明治大学総合数理学部

2024 年度

卒　業　研　究

多次元データに対する局所差分プライバシー手法 CALM の評価

学位請求者　先端メディアサイエンス学科

山本　悠太

目次

第 1 章

研究背景

デジタル化の進展に伴い, 個人データの収集と分析が広範囲に行われるようになり, 企業や政府機関が収集するデータの量が増大している [5]. それらのデータはターゲット広告やマイナンバーカードなどの技術に利用されている一方, 同意のない情報提供などのプライバシー侵害のリスクとなるケースも多い. データの有用性を損なわず, プライバシー侵害のリスクを軽減するための技術的対策として局所差分プライバシー [2] が注目されている.

単次元のデータに関する局所差分プライバシー手法に関しては Randomized Response[2], Count Mean Sketch[7], Optimized Unary Encoding[6], Optimized Local Hash[6] など様々な手法が提案されているが, 多次元データに対する局所差分プライバシー手法に関しては十分な提案がなされているとは言えない. その理由として, データの次元数に比例して大きなプライバシー予算がかかることが大きな課題となっていることがあげられる. 既存手法として知られる, Zhang らによる CALM[1] では, 求めたい多次元データより低次元の周辺頻度を表現する, 分割表の制約を無矛盾とする多次元周辺頻度をエントロピー最大となる凸最適化問題を解決することにより算出する. CALM[1] の問題として、凸最適化ソルバーの精度、ノイズの大きさや次元数によっては解不定となってしまうことが Zhang ら, 菊池 [3] により指摘されている.

本研究では多次元データに対する局所差分プライバシー手法とである, CALM[1] の再実装を行い, 評価することを目的とする. 特に, 2 次元の分割表から 5 次元の分割表の確率分布を求めるケースについて実験を行い, CALM を適用, 評価した.

第 2 章

先行研究

2.1 局所差分プライバシー

局所差分プライバシー (Local Differential Privacy, LDP) は, データが収集される前に個人のプライバシーを保護するための枠組みである. 具体的には, データ提供者が自身のデータにランダム化処理を施すことで, データ収集者に渡される情報が個々の元データを推測することが難しくなるように設計されている.

形式的には, 任意の入力 $x, x' \in \mathcal{X}$ と任意の出力 $y \in \mathcal{Y}$ に対して, メカニズム $\mathcal{M}$ が以下を満たすとき, $\mathcal{M}$ は ϵ-局所差分プライバシーを満たす：

$$\Pr[\mathcal{M}(x) = y] \leq e^{\epsilon} \Pr[\mathcal{M}(x') = y], \tag{2.1}$$

ここで, $\epsilon \geq 0$ はプライバシー予算 (privacy budget) と呼ばれ, プライバシーと有用性のトレードオフを制御するパラメータである. ϵ が小さいほどプライバシーは強くなるが, 出力結果の有用性は低下する可能性がある.

局所差分プライバシーの基本的な仕組みでは, データ提供者がデータを収集者に送信する前に、ランダム化によって情報を保護する. これにより, 収集者は統計的な分析を行うために必要な集約データを得ることができる一方で, 特定の個人のデータに関する情報を直接推測することが難しくなる.

2.1.1 多次元の局所差分プライバシー

一般に想定される局所差分プライバシーは 1 次元のデータに対してのものが多い. 多次元に対する局所差分プライバシーには困難点が存在するためである. 考慮する属性数が増えることにより, プライバシーコストが増大する点や, また属性間の依存関係について考慮する必要があり, 計算コストも増大する点などがあげられる. 代表的な手法として LoPub[8] などが存在し, 今回主に言及する CALM[1] は先述の困難点をある程度解消することができる手法である.

2.2 CALM

CALM[1] では多次元のデータに対し, 属性ごとの頻度を計算する分割表を複数作成し, これらの分割表から各属性の周辺確率を得, そこから全体の確率分布に対する制約条件を求める. 求めた制約条件が無矛盾となり, かつエントロピー最大となるように凸最適化問題を解くことによって, 各ユーザのデータが秘匿された状態で全体の確率分布を求めることができる.

いかに例を示す. 今, $n = 32$ 人のユーザ v_1 から v_32 が, "Gender", "Age", "Marital" という $k = 3$ 次元の情報を持っている. これらの情報は Gender = {Male, Female}, Age = {Adult, Elder}, Marital = {Yes, No} からなり, 表 2.1 に示す 3 次元頻度分布から, 全体確率分布 $F(V)$ を収集したいケースを考える. この時, 元データを直接サーバに送信してしまうと, ユーザのプライバシーが秘匿されないため, より低い $\ell < k$ 次元の周辺頻度の集計をする分割表から表 2.1 の頻度分布を算出することを試みる.

表 2.2, 2.3 に分割表の例を示す. 表 2.2 では Gender, Age の周辺頻度, 表 2.3 では Gender, Marital の周辺頻度を表す. これらの頻度は 2 属性間の関係を表している. x_1 から x_8 を 3 次元の属性の組み合わせだとすると, 表 2.2 からは以下のような関係を得られ,

$$\begin{cases} x_1 + x_5 = 6 \\ x_2 + x_6 = 8 \\ x_3 + x_7 = 10 \\ x_4 + x_8 = 12 \end{cases} \tag{2.2}$$

表 2.3 からは以下のような関係を得る.

$$\begin{cases} x_1 + x_3 = 4 \\ x_2 + x_4 = 6 \\ x_5 + x_7 = 12 \\ x_6 + x_8 = 14 \end{cases} \tag{2.3}$$

CALM ではユーザはすべてのデータを送信するのではなく, 属性を選択して送信する. 例えば n/2 のユーザは Gender, Age について RR を行い，表 2.2 を推定し，残りの n/2 のユーザは, Gender,Marital について表 2.3 の推定を与える, といった属性の送信方法が考えられる.

本研究では送信するデータの選択は, サーバ側がユーザ側に対し, データ送信前に送る割り当てを行うものとする.

エントロピーを最大化する凸最適化ソルバー CVXPY[9] を用いて, x_1 から x_8 についてエントロピー最大となるように, (2.2) と (2.3) を制約条件とした凸最適化問題を解くことで全体の確率推定を行う.

表 2.1　全体の頻度分布, $k = 3$

	Married		Unmarried	
	Adult	Elder	Adult	Elder
$Male$	1	3	5	7
$Female$	2	4	6	8

表 2.2　Gender, Age の分割表, $\ell = 2$

	Adult	Elder
$Male$	6	10
$Female$	8	12

表 2.3　Gender, Marital の分割表, $\ell = 2$

	Yes	No
$Male$	4	12
$Female$	6	14

第 3 章

実験

3.1 実験方法

$n = 64$ 人のユーザ v_1 から v_64 が, a_1, a_2, a_3, a_4, a_5 という $k = 5$ 次元の情報を持つ表 3.1 のようなデータセットを作成する. 各属性はバイナリデータであり, ユーザの持つ各属性値は $\{0, 1\} = \{0.9, 0.1\}$ の確率でランダムに決定される. 求める元の確率分布を表 3.2 に示す.

表 3.1 作成したデータセット

	a_1	a_2	a_3	a_4	a_5
v_1	0	1	0	0	0
v_1	0	0	0	0	0
v_2	0	0	0	0	0
...	...	...	...	...	...
v_{32}	0	0	1	0	0

簡単のため, 本実験ではプライバシーコスト $\varepsilon \to \infty$ のケースを考える. そのため摂動を加えた後のデータは元データと全く同じ確率分布となる.

表 3.3 のような 2 次元の分割表 ($\ell = 2$) から, 表 3.2 の元データの確率分布 ($k = 5$) を算出することを考える. 属性数が 5, 各属性がバイナリであることから, 分割表の個数は ${}_5C_2 = 10$ 個である. また, 表 3.3 から (3.1) のような, 5 次元の確率分布を表現する変数 x_n に対する制約条件を得ることができる. 得られる表は 10 個であり, 1 つの表から 4 本の式を得られるため, 得られる制約条件は最大 40 個である. エントロピーを最大化する凸最適化ソルバーを用いて, x_0 から x_{31} についてエントロピー最大となるように, 分割表から得られた式を制約条件とした凸最適化問題を解き, 元データとの確率の誤差を MAE を用いて評価する.

エントロピーを最大化する凸最適化ソルバーを用いて, x_0 から x_{31} についてエントロピー最大となるように, 分割表から得られた式を制約条件とした凸最適化問題を解き, 元データとの確率の誤差を MAE を用いて評価する. 最適化問題のソルバーには, Python 用ソルバーライブラリである CVXPY[9] を使用し, 得られた解について手動で計算を行い誤差を算出した.

$$\begin{cases} x_0 + x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 = 51 \\ x_8 + x_9 + x_{10} + x_{11} + x_{12} + x_{13} + x_{14} + x_{15} = 5 \\ x_{16} + x_{17} + x_{18} + x_{19} + x_{20} + x_{21} + x_{22} + x_{23} = 8 \\ x_{24} + x_{25} + x_{26} + x_{27} + x_{28} + x_{29} + x_{30} + x_{31} = 0 \end{cases} \tag{3.1}$$

3.2 結果

得られた推定結果を表 3.4 に示す．$\varepsilon \to \infty$ のケースでは解不定とはならず, 2 次元の分割表から 5 次元の確率分布を算出することが可能であることが確認できた．表 3.2 と表 3.4 との誤差を算出した結果,$MAE = 0.0064$ となった．最大の誤差となったのは x_0=<0, 0, 0, 0, 0> の項であり, 元の確率と推定結果とで 0.016 の誤差が観測された．また, $x_1$6=<1, 0, 0, 0, 0> の項の推定については誤差 0 で推定することができていた．

表 3.2　元の確率分布

$< a_1, a_2, a_3, a_4, a_5 >$	確率
<0, 0, 0, 0, 0>	0.500
<0, 0, 0, 0, 1>	0.047
<0, 0, 0, 1, 0>	0.093
<0, 0, 0, 1, 1>	0.016
<0, 0, 1, 0, 0>	0.141
<0, 1, 0, 0, 0>	0.047
<0, 1, 0, 0, 1>	0.016
<0, 1, 1, 1, 0>	0.016
<1, 0, 0, 0, 0>	0.109
<1, 0, 0, 1, 0>	0.016
他	0

表 3.3　a_1, a_2 の分割表, $\ell = 2$

	0	1
0	51	5
1	8	0

表 3.4　分割表から得た確率分布

$< a_1, a_2, a_3, a_4, a_5 >$	確率
<0, 0, 0, 0, 0>	0.516
<0, 0, 0, 0, 1>	0.051
<0, 0, 0, 1, 0>	0.082
<0, 0, 0, 1, 1>	0.008
<0, 0, 1, 0, 0>	0.140
<0, 1, 0, 0, 0>	0.046
<0, 1, 0, 0, 1>	0.005
<0, 1, 1, 1, 0>	0.003
<1, 0, 0, 0, 0>	0.109
<1, 0, 0, 1, 0>	0.017
他	0

第 4 章

考察

実験を通して 2 次元の分割表から 5 次元の確率分布を得るような CALM が機能することが確認できた. 精度の面では, プライバシーコスト $\varepsilon = \infty$ を想定しているのにもかかわらず誤差が出る結果となってしまったが, これは制約条件が重複しており, 十分に値を決定しきれなかったことが原因だと推測する. また, 用いたソルバーの計算過程において誤差が出た可能性も検討し, 用いるソルバーを変更した実験や, 計算を単純化できるケースを考えた実験を行い, さらなる検証を行うことは不可欠であると考える.

また, 今回の実験ではプライバシーコスト $\varepsilon = \infty$ を想定して動作検証を行ったため, 元データと全く同じデータに対し CALM を適用している. そのため摂動を加えた時の動作検証はできていない. 既存研究 [1][3] では摂動を加えた際にはその大きさ次第で, 最適化問題が無矛盾となる解を出せず解不定となる可能性が指摘されているため, 本実験で考えた 2 次元の分割表から 5 次元の確率分布を算出するケースに関してもプライバシーコストを変化させたときの動作についてさらなる調査が必要である.

また取得する分割表の大きさについても検討の余地がある. 例えば $k = 5$ 次元のデータに対して, 今回は $k = 2$ 次元の分割表を取得し, これを用いて周辺確率を算出したが, $k = 3$ 次元の分割表, $k = 4$ 次元の分割表を取得し, そこから周辺確率を算出する方法も存在する. このように高次元データに対しての適切な分割表の取得方法に関しては自明ではないため, この点に関して更なる実験が必要である.

第 5 章

おわりに

本研究では CALM の再実装を行い, 実験を行い簡易的な評価を行ったが, CALM への理解とプログラムの実装があいまいなまま実験を進めた結果, 極めて単純なケースのみを想定して実験を行うこととなった. その結果, 摂動の強さを変えた時の CALM の評価は行うことができなかった. 今後の実験として, プライバシーコストを変化させたときの精度の調査, 分割表の次元の大きさを変化させたときの誤差の変化の調査を行い, 本論文をさらに補完していく予定である.

謝辞

本研究を行うにあたり，多くの方より御指導いただきました．特に，多大なる御指導を受け賜りました，明治大学総合数理学部先端メディアサイエンス学科，菊池浩明教授に深く感謝申し上げます．予備実験等に協力してくださった菊池研究室の皆様並びに先端メディアサイエンス学科の方々に深く感謝の意を表するとともに，謝辞とさせていただきます．

参考文献

[1] Zhikun Zhang, Tianhao Wang, Ninghui Li, Shibo He, Jiming Chen, CALM: Consistent Adaptive Local Marginal for Marginal Release under Local Differential Privacy, CCS’18, 2018.

[2] J. C. Duchi, M. I. Jordan and M. J. Wainwright, ” Local Privacy and Statistical Minimax Rates,” 2013 IEEE 54th Annual Symposium on Foundations of Computer Science, Berkeley, CA, USA, pp. 429-438,2013

[3] 菊池浩明, 高次元データの局所差分プライバシーの推定精度を向上する擬似逆行列ベース手法, Computer Security Symposium 2023, pp.282-pp.287, 2023

[4] Movielens, (https://grouplens.org/datasets/movielens/, 2024 年 9 月参照)

[5] 個人情報保護委員会,(https://www.ppc.go.jp/personal_info/legal/ fundamentalpolicy/, 2024 年 12 月参照)

[6] Tianhao Wang, Jeremiah Blocki, and Ninghui Li, Somesh Jha, Locally Differentially Private Protocols for Frequency Estimation, 26th USENIX Security Symposium, 2017.

[7] Differential Privacy Team, Apple, Learning with Privacy at Scale, 2017.

[8] Xuebin Ren, Chia-Mu Yu, Weiren Yu, Shusen Yang, Xinyu Yang, Julie A. McCann, and Philip S. Yu, “LoPub : High-Dimensional Crowdsourced Data Publication With Local Differential Privacy,” in IEEE Transactions on Information Forensics and Security, vol. 13, no. 9, pp. 2151-2166, Sept. 2018.

[9] CVXPY, (https://github.com/cvxpy/cvxpy/, 2025 年 1 月参照)

付録 A

Phishing URL 攻撃パターンの自動分類とその評価

付録 B

研究背景

近年フィッシング攻撃による被害は増加傾向を取り続けている. 特に被害の中心となっているのは大手 EC サイトを模倣した偽造サイトや, 金融機関を偽ったサイトなどの URL を被害者に対し送り付け, そこから信用情報などを盗む手口である. そういった詐欺被害を防ぐために送付される Phishing URL に対して URL ベースでの解析を行い, 攻撃を事前検知する方法は複数提案されている. Kim[1] らは, ドメインや URL, IP アドレス等からなる異種混合ネットワークを作成し, 予測するという, 検知を回避するような構造をしている Phishing URL も十分に検知できる手法を提案している. 小出 [2] らは, ChatGPT を用いた検出手法を用い, 著名なサイトの偽装サイトについて 98 %を超える検出精度を記録している. しかしこれらの手法はいずれも Phishing URL かどうかの検知までを行っており, その Phishing URL が用いている攻撃手法については言及されていない.

そこで, 攻撃手法パターンについての高速分析を行い攻撃者の傾向分析を行い, 流行している攻撃手法を割り出すことで新たな視点からの有効なフィッシング対策手法を提案できるのではないかと考えた. 本研究では JPCERT より提示された最新の Phishing URL 群 [6] に対し攻撃パターンを自動検出, 分類することで, 攻撃者の攻撃傾向を大量の Phishing URL 群から読み解き, 流行している攻撃手法を把握することを目的とする. 類似した研究として小嶋 [3] らが行った研究が挙げられる.

目的達成のために本稿では, JPCERT によって提示された最新の Phishing URL 群について, 作成した検知モジュールを含むシステムにより攻撃手法を自動判別, 分類する実験を行い, 攻撃者の傾向を掴む. 加えて, Phishtank によって提示されたグローバルな URL 群についても同様の実験を行い, 海外と日本の傾向の比較を行う. また, 作成した検知機構の精度分析を行い, 問題点, 改善点について検討する.

付録 C

攻撃パターン自動分類機構

C.1　システム構成

本研究では Phishing URL 群が提示された際, 4 つのプロトコルからなる攻撃パターン自動分類機構により自動的に攻撃パターンを検出, 分類する. 想定されるシステム稼働例を図 C.1 に示す.

C.2 Phishing URL 分類手法

本研究では与えられた Phishing URL に対し, その攻撃手法が「タイポスクワッティング」, 「コンボスクワッティング」, 「IP アドレス挿入」, 「ランダム文字」「その他」の 5 つの分類のうちどれに属するかを判別する. なおこの 5 分類は先行研究に従って作成された分類 [4] である.

分類には「その他」の分類を除く各分類につき 1 つずつ作成した, 4 つの検出モジュールを用いる. 各検知モジュールにより, 与えられた Phishing URL の攻撃手法が分類された時点でシステムはその Phishing URL への処理を停止し, 次の URL の分類へと移行するものとなっている. 以下, 本項にて 4 つの検知モジュールへの説明を行う.

C.2.1 タイポスクワッティング検知モジュール

タイポスクワッティングとは, 攻撃者が正規のブランド名へとアクセスする際の打ち間違いを期待して, 正規のブランド名から一部分を変更した文字列を URL 内に含むことにより, 偽サイトへ誘導する攻撃手法のことである。

タイポスクワッティング検知へのアプローチとして, 正規ブランドと Phishing URL のドメイン部間のレーベンシュタイン距離 d をとり, d が閾値を下回る場合をタイポスクワッティング攻撃と分類する機構を作成し, これにより分類することとした. なおレーベンシュタイン距離とは, 2 つの文字列 x, x' について, x を x' に変換するのに必要な削除, 挿入, 置換の数である. そのため, レーベンシュタイン距離が大きいほど文字列が異なっているといえる. 例えば x ='levenshtein', x ='levanshte' とすると, x を x' にするために必要な操作は置換 1 回, 挿入 2 回であるため, レーベンシュタイン距離 d は $d = 3$ となる. [5]

以下がタイポスクワッティングを用いた Phishing URL のドメイン部が満たす, 正規ブランド名とのレーベンシュタイン距離 d についての関係式である。

$$Levenshteindistance : d \leq 2$$

また, 本稿で用いた正規ブランド名のリストは Python ライブラリの request と Beautifulsoup を用いて, 与えられた Phishing URL の偽装元である正規ブランド名について, google 検索を行い, その結果の一番上位のサイトのドメイン部を自動収集する方法で作成した. 表 C.1 に作成したブランド名の単語リストの一部を示す.

C.2.2　コンボスクワッティング検知モジュール

コンボスクワッティングとは, 正規のブランド名に一見信頼できる単語を付加することであたかも信頼できるサイトかのように偽サイトへと誘導する攻撃手法のことである。

コンボスクワッティング検知へのアプローチとして, 正規ブランド名にコンボスクワッティングで比較的高頻度で用いられる単語を付加していた場合に検知する機構を作成し, これにより分類することとした. なおここで検知する単語とは, 表 C.2 で示されているような, コンボスクワッティングの際に高頻度で用いられる傾向のある単語を指す. この単語リストは自らが過去に行った URL 分析の結果を踏まえて製作されている. 検知対象となる単語は表 C.2 で示されているものを含め 31 件である.

C.2.3 IP アドレス挿入

IP アドレス挿入とは, URL に直接 IP アドレスを挿入する URL 作成手法のことを指す。この攻撃手法は URL 文字列を注視すれば簡単に Phishing URL であると気づくことができるため, 不注意なユーザをターゲットとしていると考えられる.

IP アドレス挿入へのアプローチとして, URL が数字と特殊文字列のみで構成されている場合に検知する機構を作成し, これにより分類することとした.

C.2.4 ランダム文字

ランダム文字とは, Phishing URL をランダムに生成された文字列から作成する手法のことを指す。この攻撃手法は URl 文字列を注視すれば簡単に Phishing URL であると気づくことができるため, 不注意なユーザをターゲットとしていると考えられる. また, 大量の Phishing URL を短時間で生み出すことができることも大きな特徴である.

ランダム文字へのアプローチとして, URL 文字列のドメイン部について, 文字遷移確率 p を計算する Python ライブラリを用いて文字遷移確率 p を計算し, その値が閾値を下回っていた場合をランダム文字として検知する機構により, 分類することとした. ランダム文字手法を用いた Phishing URL の文字遷移確率 p が満たす式は以下の通りである.

$$p \leq 0.5$$

使用する Python ライブラリは texttrans という, 学習した単語帳との類似度を比較し文字遷移確率を算出するものである. 学習させた単語帳はデフォルトで学習させられていた英単語帳を使用している.

なお, 以上 4 つの分類に含まれない Phishing URL については本稿では「その他」の分類として扱うこととしている.

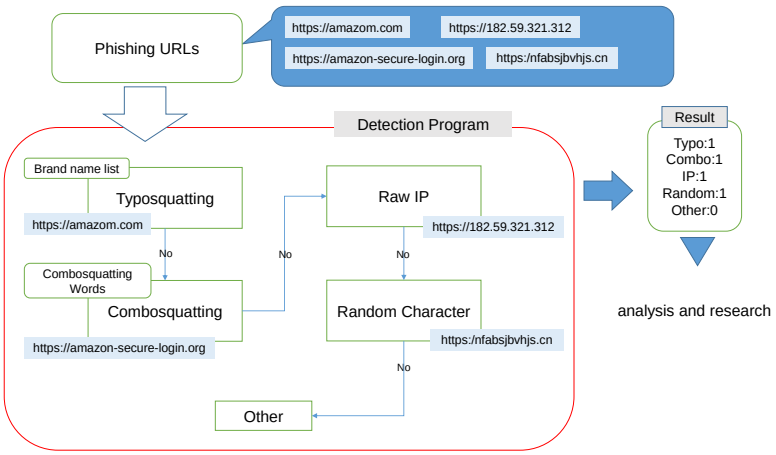

図 C.1　システム稼働例

表 C.1　ブランド名の単語リストの一部 (大文字, 小文字の判別はしない)

ionos	aeonbank	gmo-aozora	globalcu
australia	minatobk	dream	eonet
weblio	akita-bank	eposcard	usaa

表 C.2　コンボスクワッティングで用いられる単語の一部 (大文字, 小文字の判別はしない)

support	secure	login	help
account	verify	service	find
online	update	web	maps
alert	mail	verification	signin

付録 D

実験

D.1　プログラムと実験の流れ

本実験は JPCERT によって提示された最新の Phishing URL 群 [6] を用いて実験を行う. 2019 年 3 月から 2023 年 9 月までの 55 か月間のデータから偽装に用いられている正規ブランド名リストを作成し, 検知のためのリストとして使用した. 同じく 55 か月間のデータに含まれている Phishing URL 群について先述したモジュールによる自動分類を行った. そして得られた分析結果を月ごとの積み上げ棒グラフとして表現し, 攻撃手法の変遷を分析した. また, Phishtank から得た 2023 年 12 月以前の最新 38962 件の Phishing URL 群についても同様の分類を行った。

検知機構の精度分析については目視により攻撃手法を分類し, それを正しい攻撃手法の分類とした. 目視により分析した 102 件の Phishing URL について検知機構との分類結果を比較し, 精度の検証を行った.

D.2　結果

まず攻撃手法の自動分類の結果は図 2, 図 3 のようになった。

図 D.1, D.2 に示すように, 2019 年, 2023 年ともに最も大きい割合を占める手法はランダム文字であるということがわかった．また, 2019 年と 2023 年で攻撃手法の割合を比較すると差異があることがうかがえる．2019 年度では 20 %程度の割合を占めていたタイポスクワッティング, コンボスクワッティングの手法が, 2023 年度では 10 %程度の割合にまで減少している．また今回の検知機構で判別できなかった, その他の攻撃手法が 2019 年から 2023 年にかけて 2 倍程度に増加している．

続いて, Phishtank, JPCERT から提示された URL についての分析結果の比較を表 D.1 に示す．用いた URL として, Phishtank は 12 月 8 日時点で収集した 38962 件を用いた．JPCERT は提示されている中で最新の, 2023 年度 9 月度のものを使用した。URL 件数は 6024 件であった．表 D.1 は各攻撃手法ごとの占める割合のパーセンテージが示されている．例えば, JPCERT におけるタイポスクワッティングの割合は 3.3 %であった．表 D.1 からわかる通り, Phishtank と JPCERT では大きく攻撃傾向に乖離があることがわかった．タイポスクワッティング攻撃の割合が JPCERT では 5.4 %程度であるのに対し, Phishtank では 24.8 %と大きく Phishtank の割合が上回った．また, ランダム文字攻撃では JPCERT では 82.6 %を占めているのに対し, Phishtank では 59.3 %とこちらは JPCERT が 20 %近く上回る結果となった．

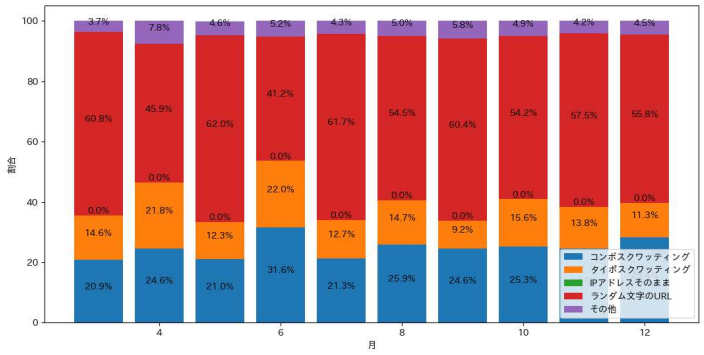

図 D.1　2019 年の攻撃手法比率

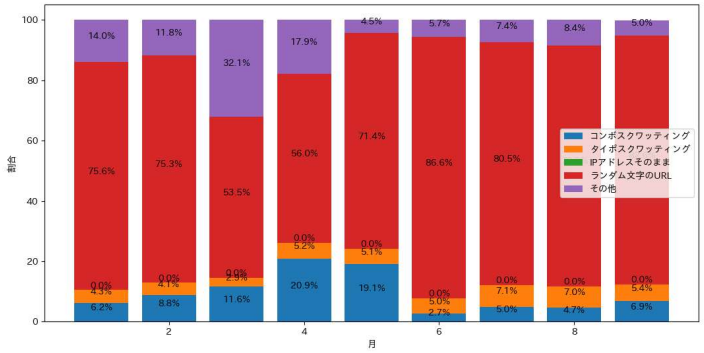

図 D.2　2023 年の攻撃手法比率

表 D.1　Phishtank と JPCERT 最新の攻撃手法割合比較 (単位: %)

	Typo	Combo	IP	Random	Other
JPCERT	5.4	6.9	0.0	82.6	5.0
Phishtank	24.8	6.8	0.0	59.3	9.1

D.3　評価

次に, 検知機構の精度分析の結果について示す. まず 102 件の正誤を可視化するために, 図 D.3 の混同行列を作成した. 縦軸が正解の分類ラベル, 横軸が検知機構が予測した分類ラベルである. (まだラベル直してない)

また, 今回の実験において IP アドレス挿入による攻撃手法が検知された回数は 0 回であったため, 今回の混同行列, 精度分析では IP アドレス挿入への分析は行わない.

図 D.3 を用いて, 各精度指標を算出した. その結果を表 D.2 に示す. また全体の精度 acc は以下のようになった.

$$acc = 0.61$$

図 D.3, 表 D.2 から得られた精度としては, ランダム文字への $Recall$ 値が 78 件の URL に対し, 約 0.936 の数値を得ることができた. また, 全体の精度 acc は 0.61 と 60 %程度の Phishing URL に対して正しい攻撃手法分類を行えていることが分かった.

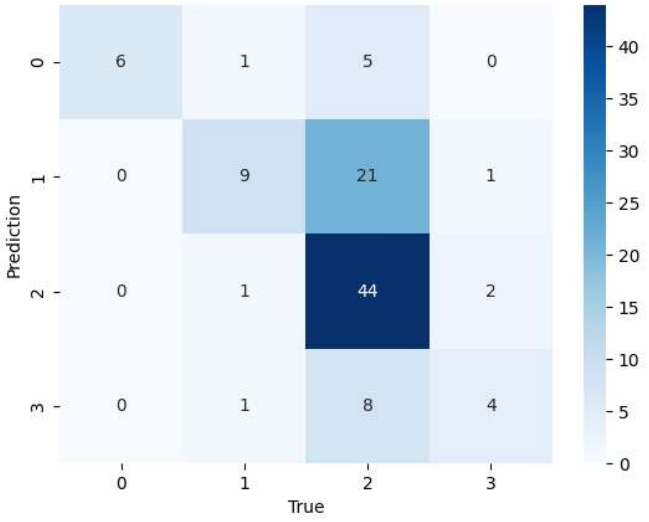

図 D.3　混同行列

表 D.2　精度の指標

	Typo	Combo	Random	Other
Recall	0.5	0.29	0.94	0.31
Precision	1	0.75	0.56	0.57
F-1	0.67	0.42	0.70	0.40

付録 E

考察

実験から得られた結果から, 攻撃者の用いている Phishing URL 生成手法は大きく変遷していることがわかった. [4] で示されていた結果と比較すると, ランダム文字攻撃, その他の割合が増加していることが伺える. 特にその他の割合の増加については, 既存手法では分類できない手法での攻撃が多く出現していることが原因と推察される. 例えば, ランダムな単語を複数挿入するなどといった攻撃が, 今回の実験で用いたデータ内で観測された. こういった新しい攻撃手法に対してどのように対策していくかは目下の課題といえるだろう.

また, 海外と日本国内の Phishing URL 攻撃手法の傾向には大きく差があるといえる. ランダム文字攻撃が主流の攻撃であるという点は同じであるが, その割合は国内が多かった. また, タイポスクワッティング攻撃の割合ではその逆で海外が多かった. 今回得られた海外と日本国内の攻撃手法の傾向の差異は, 攻撃者の発信元を特定したり, 有効なフィッシング対策を立てるうえで有用な情報となりえるのではないだろうか. また, 今回の実験精度についてはあまり良い結果とは言えないものとなった. コンボスクワッティング判別に用いた単語リストの調整等, 改善点は多く存在する. 特に誤検知が発生した主な原因として, 検知モジュールが並列構造でなかったということが挙げられる. 各モジュールに順番に通していく構造のシステムを開発したため, 複数手法を用いた方式の URL についての対応が困難であった. 今回は実現することができなかったが, 並列での検知をできるようなシステムを開発することを課題とする.

付録 F

おわりに

本稿では自動検出機構を用いた Phishing URL 攻撃手法パターンの自動判別を行った. 実験結果からうかがえた, その他に分類される攻撃手法についての自動判別, また検知機構の精度向上を今後の課題とする.

参考文献

[1] Taeri Kim, Noseong Park, Jiwon Hong, Sang-Wook Kim, Phishing URL Detection: A Network-based Approach Robust to Evasion”, CCS ’ 22, 2022.

[2] Takashi Koide, Naoki Fukushi, Hiroki Nakano, and Daiki Chiba, Detecting Phishing Sites Using ChatGPT, 2023.

[3] 小嶋 彬彦, 加藤 雅彦, メールフィッシングに使用される手法の分類とその検知に関する研究, CSS ’23, 2023.

[4] 佐野 智弥, “最近のフィッシング URL の生成手口を分析！よくある攻撃パターンとは？” (https://www.lac.co.jp/lacwatch/people/20230303_00 3297.html, 2023 年 10 月参照)

[5] Rishin Haldar, Debajyoti Mukhopadhyay, Levenshtein Distance Technique in Dictionary Lookup Methods: An Improved Approach, 2011.

[6] 中井 尚子, “JPCERT/CC が確認したフィッシングサイトの URL を公開” (https://blogs.jpcert.or.jp/ja/2022/08/phishurl-list.html, 2023 年 10 月参照)