃者のアクセスログ等をとることによる被解析検知を逃れられることがあげられる．一方で，文字列のみを用いる場合，後述する研究のような Web サイトにアクセスして情報を取得する場合に比べると取得できる情報に限りがあり，精度の面ではやや劣る傾向が見られる．

(2) 非悪性サイトへのアクセスが必要なもの

PREDETOR [3] は，ドメインの文字列や登録情報を用いて悪性か否かを分類するものである．文献中で，悪用されるドメインの登録にはバースト性があることや失効したドメインが即時再取得された場合は，攻撃者による取得である可能性が比較的高いこと等が述べられており，悪性ドメインの分類に寄与することが実証されている．文献 [11] は，URL 文字列やホスト情報を用いた悪性サイトの推測器を提案している文献であり，フィッシングに関連する URL は，そうでない URL に比べて URL が長いこと，ドメイン名が長いこと，およびドメインの生存期間が短いこと等が検証結果として示されている．文献 [14] は，クライアント型ハニーポットでの URL の巡回を最適化するために，巡回候補 URL の悪性度を算出し，その高い順に巡回するものである．この悪性度を算出するために，SVM を利用し，特徴量には WHOIS 情報や FQDN 文字列の特徴を採用している．文献 [16] は，悪性 Web サイトが属する IP アドレ

スブロックとドメイン登録に用いたレジストラに着目し，両情報が既知の悪性ドメインのものと類似している場合，信頼性が低いドメインであるとして，ブラックリストに追加する手法を提案している．EXPOSURE [18] は，主にDNS 情報を用いて決定木で悪性ドメインを見つけるものである．Woodpecke [39] は，正規ドメインを侵害し，同ドメインのサブドメインとして作成される Shaded Domain を IP アドレスやサブドメイン等の特徴を用いて，ランダムフォレストによって検出するものである．文献 [40] の手法は，DNS 情報等に加えてサイトへアクセスを試みたプロセス情報も用いることによって，高精度にアクセス先が悪性か否か判定する．

これらの研究は，Web アクセスが不要な研究よりも推測に利用できる情報が多いため，精度が比較的高い傾向が見られる．一方の欠点としては，Web アクセスが不要な (1) に比べると，推測に要する時間が比較的長いことや対象の Web サイトにアクセスする (3) に比べると取得できる情報に限りがあり，精度が劣る傾向が見られる点があげられる．

(3)　悪性サイトへのアクセスが必要なもの

EvilSeed [13] は，既知の悪性サイト情報を基に，クライアント型ハニーポットが利用する効率的な巡回クエリを生成し，悪性サイトを収集するものである．既知の悪性サイトリストを基（Seed）にして，リンクや DNS 情報等が類似しているページを検索するクエリを生成する．そのクエリを基に巡回したページの性質を Oracle と呼ばれる Google Safe Browsing 等からなるコンポーネントを用いて判定し，悪性と判断されたものは，再度検索クエリ作成の Seed とすることで効率的に悪性サイトを巡回することが可能となる．本研究のように，実際に悪性サイトへアクセスするものは，比較的時間を要する・クローキングへの対策が必要である等の懸念があるものの，精度の面で優れている傾向にある．

提案手法は，複数の推定器を組み合わせるため，上述した手法等をその構成要素の 1 つとして活用することが可能である．また，本節で述べたような Web サイトの性質を判別する手法では，教師データや専門家の知見を用いて悪性/良性を判別するモデルが構築される．このモデル構築の学習フローは，新たなデータを随時オンライン学習していくものと一定のまとまったデータを用いて一からバッチ学習を行うものに大別できる．ここで，バッチ学習を前提としたモデルは，学習完了後，新たなデータに追従するには再学習を行う必要があるものの，そのコストは小さくないといった課題がある．一方で，提案手法はユーザからのフィードバックを基にして各推定器の重みを更新し，全体としての精度向上を図る．このため，バッチ学習を前提したモデルを組み込む場合であっても，バッチ処理による再学習によらない精度向上が期待できる．さらに，今回は，(1) と (2) に分類される手法のみを構成要素として用いたが，(3) にあたる手法を組み込むことによって，さらなる分類精度向上が期待できる．

6.2　ユーザの判断を利用する研究

ここでは，提案手法と同様に，セキュリティ分野において，ユーザの判断を用いる研究について述べる．

AI^2 [41] は，教師なし学習と教師あり学習の組合せによって効率的に不正ログインを検出するシステムである．まず，教師なし学習モジュールによって希少なものを検出・ランク付けする．これらは，分析官にも提示され，それぞれに正常なものか攻撃らしいかのラベル付けが行われる．ここでラベル付けされたデータは教師あり学習モジュールの学習に用いられ，ここで作成されたモデルは新規データのラベル推測に教師なしモデルとともに用いられる．このように，両モジュールの組合せによって不正イベントを検出した後，検出したデータのみに分析官がラベル付けを行い，モデルにフィードバックすることで，ラベル付けのコストを抑制しつつ，高精度な不正ログイン検出を可能としている．文献 [42], [43] は，分析者のフィードバックを利用する異常検知アルゴリズム Active Anomaly Detection（AAD）を提案している．AAD は，LODA アルゴリズム [44] を用いて各データの異常度をランク付けし，異常度の高いものを分析官に提示する．分析官は提示されたデータが異常か否かをフィードバックし，AAD はその結果を基に，異常なデータがより異常らしくなるように自身を更新することで，精度の向上を図る．MADE [45] は，エンタプライズシステムにおける悪性通信を検出するシステムである．プロキシログや FW ログ等，複数のアプライアンスのログを機械学習モデルに投入し，異常らしきものを検出した際には SOC/CSIRT に通知するとともに，真に悪性だったかを SOC/CSIRT が判断し，その結果をモデルにフィードバックすることによって精度向上を図る．

上述の研究は，いずれもユーザの判断を用いて認識精度や分析効率を向上している．ただし，エキスパートの存在を前提としており，人材確保が容易ではないことやスケール性に欠けることが課題としてあげられる．これに対して提案手法は，ユーザの判断を用いることによって Web サイトの判別精度を向上させることが期待される点は同様に，エキスパートに限らないユーザの集合知や追加認証によって自動的に遮断するマルウェアの悪性通信情報を利用することにより，上述の課題を緩和可能である．

7.　おわりに

本稿では，機械学習を用いた自律進化型悪性サイトアクセス抑制手法の設計と実装を述べた．提案手法により，既存手法の課題であったリスト管理のコストや良性サイトへのアクセスを誤って遮断してしまうことによる業務への悪影響を抑制しつつ，悪性サイトへのアクセスの遮断が可能

となる．また，追加認証の結果を機械学習モデルにフィードバックすることにより，自律的なモデルの精度改善が期待される．

評価では，提案手法のプロトタイプを実装し，サイトの性質を最大 92.03%の精度で推定できること，認証結果をフィードバックすることにより，運用のなかでの自律的な精度向上や精度低下抑制が可能なことを確認した．また，処理性能の面でも，93.8%が 3 秒以内に処理を完了することから，実用に耐えうる範囲であることを示した．

今後の課題としては，大規模環境における実運用を通しての精度向上がある．

参考文献

[1] Norton: What Are Malicious Websites?, available from ⟨https://us.norton.com/internetsecurity-malware-what-are-malicious-websites.html⟩ (accessed 2019-08-30).

[2] Zhao, B.Z.H., Ikram, M., Asghar, H., Kaafar, M.A., Chaabane, A. and Thilakarathna, K.: A decade of mal-activity reporting: A retrospective analysis of internet malicious activity blacklists, *Proc. 14th ACM Asia Computer Communication and Security* (*ASIA CCS '19*), pp.1–13 (2019).

[3] Shuang, H., Alex, K., Brad, M., Vern, P. and Nick, S.: PREDATOR: Proactive Recognition and Elimination of Domain Abuse at Time-Of-Registration, *Proc. 2016 ACM SIGSAC Conference on Computer and Communications Security* (*CCS '16*), pp.1568–1579 (2016).

[4] João, G., Indrė, Ž., Albert, B., Mykola, P. and Abdelhamid, B.: A survey on concept drift adaptation, *ACM Comput. Surv.*, Vol.46, No.4, pp.1–37 (2014).

[5] 仲小路博史，藤井康広，磯部義明，重本倫宏，鬼頭哲郎，川口信隆，林　直樹，下間直樹，菊池浩明：人間行動を用いた自律進化型防御システムの提案，2016 年暗号と情報セキュリティシンポジウム（SCIS2016），pp.1–8 (2016).

[6] Nakakoji, H., Fujii, Y., Isobe, Y., Shigemoto, T., Kito, T., Hayashi, N. Kawaguchi, N., Shimotsuma, N. and Kikuchi, H.: Proposal and Evaluation of Cyber Defense System Using Blacklist Refined Based on Authentication Results, *The 19th International Conference on Network-Based Information Systems* (*NBiS2016*), pp.135–139 (2016).

[7] 重本倫宏，藤井翔太，来間一郎，鬼頭哲郎，仲小路博史，藤井康広，菊池浩明：ホワイトリストを用いた自律進化型防御システムの開発，情報処理学会論文誌，Vol.59，No.3，pp.1050–1060 (2018).

[8] 孫　博，秋山満昭，八木　毅，森　達哉：既知の悪性 URL 群と類似した特徴を持つ URL の検索，コンピュータセキュリティシンポジウム 2014（CSS2014）論文集，pp.1–8 (2014).

[9] Michael, D., Greg, H., Gilad, G., Aravind, A. and Prabaharan, P.: A Lexical Approach for Classifying Malicious URLs, *2015 International Conference on High Performance Computing* (*HPCS 2015*), pp.195–202 (2015).

[10] Sujata, G., Niels, P., Monica, C. and Aviel D.R.: A framework for detection and measurement of phishing attacks, *Proc. 2007 ACM Workshop on Recurring Malcode*, pp.1–8 (2007).

[11] Ma, J., Saul, L.K., Savage, S. and Voelker, G.M.: Beyond blacklists: Learning to detect malicious Web sites from suspicious URLs, *Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (*KDD 2009*), pp.1245–1254 (2009).

[12] Mark, F., Christian, K. and Vern, P.: On the potential of proactive domain blacklisting, *Proc. 3rd USENIX Conference on Large-scale Exploits and Emergent Threats: Botnets, Spyware, Worms, and More* (*LEET '10*), pp.1–8 (2010).

[13] Invernizzi, L., Benvenuti, S., Cova, M., Comparetti, P.M., Kruegel, C. and Vigna, G.: EvilSeed: A Guided Approach to Finding Malicious Web Pages, *Proc. IEEE Symposium on Security and Privacy*, pp.428–442 (2012).

[14] 千葉大紀，森　達哉，後藤滋樹：悪性 Web サイト探索のための優先巡回順序の選定法，コンピュータセキュリティシンポジウム 2012（CSS2012）論文集，pp.805–812 (2012).

[15] 福島祥郎，堀　良彰，櫻井幸一：ドメイン情報に着目した悪性 Web サイトの活動傾向調査と関連性分析，コンピュータセキュリティシンポジウム 2010（CSS2010）論文集 (2012).

[16] 福島祥郎，堀　良彰，櫻井幸一：悪性 Web サイト間の関連性に着目した信頼性評価によるブラックリスト方式の検討，情報処理学会研究報告，Vol-CSEC-52, No.38, pp.1–8 (2011).

[17] Ricardo, V.S. and Jose, C.B.: Identifying Botnets Using Anomaly Detection Techniques Applied to DNS Traffic, *5th IEEE Consumer Communications and Networking Conference*, pp.476–481 (2008).

[18] Leyla, B., Engin, K., Christopher, K. and Marco, B.: EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis, *18th Annual Network and Distributed System Security Symposium* (*NDSS '11*) (2011).

[19] Pallabi, P., Zackary, W., Bhavani, T., Kevin, W.H. and Latifur, K.: Supervised Learning for Insider Threat Detection Using Stream Mining, *Proc. International Conference on Tools with Artificial Intelligence* (*ICTAI*), pp.1032–1039 (2011).

[20] Freund, Y. and Schapire, R.E.: Experiments with a New Boosting Algorithm, *Proc. 13th International Conference on International Conference on Machine Learning* (*ICML 96*), pp.148–156 (1996).

[21] scikit-learn: Machine Learning in Python, available from ⟨http://scikit-learn.org/stable/⟩ (accessed 2018-02-23).

[22] joepie91: GitHub - joepie91/python-whois: A python module for retrieving and parsing WHOIS data, available from ⟨https://github.com/joepie91/python-whois⟩ (accessed 2018-02-23).

[23] Nominum: dnspython home page, available from ⟨http://www.dnspython.org/⟩ (accessed 2018-02-23).

[24] Amazon Web Services, Inc.: AWS | Alexa Web Information Service – Traffic Metrics for any Website, available from ⟨https://aws.amazon.com/jp/awis/⟩ (accessed 2018-02-23).

[25] MAXMIND: IP Geolocation and Online Fraud Prevention, available from ⟨https://www.maxmind.com/en/home⟩ (accessed 2018-02-23).

[26] dmoz: DMOZ – The Directory of the Web, available from ⟨http://dmoztools.net/⟩ (accessed 2018-02-23).

[27] hpHosts: hpHosts Online – Simple, Searchable & FREE!, available from ⟨http://www.hosts-file.net/⟩ (accessed 2018-02-23).

[28] SPAMHAUS: The Spamhaus Project, available from ⟨http://www.spamhaus.org/⟩ (accessed 2018-02-23).

[29] Malware Domain List: MDL, available from ⟨http://www.malwaredomainlist.com/⟩ (accessed 2018-02-23).
[30] aguse：aguse.jp：ウェブ調査，入手先 ⟨https://www.aguse.jp/⟩（参照 2018-02-23）.
[31] SMARTBEAR: The Cost of Poor Web Performance, available from ⟨https://smartbear.com/blog/test-and-monitor/the-cost-of-poor-web-performance-infographic/⟩ (accessed 2019-03-29).
[32] Gao, H., Yan, J., Cao, F., Zhang, Z., Lei, L., Tang, M., Zhang, P., Zhou, X., Wang, X. and Li, J.: A Simple Generic Attack on Text Captchas, *23rd Network and Distributed System Security Symposium* (*NDSS '16*) (2016).
[33] Guixin, Y., Zhanyong, T., Dingyi, F., Zhanxing, Z., Yansong, F., Pengfei, X., Xiaojiang, C. and Wang, Z.: Yet Another Text Captcha Solver: A Generative Adversarial Network Based Approach, *Proc. 2018 ACM SIGSAC Conference on Computer and Communications Security* (*CCS '18*), pp.332–348 (2018).
[34] Sujata, G., Niels, P., Monica, C. and Aviel, D.R.: A framework for detection and measurement of phishing attacks, *Proc. 2007 ACM Workshop on Recurring Malcode*, pp.1–8 (2007).
[35] McGrath, D.K. and Gupta, M.: Behind Phishing: An Examination of Phisher Modi Operandi, *Proc. 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats* (*LEET '08*), pp.1–8 (2008).
[36] Yue, Z., Jason, I.H. and Lorrie, F.C.: CANTINA: A Content-Based Approach to Detecting Phishing Web Sites, *Proc. 16th International Conference on World Wide Web* (*WWW '07*), pp.639–648 (2007).
[37] Guang, X. and Jason, I.H.: A hybrid phish detection approach by identity discovery and keywords retrieval, *Proc. 18th International Conference on World Wide Web* (*WWW '09*), pp.571–580 (2009).
[38] Pawan, P., Manish, K., Ramana, R.K. and Minaxi, G.: PhishNet: Predictive Blacklisting to Detect Phishing Attacks, *Proc. 29th Conference on Information Communications* (*INFOCOM'10*), pp.346–350 (2010).
[39] Liu, D., Li, Z., Du, K., Wang, H., Liu, B. and Duan, H.: Don't Let One Rotten Apple Spoil the Whole Barrel: Towards Automated Detection of Shadowed Domains, *Proc. 2017 ACM SIGSAC Conference on Computer and Communications Security* (*CCS '17*), pp.537–552 (2017).
[40] Suphannee, S., Kangkook, J., Yixin, S., Lauri, K.P., Zhichun, L., Cristian, L., Lu-An, T. and Ding, L.: Countering Malicious Processes with Process-DNS Association, *26th Network and Distributed System Security Symposium* (*NDSS '19*) (2019).
[41] Kalyan, V., Ignacio, A. and Vamsi, K.: AI^2: Training a Big Data Machine to Defend, *Proc. IEEE Int'l Conf. on Big Data Security'16* (2016).
[42] Das, S., Wong, W.K., Dietterich, T., Fern, A. and Emmott, A.: Incorporating Expert Feedback into Active Anomaly Discovery, *Conference: Conference: 2016 IEEE 16th International Conference on Data Mining* (*ICDM*), pp.853–858 (2016).
[43] Das, S., Wong, W.K., Fern, A., Dietterich, T. and Siddiqui, A.: Incorporating Feedback into Tree-based Anomaly Detection, *KDD 2017 Workshop on Interactive Data Exploration and Analytics* (*IDEA'17*) (2017).
[44] Pevny, T.: Loda: Lightweight on-line detector of anomalies, *Machine Learning*, Vol.102, No.2, pp.275–304 (2015).
[45] Alina, O., Zhou, L., Robin, N. and Kevin, B.: MADE: Malicious Activity Detection in Enterprises, *Proc. 34th Annual Computer Security Applications Conference* (*ACSAC2018*), pp.124–136 (2018).

藤井 翔太 （正会員）

2016 年岡山大学大学院自然科学研究科電子情報システム工学専攻修士課程修了．同年（株）日立製作所システムイノベーションセンタ入所．以来，ネットワークセキュリティ技術に関する研究開発に従事．

鬼頭 哲郎 （正会員）

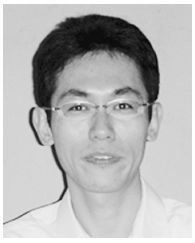
2005 年東京大学大学院情報理工学系研究科電子情報学専攻修士課程修了．同年（株）日立製作所システム開発研究所（現，システムイノベーションセンタ）に入所．以来，ネットワークセキュリティ技術に関する研究開発に従事．現在，セキュリティ事業統括本部にて日立グループ内のセキュリティ監視およびインシデントレスポンス業務に従事．

重本 倫宏 （正会員）

2006 年大阪大学大学院基礎工学研究科システム創成専攻修士課程修了．同年（株）日立製作所システム開発研究所（現，システムイノベーションセンタ）入所．以来，ネットワークセキュリティ技術に関する研究開発に従事．現在，システムイノベーションセンタセキュリティ研究部主任研究員．明治大学大学院先端数理科学研究科先端メディアサイエンス専攻博士後期課程在籍．

仲小路 博史 （正会員）

2001 年東京理科大学大学院理工学研究科情報科学専攻修士課程修了．同年（株）日立製作所システム開発研究所（現，システムイノベーションセンタ）入所．サイバー攻撃対策技術の研究開発に従事．2017 年明治大学大学院先端数理科学研究科現象数理学専攻博士後期課程修了．同年より Hitachi Europe Ltd. European R&D Centre の Senior Researcher としてフィジカルセキュリティの研究開発に従事．博士（理学）．



藤井 康広

2001 年東京大学大学院理学系研究科博士課程修了（物理学）．同年（株）日立製作所システム開発研究所（現，システムイノベーションセンタ）入所．以来，情報セキュリティ技術の研究開発に従事．現在，（株）日立製作所ライフ事業統括本部兼日立オートモティブシステムズ（株）にて，コネクテッドサービス事業の企画立案および自動車セキュリティ標準化活動・法制化対応に従事．博士（理学）．