

Website Fingerprinting 攻撃による Hidden Service の特定

西村 怜二¹ 吉浦 紀晃¹

概要: 近年, インターネット利用時における匿名通信が注目を集めている. 最も普及している匿名通信システムとして Tor が挙げられる. Tor には, Tor ユーザと Web サイト, 両方の匿名性を実現している Hidden Service と呼ばれるものがある. この匿名性を悪用した犯罪が多発しており, 違法薬物の取引や児童ポルノの拡散の温床となっている. Tor を利用した違法行為を取り締まるために Tor ユーザのアクセス先を特定する技術が必要とされている. その技術の 1 つであり, Tor に対する最も主流な攻撃に Website Fingerprinting Attack (WF 攻撃) がある. Hidden Service ではない Web サイトに対しては既存の WF 攻撃が十分な成功を収めているが, Hidden Service に対する WF 攻撃の例は少ない. 本研究では, Hidden Service に対して k-Nearest-Neighbor Algorithm (k-NN) を用いた WF 攻撃と Convolutional Neural Network (CNN) を用いた WF 攻撃を行い, その有効性を確かめた. その結果, 学習にかかる時間や計算資源を考慮しないのであれば CNN を用いた WF 攻撃が Hidden Service に対して一番有効であることがわかった. また, データ整形やどのようなデータセットを扱うかにより WF 攻撃の精度は大きく変化することがわかった.

Identifying Hidden Services Using Website Fingerprinting Attack

REIJI NISHIMURA¹ NORIAKI YOSHIURA¹

Abstract: Anonymous communication has recently been getting attention when using the Internet. Tor is one of the most famous anonymous communication systems. Tor has what is called a Hidden Service, which allows both Tor users and websites to be anonymous. This anonymity has been exploited in a number of crimes, and has become a hotbed of illegal drug trafficking and the spread of child pornography. Prevention of these crimes requires technologies of identifying where Tor users are accessing. One of these technologies, the most common attack against Tor, is the Website Fingerprinting (WF) attack. Existing WF attacks have been sufficiently successful against non-Hidden Service websites, and there are few examples of WF attacks against Hidden Services. This paper tested the effectiveness of existing WF attacks using k-Nearest-Neighbor Algorithm (k-NN) and using Convolutional Neural Network (CNN) against the Hidden Service. As a result, this paper found that the CNN-based WF attack is the most effective against the hidden service without considering the learning time and computational resources. This paper also found that the accuracy of the WF attack varies greatly depending on the data format and the type of data set.

1. はじめに

近年, インターネットにおいて様々な情報を交換することは必要不可欠になった. しかし利便性が向上した反面, 情報を交換する上でメールアドレスや住所などといった個

人情報が漏洩する可能性が高まった. そこで情報を隠したい人がその情報を隠すことができる技術の需要が高まってきた. その際たる技術が暗号化通信である. 暗号化通信では通信の内容を暗号化し, たとえ通信を傍受されても内容が漏れないようになるという技術である. しかし暗号化通信では, ユーザがどの Web サイトを閲覧しているかの情報は秘匿されていない. この接続先の情報は, 匿名化通信を用いることで秘匿することができる. 最も普及してい

¹ 埼玉大学大学院理工学研究科数理電子情報部門情報領域
Department of Information and Computer Sciences, Saitama University

る匿名通信システムに The Onion Routing (Tor) がある。Tor は、自身の身元を明かすことなくインターネット上で活動することができるため、抑圧的な体制下にある政治活動家やジャーナリスト、内部告発者などが検閲や監視を掻い潜って情報を発信する場合や、健康相談や電子投票等の誰がどこに送信したかということを知られたくない場合など、様々な目的で利用されている。Tor には、更に通信の匿名性を高めた Hidden Service と呼ばれるものがある。しかし、Hidden Service には本来の目的と違い、違法薬物売買やパスポート偽装、違法薬物売買等の悪質な Web サイトも存在しており、そのサイトを利用して違法行為を行う Tor ユーザも存在する。このような Hidden Service を悪用した違法行為を取り締まるためには Tor ユーザのアクセス先を特定できる技術が必要とされている。本研究では、Hidden Service のデータを収集し、Tor ユーザのアクセス先を特定する攻撃である Website Fingerprinting Attack (WF 攻撃) を行う。

2. The Onion Routing (Tor)

Tor は Web 閲覧の際のプライバシーの保護と匿名化を目的としたオープンソースのソフトウェアであり、元々はアメリカ海軍調査研究所 (USNRL) によって開発された。現在は Tor Project によって開発が続いている [1]。

Tor は Onion Routing と呼ばれる暗号化通信方式と複数の Relay ノードと呼ばれる中継ノードを経由して通信を行うことで匿名性を保っている [2]。Tor の通信の仕組みを図 1 に示す。Tor ネットワークに接続するにはまず、Tor クライアント (ここでは Tor ユーザ) が Directory Authority (DA) と呼ばれるノードから Relay ノードの情報が含まれている Consensus File をダウンロードする。クライアントはこの中から通信を経由する Relay ノードを世界中からランダムに選択する (初期設定では 3 つ)。次にクライアントはそれぞれの Relay ノードと鍵共有した後、メッセージを多重に暗号化して通信を行う。この構成において、各ノード (クライアントとサーバ含む) は直接接続している前後のノードの IP アドレスのみを知ることができる。Tor ネットワークにおいてクライアントに一番近い Relay ノードは Entry ノードと呼ばれる。クライアントが接続するサーバのアドレスはわからないが、クライアントのアドレスを知っているため匿名化通信においては非常に重要なノードと言える。一方、サーバに一番近い Relay ノードを Exit ノードと呼び、間の Relay ノードは Middle ノードと呼ばれる。理論上、世界各国の Relay ノードそれぞれに対して順に情報の開示を要求することで通信元の特特定が可能だが現実的に不可能である。Tor ネットワーク内のノードは一定時間ごとに更新される。ノードが不正アクセスを受けていた場合でも一定時間でノードが更新されるため Tor ユーザの特特定は困難である。また Tor では、経由するノ

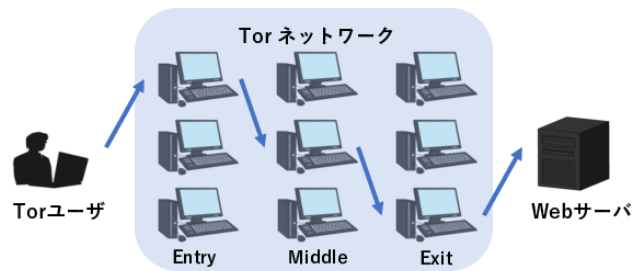


図 1 Tor ネットワーク

Fig. 1 Tor Network

ドの数、ノードを更新するまでの時間などを設定できる。Tor では上記のような通信の仕組みで匿名性を保っている。

2.1 Hidden Service

Hidden Service は Tor Network を用いて、サービスを提供しているサーバの IP アドレスを隠すシステムである。Tor ネットワークを経由した通信でなければアクセスすることができず、"半角英数字 16 桁. onion" という特殊なアドレスを持っている。図 2 に Hidden Service の全体図を示す。ユーザは Hidden Service にアクセスする際には RP を経由して通信を行う。各通信には全て上記した Tor ネットワークの通信方式で行われている。

- (1) Hidden Service のサーバが Tor ネットワークに IP (Introduction Point) を 3 つと Hidden Service の公開鍵を HSD (Hidden Service Directory) に通知
- (2) HSD は 3 つの IP と Hidden Service の公開鍵を登録、公開鍵から派生した 16 文字に ".onion" ドメインをつける。
- (3) ユーザは ".onion" のアドレスを知り、HSD から IP と Hidden Service の公開鍵の情報を得て、Tor ネットワーク内のノードからランダムに RP (Rendezvous Point) を設定する。
- (4) RP はワンタイムパスワードを設定し、その情報をユーザに送る。
- (5) ユーザは RP の IP アドレスと RP のワンタイムパスワードを Hidden Service の公開鍵で暗号化し、それを IP に送る。
- (6) IP は 5 で送られてきたものを Hidden Service に送る
- (7) Hidden Service は 6 で送られてきたものを自分の秘密鍵で復号する
- (8) Hidden Service から RP に RP のワンタイムパスワードを送る
- (9) RP はワンタイムパスワードを確認し、接続完了をユーザに通知する

Hidden Service ではこのようにして匿名性が保たれている。また、Hidden Service の Web サイトは作られてから 1 週

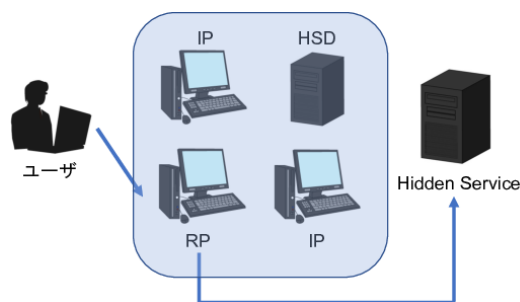


図 2 Hidden Service の全体図
Fig. 2 Overview of Hidden Service

間ほどで閉鎖されることもあれば、時間によっては Web サイトが稼働していない等、Web サイトが不安定であるために大量のデータ収集が困難である。通常の Web サイトの URL は Alexa のような大量の Web サイトのリストを公開しているもので集めることができる [3]。しかし、Hidden Service では大量の Web サイトのリストを公開しているものはない。Ahmia 等の Hidden Service のサイトを検索するための検索エンジンもあるが、全ての Hidden Service を網羅しているとは到底いえないのが現状である [4]。そのため、Hidden Service ではそもそも URL の収集すら困難であることが多い。

2.2 Tor に対する攻撃

本節では Tor の匿名性に対する既存の攻撃について説明する。

2.2.1 複数のノードが攻撃者の管理下にある場合の攻撃

1 章で述べた通り、Tor を悪用した違法行為を取り締まるためには Tor ユーザのアクセス先を特定する攻撃が必要とされている。Xiaogang らの研究や Marc らの研究では、高い特定率で Tor ユーザがどの Web サイトへアクセスしているかを特定することを可能とした [5][6]。これらの 2 つの手法に共通しているのは Entry ノードと Exit ノードが攻撃者の管理下にあるという仮定である。Xiaogang らの研究では、偽コンテンツを用意する。管理下の Exit ノードから Web サイトに要求があった場合、管理下の Exit ノードは偽コンテンツを混入させてパケットを中継する。この偽コンテンツにはリンク先のないイメージタグが挿入されているため、Web ブラウザは画像を取得しようとする。管理下の Entry ノードはこの時に発生する通信の特徴を利用して Tor ユーザと Web サイトを特定するという手法である。N. Levine らの研究は、通信は短い時間に行われることに注目した攻撃である。管理下においている Entry ノードが通信を経由した時間と Exit ノードが通信を経由した時間に強い相関が見られた場合、Entry ノードが通信しているクライアントと Exit ノードが通信しているサーバの 2 つが通信していると判定する。これらの研究は Tor の匿名

性に対する強力な攻撃を成立させているが、Tor ユーザが攻撃者の管理下にある Entry ノードと Exit ノードを同時に利用していなければならないため、実現性が低い。

2.2.2 Website Fingerprinting Attack (WF 攻撃)

WF 攻撃は現在 Tor に対する最も主流な攻撃であり、本研究においても使用した攻撃である。WF 攻撃は、Web サイトと通信した際に観測できるトラフィック・パターンを Website Fingerprint (WF) として、これを Web サイトを表す固有の特徴としてユーザがどの Web サイトへアクセスしているかを特定する攻撃である [7][8]。まず攻撃者は Tor を利用して Web サイトにアクセスし、その際に流れるトラフィックを観測する。そしてそのトラフィックから得られる情報 (パケットの総数やパケットの送信順序等) を Web サイト固有の情報として分類器 (Tor ユーザがどの Web サイトにアクセスしたかを分類) に学習させる。改めて攻撃対象となる Tor ユーザのトラフィックを観測し、分類器が事前に収集したデータと比較を行うことでユーザがどの Web サイトへアクセスしたかを分類し、そのサイトを特定する。この攻撃はデータの暗号化が行われている通信に対しても有効である。なお、Tor でのパケットは固定長にパディングされているため、トラフィックを観測して得られる情報はパケットの総数やパケット単体のパケット長、パケットの向き、時刻だけである。Tor では、ユーザは Entry ノードとしか通信を行わない。そのため、WF 攻撃を行う際は、ユーザの IP アドレスを知ることができるユーザと Entry ノード間の通信をキャプチャする必要がある。このような通信をキャプチャする方法は 2 つある。1 つ目は、ISP などのネットワーク上のパケットをキャプチャできる攻撃者がユーザと Entry ノード間のパケットをキャプチャする方法である。2 つ目は、攻撃者が Entry ノードを管理下に置き、そこを通るパケットをキャプチャする方法である。Tor ユーザが通信する Entry ノードはランダムに決定されるため攻撃者が設置したノードに接続する可能性は低い。そのため、前者の方法を用いるのが現実的な手法である。

3. 関連研究

本節では、本研究でも使用した WF 攻撃についての研究を紹介する。これまでの WF 攻撃のほとんどは、手動で抽出した特徴を使用して攻撃を実行している [9]。つまり、攻撃者は HTTP や Tor などのさまざまなプロトコルを慎重に調査し、ネットワークトレースから Web サイトを識別する可能性のある特徴 (パケットの総数やバースト情報等) を手動で決定する。これらの攻撃は多くの環境で高い精度を達成したが、攻撃者が手動で特徴を指定しなければならないため、異なるプロトコル間で攻撃を一般化することや、攻撃の強さを推論することは困難である。また、Tor

のバージョン変更やコンテンツの更新等による Web サイトの変化にも弱いため、実用性は低い。最近では、WF 攻撃に deep learning やニューラルネットワークといった機械学習も利用されてきている。ニューラルネットワークを利用する主な利点は、これまでの手動で特徴を決定するという手法とは違い、入力と出力のペアを分析するだけで、特徴を自動的に学習する能力を持っていることである。これにより、以前に知られていたよりも強力な特徴をカバーし、より高い精度を達成している。また、手動で特徴を指定する従来の方法と違い、自動的に特徴を捉えるため、Tor のバージョン変更や Web サイトの動的変化にも対応しやすい。WF 攻撃の精度の評価方法では、closed-world と open-world の 2 種類のシナリオについて議論されることが多い。closed-world は、ユーザが Monitored Site (アクセスを特定したい Web サイトの集合) の中の 1 つの Web サイトにアクセスするというシナリオである。学習させるデータは Monitored Site のデータのみである。例えば、Monitored Site が 10 個あるとすると、ユーザはそのうちの 1 つの Web サイトにアクセスを行う。攻撃者はユーザがアクセスしたサイトが Monitored Site のうちのどのサイトにアクセスしたかを特定する。open-world は、ユーザが Monitored Site 以外のあらゆる Web サイトにもアクセスするというシナリオである。学習させるデータは Monitored Site とそれ以外のあらゆる Web サイトである。open-world のシナリオの方が現実的であると言われることもあるが、Hidden Service においてはそうであるとは言えない。それは、あらゆる Web サイトのデータ収集というものが現実的ではないからである。そもそも Hidden Service ではなくとも、あらゆる Web サイトのデータ収集というものは今の広大なインターネットの世界においては難しい。特に、2.1 節で述べた通り Hidden Service では、その匿名性やサイトの稼働状況の不安定さから大量のデータの収集が困難である。また、Juarez らは、Monitored Site に含まれない Web サイトにもアクセスするという open-world のシナリオではそもそも WF 攻撃に実用性がないと指摘している [10]。

現在 Tor を用いて違法行為を行うユーザを WF 攻撃の結果のみから取り締まることは現実的に厳しい。実際に WF 攻撃を利用するには、ある程度 Tor ユーザがどの Web サイトにアクセスしているかの推測がついており、かつ違法行為を行った証拠の補強等に使用されることが考えられる。また、本研究の WF 攻撃の対象でもある Hidden Service は、人権運動組織や内部告発サイト、違法薬物や児童ポルノといった Tor ユーザが何か明確な目的を持ってアクセスされることが多いと考えられるため、Tor ユーザが多く Hidden Service の Web サイトにアクセスして違法行為を行うシナリオは少ないと想定される。これらの理由のため、本研究では closed-world かつ Monitored Site の数は少なくてもよいという前提に立つ。

3.1 k-NN を用いた WF 攻撃

Albert Known らは 1 つの Web サイトに 50 のデータ、50 の Hidden Service の Web サイトに対して k-NN を用いた WF 攻撃を行い、94.7%の精度を記録した。k-NN に用いた特徴は Wang らが用いた特徴と同様のものを利用しており以下の通りである [11][12]。

一般的な特徴

パケットの総送信サイズ、送信時間、受信パケット数、送信パケット数

パケットの順序

各送信パケットの位置

バースト情報

同じ方向のパケットが連続したというバースト情報。
受信バースト、送信バースト、最大バースト長

3.2 CNN を用いた WF 攻撃

Albert Known らは 1 つの Web サイトに 100 のデータ、100 の Hidden Service ではない Web サイトに対して CNN を用いた WF 攻撃を行い、97.8%の精度を記録した [13]。データセットには、Rimmer らの現存する最大かつ最新の WF データセットを利用した [14]。CNN のモデルには、特に画像認識の分野で優れた性能を発揮しているニューラルネットワークである Resnet18 を用いた [15]。また、畳み込みには通常の畳み込みではなく拡張畳み込みを用いている [16]。通常の畳み込みでは小さな領域を細かく見て特徴を捉えるが、拡張畳み込みを行うことにより広い領域を捉えることを可能にした。これによりパケットのような時系列データの特徴を捉えることに成功した。

4. 実験

4.1 実験概要

本研究では、Hidden Service に対して CNN を用いた WF 攻撃と k-NN を用いた既存の WF 攻撃を行う。2.1 節でも述べた通り、Hidden Service の Web サイトは稼働状況が不安定であるために既存のデータセットは存在しない。そのため、Hidden Service に対する WF 攻撃は次の 4 ステップで行う。

- (1) データ収集
- (2) データ整形
- (3) 分類器作成
- (4) 分類器の性能評価

4.2 データ収集

データ収集には、Tor 経由でのブラウジング自動化に firefox, python, selenium を使い、パケットキャプチャに tshark を用いた。まず Monitored Site とする Hidden Service の Web サイトを決めるために 200 のサイトにアクセスしパケットキャプチャを行った。この 200 サイト

は、Hidden Service の URL を探す際に確認できた Hidden Service の URL が載っている最大のリンクサイトである DeepLink Onion Directory に含まれていたサイトである [17]. Hidden Service は Web サイトの読み込みが遅いとはいえ、60 秒で読み込みが終わる．そのためパケットキャプチャはサイトにアクセスしてから 60 秒間行い、これを 1 個のデータとした．パケットキャプチャの結果、200 サイトのうち 123 サイトはパケット総数が 3000～4000 であることがわかった．既存の研究では、監視対象とする Web サイトのパケット総数に着目した WF 攻撃は行われていない．そこで本研究では監視対象とする Hidden Service の Web サイトを選ぶ際に 2 種類のデータセットを用いた．それぞれ、1 つのサイトにつき 300 個のデータを収集した．パケット総数類似サイト

複数のサイトどうしでのパケット総数が類似しているサイト、パケット総数は 3000～4000、サイトの数は 20
パケット総数が類似していないサイト

複数のサイトどうしでのパケット総数が類似していないサイト、パケット総数が 5000 のサイトもあればパケット総数が 10000 を超えるサイトもある．サイトの数は 20

この 2 種類のデータセットを用いることにより、パケット総数の特徴を利用しやすい場合とパケット総数の特徴を利用しにくい場合の WF 攻撃の精度を評価することが可能になる．

4.3 データ整形

データ整形では、3.2 節で述べた既存する最大かつ最新の Rimmer らの WF データセットと同様の整形法であり、近年の WF 攻撃においては一般的なデータ整形法を用いた [14]. パケットの特徴としてパケットの送信順序を用いる．ユーザから Entry ノードへの送信パケットを 1、ユーザが Entry ノードから受信するパケットを -1 として、時系列に沿って連結することで 1 次元のベクトルデータとする．CNN へのデータ入力では入力データ長は一定である必要がある．そのため、あるデータについて、パケットの総数が予め設定した値よりも小さい場合は 0 でパディングを行い、大きい場合は設定した値以降のパケットを無視する．k-NN では上記のデータ整形と少し異なる．Wang らは、k-NN のデータ整形においては、1、-1 としてデータを扱うよりパケットの送信位置を利用した方がパケットの送信順序の特徴を捉えることができ、k-NN の精度が上がるとしている [11]. 以下の図 3 は CNN に利用する Rimmer らのデータ整形と k-NN におけるデータ整形を表したものである．この図 3 においては、Rimmer らの初めの送信パケット（1 が送信パケット、-1 が受信パケット）に着目する．この送信パケットの前にいくつかのパケットが存在

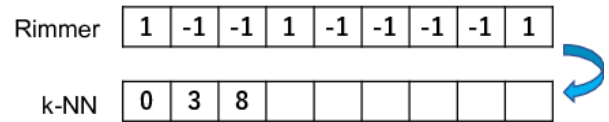


図 3 データ整形
Fig. 3 data format

するかという 0 個である．Rimmer らの 2 つ目の送信パケットに着目する．この送信パケットの前には計 3 つのパケットが存在する．このように k-NN では各送信パケットにおいて、それより前にいくつかのパケットが存在したかでパケットの送信位置を記録し、パケットの送信順序の特徴を利用する．これによりパケットの送信順序の特徴を利用しつつ入力データ長を大きく減らすことができる．

4.4 分類器作成

分類器は 2 種類作成した．1 つ目は Hidden Service に対する既存の WF 攻撃である k-NN を用いたもの、2 つ目は Hidden Service ではない Web サイトに対する WF 攻撃で高い精度を記録した CNN を用いたものである．

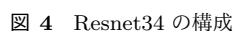
4.4.1 k-NN を用いた分類器

k-NN を用いた分類器は Albert Known らや Wang らと同様の特徴を利用しており、以下の特徴を用いた．一般的な特徴として、パケットの総送信サイズ、受信パケット数、送信パケット数を用いた．パケットの送信順序には 4.3 節で説明したパケットの送信位置を用いた．バースト情報には、受信バースト数、送信バースト数、最大バースト長、平均バースト長を用いた．CNN を用いた分類器では入力データ長を制限するが、k-NN を用いた分類器においては、入力データ長を考えず、全てのパケットの情報を利用出来るものとする．k-NN では計算資源を多く使うこともなく、また計算時間も短いからである．

4.4.2 CNN を用いた分類器

Albert Known らは Resnet18 をベースに、各畳み込み層で拡張畳み込みを利用することで WF 攻撃において高い精度を記録した CNN のモデルを作成した．本研究では、Albert Known らの Resnet18 を参考に更に層を深くした Resnet34 を用いた．実装には tensorflow を用いた．以下の図 4 に本研究で用いた Resnet34 の構成を示す．

各畳み込み層のインデックスは、カーネルサイズ、フィルタ数、拡張率である．/2 は、ストライドサイズを 2 にしてダウンサンプルしていることを示す．U 字の接続はブロックの入力をその出力に追加するスキップ接続を示している．点線のスキップ接続は、入力の次元がブロックの出力の次元と一致するようにダウンサンプリングされてから出力されていることを示している．ResNet34 は 4 つのブロックに分かれており、それぞれが複数の畳み込み層で構成されている．各畳み込み層には、BatchNormalization と



活性化関数 Relu 関数が含まれている [18]．出力層にはクラス分類における各クラスに属する確率を出力する softmax 関数を用いた．畳み込み層の拡張率は Albert Known らと同様に層が増える毎に 1, 2, 4, 8 と 2 倍ずつ増やした．しかし拡張率は増やせば増やすほど、モデルの学習時間と使用されるメモリの量は増加する．また、CNN では入力データ長を大きくすることでも、学習時間と使用されるメモリの量は増加する．そのため、今回の実験環境では、メモリの容量の関係で拡張率は最大 64 まで、入力データ長は最大 10000 までとしている．

実験 1 では 4.2 節で説明した 2 種類のデータセットに対して k-NN, CNN それぞれを用いた分類器で 5 分割交差検定を行い分類精度の検証を行なった. 表 1 はそれぞれの分類器の分類精度を示している. 分類精度の隣の括弧内の数値は分類器作成にかかった学習時間を示している. m=分, h=時間を示している. CNN への入力データ長は Rimmer らのデータセットや Albert Known らの CNN による WF 攻撃と同様に 5000 に設定しており, 4.3 節で説明した通りパケット総数が 5000 に満たないデータは 0 パディングを行い, パケット総数が 5000 以上の場合はそれ以降のパケッ

Table 1 Results of Experiment 1.

トを無視している.

表 1 から分かる通り、パケット総数が類似しているサイトでの分類は、既存の k-NN を用いた WF 攻撃では分類精度が 64.5% に対し、本研究で使った CNN を用いた WF 攻撃では 74.8% と CNN を用いた WF 攻撃の方が分類精度が高いことがわかった。パケット総数が類似していないサイトでの分類は既存の k-NN を用いた WF 攻撃では分類精度が 84.48% に対し、CNN を用いた WF 攻撃では 77.13% と k-NN を用いた既存の WF 攻撃の方が分類精度が高い。分類器作成にかかった時間は k-NN は 10 分ほどで終わるのに対し、CNN では 16 時間かかった。

CNN でのパケット総数類似サイトとパケット総数が類似していないサイトでの分類精度が大きく変わらないのには以下の理由が考えられる。CNN では、パケットの送信順序を 1, -1 の並び、パケット総数を 0 パディングの数でそれぞれ特徴として捉えている。しかし実験 1 においては入力データ長を 5000 に設定したため、パケット総数 5000 以上のサイトでは 0 パディングは行われておらず、パケット総数の特徴を利用出来ない可能性が考えられる。ま

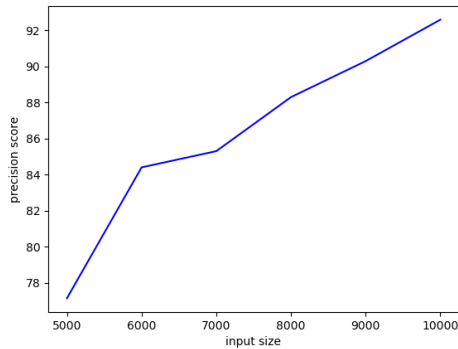


図 5 入力データ長を変化させた場合の CNN 分類精度

Fig. 5 CNN classification accuracy with varying input size

た、パケット総数だけではなく、5000 以降のパケットを無視していることから、5000 以降のパケットの送信順序の特徴も利用出来ていないため、サイトのパケットの特徴を全て利用出来る k-NN に比べて精度が低いことが考えられる。

4.7 実験 2

実験 2 では、パケット総数が類似していないサイトに対して、CNN への入力データ長を 5000 から 10000 へと 1000 ずつ増やして実験を行なった。CNN への入力データ長を増やすことにより実験 1 よりパケットの送信順序とパケット総数の特徴を多く利用出来るため、分類精度の向上が期待できる。実験 1 と同様、パケット総数が入力データ長を上回る場合は、それ以降のパケットの無視と 0 パディングを行っている。図 5 は実験 2 の結果を示したものである。

4.8 実験 2 の結果と考察

入力データ長を 1000 ずつ増やした結果、増やす度に分類の精度が上がり入力データ長 10000 では分類精度が 92.6% にまで上がった。また、入力データ長を増やすほど分類器作成にかかる時間も多くなった。この結果から、パケット総数が類似していなく、パケット総数が 5000 以上のサイトでの CNN を用いた WF 攻撃は入力データ長を大きくすれば分類精度が上がる事がわかる。これは、実験 1 では出来なかった、パケット総数の特徴とパケットの送信順序の特徴を多く利用出来ているからだと考えられる。4.2 節で述べた通りパケット総数が類似していないサイトの中には、パケット総数が 10000 より大きいサイトもいくつか存在する。それらのサイトは 0 パディングが行われていないため、パケット総数の特徴を利用出来ているとはいえない。このことから言えるのは、本研究の実験環境では出来なかったが、サイトの全ての情報を利用できる入力データ長（今回のデータセットでは最大パケット総数 17000）で CNN を用いた WF 攻撃を行えば、更に高い精度を期待することができるということである。

4.3 節で述べた k-NN と同様のデータ整形方法を用いる

表 2 k-NN, CNN を用いた WF 攻撃の実験結果

Table 2 Experimental results of WF attack using k-NN and CNN.

	k-NN	CNN
パケット総数 3000~4000	64.5%(10m)	74.8%(16h)
パケット総数 5000 以上	84.48%(10m)	92.6%(28h)

ことで、全てのサイトのパケット総数の特徴とパケットの送信順序の特徴を利用できると考えて実験を行なったが、分類精度は 68% となり分類精度は向上しなかった。k-NN では有効であるとされているデータ整形方法であったが、本研究で作成した CNN に用いた WF 攻撃には効果がなかった。

パケット総数が類似しているサイトでの分類とパケット総数が類似していないサイトでの k-NN を用いた既存の WF 攻撃と本研究の CNN による WF 攻撃の最終的な分類精度を表 2 に示す。

5. 考察

実験 1, 実験 2 から、今後 WF 攻撃を実際に用いる場合はどのような手法を用いるかは勿論のこと、監視対象とする Web サイトの選び方は非常に重要であるといえる。なぜなら、監視対象とする Web サイトの選び方に着目しない場合、パケット総数が類似していないサイトだけを選んで WF 攻撃の精度を上げることも可能であるため、WF 攻撃の精度を正しく検証出来ない可能性が高いからである。k-NN を用いた本研究での WF 攻撃はデータセットが違うとはいえ、既存の k-NN を用いた WF 攻撃より 10% ほど精度が下がった。これは Tor を開発する Tor Project が k-NN に利用されているサイト毎の特徴（受信パケット数や送信パケット数）を捉えにくくする等によって、k-NN を用いた WF 攻撃に対する対策をとった可能性が高いからである。このように k-NN のような手動で特徴を指定する WF 攻撃は対策を取ることが容易であると想像できる。本研究で行なった、k-NN を用いた WF 攻撃と CNN を用いた WF 攻撃からも Hidden Service に対する WF 攻撃は CNN 等のニューラルネットのように自動で特徴を学習する攻撃が適していると考えられる。

6. おわりに

本研究では Hidden Service の中でもパケット総数類似サイトとパケット総数が類似していないサイトそれぞれ 20 サイトに対して、k-NN を用いた既存の WF 攻撃と本研究で用いた CNN による WF 攻撃を行ない、有効性を検証した。どちらのデータセットに対しても分類器作成にかかる時間と計算資源を考慮しない場合、k-NN を用いた既存の WF 攻撃より本研究で用いた CNN による WF 攻撃の方が分類精度が優れていることがわかった。また、パケット総

数が類似しているサイトとそうでないサイトに対する WF 攻撃は、同じ手法を用いた場合でもパケット総数が類似している場合、分類精度が大きく落ちることがわかった。

今後の課題として、パケット総数が類似しているサイトでの分類精度を上げるための手法の構築や少ない計算資源で分類器を作成する手法の構築が挙げられる。

参考文献

- [1] Tor Project:Anonymity Online
<https://www.torproject.org>
- [2] David M. Goldschlag, Michael G. Reed, and Paul F. Syverson. Hiding Routing Information. In Proc. of the First International Workshop on Information Hiding, 1996.
- [3] Alexa - The Web Information Company.
<https://www.alexa.com>.
- [4] Tor Hidden Service Search engine.
<https://ahmia.fi>.
- [5] Xiaogang Wang, Junzhou Luo, Ming Yang, and Zhen Ling: " A potential HTTP-based application-level attack against Tor",Future Generation Computer Systems, pp.67-77,2011
- [6] Marc Liberatore and Brian N.Levine. Inferring the Source of Encrypted HTTP Connections. In Proceedings of the 13th ACM Conference on Computer and Communications Security, pages 255-263, 2006.
- [7] Xiang Cai, Xin Cheng Zhang, Brijesh Joshi, and Rob Johnson. Touching from a distance: website fingerprinting attacks and defenses. In Proc. of the 2012 ACM SIGSAC Conference on Computer and Communications Security, pages 605-616, 2012.
- [8] Andriy Panchenko, Lukas Niesse, Andreas Zinnen, and Thomas Engel. Website fingerprinting in onion routing based anonymization networks. In Proc. of the 10th ACM workshop on Privacy in the Electronic Society, pages 103-114, 2011.
- [9] Dominik Herrmann, Rolf Wendolsky, and Hannes Federath. Website Fingerprinting: Attacking Popular Privacy Enhancing Technologies with the Multinomial Naïve-Bayes Classifier. In Proceedings of the ACM Workshop on Cloud Computing Security, pages 31-42, 2009.
- [10] Marc Juarez, Sadia Afroz, Gunes Acar, and Claudia Diaz. A Critical Evaluation of Website Fingerprinting Attacks. In Proc. of the 2014 ACM Conference on Computer and communications security, pages 263-274, 2014
- [11] Tao Wang, Xiang Cai, Rishab Nithyanand, Rob Johnson, and Ian Goldberg. Effective attacks and provable defenses for website fingerprinting. In Proc. of the 23rd USENIX Security Symposium, pages 143-157, 2014.
- [12] Albert Kwon, Mashael AlSabah, David Lazar, Marc Dacier, and Srinivas Devadas. Circuit Fingerprinting Attacks: Passive Deanonimization of Tor Hidden Services. In Proc. of the 24th USENIX Security Symposium, 2015.
- [13] S. Bhat, D. Lu, A. Kwon, and S. Devadas, "Var-CNN: A data-efficient website fingerprinting attack based on deep learning," Privacy Enhancing Technologies Symposium (PETS), vol. 2019, no. 4, pp. 292-310, 2019.
- [14] Vera Rimmer, Davy Preuveneers, Marc Juarez, Tom V. Goethem, and Wouter Joosen. Automated Feature Extraction for Website Fingerprinting through Deep Learning. In Proceedings of the Network and Distributed System Security Symposium, 2018.
- [15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. arXiv preprint arXiv:1512.03385, 2015.
- [16] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. WaveNet: A Generative Model for Raw Audio. arXiv preprint arXiv:1609.03499, 2016.
- [17] DeepLink Onion Directory
<http://deeplinktx532i7z.onion>
- [18] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Proceedings of the 32nd International Conference on Machine Learning, 2015.