HOKKAIDO UNIVERSITY

| | |
|---|---|
| Title | 階層的な機械学習を用いたDoHトラフィック解析における悪意のあるDNSトンネルツールの識別 |
| Author(s) | 三橋, 力麻; 金, 勇; 飯田, 勝吉; 品川, 高廣; 高井, 昌彰 |
| Citation | 電子情報通信学会技術研究報告, 121(300), 85-92 |
| Issue Date | 2021-12-09 |
| Doc URL | http://hdl.handle.net/2115/86958 |
| Type | article |
| File Information | IA2021-48.pdf |

Instructions for use

Hokkaido University Collection of Scholarly and Academic Papers : HUSCAP

一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
IA2021-48(2021-12)

# 階層的な機械学習を用いたDoHトラフィック解析における悪意のあるDNSトンネルツールの識別

三橋　力麻[†,††]　金　　勇[†††]　飯田　勝吉[††]　品川　高廣[††††]　高井　昌彰[††]

† 東京大学大学院情報理工学系研究科　〒113–8658 東京都文京区弥生 2–11–16
†† 北海道大学　〒060–0811 北海道札幌市北区北 11 条西 5 丁目
††† 東京工業大学　〒152–8550 東京都目黒区大岡山 2–12–1
†††† 東京大学　〒113–8658 東京都文京区弥生 2–11–16
E-mail: †mitsuhashi@os.ecc.u-tokyo.ac.jp, ††{iida,ytakai}@iic.hokudai.ac.jp, †††yongj@gsic.titech.ac.jp,
††††shina@ecc.u-tokyo.ac.jp

**あらまし**　DNS over HTTPS（DoH）プロトコルは, プライバシー保護や改ざん防止などが期待できる一方で, マルウェアや悪意のある DNS トンネルツールによって生成されたトラフィックの検知が困難になる問題がある. 本研究では機械学習技術を用いた階層的な分類方法により, Web アクセスなど一般的な HTTPS トラフィックからの DoH トラフィックの分別に加え, DoH トラフィックを生成した悪意のある DNS トンネルツールを識別するシステムを提案する.
**キーワード**　DNS over HTTPS, DoH, トラフィック分類, 機械学習, 悪意のある DNS トンネルツールの識別

# Recognition of Malicious DNS Tunnel Tools by DoH Traffic Analysis Using Multi-stage Machine Learning Technology

Rikima MITSUHASHI[†,††], Yong JIN[†††], Katsuyoshi IIDA[††], Takahiro SHINAGAWA[††††], and Yoshiaki TAKAI[††]

† Graduate School of Information Science and Technology, the University of Tokyo　2–11–16 Yayoi, Bunkyo-ku, Tokyo, 113–8658 Japan
†† Hokkaido University　Kita11, Nishi5, Kita-ku, Sapporo, Hokkaido, 060–0811 Japan
††† Tokyo Institute of Technology　2–12–1 Ookayama, Meguro-ku, Tokyo, 152–8550 Japan
†††† The University of Tokyo　2–11–16 Yayoi, Bunkyo-ku, Tokyo, 113–8658 Japan
E-mail: †mitsuhashi@os.ecc.u-tokyo.ac.jp, ††{iida,ytakai}@iic.hokudai.ac.jp, †††yongj@gsic.titech.ac.jp,
††††shina@ecc.u-tokyo.ac.jp

**Abstract**　The DNS over HTTPS (DoH) protocol is expected to protect privacy of Internet users and prevent data tampering. However, the DoH also causes a problem in that network administrators can hardly detect the suspicious DoH traffic generated by malware and malicious DNS tunnel tools. In this research, we propose a machine learning based system using multi-stage classification model in order to not only filter the DoH traffic from the HTTPS traffic such as web access but also recognize the malicious DNS tunnel tools that generates the suspicious DoH traffic.
**Key words**　DNS over HTTPS, DoH, Network traffic classification, Machine learning, malicious DNS tunnel tool recognition

## 1. はじめに

プライバシー保護や改ざん防止などを目的として, インターネット上の DNS トラフィックを暗号化しようという機運が高まっている. 暗号化の手法としては, SSL/TLS プロトコルを使用し, RFC8484 [1] で標準化されている DNS over HTTPS（DoH）が有望視されている. そのため近年, DoH をサポートするソフトウェアが急速に増加している. 例えば, Firefox, Chrome, Edge などの主要な Web ブラウザでは, DoH が既に実装されている. クライアントシステムでは, Cloudflare Tunnel (cloudflared) [2], DNS-over-HTTPS [3], DNS Over HTTPS Proxy [4], DNSCrypt [5], doh-client [6] などのプロキシソフトウェアをインストールする



This article is a technical report without peer review, and its polished and/or extended version may be published elsewhere.
Copyright ©2021 by IEICE

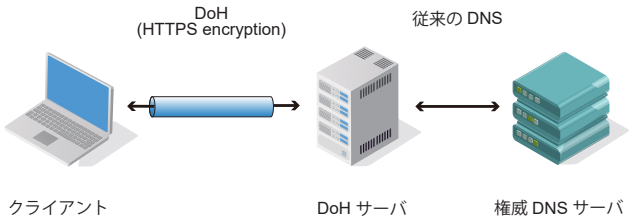


図 1　DoH によるドメイン名解決

ことで, DoH を利用できるようになる. さらに OS レベルでは, Windows 11, MacOS11, iOS14 などの最新バージョンで DoH が利用可能となっている. DoH によるドメイン名解決を図 1に示す. DoH は, クライアントと DoH サーバとの間で DNS トラフィックを暗号化する. また, DoH サーバはインターネット上の権威 DNS サーバとの間で従来の DNS プロトコルを使用してドメイン名解決を行う.

DoH を用いて DNS トラフィックを暗号化することにより, インターネットユーザは訪問したウェブサイト名を盗聴されるなどのリスク低減が期待できる. 一方で DoH は, ネットワーク管理者がセキュリティサービスを提供することを目的としたトラフィック分析が困難になる問題がある. 例えば, マルウェアが DoH を利用してインターネット上の Command and Control (C&C) サーバと通信した場合, 接続先のドメインが暗号化されているため, 接続先を検知することが難しくなる. 既に DoH を用いて C&C サーバと通信するマルウェアの存在が確認されている [7].

こうした状況下で, 悪意のある DoH トラフィックを生成するアプリケーションを特定することは, ネットワーク管理者にとって重要といえる. 悪意のある DoH トラフィックを生成するアプリケーションを特定することによって, 1) アプリケーションが外部 Web サイトへ接続する通信を遮断すること, 2) アプリケーションをダウンロードするサイトへのアクセスをブロックすること, 3) アプリケーションの使用を禁止するセキュリティ・ポリシーを作成することなどが可能になる.

我々は悪意のある DoH トラフィックを生成するアプリケーションとして, DNS トンネルツールに着目した. DNS トンネルツールは, クライアントとサーバ間で DNS プロトコルを使って一般的な通信 (HTTP や SSL など) を行うことが可能なプログラムである. DNS トンネルツールを利用することにより, クライアントはファイアウォールを越えて, インターネット上のサーバと通信することができる. また, クライアントがマルウェアに感染した場合, 攻撃者に自由に操作される危険性もある. 先行研究では, HTTPS トラフィックの中から DNS トンネルツールによって生成された DoH トラフィックを検出するアプローチが提案されている [8] [9]. しかし, それらの提案では, DoH トラフィックを生成した DNS トンネルツールを特定することはできない.

機械学習技術は, DNS トンネルツールの特徴に応じて DoH トラフィックを自動的に分類することができるため, DNS トンネルツールの識別を実現するための有効な手段となり得る. しかしながら, DNS トンネルツールの特徴を学習するためには, DNS トンネルツールを実際に稼働させて, 長期間にわたって大量のデータを収集する必要がある. また, DNS トンネルツールを識別するための特徴量は, パケットヘッダ, パケット番号, パケット長, パケット方向, パケット間の到着間隔などを使って, DoH トラフィックの中から有効に機能するものを探し出す必要がある. このように機械学習技術を用いて DNS トンネルツールを識別するためには時間と手間がかかるが, ひとたび機械学習モデルが稼働すれば DoH トラフィックを自動的に解析することができる.

本研究では, DoH トラフィックを分類する機械学習技術を用いて, 悪意のある DNS トンネルツールを識別する新しいシステムを提案する. 図 2に示すように, 提案システムは階層的なネットワークトラフィックの分類を用いる. 提案システムの特徴的なプロセスは, 悪意のある DNS トンネルツールを特定する第 3 段階にある. 各段階の分類に適したモデルをパラメータチューニングによって作成することで, より精度の高い DNS トンネルツールの識別を目指す.

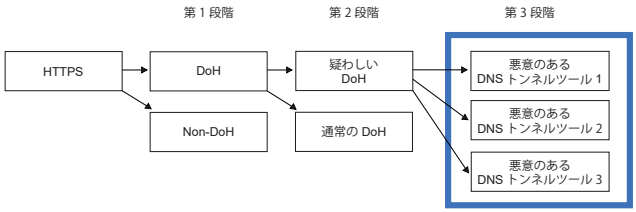


図 2　階層的なネットワークトラフィック分類の概要

提案システムを CIRA-CIC-DoHBrw-2020 データセット [10] で評価したところ，第 1 段階で HTTPS から DoH トラフィックを抽出したところ, Accuracy は 99.81%，F-score は 99.87% であった．第 2 段階で DoH トラフィックから疑わしい DoH トラフィックを検出したところ，Accuracy と F-score はそれぞれ 99.99% だった. また, 第 3 段階で疑わしい DoH トラフィックから悪意のある DNS トンネルツールを識別したところ, Accuracy は 97.22% であり，F-score は 95.19% だった. このように, DoH のトラフィックを解析することで, 悪意のある DNS トンネルツールを特定できる可能性を示した報告は, 我々の知る限り初めてのものである. 本報告は ISC2021 [11] の国際会議で発表した内容に対して, DoH の最新動向, 評価結果の考察, 今後の予定等を加筆したものである.

## 2. 関 連 研 究

### 2.1 ネットワークトラフィックの分類

ネットワークトラフィックの分類は，現在, 非常に活発な研究分野である．特に，機械学習技術を用いたアプローチが数多く提案されている [12]．また，暗号化されたネットワークトラフィックの分類についても多くの報告がある [13]．しかし，DoH 技術は歴史が浅く，まだ完全には実用化されていないた

め，DoH のトラフィック分類に関する研究報告はサーベイ論文等にはとんど含まれていない．

最近報告された DoH ネットワークの分類に関する研究を見ると, D. Vekshin ら [14] は, 機械学習を用いて HTTPS トラフィックを分類することで その中に含まれる DoH トラフィックを抽出した．また, DoH トラフィックを分類することで, Chrome, Firefox, Cloudflare などの DoH クライアントを特定した．機械学習モデルには Ada-boost を使用し，99.9% の分類精度を得た．データセットには，Alexa [15] が提供する上位 100 万の Web サイトを使って, ドメイン名へのアクセスデータを収集した．M. MontazeriShatoori ら [8] は，機械学習技術を用いて HTTPS トラフィックから DoH トラフィックを抽出し，その後, DoH トラフィックを良性の DoH トラフィックと悪性の DoH トラフィックに分類する仕組みを提案した．この悪性の DoH トラフィックとは, DNS トンネルツールによって生成されたトラフィックと定義している．どちらの分類にも Random forest モデルを使用し, 前者は 99.3% の F-score, 後者は 99.9% の F-score を得た．評価には CIRA-CIC-DoHBrw-2020 データセットを使用した．S. K. Singh ら [9] は，CIRA-CIC-DoHBrw-2020 データセットを用いて，良性および悪性の DoH トラフィックを分類する際の精度を向上させた．彼らは，Gradient boosting モデルを使って, ホールドアウト法で 100 %の分類精度を得た．

### 2.2 DNS トンネルツールの検知

DNS に対する攻撃手法や対策については, これまで多くの研究が報告されており [16] [17] [18] [19], 機械学習の発展に伴って DNS のトンネル検出の研究分野は近年注目を集めている．DNS によるドメイン名解決はインターネット上で必要不可欠なサービスであるため, DNS プロトコルを用いた DNS トンネルは, 攻撃者が C&C ノードを構築したり, 機密データを流出させたりするための共通的な手法となっている [20].

DNS トンネルの検出に関する最近の報告として, P.Yang ら [21] は, スタッキング・モデルを用いて DNS トンネルツールが生成した DNS トラフィックの検出を試みた．DNS トラフィックは, dns2tcp, dnscat2, DeNiSe, Heyoka などのツールを用いて生成した．機械学習モデルは, K 近傍法（K-NN）, サポートベクターマシン（SVM）, ランダムフォレストの 3 つを組み合わせたスタッキングモデルを使用した．

A.L. Buczak ら [20] は, 侵入テストの結果から特徴量を抽出し, ランダムフォレストを用いて, 正常な DNS トラフィックと DNS トンネルのトラフィックを区別した．D.Lambion ら [22] は, 畳み込みニューラルネットワーク（CNN）, ランダムフォレストおよびアンサンブル分類器を使用して, DNS トラフィックから悪意のある DNS トンネルを検出した．実験では一日分の実トラフィックデータを分類器に読み込ませて, 性能と堅牢性を評価した．A.Chowdhary ら [23] は, DNS トラフィックから DNS トンネルツールが生成したクエリを検出するための二つの方法を発表した．一つ目の方法は DNS フルサービスリゾルバのキャッシュミスを利用するものであり, 二つ目の方法は機械学習技術を利用して, 与えられた DNS クエリを分類するものであった．

K. Wu ら [24] は, DNS クエリの文字の特徴量に基づいた 3 段階の DNS トンネル識別法である FTPB を導入した．第 1 段階では, FTPB は DNS トンネルによって生成された DNS クエリと, DGA によって生成された DNS クエリと区別する．第 2 段階では逆文書頻度（TF-IDF）を用いて, 抽出 DNS クエリをベクトルに変換した後, 主成分分析（PCA）を用いて次元を 2 に下げる．第 3 段階では Bagging with J48, ランダムフォレスト, Ada-boost, Gradient boosting などの機械学習分類器を用いて, バイナリベクトルを分類する．FTPB は 3 つの段階を用いてるが, その機能は我々の提案手法とは異なっている．

以上をまとめると, ネットワークトラフィックの分類と DNS トンネルツールの検知の研究分野ではいずれも, DoH トラフィックの分類によって悪意のある DNS トンネルツールを識別する手法については報告されていない．我々は, CIRA-CIC-DoHBrw-2020 データセットを用いた実験で, 第 1 段階および第 2 段階の分類において, 従来の手法と同等以上の精度を達成するとともに, 第 3 段階において, DNS トンネルツールの識別を目指す．

## 3. 設　　計

2 節では, ネットワークトラフィックの分類に関する関連研究を紹介し, DNS トンネルツールの検知手法を調査した．先に述べたとおり, ネットワーク管理者がネットワークセキュリティを維持するためには, DoH トラフィックの中から悪意のある DNS トンネルツールを特定する必要がある．本節では, 提案システムの設計について説明する．

### 3.1 システムの全体像

DoH トラフィックから悪意のある DNS トンネルツールを識別するために，階層的な分類手法を導入する．この手法のポイントは，各段階におけるネットワークトラフィックの分類に最も適した機械学習モデルを決定することである．図 3に示すように，提案システムは複数のブロックで構成されている．以下では各ブロックの詳細を説明する．

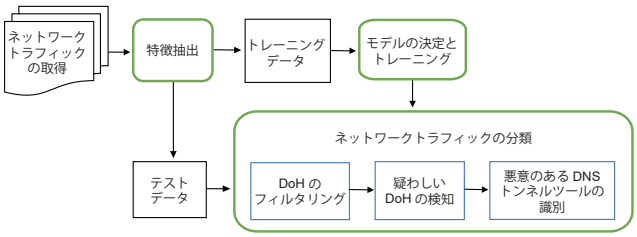


図 3　提案システムの全体像

### 3.2 ネットワークトラフィックの取得と特徴抽出

DoH は SSL/TLS プロトコルを用いて DNS トラフィックを暗号化しているため, 提案するシステムは HTTPS トラフィックを入力データとする．ネットワーク上には様々な種類のトラフィックが流れているが, パケットの送信元または送信先のポート番号から, HTTPS によって生成されたトラフィックであるかどうかを判定することができる．HTTPS トラフィックは図 4に示したポイントで収集する．クライアントが Web サーバに接続する目的は, web コンテンツを取得することである．また, クラ

イアントはドメイン名を解決するために, DoH サーバを経由して通常の DNS サーバに接続する. クライアント上の悪意のある DNS トンネルツールは, 攻撃の指示を受けたり, 機密情報を送信したりするために, 疑わしい DNS サーバに接続する可能性がある.

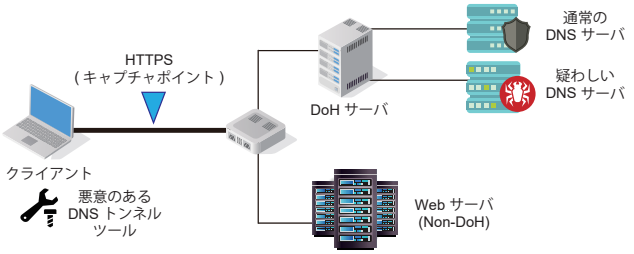


図 4　ネットワーク接続とキャプチャポイント

取得したネットワークトラフィックを機械学習モデルで分類するために，双方向通信の HTTPS トラフィックフローを定義する. トラフィックフローは，送信元 IP アドレス，送信先 IP アドレス，送信元ポート番号，送信先ポート番号で決定される. これらはパケットのヘッダに含まれている情報であり, 暗号化されていないため利用できる. トラフィックフローの統計的な特徴量は, 例えば, パケット数, パケットの方向, パケットの到着時間, パケットの長さなどから抽出する. ただし, パケットのペイロードは暗号化されているため, その内容は利用しない.

### 3.3　モデルの決定と学習

本節では, 悪意のある DNS トンネルツールを識別するための階層的な分類で使用する機械学習モデルを決定する方法について説明する. 提案システムは XGBoost [25], LightGBM [26], CatBoost [27] を使用する. これらのライブラリはいずれも GBDT（Gradient Boosting Decision Tree）アルゴリズムを用いている. また, S.R ら [28] によると，これらのライブラリは現在のところ，他の機械学習アルゴリズムと比べて，柔軟なパラメータチューニングが可能となっている. また, 高い分類精度が期待できるため，Kaggle [29] などの機械学習コンテストで広く用いられている.

機械学習技術で高い分類精度を得るためには，高性能な機械学習ライブラリの使用に加えて，データセットに適したパラメータチューニングが重要である. ネットワークトラフィック分類に適した機械学習モデルを決定する方法を図 5に示す. まず，パラメータチューニングしたモデルを使ってトレーニングデータを学習する. 次に，学習済モデルを用いて検証データを分類する. そこで得られた分類精度を比較することで，そのデータセットに最適なパラメータチューニングモデルを決定することができる. そして，決定されたモデルは，学習データと検証データを使って再度学習し, テストデータの分類器として使用される.

ここで注意したいのが, オーバーフィッティングの問題である. オーバーフィッティングとは，あるデータセットに特化した機械学習モデルが，それ以外の追加データを正確に分類できなくなることを意味する. つまり, パラメータチューニングしたモデルが検証データにオーバーフィットした場合, そのモデルはテストデータを十分な精度で分類できなくなる. したがって, パラメータチューニングの良し悪しは, 検証データの分類結果だけではなく, テストデータの分類結果も用いて分析する必要がある.

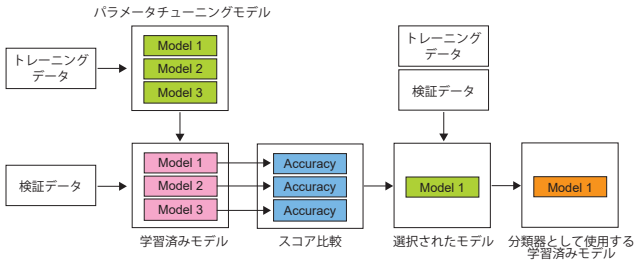


図 5　トラフィック分類に適したモデルの決定プロセス

各モデルのチューニングに使用したパラメータを表 1に示した. これらのパラメータは，各モデルのドキュメントで分類精度の向上に効果があるとされているものである [30] [31] [32]. パラメータの値は，デフォルトの値から一定の範囲を広げて作成した. そして, 検証データを最も高い精度で分類できる組み合わせをグリッド探索で見つけることにした. パラメータが異なるモデルとして, XGBoost に 35 種類, LightGBM に 56 種類, CatBoost に 48 種類, 合計 139 種類のモデルを用意した.

表 1　グリッドサーチのパラメータ（下線はデフォルト値））

| XGBoost | LightGBM | CatBoost |
|---|---|---|
| max_depth:<br>2, 4, 6, 8, 10, 12, 14 | num_leaves:<br>7 ,15, 31, 63, 127, 255, 511 | max_depth:<br>2, 4, 6, 8, 10, 12, 14, 16 |
| max_bin:<br>128, 256, 512, 1024, 2048 | max_bin:<br>127, 255, 511, 1023, 2047, 4095, 8191, 16383 | l2_leaf_reg:<br>1, 2, 3, 4, 5, 6 |

### 3.4　ネットワークトラフィックの分類

提案システムでは, 悪意のある DNS トンネルツールを特定するために, ネットワークトラフィックを 3 段階で分類する. 各段階のネットワークトラフィックを詳細に分析することで, より正確な分類が可能となる. 各段階で使用する分類器は, 3.3 節のプロセスで決定されたものを使用する. 分類器の決定は 3 段階のそれぞれで行うため, 結果的に 3 つの分類器が決定される.

図 6に 3 段階の分類器の入力データと出力データを示す. 第 1 段階の学習済み分類器にテスト・データが入力されると，DoH トラフィックと non-DoH トラフィックを分類する. 第 2 段階の学習済み分類器は DoH トラフィックから疑わしい DoH トラフィックを検出する. 第 3 段階の学習済み分類器は疑わしい DoH トラフィックを生成した悪意のある DNS トンネル・ツールを識別する. 第 3 段階に関しては, ozymanDNS, DeNiSe, Heyoka, DNScapy など, 数多くの DNS トンネルツールが存在することに注意する必要がある. リスクベースのアプローチの観

点からは, リスクの高いものを識別し, 段階的に対象を増やしていくのが良いと考えられる. そこで, 本研究ではまず, 知名度が高く頻繁に使用される DNS トンネルツールである dns2tcp [33], dnscat2 [34], iodine [35] の識別に焦点を当てることにした.

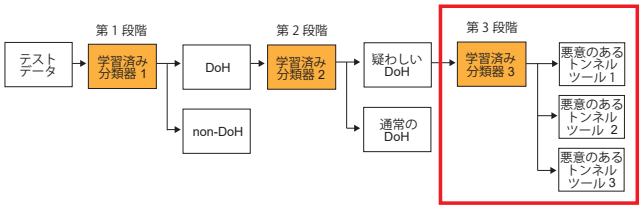


図 6　3 つの段階における入出力データ

## 4. 評　　価

3 節では，提案システムの全体像を示し，階層的な分類に使用する機械学習モデルの決定方法を説明し，各段階での分類対象を説明した. 本節では，提案システムを実装した際の分類性能を評価し，重要な特徴量を分析する.

### 4.1 実　　装

3 節で提案した設計をもとに，提案システムを以下のように実装した. ハードウエア環境としては，一台のマシンを用意し, Intel Xeon Silver 4210R の CPU, 96 ギガバイトのメモリ, Nvidia GeForce RTX 3080 の GPU を搭載した. ソフトウェア環境は, Ubuntu 20.04 に singularity 3.7.3 と Nvidia TensorFlow Release21.02 Container を導入した. 使用した機械学習ライブラリは, XGBoost 1.3.3, LightGBM 3.2.1, CatBoost 0.25.1 である. なお, XGBoost と CatBoost は GPU 上で, LightGBM は CPU 上で実行した.

### 4.2 Dataset

実験では，CIRA-CIC-DoHBrw-2020 データセットを使用した. このデータセットに含まれるラベル数とトラフィック量を表 2に示す. このデータセットは各段階のトラフィック量に偏りがあるため，全体の分類結果だけでなく，トラフィック量が少ない場合の分類結果を把握することが重要である. 表 3に示すように，データセットから 29 個の統計的な特徴量を抽出する.

表 2　データセットのラベルとトラフィック量

| | ラベル | トラフィック量 |
|---|---|---|
| 第 1 段階 | Non-DoH | 897494 |
| (DoH の抽出) | DoH | 269643 |
| 第 2 段階 | normal DoH | 19807 |
| (疑わしい DoH の検知) | suspicious DoH | 249836 |
| 第 3 段階 | dns2tcp | 167486 |
| (悪意のある DNS | dnscat2 | 35770 |
| トンネルツールの識別) | iodine | 46580 |

データセットのネットワークトラフィックを分類するために, 層化 10 分割交差検証を使用し，トレーニングデータとテスト

表 3　トラフィックの統計的な特徴量

| | | | |
|---|---|---|---|
| 1 | Number of Flow Bytes Sent | 17 | Standard Deviation of Packet Time |
| 2 | Rate of Flow Bytes Sent | 18 | Coefficient of Variation of Packet Time |
| 3 | Number of Flow Bytes Received | 19 | Skew from Median Packet Time |
| 4 | Rate of Flow Bytes Received | 20 | Skew from Mode Packet Time |
| 5 | Mean Packet Length | 21 | Mean Request/response Time Difference |
| 6 | Median Packet Length | 22 | Median Request/response Time Difference |
| 7 | Mode Packet Length | 23 | Mode Request/response Time Difference |
| 8 | Variance of Packet Length | 24 | Variance of Request/response Time Difference |
| 9 | Standard Deviation of Packet Length | 25 | Standard Deviation of Request/response Time Difference |
| 10 | Coefficient of Variation of Packet Length | 26 | Coefficient of Variation of Request/response Time Difference |
| 11 | Skew from Median Packet Length | 27 | Skew from Median Request/response Time Difference |
| 12 | Skew from Mode Packet Length | 28 | Skew from Mode Request/response Time Difference |
| 13 | Mean Packet Time | 29 | Duration |
| 14 | Median Packet Time | | |
| 15 | Mode Packet Time | | |
| 16 | Variance of Packet Time | | |

データを 9:1 の割合で分割する. 分類結果を測定する指標として，accuracy，recall，precision，F-score を用いる. それぞれの計算式は以下のとおりである. なお，第 3 段階は多クラス分類をするため, 各指標のマクロ平均値も用いる.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F\text{-}score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

ここで，$TP$ は真陽性，$FP$ は偽陽性，$FN$ は偽陰性，$TN$ は真陰性を意味する. さらに，分類の指標として，mean time between false alarms (MTBFA) を使用する. MTBFA は，モニタリングにおける誤警報の平均的な間隔であり，以下の式で算出する. システムの分類精度が不十分なために誤警報（偽陽性）の数が増えると, MTBFA は短くなる関係にある.

$$MTBFA = \frac{Monitoring\ hours}{Number\ of\ false\ alarms}$$

### 4.3 モデルの決定

表 4は, 検証データに対してグリッドサーチを行って得られた最大 Accuracy とパラメータを示している．ここに示したパラメータ値は, 探索範囲の最大値と最小値の間に収まったため, これ以上探索範囲を広げる必要はないと判断した．各段階での分類精度を比較した結果，第 1 段階ではパラメータチューニングした XGBoost を, 第 2 段階ではパラメータチューニングした LightGBM を, 第 3 段階ではパラメータチューニングした CatBoost を使用することにした．なお，この実験では，第 3 段階の max depth: [14,16] と L2 leaf reg: [1,2,3,4,5,6] の組み合わせで GPU メモリが不足したため, CPU を使用して計算した.

表 4　グリッドサーチで得られた最大 Accuracy とパラメータ

| | | 第 1 段階 | 第 2 段階 | 第 3 段階 |
|---|---|---|---|---|
| XGBoost | accuracy | 0.9981 | 0.99998 | 0.9719 |
| | max_depth | 12 | 4 | 6 |
| | max_bin | 1024 | 512 | 1024 |
| LightGBM | accuracy | 0.9981 | 0.99997 | 0.9721 |
| | num_leaves | 255 | 15 | 63 |
| | max_bin | 511 | 255 | 8191 |
| CatBoost | accuracy | 0.9979 | 0.99999 | 0.9690 |
| | max_depth | 14 | 4 | 10 |
| | L2_leaf_reg | 5 | 2 | 4 |

表 5　テストデータを用いた DNS トンネルツールの識別

| | 分類器 | Accuracy | Precision | Recall | F-score | MTBFA |
|---|---|---|---|---|---|---|
| 第 1 段階 | XGBoost | 0.9981 | 0.9981 | 0.9994 | 0.9987 | 181 分 |
| | LightGBM | 0.9981 | 0.9980 | 0.9995 | 0.9987 | 111 分 |
| | CatBoost | 0.9979 | 0.9978 | 0.9995 | 0.9986 | 101 分 |
| 第 2 段階 | CatBoost | 0.9999 | 1.0 | 0.9999 | 0.9999 | 80683 分 |
| | LightGBM | 0.9999 | 0.9999 | 0.9999 | 0.9999 | 48410 分 |
| | XGBoost | 0.9999 | 0.9999 | 0.9999 | 0.9999 | 30256 分 |
| 第 3 段階 | LightGBM | 0.9722 | 0.9497 | 0.9543 | 0.9519 | 33 分 |
| | XGBoost | 0.9706 | 0.9473 | 0.9518 | 0.9495 | 32 分 |
| | CatBoost | 0.9691 | 0.9446 | 0.9494 | 0.9469 | 29 分 |
| 3 段階連続 | | 0.9703 | 0.9487 | 0.9503 | 0.9494 | 31 分 |

表 6　ホールドアウト法による先行研究との比較

| | 分類器 | Precision | Recall | F-score |
|---|---|---|---|---|
| 第 1 段階 | Random Forest [8] | 0.993 | 0.993 | 0.993 |
| | XGBoost (ours) | **0.9982** | **0.9995** | **0.9989** |
| 第 2 段階 | Random Forest [8] | 0.999 | 0.999 | 0.999 |
| | Gradient Boost [9] | 1.0 | 1.0 | 1.0 |
| | CatBoost (ours) | **1.0** | **1.0** | **1.0** |
| 第 3 段階 | LightGBM (ours) | 0.952 | 0.956 | 0.954 |

### 4.4 悪意のある DNS トンネルツールの識別結果

4.2 節で決定した分類器を用いてとテストデータを分類した結果を表 5に示す．第 3 段階で悪意のある DNS トンネルツールを識別した結果は，Accuracy が 97.22%，F-score が 95.19% であった．また，第 1 段階の DoH トラフィックのフィルタリングの結果は, Accuracy が 99.81%，F-score が 99.87% であった. 第 2 段階の疑わしい DoH トラフィックの検出の結果は, Accuracy が 99.99%，F-score が 99.99% であった．これらの結果から，提案システムの性能は，ネットワーク管理者によるネットワークセキュリティ維持の取り組みを支援するのに十分なものであることがわかる．また，提案手法で決定したモデルの結果は，他の最終候補モデルの結果と同等かそれ以上であったことから, パラメータチューニングに伴うオーバーフィッティングの問題は発生していないと考えられる.

第 2 段階の MTBFA を見ると, 第 1 段階, 第 3 段階に比べて非常に長い時間になっている. これは, 第 2 段階の分類精度が高いためであり, 大規模な実ネットワーク環境にも適用できると評価できる. 第 3 段階の MTBFA は約 30 分であったが, 提案システムを実ネットワークで使用するためには, もう少し長いことが望ましい. これは，第 3 段階での分類精度を向上させることで実現できる．3 段階連続の結果は, 第 1 段階と第 2 段階で発生した偽陽性と偽陰性の影響を第 3 段階で使用するデータセットに反映して計算した. 各段階の分類器は，第 1 段階は XGBoost, 第 2 段階は CatBoost，第 3 段階は LightGBM を使用した．第 1 段階と第 2 段階の分類精度は比較的高かったため，第 3 段階の評価指標はほとんど影響を受けなかった.

提案システムと先行研究のシステムとの性能比較を表 6に示す．先行研究はいずれも, データセットをトレーニングデータとテストデータに 1 回だけ分割して評価するホールドアウト法を用いている．これに対して，我々が用いた層化 10 分割交差検証は，トレーニングデータとテストデータを 10 回分割して評価を行い，その平均値を算出している．比較の基準を揃えるために, 我々は 10 回の評価の中から最も分類精度が良い結果を選択した. その結果，第 1 段階の分類では，Precision，Recall, F-score ともに, 本システムが最も高い値を示した．第 2 段階の分類では，既存の研究と同様に，Precision，Recall, F-score ともに 1.0 に達した．ホールドアウト法では，データセットをトレーニングデータとテストデータに分割する際のサンプルの選び方によっては, これらの指標値が低下する可能性があることに注意する必要がある．我々の提案システムにおける交差検証の結果は, Precision，Recall, F-score はそれぞれ 1.0，0.9999, 0.9999 であり，分類精度が大きく低下することはなかった. なお, 第 3 段階で悪意のある DNS トンネルツールを識別する試みは, 我々の知る限りではこれが初めてであり, 得られた結果は実用上十分な分類精度を提供できている.

### 4.5 重要な特徴量の分析

階層的なトラフィックデータの分類を可能にした背景を分析するために，各分類器が重要とみなした特徴量を表 7に示す. 第 1 段階では，XGBoost は “Mode Packet Length” を最も重要な特徴量として使用した. “Mode Packet Length” はトラフィック

フローのなかでもっとも頻繁に出現するパケットの長さを意味する. “Mode Packet Length” の値は次に続く “Mean Packet Time” の値よりもはるかに大きく, 前者の特徴量が非常に重要であることを示している. データセットに含まれる第 1 段階の “Mode Packet Length” の平均値は, non-DoH トラフィックでは 164.0 であり，DoH トラフィックでは 68.0 だった. この差は，non-DoH トラフィックにはサイズの大きなデータである web コンテンツが多く含まれていることによって生じている.

第 2 段階では, CatBoost は “Mode Packet Length” を最も重要な特徴量として使用した. データセットに含まれる第 2 段階の “Mode Packet Length” の平均値は，normal DoH トラフィックでは 74.1 であり，suspicious DoH トラフィックでは 67.5 であった. この差は，normal DoH トラフィックには，クライアントと DoH サーバーとの間で行われた SSL/TLS 鍵交換データが多く含まれていることによって生じている. 一方, 悪意のある DNS トンネルツールの多くは DoH サーバと長時間接続しているため, suspicious DoH トラフィックにはこのデータはほとんど含まれていない.

第 3 段階では, LightGBM は “Median Request/response Time Difference” を最も重要な特徴量として使用した. “Request/response Time Difference” は, DoH トラフィックの上で行われた DNS クエリに対する応答時間の間隔を意味している. データセットに含まれる第 3 段階の “Median Request/response Time Difference” の平均値は, dns2tcp が 0.2, dnscat2 が 2.7, iodine が 1.4 となっている. この差はクライアントからのパケット送信間隔と, 疑わしい DNS サーバの処理負荷によって生じていると推測される. なお, クライアントから疑わしい DNS サーバまでの地理的な距離がこの差の原因になっている可能性も検討したが, CIRA-CIC-DoHBrw-2020 データセットのドキュメント [10] によると, データセットに含まれる すべての DNS トンネルツールは, ローカルネットワーク環境に設置された 1 つの疑わしい DNS サーバに接続していたため, この仮説は却下した.

表 7　階層的な分類における重要な特徴量

| | 重要な特徴量 | 値 |
|---|---|---|
| 第 1 段階 | Mode Packet Length | 0.7757 |
| | Mean Packet Time | 0.0819 |
| 第 2 段階 | Mode Packet Length | 68.9465 |
| | Median Packet Length | 13.5604 |
| 第 3 段階 | Median Request/response Time Difference | 3694 |
| | Skew from Median Request/response Time Difference | 3304 |

### 4.6 考　　察

本節では, 実施した評価に関する考察点を列挙する. まず, 攻撃者に利用される可能性が高いことを考慮して, 最も有名な DNS トンネルツールを評価に用いた. 4.3 節において, 我々は 3 つの DNS トンネルツールを高い精度で識別した. ただし, 悪意のある DNS トンネルツールには他にも様々な種類があり, その数は増加する可能性があることに注意する必要がある. 新しいツールを認識するためには定期的な更新が重要となるが, 提案システムを更新するまでに必要な期間の検証や, 更新が完了するまでの間に必要となる補完機能の導入については, 今後の課題としている.

次に, ローカルネットワーク環境で評価を行い, 提案システムの有効性を確認した. 4.5 節では, DoH トラフィックの上で行われた DNS クエリに対する応答時間の間隔を特徴量として, 3 つの悪意のある DNS トンネルツールを区別できることを示した. 実験に使用したデータでは, 各 DNS トンネルツールが接続する疑わしい DNS サーバは, 共通のネットワーク内にあり, 性能仕様も同様であった。そのため、これと同一条件下ではこの特徴量を用いた分類はうまく機能すると考えられるが, 条件が変わった場合の影響については, 実ネットワークを使って今後調査する必要がある. 実ネットワーク環境での評価については, 提案システムを学内ネットワークに導入し, その有効性を確認する予定である.

一方で, よく知られた DNS トンネルツールの一部が改変をしたり, 新たな機能を追加したりする攻撃者が現れた場合, それらのツールは提案システムの対象外となる可能性がある. さらに, DoH サーバへの接続時間や DoH サーバの処理性能はサービス事業者ごとに異なる. そのため, 提案システムを実ネットワークに導入するためには, これらの環境変化を把握できるような特徴量を追加する必要がある.

## 5. おわりに

DoH は, DNS トラフィックを暗号化することで, インターネットユーザにセキュリティとプライバシーを提供するために開発された. しかし, DoH には, ネットワーク管理者がネットワークセキュリティを確保するためのトラフィック分析が困難になる問題がある. 暗号化されたネットワークトラフィックの分類や DNS トンネルの検出に関する研究は数多く報告されているが, DoH は新しいプロトコルであるため, これまでの研究成果をそのまま適用することはできない.

本研究では, 悪意のある DNS トンネルツールによって生成された DoH トラフィックを特定することを試みた. DoH トラフィックのペイロードは暗号化されているため, パケットペイロードに依らない統計的な特徴量を利用してトラフィックを詳細に分析した．我々のアプローチは階層的なトラフィック分類であり，各段階でのネットワークトラフィック分類に適したパラメータチューニングモデルを使用する．3 段階の階層的なネットワークトラフィック分類を行うシステムを設計，実装，評価した．プロトタイプでは，高い分類精度が期待できる機械学習ライブラリである XGBoost，LightGBM，CatBoost のパラメータを調整し，139 種類のモデルを用意した．本システムが悪意のある DNS トンネルツールを識別できることを証明し, その性能を評価するために, CIRA-CIC-DoHBrw-2020 データセットを用いて一連の実験を行った. その結果, 本システムは 97.22% の精度で悪意のある DNS トンネルツールを識別した. また, HTTPS トラフィックから DoH トラフィックを 99.81% の精度で抽出し, DoH トラフィックから疑わしい DoH トラフィッ

クを 99.99% の精度で検出できることを示した．また，ネットワークトラフィックの分類時に機械学習モデルが重視した特徴量を示し，高い分類精度が得られた理由を議論した．さらに，実施した評価に関する考察を述べた．

## 文　献


[1] P. Hoffman and P. McManus, "DNS Queries over HTTPS (DoH)," RFC 8484, Oct. 2018.

[2] "Cloudflare tunnel (cloudflared)". https://developers.cloudflare.com/cloudflare-one/connections/connect-apps.

[3] "DNS-over-HTTPS". https://github.com/m13253/dns-over-https.

[4] "DNS Over HTTPS Proxy". https://github.com/facebookarchive/doh-proxy.

[5] "DNSCrypt". https://github.com/DNSCrypt.

[6] "doh-client". https://docs.rs/crate/doh-client/1.1.5.

[7] "First-ever malware strain spotted abusing new DoH (DNS over HTTPS) protocol". https://www.zdnet.com/article/first-ever-malware-strain-spotted-abusing-new-doh-dns-over-https-protocol/.

[8] M. MontazeriShatoori, L. Davidson, G. Kaur, and A. Habibi Lashkari, "Detection of DoH Tunnels using Time-series Classification of Encrypted Traffic," Proceedings of 2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), pp.63–70, 2020.

[9] S.K. Singh and P.K. Roy, "Detecting Malicious DNS over HTTPS Traffic Using Machine Learning," Proceedings of 2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies, pp.1–6, 2020.

[10] "CIRA-CIC-DoHBrw-2020". https://www.unb.ca/cic/datasets/dohbrw-2020.html.

[11] R. Mitsuhashi, A. Satoh, Y. Jin, K. Iida, T. Shinagawa, and Y. Takai, "Identifying Malicious DNS Tunnel Tools from DoH Traffic Using Hierarchical Machine Learning Classification," Proceedings of the 24th Information Security Conference (ISC), no.71, pp.••–••, 2021.

[12] F. Pacheco, E. Exposito, M. Gineste, C. Baudoin, and J. Aguilar, "Towards the Deployment of Machine Learning Solutions in Network Traffic Classification: A Systematic Survey," IEEE Communications Surveys Tutorials, vol.21, no.2, pp.1988–2014, 2019.

[13] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapé, "Mobile Encrypted Traffic Classification Using Deep Learning: Experimental Evaluation, Lessons Learned, and Challenges," IEEE Transactions on Network and Service Management, vol.16, no.2, pp.445–458, 2019.

[14] D. Vekshin, K. Hynek, and T. Cejka, "DoH Insight: Detecting DNS over HTTPS by Machine Learning," Proceedings of the 15th International Conference on Availability, Reliability and Security (ARES), pp.••–••, 2020.

[15] "Amazon Alexa Voice AI". https://developer.amazon.com/en-US/alexa/.

[16] S. Ajmera and T.R Pattanshetti, "A Survey Report on Identifying Different Machine Learning Algorithms in Detecting Domain Generation Algorithms within Enterprise Network," Proceedings of 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp.1–5, 2020.

[17] H. Ichise, Y. Jin, and K. Iida, "Analysis of DNS TXT Record Usage and Consideration of Botnet Communication Detection," IEICE Transactions on Communications, vol.E101, no.1, pp.70–79, 2018.

[18] H. Ichise, Y. Jin, K. Iida, and Y. Takai, "NS record History Based Abnormal DNS traffic Detection Considering Adaptive Botnet Communication Blocking," IPSJ Journal of Information Processing, vol.28, pp.112–122, 2020.

[19] Y. Iuchi, Y. Jin, H. Ichise, K. Iida, and Y. Takai, "Detection and Blocking of DGA-based Bot Infected Computers by Monitoring NXDOMAIN Responses," Proceedings of 2020 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2020 6th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), pp.82–87, 2020.

[20] A.L. Buczak, P.A. Hanke, G.J. Cancro, M.K. Toma, L.A. Watkins, and J.S. Chavis, "Detection of Tunnels in PCAP Data by Random Forests," Proceedings of the 11th Annual Cyber and Information Security Research Conference (CISRC), pp.••–••, 2016.

[21] P. Yang, X. Wan, G. Shi, H. Qu, J. Li, and L. Yang, "Naruto: DNS Covert Channels Detection Based on Stacking Model," Proceedings of the 2020 The 2nd World Symposium on Software Engineering (WSSE), p.109–115, 2020.

[22] D. Lambion, M. Josten, F. Olumofin, and M. De Cock, "Malicious DNS Tunneling Detection in Real-Traffic DNS Data," Proceedings of 2020 IEEE International Conference on Big Data (Big Data), pp.5736–5738, 2020.

[23] A. Chowdhary, M. Bhowmik, and B. Rudra, "DNS Tunneling Detection using Machine Learning and Cache Miss Properties," Proceedings of 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), pp.1225–1229, 2021.

[24] K. Wu, Y. Zhang, and T. Yin, "FTPB: A Three-Stage DNS Tunnel Detection Method Based on Character Feature Extraction," Proceedings of 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp.250–258, 2020.

[25] C. Tianqi and G. Carlos, "XGBoost: A Scalable Tree Boosting System," Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, p.785–794, 2016.

[26] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," Proceedings of Advances in Neural Information Processing Systems, vol.30, pp.••–••, 2017.

[27] L. Prokhorenkova, G. Gusev, A. Vorobev, A.V. Dorogush, and A. Gulin, "CatBoost: unbiased boosting with categorical features," Proceedings of Advances in Neural Information Processing Systems, vol.31, pp.••–••, 2018.

[28] S. R, S.S. Ayachit, V. Patil, and A. Singh, "Competitive analysis of the top gradient boosting machine learning algorithms," Proceedings of 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), pp.191–196, 2020.

[29] "Kaggle". https://www.kaggle.com/.

[30] "XGBoost Documentation - Xgboost Parameters". https://xgboost.readthedocs.io/en/latest/parameter.html.

[31] "LightGBM Documentation - Parameters". https://lightgbm.readthedocs.io/en/latest/Parameters-Tuning.html.

[32] "CatBoost Documentation - Parameters". https://catboost.ai/en/docs/concepts/parameter-tuning.

[33] "dns2tcp". https://github.com/alex-sector/dns2tcp.

[34] "dnscat2". https://github.com/iagox86/dnscat2.

[35] "iodine". https://code.kryo.se/iodine/.