ブラックリストに基づく検出の効率化に向けた悪性DNSクエリ分類手法

佐藤 彰洋[1,a)]　中村 豊[1,b)]　小倉 光貴[1,c)]　野林 大起[1,d)]　池永 全志[1,e)]

概要：マルウェアはインターネットにおける重大な脅威のひとつである．ネットワーク内の感染端末を検出するためには，ブラックリストを利用した通信の監視が一般的である．しかしながら，ブラックリストに基づく検出は，(1) ブラックリストは必ず幾つかの誤りを含むこと，(2) 検出結果の正誤の判断が困難であることが問題となる．本稿では，ブラックリストによる検出結果の効率的な分析のため，悪性 DNS クエリ分類手法を提案する．本手法は，従来のドメイン文字列による表層的な類似性に基づく分類とは異なり，悪性クエリとそれに付随するクエリ群が潜在的に示す原因に基づく分類を実現する．実験により，ブラックリストにより検出された 388 のクエリを 3 のクラスタに分類できること，各クラスタが共通の原因のクエリのみで構成されることを確認した．

キーワード：マルウェア，ブラックリスト，DNS クエリ，機械学習

A Malicious DNS Query Clustering Approach for Blacklists based Detection

Akihiro Satoh[1,a)]　Yutaka Nakamura[1,b)]　Mitsutaka Ogura[1,c)]　Daiki Nobayashi[1,d)]　Takeshi Ikenaga[1,e)]

Abstract: Malware is some of the most serious threats to network security. One common way for detecting infected machines in a network is by monitoring communications based on blacklists. However, the detection is problematic in that (1) none of the blacklists is completely reliable, and (2) a blacklist doesn't provide the sufficient evidence to determine the validity and accuracy of detection results. In this paper, we propose a malicious DNS query clustering approach for blacklists based detection. Unlike conventional classification based on the superficial similarity of character strings in domain names, our approach realizes cause-based classifications latently indicated by malicious queries and their accompanying queries. In experiments, we confirmed that this approach could classify the 388 malicious queries detected through blacklists into the 3 clusters consisting of queries with common cause.

Keywords: Malware, Blacklist, DNS Query, Machine Learning

1. はじめに

マルウェアはインターネットにおける重大な脅威のひとつである．サイバー犯罪者は，マルウェアに感染した端末を介して，機密情報窃取，フィッシング詐欺，標的型攻撃などの悪意ある活動を試みる．米 McAfee 社の報告によると，約 30 万のマルウェアが日々誕生しており，それによる 2017 年の被害総額は 6000 億ドルを超える [1]．そのため，マルウェアに対抗するための技術の確立が急務である．

マルウェアによる被害の抑止のため，管理者は自身のネットワークに内在する感染端末を迅速に排除する必要がある．ネットワーク内の感染端末を検出するためには，ブラックリストを利用した通信の監視が一般的である．マルウェアによる通信先の変更に対する迅速な追従のため，これまでに機械学習を用いたブラックリストの自動生成などの試みがなされてきた [2], [3]．しかしながら，ブラックリ

[1] 九州工業大学
Kyushu Institute of Technology, 1-1 Sensuicho, Tobata, Kitakyushu, 804–8550, Japan
[a)] satoh@isc.kyutech.ac.jp
[b)] yutaka-n@isc.kyutech.ac.jp
[c)] ogura.mitsutaka307@mail.kyutech.ac.jp
[d)] nova@ecs.kyutech.ac.jp
[e)] ike@ecs.kyutech.ac.jp

ストに基づく検出は，(1) ブラックリストは必ず幾つかの誤りを含むこと，(2) 検出結果の正誤の判断が困難であることが問題となる．故に，単純にブラックリストに合致するか否かでは済まず，管理者による通信の調査と原因の特定が必須となる．

本稿では，ブラックリストに基づいて検出された悪性 DNS クエリを原因ごとに分類することを目指す．DNS に着目した理由は，マルウェアによる通信に先んじて必ず名前解決が生じるためであり，そのようなマルウェアに起因する名前解決を悪性 DNS クエリと定義する．この分類の実現により，分析の必要な悪性 DNS クエリの数を大幅に削減することが可能となる．本稿の構成は次の通りである．まず，2 章で既存研究とその問題点を整理する．3 章で悪性 DNS クエリ分類手法を提案した後，4 章で提案手法の有効性を議論する．最後に 5 章で本研究の貢献と課題を纏める．

2. 関連研究

Soldo らは，複数の参加者から提供される過去の攻撃ログに基づいて，新たにブラックリストを生成する方法を提案している [4]．一方，Sun らが開発した AutoBLG [2] や，Rahbarinia らが開発した Segugio [3] は，既存のブラックリストから新たにブラックリストを自動生成するシステムである．これらの違いは，自動生成のために AutoBLG がウェブのクローリング結果を利用するのに対し，Segugio は受動的なトラフィックの観測結果を利用する点である．このように，ブラックリストの高度化については頻繁な研究が行われており，現在もネットワークにおける脅威防御戦略の中核を成している．

Kheir らは，ブラックリストには考慮すべき量の誤りが含まれることを示した [5]．これはクラウドやホスティング，ダイナミック DNS，広告ネットワークなど，良性と悪性が混在するドメインが原因と考えられる．加えて，自動生成されたブラックリストはこの問題を深刻化する要因と成り得る．著者らは，複数のブラックリストを相互照合することにより，ブラックリストの正確性の改善を試みている．しかしながら，ブラックリストはそれぞれの保護範囲が異なるため，それらの相互照合は保護範囲を大幅に狭めることになる．

Kührer らは，未登録のドメイン，パーキングドメイン，シンクホールドメインなどを考慮して，19 種のブラックリストの有効性を評価した [6]．その評価には，良性と悪性で予めラベル付けされたデータセットのみが用いられている．この理由は，通信の詳細な調査が必要となること，検出数に比例して調査範囲が拡大することから，ブラックリストによる検出結果の検証に煩雑さが伴うためである．この問題については，文献 [7] でも同様の指摘がなされている．

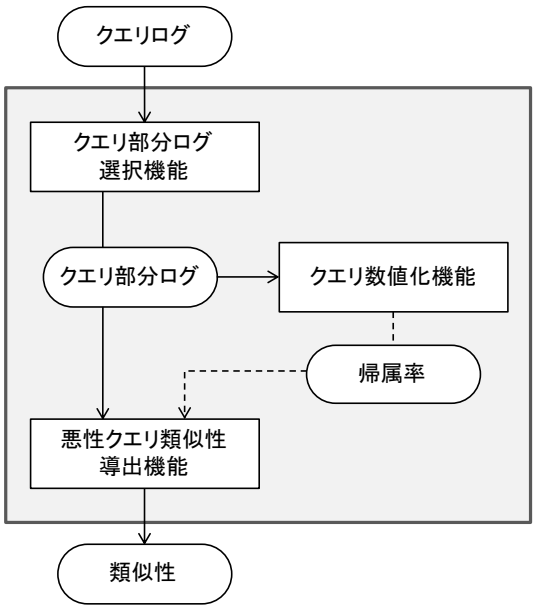


図 1 悪性 DNS クエリ分類手法の概要

これまでに，DNS のクエリとレスポンスの分析に着目した幾つかの研究が発表されている．Wang らは，クエリの類似性に応じた端末の分類と，そのクエリの統計特性に応じた悪性の判別を実現する DBod を開発した [8]．しかしながら，このシステムはドメイン生成アルゴリズムに基づくマルウェアに特化しており，他に対しては全く効果が期待できない．Berger らは，クエリにおけるアドレスとドメインの関係の変化を継続的に学習する DNSMap を開発した [9]．このシステムは，階層構造を考慮してドメイン文字列の類似性を導出する機能を有すが，その表層的な類似性では，悪性クエリの分類において十分な性能を発揮できない．

3. 提案

本稿では，ブラックリストによる検出結果の効率的な分析のため，悪性 DNS クエリ分類手法を提案する．その独自性は，(1) ブラックリストと合致したクエリの前後には，その原因の推定を助けるクエリ群が存在することに着目したこと，(2) クエリの数値表現のため，クエリ間の共起関係を利用したこと，(3) クエリに対する重みの付与のため，一般的なマルウェアの性質，すなわち同一マルウェアファミリに感染した端末は，共通の悪性ドメイン群と繰り返し通信する性質を考慮したことの 3 点である．これにより，従来のドメイン文字列による表層的な類似性に基づく分類とは異なり，悪性クエリとそれに付随するクエリ群が潜在的に示す原因に基づく分類を実現する．加えて，ネットワークへの導入が非常に容易な点を留意すべきである．具体的には，DNS の名前解決にのみ着目するため，本手法の導入はネットワーク内の再帰 DNS に対する名前解決，または名前解決のログを転送するのみである．

```
11-Jan-2018 05:00:53.265 queries: info: client 192.168.10.240#35704 ↩
                (210.152.241.100.in-addr.arpa): query: 210.152.241.100.in-addr.arpa IN PTR + (192.168.0.1)
11-Jan-2018 05:00:53.467 queries: info: client 192.168.20.120#54171 ↩
                (smtp.kyutech.ac.jp): query: smtp.kyutech.ac.jp IN SOA + (192.168.0.1)
11-Jan-2018 05:00:53.470 queries: info: client 192.168.20.120#54311 ↩
                (www.ipsj.or.jp): query: www.ipsj.or.jp IN A + (192.168.0.1)
11-Jan-2018 05:00:53.470 queries: info: client 192.168.10.240#49193 ↩
                (ieeexplore.ieee.org): query: ieeexplore.ieee.org IN AAAA + (192.168.0.1)
11-Jan-2018 05:00:53.473 queries: info: client 192.168.30.100#54015 ↩
                (analytics.google.com): query: analytics.google.com IN TXT + (192.168.0.1)
```

図 2 再帰 DNS に対するクエリログの例

図 1 に提案手法の概要を示す．本手法は，(1) クエリ部分ログ選択機能，(2) クエリ数値化機能，(3) 悪性クエリ類似性導出機能により構成される．以降，各機能の詳細について述べる．

3.1 クエリ部分ログ選択機能

本機能は，ブラックリストとの比較により，クエリログから悪性クエリを検出する．次いで，悪性クエリを含む前後をクエリ部分ログとして選択する．この理由は，ブラックリストと合致したクエリの前後には，その原因の推定を助けるクエリ群が存在するためである．

本機能への入力となるクエリログは，ネットワーク内の端末群から再帰 DNS に対する名前解決要求の記録である．図 2 に，アドレスが 192.168.0.1 の再帰 DNS に対するクエリログの例を示す．これにおいて，各クエリは発生時間，送信元アドレス，ドメイン名，ドメインのクラスやタイプなどの値を持つ．留意すべきは，クエリログにおけるドメイン名をプライマリドメイン名に短縮する点である．プライマリドメイン名は登録可能な最高レベルのドメイン名である．具体的には，`www.ipsj.or.jp` と `smtp.kyutech.ac.jp` のプライマリドメインは，それぞれ `ipsj.or.jp` と `kyutech.ac.jp` となる．

先ず，クエリ x_n^* のドメイン名がブラックリスト L_B のエントリと合致した場合，クエリ x_n^* を悪性として検出する．ここで，クエリログから検出された悪性クエリの数を N とする．次いで，悪性クエリ x_n^* と同一送信元アドレス且つその前後 t_α 秒未満のクエリ群を，クエリ部分ログ X_n として抽出する．本機能の出力は，N の悪性クエリから得られた N のクエリ部分ログの集合 $\mathbb{X}$ となる．

3.2 クエリ数値化機能

本機能は，Word2Vec とソフトクラスタリングの 2 種類の機械学習手法により，クエリ間の共起関係に基づいたクエリの数値化を試みる．この理由は，全クエリ部分ログに含まれる膨大な数のクエリの特徴を効率的に表現するためである．

共起関係に基づいたクエリの分散表現のために，全クエリ部分ログ $\mathbb{X}$ に Word2Vec を適用する．ここで分散表現とは，1 つのデータをベクトル空間の 1 点に対応付ける表現方法を意味する．Word2Vec は，自然言語処理の分野で注目されている分散表現のための手法で，「文中の各単語はその周辺語と強い関係性がある」という仮定に基づき各単語の特徴をベクトルで表現する [10]．文中の単語をクエリ部分ログ中のクエリに対応付けるための，我々による Word2Vec の改良は，(1) 単語に代わりクエリのドメイン名を用いたこと，(2) 共起関係を測る周辺語の範囲を単語数から時間，t_β 秒未満に変更したことである．

次いで，クエリの分散表現にガウス混合モデル基づくソフトクラスタリングを適用する．ソフトクラスタリングは，各データポイントが各クラスタに属する確率を導出する手法である [11]．各クラスタには，共起関係が類似したクエリ，すなわち機器間の通信において担うタスクが類似したクエリの集約が期待できる．次式に，本機能の出力となるクエリの帰属率を示す．

$$\vec{p}(x_i) = \left(p(c_1|x_i), \cdots, p(c_m|x_i), \cdots, p(c_M|x_i) \right)$$

ここで，M はクラスタの数，$p(c_m|x_i)$ はクエリ x_i がクラスタ c_m に属する確率を意味する．

3.3 悪性クエリ類似性導出機能

本機能は，クエリ部分ログに含まれるクエリの帰属率から特徴ベクトルを導出する．その特徴ベクトルの類似性をコサイン距離で比較することにより，悪性クエリとそれに付随するクエリ群が潜在的に示す原因に基づく分類を実現する．留意すべきは，一般的なマルウェアの性質を考慮して，複数のクエリ部分ログに共通して出現するクエリを重視した点である．

上述のように，帰属率はクエリが通信において担うタスクを暗に意味する．加えて，クエリ部分ログにおいて，そのクエリ群が担うタスクの類似性は，その原因の類似性に強く依存すると考えられる．そこで，クエリの帰属率の加重和からクエリ部分ログの特徴ベクトルを導出する．

$$\vec{X}_n = \sum_{x_i \in X_n} w_\alpha(x_i)\ w_\beta(x_i)\ w_\gamma(x_i)\ \vec{p}(x_i)$$

この式において，クエリ x_i のドメイン名がホワイトリスト L_W に含まれる場合は $w_\gamma = 0$，それ以外は 1 となる．ま

た，一般的なマルウェアの性質を考慮したクエリ x_i の重み，w_α と w_β を次式で示す．

$$w_\alpha(x_i) = \frac{|\mathcal{F}_{addr}(x_i,\mathbb{X}) \cap \mathcal{F}_{name}(x_i,\mathbb{X})|}{|\mathcal{F}_{addr}(x_i,\mathbb{X})|}$$
$$w_\beta(x_i) = \frac{|\mathcal{F}_{list}(x_i,\mathbb{X}) \cap \mathcal{F}_{name}(x_i,\mathbb{X})|}{|\mathcal{F}_{list}(x_i,\mathbb{X})|}$$

ここで，$\mathcal{F}_{name}(x_i,\mathbb{X})$，$\mathcal{F}_{addr}(x_i,\mathbb{X})$，$\mathcal{F}_{list}(x_i,\mathbb{X})$ は集合 $\mathbb{X}$ の 3 つの異なった部分集合である．$\mathcal{F}_{name}(x_i,\mathbb{X})$ は，クエリ x_i と同一ドメイン名のクエリを含むクエリ部分ログから成る集合である．一方，$\mathcal{F}_{addr}(x_i,\mathbb{X})$ と $\mathcal{F}_{list}(x_i,\mathbb{X})$ は，クエリ x_i を含むクエリ部分ログ X_n と比較して，同一送信元アドレスのクエリ部分ログと，ブラックリストの同一エントリに合致したクエリ部分ログから成る集合である．また，$|\cdot|$ は集合の要素数を意味する．本機能の出力は，クエリ部分ログの特徴ベクトルをコサイン距離で比較することで得られた悪性クエリの類似性である．

4. 評価

本章では，実験を通じた提案手法の評価により，悪性クエリを原因ごとに分類できること，それによりブラックリストの検出結果を効率的に分析できることを示す．4.1 節で実験の諸元について述べた後，4.2 節と 4.3 節で結果について議論する．

4.1 諸元

実験に用いたデータセットは，キャンパスネットワークで 2017 年 1 月から 2018 年 2 月までに観測されたクエリ群である．ブラックリスト L_B は一般公開されている 3 種を [13], [14], [15]，ホワイトリスト L_W は自ネットワークのドメインに加え，Alexa が公開するアクセス数の上位 1,000,000 件を採用した [16]．

提案手法における各種パラメタを，経験的に $t_\alpha = 90$，$t_\beta = 1.0$ に設定した．また，Word2Vec の次元数，学習係数，反復数をそれぞれ 100，0.0005，250,000 に設定した．これらの最適化は今後の課題とする．

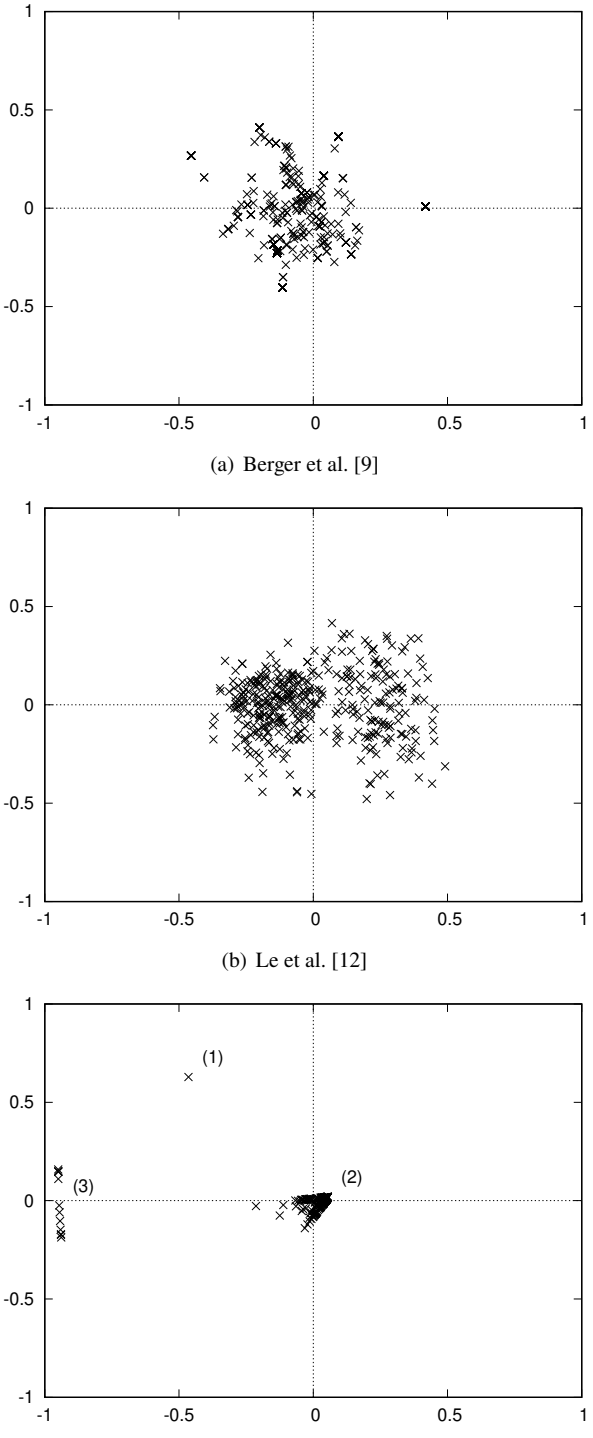
提案手法との比較のため，文献 [9], [12] を参考に悪性クエリを分類する 2 つの手法を実装した．第 1 の実装は，ドメイン文字列の類似性を用いた手法であり，第 2 の実装は，Word2Vec の最も有名な拡張のひとつである Doc2Vec を用いた手法である．Word2Vec が単語の特徴ベクトルを導出するのに対し，Doc2Vec は単語群，すなわち文章の特徴ベクトルを導出する．Doc2Vec の各種パラメタは，共起関係を測る周辺語の範囲を 5，その他を提案手法と同値に設定した．これらの比較のために用いた機器の構成を，表 1 に示す．

表 1 評価に用いた機器の構成

CPU	Intel(R) Core(TM) i5-4460 3.20GHz
RAM	DDR3 16GB
GPU	NVIDIA GeForce GTX 750 Ti
Kernel	Linux 3.10.0-693.11.1.el7.x86_64
Software	TensorFlow 1.3.0, CUDA Toolkit 8.0 with cuDNN 6.0

4.2 分類精度の比較

文献 [17], [18] では，多くのマルウェアが TXT タイプのクエリを介して通信することが報告されている．そこで，ドメイン名がブラックリストのエントリと合致する，且



(a) Berger et al. [9]
(b) Le et al. [12]
(c) Our work

図 3 各手法における分類結果

表 2 各手法における計算時間 [sec]

Berger et al. [9]	Le et al. [12]	Our work
1.505	1418.474	2074.312

つドメインタイプが TXT のクエリを悪性とした．データセットから検出されたのは，総クエリ数が 388，その総ドメイン数が 158 である．

図 3 に実験結果を示す．各手法により導出された悪性クエリの類似性を可視化するために，多次元尺度構成法を適用した．ここで，各シンボルは悪性クエリを，シンボル間の距離は悪性クエリの類似性を意味する．図 3(a) と 3(b) では，各シンボルが散乱しているため，これらから悪性クエリの類似性を判断することが困難である．2 つの実装において性能が低下したそれぞれの理由は，ドメイン文字列の表層的な類似性により悪性クエリを分類することの限界と，悪性クエリの前後に含まれる原因の推定に不要なクエリ群の影響である．一方，提案手法は，388 の悪性クエリを 3 のクラスタに明確に分類していることから，それによる分析の効率化が期待できる．

図 3(c) のクラスタ (1), (2), (3) に分類された悪性クエリの数は，それぞれ 1，12，375 であった．これら悪性クエリとそれに付随するクエリ群を，ウェブ検索や WHOIS，各種ドメイン評価サービスで調査することにより [19], [20], [21], その原因の特定を行なった．1 つの悪性クエリのみクラスタ (1) に分類された原因は，そのクエリの前後に殆ど通信が発生しなかったためである．具体的に，その期間に観測されたのは，総クエリ数が 7，その総ドメイン数が 2 のみであった．このような場合では，提案手法による類似性の導出は困難となる．クラスタ (2) では，悪性クエリの前後にドメインレピュテーションのための通信が多発していた．例えば，その通信先は `spamhaus.org`，`abuseat.org`，`barracudacentral.org` などである．このことから，クラスタ (2) の悪性クエリは，セキュリティ機器による通信を誤検出したことが原因と考えられる．一方，クラスタ (3) では，悪性クエリの前後に BitTorrent の Tracker に対する通信が発生していた．例えば，その通信先は `opentrackr.org`, `asnet.pw`, `blackunicorn.xyz` などである．ブラックリストに登録のあるドメインへの通信であること，P2P を介して通信するマルウェアが報告されていることから [22], [23]，これらの悪性クエリはマルウェアの感染が原因と考えられる．上述の結果，すなわち提案手法による分類において各クラスタが共通の原因のクエリのみで構成されることから，それによる分類の正確性が確認できた．

4.3 計算時間の比較

表 2 に各手法の計算時間を示す．ここで留意すべきは，事前の処理が可能であるため，クエリ部分ログの選択に要する時間は除外した点である．2 つの実装が 1.505 秒と 1418.474 秒であるのに対し，提案手法は最も悪く 2074.312 秒となった．提案手法で計算時間を必要とするのは，クエリ数値化機能における Word2Vec の処理であり，全体の約 70 %を占めていた．この結果を踏まえ，その高速化のための改良点について調査を行った．

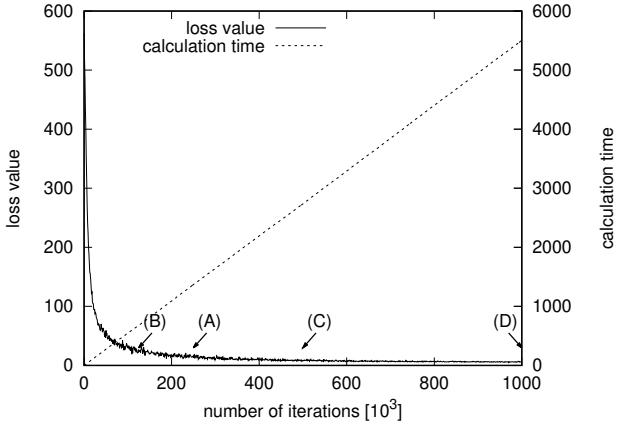


図 4 反復数に対する損失値と計算時間の関係

図 4 に Word2Vec における損失値，計算時間，反復数の関係を示す．ここで，損失値は共起関係の導出における予測値と正解値との乖離の総和である．図中の実線が損失値と反復数の関係を，破線が計算時間と反復数の関係を示す．この破線の関係が線形であることから，計算時間の短縮のために反復数を抑えることの重要性が読み取れる．図中の矢印 (A) は，これまでの実験で用いた反復数である 250,000 の箇所を，矢印 (B)，(C)，(D) は，それぞれ 125,000，500,000，1,000,000 の箇所を指している．矢印 (A)，(B)，(C)，(D) における損失値は，それぞれ 12.844，25.892，8.585，5.876 であり，矢印 (B) の箇所で損失値がほぼ収束していることが見て取れる．また，矢印 (B)，(C)，(D) の箇所の分類結果を確認したところ，それらと矢印 (A) の箇所の分類結果との間に差異は見られず，図 3(c) と同等の結果であった．この理由は，この処理に次ぐソフトクラスタリングにより，多少の共起関係の差は吸収されるためである．この結果から，損失値を基準に反復を終了することで，計算時間を大幅に短縮できることが明らかになった．

5. おわりに

本稿では，ブラックリストによる検出結果の効率的な分析のため，悪性 DNS クエリを原因ごとに分類することを試みた．また実験を通じて，ブラックリストに基づいて検出された 388 のクエリを 3 のクラスタに分類できること，各クラスタが共通の原因のクエリのみで構成されることを確認した．これにより，ネットワークに内在するマルウェアへの迅速な対処が可能となるため，ネットワークの運用において安全性の向上が期待できる．

今後の予定は，ブラックリストの検出を TXT タイプのクエリに限らず，他のクエリにも提案手法を適用することで，その有効範囲を明らかにする．また，マルウェアの通信に関係しないクラスタを除外するための，新たな機能の追加を検討する．

謝辞 本研究は JSPS 科研費 JP18K11296 の助成を受けたものである．また，本研究の一部は東北大学電気通信研究所 共同プロジェクト研究によるものである．ここに深く謝意を示す．

参考文献

[1] J. A. Lewis: Economic Impact of Cybercrime — No Slowing Down, https://www.csis.org/analysis/economic-impact-cybercrime (2018).
[2] B. Sun et al.: Automating URL Blacklist Generation with Similarity Search Approach, *IEICE Transactions on Information and Systems*, Vol. E99.D, No. 4, pp. 873–882 (2016).
[3] B. Rahbarinia et al.: Efficient and Accurate Behavior-Based Tracking of Malware-Control Domains in Large ISP Networks, *ACM Transactions on Privacy and Security*, Vol. 19, No. 2, pp. 4:1–4:31 (2016).
[4] F. Soldo et al.: Blacklisting Recommendation System: Using Spatio-Temporal Patterns to Predict Future Attacks, *IEEE Journal on Selected Areas in Communications*, Vol. 29, No. 7, pp. 1423–1437 (2011).
[5] N. Kheir et al.: Mentor: Positive DNS Reputation to Skim-Off Benign Domains in Botnet C&C Blacklists, *Proceedings of the International Conference on ICT Systems Security and Privacy Protection*, pp. 1–14 (2014).
[6] M. Kührer et al.: Paint It Black: Evaluating the Effectiveness of Malware Blacklists, *Proceedings of the International Symposium on Research in Attacks, Intrusions and Defenses*, pp. 1–21 (2014).
[7] L. Bilge et al.: Exposure: A Passive DNS Analysis Service to Detect and Report Malicious Domains, *ACM Transactions on Information and System Security*, Vol. 16, No. 4, pp. 14:1–14:28 (2014).
[8] T. S. Wang et al.: DBod: Clustering and Detecting DGA-based Botnets using DNS Traffic Analysis, *Computers & Security*, Vol. 64, pp. 1–15 (2017).
[9] A. Berger et al.: Mining Agile DNS Traffic using Graph Analysis for Cybercrime Detection, *Computer Networks*, Vol. 100, pp. 28–44 (2016).
[10] T. Mikolov et al.: Distributed Representations of Words and Phrases and their Compositionality, *Advances in Neural Information Processing Systems*, pp. 3111–3119 (2013).
[11] L. Scrucca et al.: mclust 5: Clustering, Classification and Density Estimation using Gaussian Finite Mixture Models, *The R Journal*, Vol. 8, No. 1, pp. 205–233 (2016).
[12] Q. Le et al.: Distributed Representations of Sentences and Documents, *Proceedings of the International Conference on Machine Learning*, pp. 1188–1196 (2014).
[13] DNS-BH: https://www.malwaredomains.com.
[14] hpHosts Online: https://hosts-file.net.
[15] abuse.ch: https://abuse.ch.
[16] Alexa: http://www.alexa.com.
[17] H. Ichise et al.: Analysis of DNS TXT Record Usage and Consideration of Botnet Communication Detection, *IEICE Transactions on Communications*, Vol. E101.B, pp. 70–79 (2018).
[18] J. Christian et al.: On Botnets that Use DNS for Command and Control, *Proceedings of the European Conference on Computer Network Defense*, pp. 9–16 (2011).
[19] VirusTotal: https://www.virustotal.com.
[20] SimilarWeb: https://www.similarweb.com.
[21] NetValuator: http://www.netvaluator.com.
[22] R. Cuevas et al.: TorrentGuard: Stopping Scam and Malware Distribution in the BitTorrent Ecosystem, *Computer Networks*, Vol. 59, pp. 77–90 (2014).
[23] A. D. Berns et al.: Searching for Malware in BitTorrent, *University of Iowa Computer Science Technical Report*, pp. 1–10 (2008).