明治大学総合数理学部

2023 年度

卒　業　研　究

フィッシング検出方式の提案と JPCERT/CC フィッシングデータセットを用いた評価

学位請求者　先端メディアサイエンス学科

YANG LIYI

目次

第 1 章

はじめに

1.1　研究背景

近年, フィッシング攻撃による被害が増加の一途をたどっている [2]. 特に, 大手 EC サイトや金融機関を模倣した偽造サイトが攻撃の主流となり, 標的に対してリンクを送り, 個人情報や認証情報を盗む手法が蔓延している. フィッシング攻撃の仕組みを図 1.1 に示す.

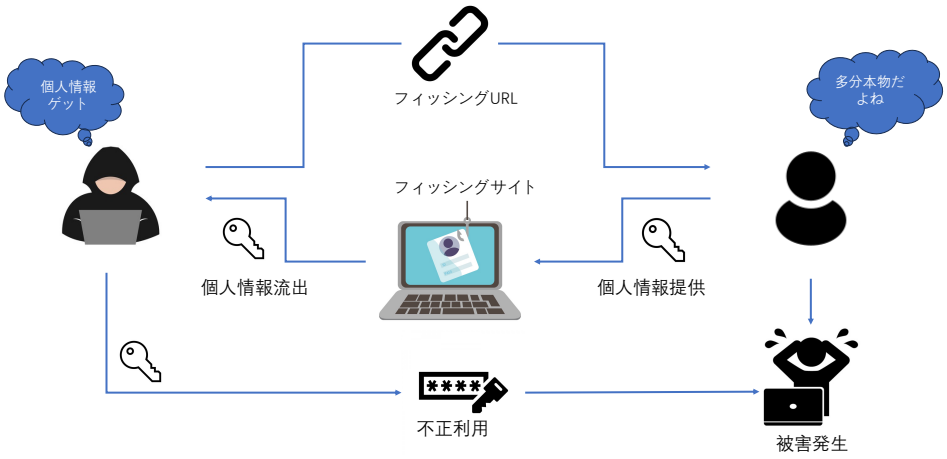

図 1.1　フィッシング攻撃の仕組み

そういった被害を防ぐために, 送付される URL に対して解析を行い, Netcraft[2] や VirusTotal[3] などのような多くのフィッシング URL を検出するサービスがある.

フィッシング攻撃に関する研究は非常に盛んであり, これまで多くのフィッシング URL の検出手法が提案されてきた. 中村ら [4] は Netcraft を利用した HTTP リクエスト解析によるフィッシングサイト検出システムを提案した. 彼らは既存サービスを応用し, HTTP リクエストの内容を解析し, 検出手法を設計した.

しかし, 今までの多くの研究やサービスは外国の企業が提供しており, 日本特有のフィッシングサイトを正確に検出できるかどうかはまだ不確かであった.

1.2 研究目的

そこで, 本研究ではまず既存サービス Netcraft を調査し, 海外のフィッシング URL と日本のフィッシング URL の違いを明らかにする. 観察した固有の件数に基づき, URL のキーワードリスクと Splited length などの新しい特徴量をいくつか提案し, 日本のフィッシング URL に特化したフィッシング検出システムの開発を目的とする.

第 2 章

調査

本章では, 既存サービス Netcraft を使用し, 日本のフィッシング URL と海外のフィッシング URL について調査した結果を報告する. また, URL の特徴について分析する.

2.1 調査対象

2.1.1 Netcraft

Netcraft はイギリスのロンドンに拠点を置くインターネットサービス会社であり, 主にウェブサイト調査やフィッシングサイト検出などのようなサービスを提供している.

2.1.2 JPCERT/CC

JRCERT/CC[5](以下 JPCERT と略す) はインターネットによる侵入や不正アクセスなどのサイバーセキュリティインシデントに対応する機関である. 日本国内サイトからのインシデント報告を受け付けており, 発生状況の把握, 対応の支援, 手口の分析, 再発防止対策の検討や技術的な助言を行なっている. 本研究は JPCERT に掲載された 2023 年 1 月から 5 月までのフィッシング URL のデータセットを使用する.

2.1.3 PhishTank

PhishTank[6](以下 Phishtank と略す) はインターネット上でのフィッシングサイトの情報を共有するウェブコミュニティ, およびサービスである. Phishtank はオープンなプロジェクトで, ユーザーからのフィッシング詐欺の詳細な報告を分析し, コミュニティの協力によって, フィッシングサイトのデータベースを構築している. 本研究は Phishtank に掲載された 2023 年 10 月 11 日のフィッシング URL のデータセットを使用する.

2.1.4 Kaggle

Kaggle[7] はデータサイエンティストや機械学習エンジニア向けのプラットフォームであり, データセットの提供, 機械学習モデルの構築と評価, コンペティションの参加など, データ関連の様々な活動をサポートしている. 本研究は Kaggle が提供する機械学習用の Malicious URLs dataset を利用する. このデータセットは Manu Siddhartha によって提供され, 2021 年にアップデートされたものである.

2.2 調査手法

2.2.1 ランダム文字列検出

本研究は文字遷移確率を算出することで, ランダム文字列を検出する. 遷移確率は学習データの文字遷移パターンに従って計算される. 与えられた文字列のパターンと学習データが近ければ近いほど遷移確率が大きくなる. 逆に遷移確率が小さいとランダム文字列の可能性が高いと意味する. 本研究は Python ライブラリ texttrans[8] を用いて, 文字遷移確率を算出した.

2.2.2 類似度算出

レーベンシュタイン距離は二つの文字列がどの程度異なっているかを示す距離の一種で, 文字列の類似度計算に使われている. 本研究は Python ライブラリ rapidfuzz[9] を用いて, 類似度を算出した.

第 3 章

フィッシングサイトの特徴量の提案と検出システム

3.1 Netcraft についての調査方法

この調査では Phishtank と JPCERT に掲載されたフィッシング URL を 1,200 個ずつ Netcraft で分析し, 警告が出たかどうか, 及びリスクという項目の結果を集計した.

3.1.1 警告確認

与えられた URL にマルウェアとフィッシングの危険性があると判断される場合は, 警告が提示される.

3.1.2 リスク

与えられた URL に対して Netcraft が独自な評価基準でそのリスクを評価する. 評価値は 0 から 10 であり, リスクの高さを意味する.

3.2 URL ベースの特徴量についての調査方法

この調査では PhishTank と JPCERT と Kaggle から提供されたフィッシング URL を 19,000 個ずつ用いて, いくつかの特徴量を抽出した.

3.2.1 SSL risk

指定した URL に対して, 先頭の部分が http か https かを識別する. https の場合は, 通信が暗号化されているので, SSL risk を 0, http の場合は SSL risk を 1 と定義する.

例を表 3.1 に示す.

表 3.1　SSL risk の例

URL	先頭部分	SSL risk
http://lphvnlopuh.duckdns.org/	http	1
https://lphvnlopuh.duckdns.org/	https	0

3.2.2　Random risk

指定した URL に対して, 数字とアルファベット以外の任意の文字で URL を分割する. 4 以上の文字列長の部分文字列に対して texttrans でランダム評価を行う. 評価の値が低ければ低いほどランダム文字列である可能性が高い. 本研究では, 閾値を 0.025 と 0.015 を用いて, URL の Random risk を式 3.1 と算出する. P_{URL} は分割後の部分文字列のランダム性評価の集合である. さらに, 0.025 より低い要素の数が 3 つ以上の場合は Random risk を 1 点増加する.

具体例を表 3.2 に示す.

$$Randomrisk = \begin{cases} 2 & \text{if } \min(P_{URL}) < 0.015, \\ 1 & \text{if } 0.015 \leq \min(P_{URL}) < 0.025, \\ 0 & \text{if } \min(P_{URL}) \geq 0.025. \end{cases} \tag{3.1}$$

表 3.2　Random risk の例

URL	評価最小値	Random risk
https://wzjdayup.xyz	0.0049	2
https://wwwepns.tanhehe.com/jp.php	0.0235	1
https://www.twitter.com	0.0260	0

3.2.3　Keyword risk

JPCERT データセット 2023 のフィッシングサイトの標的となる正規サイトのドメインから抽出した銘柄を用いて, リスクを評価する. 取り出した銘柄に加え”mypage” や ”password” などの個人情報に関するワードを用いて, キーワードリストを作成した. 与えられた URL に対して, 数字とアルファベット以外の文字でドメインを分割する. 分割後の各部分文字列に対して rapidfuzz でキーワードリストにある各キーワードとの類似度を評価する. 評価の値は 0 から 100 であり, 類似度の高さを意味する. 本研究では, 閾値を 70 と 85 の 2 つに設定し, URL の Keyword risk を式 3.2 と算出する. ここで, Q_{URL} は分割後の各部分文字列のキーワードリストとの類似度評価の集合である.

例を表 3.4 に示す. 作成したキーワードリストの一部を表 3.3 に示す.

表 3.3　キーワードリストの一部

Description	Keyword
エポスカード	eposcard
楽天	rakuten
ログイン	login

$$Keywordrisk = \begin{cases} 2 & \text{if } \max(Q_{URL}) > 85, \\ 1 & \text{if } 70 \leq \max(Q_{URL}) \leq 85, \\ 0 & \text{if } \max(Q_{URL}) < 70. \end{cases} \tag{3.2}$$

表 3.4　Keyword risk の例

URL	評価最大値	Keyword risk
https://www-cr-mufg-jp.kia8k.com/mufgcard/newsplus/	100.0	2
https://www.tmall.com	72.0	1
https://qwepo.xyz/	67.5	0

3.2.4　TLD count

Tranco[10] が提供する信頼できる Top1,000,000 のドメインのデータセットから, 出現回数が 100 回以上の Top Level Domain(TLD) を TLD リストとした. 与えられた URL に対して, 数字とアルファベット以外の文字でドメインを分割する. 分割後の各部分文字列を TLD リストと対照し, 一致する数をカウントする. その結果を TLD count と定義する.

例えば, ”https://newplenty.com.cn/jp” の場合はカウントされる部分文字列は com, cn, jp であり, TLD count は 3 である.

Splited length

指定した URL に対して, 数字とアルファベット以外の文字で URL を分割する. 分割された部分文字列数を Splited length と定義する.

例えば, ”https://info-e-orico.nftsgiant.com/” のドメインは info, e, orico, nftsgiant, com に分割され, Splited length は 5 である.

3.2.5　Whitelist risk

Tranco の Top1,000,000 のドメインのデータセットをホワイトリストとして利用する. 与えられた URL に対して, ドメインがホワイトリストにあるかどうかを判断する. ホワイトリストに載っている場合は Whitelist risk を 0, 載っていない場合は 1 と定義する.

具体例を表 3.5 に示す.

表 3.5　Whitelist risk の例

URL	ドメイン	Whitelist risk
http://lphvnlopuh.duckdns.org/	lphvnlopuh.duckdns.org	1
https://www.twitter.com/	twitter.com	0

3.3　検出システム

本研究では既存サービスが日本のフィッシング URL に弱い問題を解決するため, 日本のフィッシング URL に特化した検出システムを開発した.

システム構成を図 3.1 に示す.

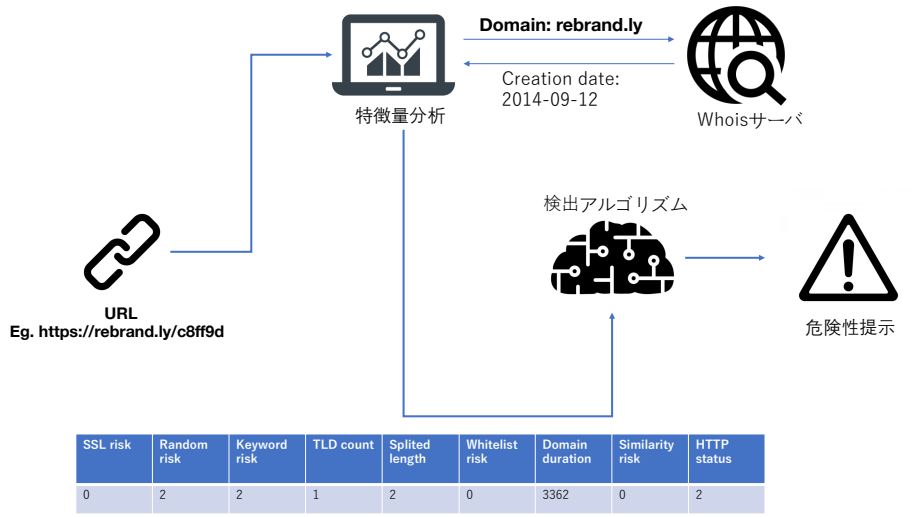

図 3.1　システム構成図

本システムは入力された URL に対して, 特徴量分析を行う. SSL risk, Random risk, Keyword risk, TLD count, Splited length, Whitelist risk のような SVM 用の特徴量以外, HTTP status と Domain duration と Similarity risk も検出用の特徴量として定義される.

3.4　検出アルゴリズム

本システムは点数加算方式でフィッシング URL を検出する. HTTP status と SVM 分類器と Domain duration と Similarity risk の 4 つの点数付け項目を設ける. 閾値を 55 に設定し, 閾値を超えた場合はフィッシング URL をみなす.

3.4.1　HTTP status

本システムでは, requests モジュールを使用し, 対象 URL の HTTP ヘッダーの情報を取得する. リクエスト成功した場合は, URL リダイレクトを確認する. 確認できた場合は危険点数を 10 点増加する. 確認できな

かった場合は危険点数を増加しない. 一方, リクエスト失敗した場合は, SVM 分類器だけで URL を判定する.

3.4.2 SVM 分類器

本システムでは 2.3 節で定めた 6 つの URL ベースの特徴量を利用し, Kaggle の安全な URL 20,000 個と JPCERT のフィッシング URL 19,714 個を学習し, 分類器を作る. 7 次元の機械学習用データセットを用いて, SVM 分類モデルに学習させる. SVC のパラメーターは, C=100, kernel='rbf' と設定する. 対象 URL を訓練した SVM モデルで判定し, 陽性が出た場合は危険点数を 40 点増加する. 陰性が出た場合は危険点数を増加しない.

3.4.3 Domain duration

本システムでは Whois サービスを利用し, 対象サイトの作成日時を調べる. 対象サイトが作成されてから, 経過した日数を Domain duration と定義する. 先行研究 [11] によると, 正規サイトの最頻値が 3,650 日であるのに対し, フィッシングサイトは 7 日と非常に期間が短い. Domain duration が 30 日以下の場合は危険点数を 25 点増加する. Domain duration が 30 日以上で 365 日以下の場合は危険点数を 15 点増加する. Domain duration が 365 日以上の場合は危険点数を増加しない.

3.4.4 Similarity risk

本システムでは, rapidfuzz モジュールを使用し, 対象 URL のドメイン部を 2.3.6 で説明したホワイトリストに載っている 1000000 個のドメインと比較し, 類似度を算出する. 類似度が 90 以上の項が存在すれば, 危険点数を 10 点増加する. 類似度が 90 以上の項が存在しなければ, 危険点数を増加しない.

第 4 章

評価実験

4.1 Netcraft についての調査結果

Netcraft の警告確認の結果を表 4.1 に示す.

表 4.1 警告確認結果

	positive	%
JPCERT	141	11.8
Phishtank	676	56.3

Netcraft のリスク値の分布を図 4.1 に示す.

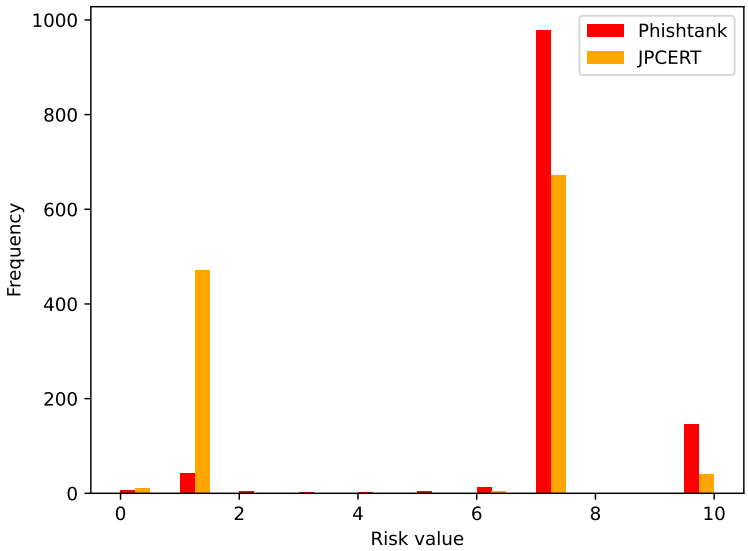

図 4.1 Netcraft のリスク値の分布

4.2　URL ベースの特徴量の調査結果

全 19,000URL の SSL リスクについての評価を表 4.2 に示す.

表 4.2　SSL リスク

	positive	%
JPCERT	2442	12.9
Phishtank	1551	8.2
Kaggle	0	0

ランダムリスクの分布を図 4.2 に示す.

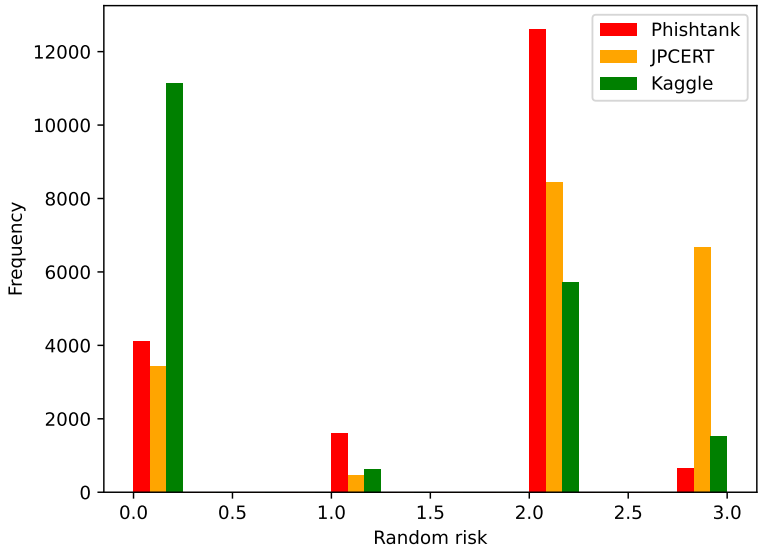

図 4.2　ランダムリスクの分布

キーワードリスクについての分布を図 4.3 に示す.

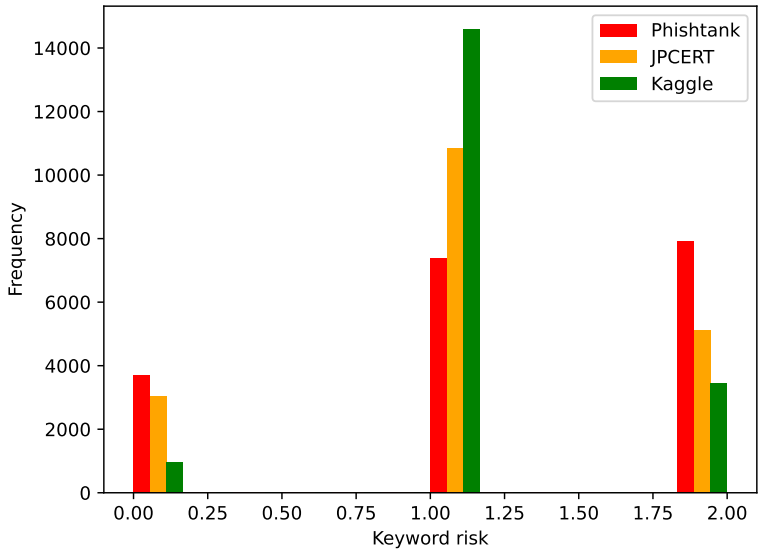

図 4.3　キーワードリスクの分布

TLD count の分布を図 4.4 に示す.

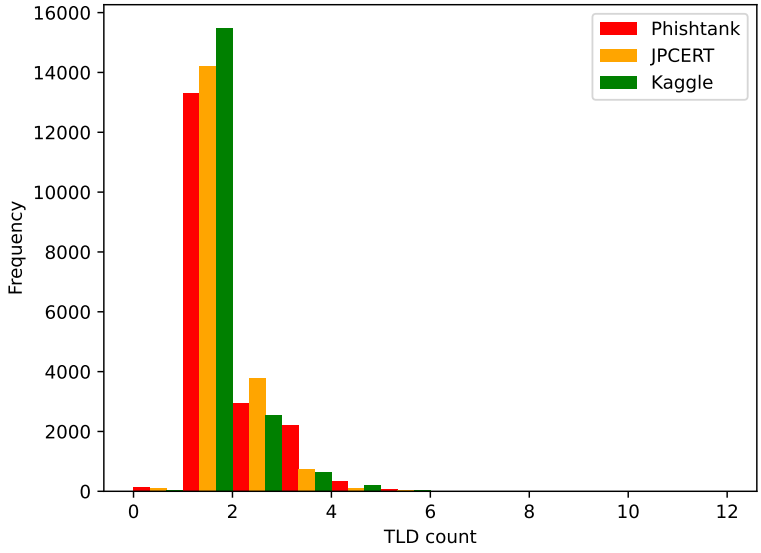

図 4.4　TLD count の分布

Splited length の分布を図 4.5 に示す.

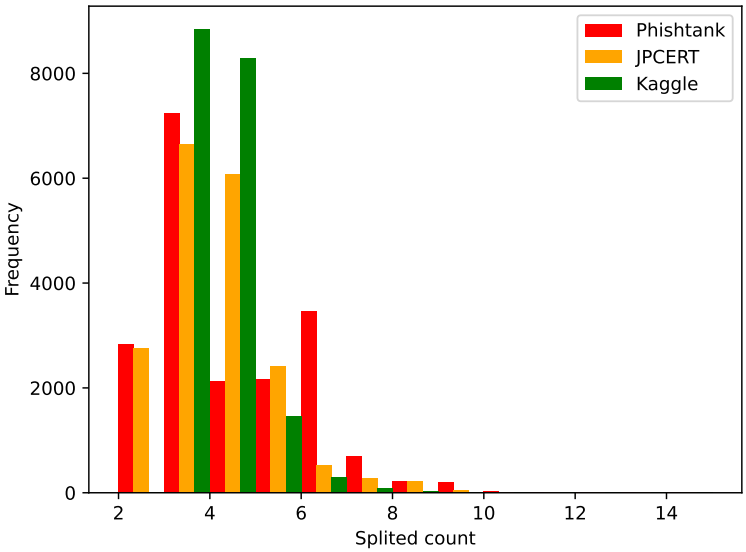

図 4.5　Splited length の分布

ホワイトリストリスクについての評価を表 4.3 に示す.

表 4.3　Whitelist リスク

	positive	%
JPCERT	18564	97.7
Phishtank	14790	77.8
Kaggle	18256	96.1

4.3　実験結果

5 分割交差検証を行った結果として, 平均精度は 0.84 であった.

本実験では JPCERT, Phishtank, Kaggle データセットから URL をランダムに 500 個ずつサンプリングして用いて, 5 のクロスバリデーションを行う.

実験結果を表 4.4 に示す. 実験結果についての精度評価を表 4.5 に示す.

表 4.4　検出精度

	平均値	最小値	最大値	標準偏差
FN(JPCERT)	16.8	13	22	2.9
FP(JPCERT)	21.8	16	25	3.1
FN (Phishtank)	65	41	92	17.3
FP(Phishtank)	21.8	16	25	3.1

表 4.5　精度評価

	JPCERT	Phishtank
Precision	0.79	0.62
Recall	0.83	0.35
F score	0.81	0.45
Accuracy	0.81	0.57

4.4　誤分類の例

誤分類の例を表 4.6 に示す.

表 4.6　誤分類の例

URL	SSL risk	Random risk	Keyword risk	TLD count	Splited length	Whitelist risk	誤分類の種類
https://funfun-kids.com/contact/23/	0	0	2	1	3	1	FN
https://tersyesg.yourtrap.com	0	0	1	1	3	1	FN
https://www.lostcanadianchildren.com/MP.html	0	2	1	1	3	1	FP

表 4.6 から, 検出システムの判断は Random risk に大きく影響されていることが分かった. URL に識別できない文字列が含まれる場合は Random risk が高くなり, 誤分類される可能性も高くなる.

4.5　考察

調査結果の図 4.1 から, Phishtank の URL の殆どはリスク値が 7 以上で, JPCERT の URL の半数ぐらいはリスク値が 2 以下であることが分かった. 故に, Netcraft は日本のフィッシングサイトに弱いと考えられる. Netcraft 以外にも VirusTotal のような人気なサービスがあるため, 他の既存サービスに対して調査を行う必要がある.

また, 提案したシステムの実験結果の表 4.5 から, Phishtank のフィッシング URL には精度が低いが, JPCERT のフィッシング URL は 8 割以上の精度で検出された. 実用できるほど精度は高くないが, 日本のフィッシング URL に対して, 海外の既存サービス Netcraft より検出精度が高いため, 日本のフィッシング URL に特化したフィッシング検出システムの開発である本研究の目的に達成できたと言えるだろう.

精度を上げるには, より適切な特徴量を選ぶべきである. 例えば, 本研究では TLD count を特徴量として利用したが, 実は調査結果の図 4.4 によると, 安全な URL とフィッシング URL の分布が差が僅かしかないため, あまり適切ではなかったと考えられる. 文字列のランダム判定に用いた texttrans は分割なしの長い文字列に弱い. 例えば, lostcanadianchildren という文字列はランダム文字列に認識される. それも精度が上がらない原因の一つであると推測される.

第 5 章

おわりに

本研究では既存のサービス Netcraft を調査し, Netcraft が日本のフィッシング URL に弱いことを明らかにした. その問題を解決するために日本のフィッシング URL に特化した検出システムを開発した. 本研究を通じて, フィッシング対策は一つのサービスに頼らず, 地域に対応した複数のサービスを運用すべきだと提案したい.

今後, URL ベースだけでではなく, 多様な方式で適切な特徴量を定め, より質の高いシステムを構築していきたいと考える.

参考文献

[1] 読売新聞オンライン, ”「フィッシング」の不正送金が急増、2 月以降の被害 9 億 6000 万円” (https://www.yomiuri.co.jp/national/20230426-OYT1T50155/, 2023 年 10 月参照)

[2] Netcraft (https://www.netcraft.com/, 2023 年 10 月参照)

[3] VirusTotal(https://www.virustotal.com/, 2023 年 10 月参照)

[4] 中村元彦, 寺田真敏, 千葉雄司, 土井範久, ”プロキシを利用した HTTP リクエスト解析によるフィッシングサイト検出システムの提案”, 情報処理学会論文誌 Vol.48 No.10, p3365-3374, 2007.

[5] JPCERT/CC (https://github.com/JPCERTCC/phishurl-list/, 2023 年 10 月参照)

[6] PhishTank (https://www.phishtank.com, 2023 年 10 月参照)

[7] Kaggle (https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset, 2023 年 10 月参照)

[8] texttrans (https://pypi.org/project/texttrans/, 2023 年 10 月参照)

[9] rapidfuzz (https://pypi.org/project/rapidfuzz/, 2023 年 10 月参照)

[10] Tranco (https://tranco-list.eu/, 2023 年 10 月参照) https://tranco-list.eu/

[11] 桜井啓多，“ドメイン情報と HTTP レスポンスヘッダに基づくフィッシングサイトの識別と評価”，2018 年度菊池研究室卒業論文，2018.

謝辞

本研究に際して, 様々なご指導をいただきました菊池浩明教授, 遠藤冴花先輩, 青山綾華先輩に深く感謝いたします. 最後に, 菊池研究室の皆様に感謝の意を表すると共に, 謝辞にかえさせていただきます.

付録 A

ブロックチェーン Ethereum とマーケットプレイス OpenSea における NFT 取引の整合性の調査

A.1　はじめに

近年, ブロックチェーンの普及とともに,Non-fungible Token (NFT) の取引も活発になってきている。NFT はブロックチェーン上に記載される一意で代替不可能なデータ単位である。ユーザはマーケットプレイスを介して,NFT の売買や譲渡などの取引を行なっている。そして, マーケットプレイスは取引記録を分散台帳ブロックチェーンに記載することで, 誰が NFT の所有者であるかを確かにして, 不法なコピーを防止している。NFT 取引の仕組みを図 1.1 に示す。

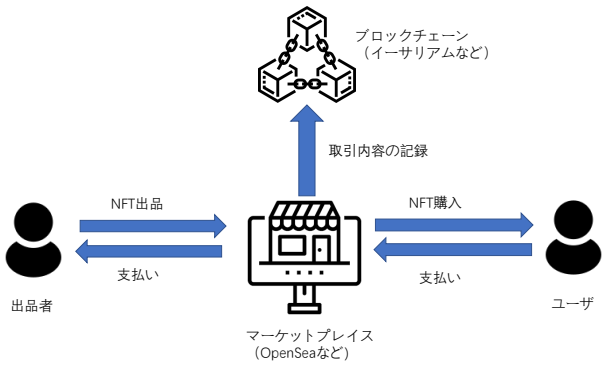

図 A.1　NFT 取引の仕組み

木村らはブロックチェーン上で管理されるデジタルアート作品の贋作を流通させるイミテーション攻撃に注目し, NFT 取引に潜むリスクを明らかにした [1]. しかしながら, それらの研究では外部からのリスクのみを想定しており, マーケットプレイス側, すなわち, 内部犯による脅威が考慮されていなかった. マーケットプレイス側は本当にユーザの取引記録をそのまま間違いなくブロックチェーンに記載しているのだろうか. 例として,2023 年 1 月に, NFT マーケットプレイスの Magic Eden のプロジェクトのコレクションページで, 偽物の

NFT が混ざって販売されていたことが発表された [2]. そこで, 本研究ではブロックチェーン Ethereum のブロック閲覧プラットフォームであるイーサスキャンを利用し, マーケットプレイスの一つである OpenSea においての取引記録が記載されているかを調査する. 不整合が確認できれば, その原因について考察する.

A.2 概要

A.2.1 イーサリアム

イーサリアム（Ethereum）は, ブロックチェーンを活用してスマートコントラクトを構築するプラットフォームで, 仮想通貨としての側面も有する. イーサリアムを利用するのに必要な通貨として, イーサ（Ether）が用いられている [3].

A.2.2 イーサスキャン

イーサスキャン（Etherscan）は,Ethereum ネットワーク用のブロックチェーン・エクスプローラーである. Ethereum ブロックチェーン上で行われた取引を検索, 確認, 検証するサービスを提供している. 通貨の取引記録も提供している [4].

A.2.3 OpenSea

OpenSea は, 最大手の NFT マーケットプレイスの一つであり, NFT の作成, 出品, 購入を提供する. ユーザの取引に関するアクティビティ履歴を提供する [5]. ユーザのアクティビティのイベントは主に sale,transfer,minted,list,offer がある. その内,transfer（譲渡）と minted（ブロックの検証と採掘）のイベントはブロックチェーンに記載される.

A.3 調査

A.3.1 調査目的

本調査では,OpenSea においてのイーサリアムのブロックチェーンを利用した NFT 取引の不整合を確認することを目的とする。

A.3.2 不整合の定義

NFT の特性により, 一つのスマートコントラクトに対して NFT の ID は唯一である. 故に,NFT 取引記録のスマートコントラクトと NFT ID から, 取引されている NFT を特定することができる. アドレスを指定すれば, イーサスキャンと OpenSea から, そのアドレスの NFT 取引記録を調べられる. それらから, 指定したアドレスがどのような NFT について取引をしたのを知ることができる.

そこで, 三つのケースがあり得る. 一つ目はイーサスキャンにも OpenSea にも記載があるケースである. それは何の問題もないケースであり, 最も多いケースであると予想する. 二つ目はイーサスキャンに記載されているが, OpenSea に記載されていないケース（case A）である. 三つ目はイーサスキャンに記載されていないのに, OpenSea に記載されているケース (case B) である. 本調査では二つ目と三つ目のケースを不整合として定義する.

OpenSea のシェア率の定義

$$\text{シェア率} = 1 - \left(\frac{caseA \text{ の合計}}{etherscan \text{ の合計}}\right) \tag{A.1}$$

未登録の定義

$$\text{未登録率} = \frac{caseB \text{ の合計}}{OpenSea \text{ の合計}} \tag{A.2}$$

A.3.3 開発したシステム

OpenSea の情報収集プログラム

OpenSea から指定したアドレスの取引記録を収集する python のプログラムである. Web ブラウザをプログラムから自動的に操作するツールである selenium webdriver を使用した.

指定した URL のページにアクセスし, タグネームで検索し, 画面中の全ての HREF タグの中のリンクを取得する. そのページの全ての HREF タグのリンクを取得するため, 画面を少しずつページの最後までスクロールさせる. 取得したリンクの重複を除いてから, 文字列”assets”が入っている要素だけを取り出す. 最後に csv ファイルに出力する.

照合プログラム

イーサスキャンからダウンロードした取引記録を読み込んで, スマートコントラクトと NFT ID の部分を取り出して, 重複の部分を取り除く. OpenSea から取得した NFT のスマートコントラクトと NFT ID の csv

ファイルを読み込んで, 二つのデータを照合する.

A.3.4　調査方法

調査方法を図 3.1 に示す. 3.3 節の開発システムを用いて, イーサスキャンから特定した NFT と OpenSea から特定した NFT を照合する.

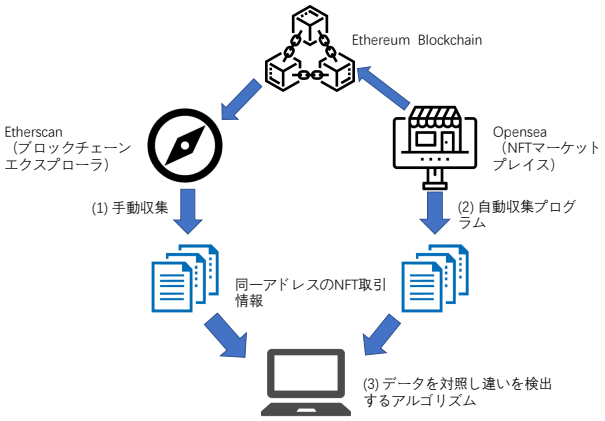

図 A.2　調査方法

(1) アドレスをイーサスキャンから収集する. NFT の取引記録が少なすぎると, 参考にならないため, 本調査では取引記録が 20 以上である任意の 10 個のアドレスを収集した. 本調査の対象アドレスを表 3.1 に示す.

表 A.1　調査対象アドレス

No.	Address
1	0x43c18046B63745aCe463909904F98F2Bfd1DeD1d
2	0x24a35c90067AcD020ea2839dF9aC49659CcE3acD
3	0x3E5814c65e415E054b186Eb7296232e116C40B71
4	0x7c4D17B7088d111A2273c944E68B60869ae690B3
5	0x40B84304f28226bF7792b734753f2Ab3f99CcDC5
6	0xdcd81eFC135E5697ce9D93d4e63fE93567Bb5cC6
7	0xa76968D9b1CC1244c2f567011D96F49F6d4CE522
8	0xa961E5b23385A84Ec35b90abE14CB80d69263a46
9	0x2AAA07a0a722AB93a27183277f5e92D50Fc248F1
10	0x3BB17B36ADafFd212D0F6deE5C18911F24892828

(2) OpenSea から NFT 取引記録を取得する. 本調査ではイーサリアムのブロックチェーンを利用した NFT の取引だけが調査対象となっている. OpenSea のイベントの中で transfer と minted のイベントしかブロックチェーンに記載されないため, OpenSea のアドレスのアクティビティページのフィルター機能を使用した.

(3) 照合プログラムを使用し, 同じアドレスのイーサスキャンからダウンロードしたファイルと OpenSea から収集した情報のファイルを比較する.

A.4　調査結果

調査の結果を表 3.2 に示す. イーサスキャンと OpenSea に記載されている NFT の数, 及びイーサスキャンに記載されているが, OpenSea に記載されていない NFT の数（case A) とその逆 (case B) を示している. 4 番と 8 番のアドレスを除けば, 全体的に case A より case B の数が少ないことが分かる.

表 A.2　対象アドレスに関する観測取引数

No.	Etherscan	OpenSea	caseA	caseB
1	61	58	5	2
2	104	97	18	11
3	174	172	2	0
4	24	27	1	4
5	62	40	28	6
6	43	29	16	2
7	35	29	6	0
8	35	40	1	6
9	120	96	25	1
10	141	136	5	0
計	799	724	107	32

A.4.1　分析

調査結果の分析結果を表 3.3 に示す.

表 A.3　結果の分析

	シェア率	未登録率
OpenSea	86.61%	4.42%
MagicEden	90.07 % [6]	5.85 % [6]

A.5　考察

A.5.1　イーサスキャンに記載されているが OpenSea に記載されていないケース B

NFT のマーケットプレイスは OpenSea 以外でもたくさんある. イーサスキャンに記載されているが OpenSea に記載されていないのはイーサリアムを利用したが, 他のマーケットプレイス, 例えば, MagicEden などで取引した可能性がある.

A.5.2 イーサスキャンに記載されていないのに OpenSea に記載されているケース A

NFT の取引は全てブロックチェーンに記載されるはずのため, このケースは明らかに不自然である. 原因としては, ブロックチェーンの登録遅延が考えられる. OpenSea 側はユーザの操作を受理したが, イーサスキャン側はまだ変更が検証されていない可能性がある.

しかし, 何ヶ月前のイーサスキャンに記載されていないのに OpenSea に記載されている NFT を見つけた. この不整合は遅延として考えにくい. 業者側の可能性があると考えられる.

A.5.3 シェア率について

本調査ではマーケットプレイスのシェア率についても計算した. 結果から見ると, OpenSea のシェア率の数値は MagicEden より低い. しかし, 本論文で定義したシェア率では NFT の有名度などの要素を十分に踏まえていないため, このシェア率だけでマーケットプレイスの規模を説明することができないと考えられる.

A.5.4 調査の不足について

本調査ではアドレスの取引記録から取引された NFT を特定する方法で整合性について評価した. しかしながら, この方法では, どのような NFT が取引されているのが分かるが, 取引の回数と時間がわからない. 調査対象のアドレスが少なく, 十分な精度がない.

A.6　おわりに

本調査では, アドレスの取引記録から取引された NFT を特定する方法でイーサスキャンを利用し, OpenSea における NFT 取引の整合性を調査した. そして, 不整合があることを確認し, その原因についても検討した.

NFT の取引が活発になっていくとともに, 不整合のリスクも高まるだろう. どうやって NFT 取引の不整合を防ぐか, あるいは正しく検出するかは今後の課題である.

参考文献

[1] 木村, 今村, 面, “NFT の信頼性にみるセキュリティリスクの考察”, 電子情報通信学会, 信学技報, vol.121, No.118, pp.123-130(2021).

[2]『yahoo ニュース』2023.01.06, 「「Magic Eden」で偽 NFT が販売、不快なポルノ NFT 画像表示に続き」.

[3] 大上, 稲葉, “Ethereum のスマートコントラクトを用いた信頼性の高いカーシェアリングシステムの提案”, 電子情報通信学会, 信学技報, vol.117, No.69, pp.37-40(2017).

[4] 佐藤, 今村, 面, “Ethereum ブロックチェーン上に潜むマルウェアなどの定量的リスク分析”, コンピュータセキュリティシンポジウム 2018 論文集, No.2, pp.862-869(2018).

[5] 藤原, 横山, “ブロックチェーンを活用した画像の部分的所有権保護についての研究”, 電子情報通信学会, 信学技報, vol.122, No.176, pp.7-12(2022).

[6] 遠藤, “ブロックチェーン Ethereum とマーケットプレイス Magic Eden での NFT 取引の整合性の調査”, 明治大学, 総合数理学部, 菊池研卒業研究 (2022).