TEMPEST: 表形式データに対するモデル抽出攻撃

辰巳 将崇[1,a)]　岩花 一輝[1]　矢内 直人[1]　宍戸 克成[2]　清水 俊也[2]　樋口 裕二[2]　森川 郁也[2]
矢嶋 純[2]

概要：モデル抽出攻撃は提供されている機械学習モデルに対するクエリとその出力を通じて, 攻撃者が同等程度の性能を持つモデルを得る攻撃である. 本稿では表形式のデータに対し, 攻撃者が攻撃対象のモデルの学習に用いたデータサンプルを一切持たないデータフリー設定を, より現実的に行う新たなモデル抽出攻撃 TEMPEST を示す. 表形式ではデータの正規化処理に起因して攻撃が画像分類よりも複雑化するが, TEMPEST では公開された統計情報を通じてクエリ用サンプルを生成することで, 初期サンプルを必要とする従来の攻撃と同等程度の性能をデータフリー設定で実現できる. さらに, TEMPEST のデータ生成では平均と分散を統計情報として用いること, また, 抽出モデルの正規化処理は可能な限り被害者モデルと同じものを用いることで性能を改善できることも確認した.

キーワード：モデル抽出攻撃, データフリー, 表形式, 機械学習

Model Extraction Attacks on Tabular Data

Masataka Tatsumi[1,a)]　Kazuki Iwahana[1]　Naoto Yanai[1]　Katsunari Shishido[2]
Toshiya Shimizu[2]　Yuji Higuchi[2]　Ikuya Morikawa[2]　Jun Yajima[2]

Abstract: Model extraction attacks are a kind of attack whereby an adversary obtains a machine learning model whose performance is identical to a model to be attacked through queries and their results. This paper presents a novel model extraction attack, named TEMPEST, on tabular data for a milder data-free setting. Whereas model extraction is more challenging on tabular data due to normalization, TEMPEST can provide the same-level performance as existing attacks, which need initial samples, by generating query samples through public statics. Moreover, we identify that the use of mean and variance for the data generation of TEMPEST and the use of the same normalization process as a victim model will improve the performance of TEMPEST.

Keywords: model extraction attack, data free, tabular data, machine learning

1. 序論

1.1 背景

機械学習は様々な分野で高い効果を生むことから, コンテンツとしての機械学習モデルの価値が高まっている. 一般に, 機械学習モデルの構築に向けた学習データの収集と学習処理は高い負荷を要する作業であり, 例えば Yang ら [1] のモデルでは 61,000 ドルから 250,000 ドルも消費している[*1]. このような機械学習モデルの価値の向上を受けて, サービスとして提供されているモデルに対するクエリとその推論結果を学習することで攻撃者が同等程度のモデルを得るモデル抽出攻撃 [2] が近年注目されている [3], [4], [5], [6], [7]. 大まかには, 攻撃対象となるモデル（被害者モデルと呼称）を踏み台に, 負荷の高いデータ収集作業および学習処理に係る手間を大幅に削減することで, 新たなモデル（抽出モデルと呼称）を得る攻撃といえる. 近年では自然言語処理

[1] 大阪大学
Osaka University
[2] 富士通株式会社
Fujitsu Limited
[a)] m-tatsumi@ist.osaka-u.ac.jp

[*1] https://syncedreview.com/2019/06/27/the-staggering-cost-oftraining-sota-ai-models/

モデルへの攻撃 [8], [9] や画像生成モデルへの攻撃 [10] など, 実際の応用技術への攻撃例も示されている. 高い費用が掛かったモデルが不正に二次利用された場合, モデルの本来の所有者における収益にも影響することから, モデル抽出攻撃は極めて重要な問題である.

一方, 既存のモデル抽出攻撃 [2], [3], [4], [5], [6], [7] は MNIST データセットなど汎用的な学術ベンチマークのみを対象に検討しており, 被害者モデルが扱うデータの形式を考慮した研究は行われていない. 一般に機械学習は与えられた学習データの性質に応じてモデルの機能が異なることから, データの形式はモデル抽出攻撃の成功可否に影響することが予想される. しかしながら, そのようなデータ形式を考慮した検討はこれまで示されてこなかった.

上述した観点に対し, 本研究では**表形式データを対象とすることで, より現実的かつ効果的なモデル抽出攻撃が可能か**明らかにする. 表形式は身長や血圧を含む医療用データセットや, 世代ごとの嗜好や商品の売れ行きを含む金融系データセットなど, 機械学習において最も一般的なデータ形式である [11]. このため, 表形式に対して上述した問いを明らかにすることで, 深層学習をはじめ様々な機械学習サービスへ応用できる知見が期待できる. 加えて, 表形式では画像や言語処理で扱われるような, 学習の収束を促す汎用手法は利用できない [12]. 多くのモデル抽出攻撃の研究 [3], [4], [5], [6], [7] は画像分類を対象としていることから, 表に対する攻撃の結果は単純には包含されない.

1.2 貢献

本稿では表形式データを扱うモデルに対する新たなモデル抽出攻撃 *TEMPEST (Tabular Extraction from Model by Public and External STatistics)* を示す. 大まかには, TEMPEST は被害者モデルが持つ学習データセットに関して平均と分散など公開された統計情報のみを攻撃者の前提知識とすることで, 攻撃者が被害者モデルの学習データ自体は一切持つことがないデータフリー設定 [5], [7] でのモデル抽出をより現実的に行える, 新たなモデル抽出攻撃のクラスといえる.

表形式に対するモデル抽出攻撃は（データフリー設定に拠らず）非自明である. 具体的には, 表形式への攻撃では, データの正規化処理を考慮しなければならない. 既存研究 [3], [4], [5], [7], [13] で対象とする画像分類において特徴量は各ピクセルの数値に閉じている. 一方, 表形式では前節にも述べた通り多様であるため, 特徴量が決まった範囲に閉じていないことが考えられる. 一般に, 学習データの各特徴量がどの範囲にあるか事前に予想することは難しい. このため, 表形式に対しては, 多様なデータ範囲を考慮したモデル抽出攻撃を新たに検討する必要がある.

TEMPEST では公開された統計情報を利用することで, モデル抽出攻撃に適したデータを攻撃者が自ら生成する. ここでいう統計情報とは学習データの平均や分散等を指しており, 詳細は 3.2 節で述べるが, これは既存のデータフリー設定 [5], [7] と比べて, より現実的な仮定といえる.

表形式データセットとして, Adult, Cancer, Diabetes, Arrhythmia を用いて実験したところ, TEMPEST では公開された統計情報を通じてクエリ用サンプルを生成することで, 初期サンプルを必要とする既存攻撃 PRADA [3] と同等程度の性能を, データフリー設定で実現できることを確認した. （詳細は 5 節を参照. ）また, TEMPEST では主な処理として統計情報からのクエリデータの生成と抽出モデルの学習があるが, このうちデータの生成は統計情報として平均と分散の利用が望ましいことを確認した. 同様に, 抽出モデルの学習においても, 抽出モデルの正規化処理は被害者モデルと同じものを利用することで TEMPEST の性能を改善できることも確認した. （詳細は 6 節を参照. ）

2. 関連研究

モデル抽出攻撃は被害モデルに対し攻撃者があらかじめ持つ知識量を減らす方向で研究されている. 初期のモデル抽出攻撃 [2] は攻撃者は被害モデルが持つ学習データを二割ほど保有する必要があった. これに対し, 後続研究 [3], [14] では攻撃開始時に持つ数個のサンプルから敵対的サンプル [15] を生成することで, 攻撃者が自らデータ分布を生成する攻撃を示している. また, 攻撃者がデータ分布を代替するデータセットを保有することで, 学習データそのものは一切なくても可能な攻撃手法 [4], [6] もある. 一方, 代替データを用意することが負荷が大きいことから, 攻撃者が一切データを持つことなく, クエリだけでモデル抽出攻撃を行うデータフリー設定が近年に示された [5], [7]. しかしながら, これらのデータフリー設定でのモデル抽出攻撃は 3.2 節に述べる観点から非現実的といえる.

被害モデルのアーキテクチャに応じてもモデル抽出攻撃の性質や性能は異なる [16]. このため, モデルの種別 [17] や学習方法 [18], [19], [20] に応じた検討もされている. 近年では画像処理 [10] や言語処理 [8], [9] など応用アプリケーションへの攻撃も示されている. なお, GPU など物理媒体へのアクセスを通じて, アーキテクチャ自体を推定する攻撃もある [16], [21], [22]. 本稿の結果を上述したアーキテクチャやモデル種別に寄らせたモデル抽出攻撃と組み合わせることで, 様々な応用事例の検討も期待できる.

3. 問題設定

本節では本稿の予備的知識として機械学習およびモデル抽出攻撃の概念について説明する.

3.1 機械学習とモデル抽出攻撃

$\mathbb{N}$ を自然数の集合, $\mathbb{R}$ を実数の集合, $\mathcal{C}$ をラベルの集合とする. 機械学習モデルは任意の $m, n \in \mathbb{N}$ において, m 次

元の特徴量を持つ入力 $x \in \mathbb{R}^m$ を n 個のラベルそれぞれに関する確率を出力する関数 M として定義される．より正確には，$M(x)$ は $i \in [1, m]$ における各ラベルを持つクラス $c_i \in \mathcal{C}$ に，x が属する確率 p_i をそれぞれ出力する．このとき，機械学習モデル M は任意のデータとラベルの組 $(x, c) \in \mathbb{R}^m \times \mathcal{C}$ を入力し M の内部パラメータ θ を鍛える学習処理と，パラメータ θ を持つモデル M_θ が未知の入力 x' を入力し上述した確率を出力する推論処理からなる．

モデル抽出攻撃における攻撃者 $\mathcal{A}$ の目的は，攻撃対象となる機械学習モデル $M_\mathcal{V}$（被害者モデルと呼称）を模倣する抽出モデル $M_\mathcal{A}$（抽出モデルと呼称）を得ることにある．攻撃者は，事前に持つラベルなしデータ $X_\mathcal{A}$ を用い，被害者モデルへクエリ $M_\mathcal{V}(X_\mathcal{A})$ することで，推論結果 $c_\mathcal{A}$ を得る．このとき，$\mathcal{A}$ は $X_\mathcal{A}, c_\mathcal{A}$ を用いて，$M_\mathcal{A}$ を訓練することで，被害者モデルと同等の性能の抽出モデル $M_\mathcal{A} \simeq M_\mathcal{V}$ を得る．モデル抽出攻撃は，抽出モデル $M_\mathcal{A}$ に関して以下の 2 つの指標から評価される．

精度: 抽出モデル $M_\mathcal{A}$ がどれだけ正確に分類できるか，すなわち評価データセット $D_{val} : X \times Y$ に対して $argmax(M_\mathcal{A}(x_{val})) = y$ となる指標である．精度を用いることで攻撃者は，抽出モデルの有用性を評価できる．

忠実度: 抽出モデル $M_\mathcal{A}$ と被害者モデル $M_\mathcal{V}$ の振る舞いが一致するかという指標であり，$Pr_{x_{val} \sim D_{val}}[argmax(M_\mathcal{A}(x_{val})) = argmax(M_\mathcal{V}(x_{val}))]$ と表す．忠実度を用いることで，攻撃者は抽出モデル $M_\mathcal{A}$ の出力がどれだけ $M_\mathcal{V}$ の出力に似ているか評価できる．

とくに，上述した 2 つの評価軸ではそれぞれで攻撃者の目的が異なる．精度は，被害者モデル $M_\mathcal{V}$ を踏み台にして，攻撃者が自身で $M_\mathcal{A}$ を用いたサービスの展開を目的とするときに，有用な評価指標である．一方，忠実度は，攻撃者が被害者モデル $M_\mathcal{V}$ の精度関係なく，モデルそのものを複製したい場合に有用な評価指標である．

3.2 既存のデータフリー設定の問題点

TEMPEST は表形式に対してデータフリー設定を現実的に行う攻撃である．詳細は次節以降に述べるが，本節では既存のデータフリー設定 [5], [7] の問題点を述べる．

既存のデータフリー設定はデータの正規化処理を考慮できていない点で問題がある．大まかには既存の攻撃では被害者モデルに合わせたデータの正規化ができている前提になっており，被害者モデルに対するクエリ用データはモデルの入力用に正規化した 0 から 1 の範囲で生成されている．しかし，一般にはデータがどのように正規化されているかはわからないため，このような正規化処理がされたことを前提としている点は，現実における攻撃の適用範囲を大幅に狭めてしまう．関連して，データが正規化されるにしても，そのデータの値が取りうる範囲も厳密には不明である．画像では 0 から 255 からなるピクセル値が特徴量となる一方，表では身長や体重など多様な特徴量が扱われる．つまり，表形式では個々の特徴量の取りうる範囲が不定になる．攻撃者が自らデータの正規化を行ったとしても，その正規化処理が被害者モデルのものと一致していない限り，抽出モデルの精度が向上しないことも起こりえる．これらを踏まえ，既存のデータフリー設定では現実のシステムに対する攻撃は難しいといえる．

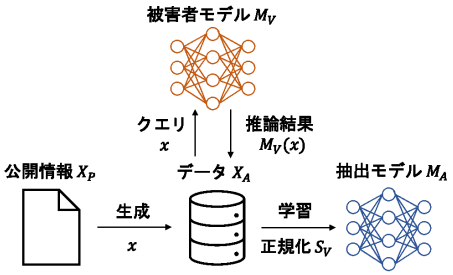


図 1: TEMPEST 攻撃の全体像

上述した問題に対し，本研究では「公開された統計情報を基にすることで，上述したデータフリー設定の問題が改善できる」という仮説を検討する．大まかには，被害モデルが利用しているであろうデータの平均や最大値・最小値など統計情報を得ることができたとき，攻撃者はデータフリー設定でのモデル抽出攻撃が可能となる．一方，例えば我が国における各年代・性別ごとに，金融データに対しては平均年収，健康データに対しては身長・体重など，様々なデータが利用できることが考えられる．すなわち，統計情報の利用は実用的な側面もあるとみなせる．

4. TEMPEST 攻撃

本節では新たなモデル抽出攻撃として TEMPEST を述べる．前述した通り，TEMPEST では既存の攻撃と比較して，公開されている統計情報を用いることで，被害者モデルへのクエリだけで高い精度を持つ抽出モデルを得ることができる．まず TEMPEST の全体像を述べたのち，攻撃アルゴリズムの詳細について述べる．

4.1 全体像

TEMPEST の全体像を図 1 に示す．主な機能として，従来の攻撃同様の被害者モデルへのクエリによる抽出モデルの学習に加え，公開情報からデータの生成も行う．

まず，攻撃者は抽出モデルの学習に必要なデータを生成する．具体的に，被害者モデルの学習データを構成する各特徴量に関して，公開されている統計情報（以降では公開情報と呼称）からデータを生成する．ここで統計情報とは，その特徴量に関する平均・分散，または，最小値・最大値などが該当する．これにより，攻撃者は被害者モデルにクエリするデータセットを自ら構築する．次に，攻撃者は構築したデータセットから選択したデータを被害者モデルにク

アルゴリズム 1 TEMPEST(データ生成)

入力: 被害者モデル $M_{\mathcal{V}}$, 抽出モデル $M_{\mathcal{A}}$, 公開情報データ $X_{\mathcal{P}}$, データ生成方法 gen_mode.
出力: 抽出モデルの学習データ $X_{\mathcal{A}}$.

1: **procedure** GENERATE_DATA($StaticalParams, gen\_mode$)
2: **if** $gen\_mode ==$ "Gen_std" **then**
3: $mean, std \leftarrow StaticalParams$
4: $X_{\mathcal{A}} \leftarrow Gaussian(mean, std)$
5: **else if** $gen\_mode ==$ "Gen_min" **then**
6: $min, max \leftarrow StaticalParams$
7: $X_{\mathcal{A}} \leftarrow Uniform(min, max)$
8: **end if**
9: **return** $X_{\mathcal{A}}$
10: **end procedure**
11: $StaticalParams \leftarrow Select\_public\_knowledge(X_{\mathcal{P}})$
12: $X_{\mathcal{A}} \leftarrow Generate\_data(StaticalParams)$

アルゴリズム 2 TEMPEST(抽出モデルの学習)

入力: 被害者モデル $M_{\mathcal{V}}$, 抽出モデル $M_{\mathcal{A}}$, 公開情報データ $X_{\mathcal{P}}$, 抽出モデルの学習データ $X_{\mathcal{A}}$, 被害者モデルの正規化パラメータ $S_{\mathcal{V}}$.
出力: $M_{\mathcal{A}}$.

1: **procedure** QUERY($M_{\mathcal{V}}, X_{\mathcal{A}}$)
2: $X^*_{\mathcal{A}} \leftarrow Victim\_scale(X_{\mathcal{A}}, S_{\mathcal{V}})$
3: $c_{\mathcal{A}} = M_{\mathcal{V}}(X^*_{\mathcal{A}})$
4: **return** $c_{\mathcal{A}}$
5: **end procedure**
6: $c_{\mathcal{A}} \leftarrow$ QUERY($M_{\mathcal{V}}, X_{\mathcal{A}}$)
7: $X^{**}_{\mathcal{A}} \leftarrow Adversary\_scale(X_{\mathcal{A}})$
8: $D_{\mathcal{A}} \leftarrow (X^{**}_{\mathcal{A}}, c_{\mathcal{A}})$
9: $M_{\mathcal{A}} \leftarrow Training(M_{\mathcal{A}}, D_{\mathcal{A}})$

エリし, 得られた推論結果を基に抽出モデルを学習する.

この一連の処理では, 攻撃者は公開情報さえあれば自らデータを生成することで, 被害者モデルの API を通じて攻撃が可能となる. 被害者モデルの正規化処理については特に攻撃者は知識を必要としない. 同様に, 抽出モデルで行う正規化処理についても, 攻撃者が得た公開情報の形式に従って複数の正規化処理を並行して実施することで, 最も効果的な正規化処理を攻撃者が選択する. これらにより, 被害者モデルの正規化処理によらず抽出モデルの学習を最適化することで, 表形式に対する現実的な攻撃が可能となる.

4.2 攻撃手法

TEMPEST の詳細な動作を以下に述べる. TEMEPST の機能はデータ生成と抽出モデルの学習の二つからなる.

まず, アルゴリズム 1 にデータ生成を示す. 攻撃者はこのアルゴリズムを通じて公開された統計情報を用いることで, 抽出モデルの学習データを生成する. このとき, 平均と分散を用いてデータを生成する方法 (Gen_std と表記), あるいは, 最小値と最大値を用いてデータを生成する方法 (Gen_min と表記) の 2 種類がある. Gen_std の処理は 2-4 行目に該当し, 公開された統計情報からそれぞれの各特徴量の平均 $mean$ と分散 std の値を算出する. 求めた平均と分散から, 各特徴量を構成するサンプルのデータ群が正規分布に従うようにデータを生成する. 一方, Gen_min の処理は 5-7 行目に該当し, 公開された統計情報から各特徴量の最小値 min と最大値 max を算出する. サンプルのデータ群が各特徴量の最小値と最大値の範囲において一様分布に従うようにデータを生成する. その生成したデータにより, 抽出モデルの学習を行う.

次に, アルゴリズム 2 に抽出モデルの学習を示す. 攻撃者は 1-4 行目の処理として, 生成したデータを被害者モデルにクエリする. このとき, 前提条件から攻撃者は被害者モデルの正規化処理 $Victim\_scale$ の詳細を知らないため, 被害者モデルと同じ正規化処理を行うことができない. このため, 7 行目で攻撃者は自らの正規化処理 $Adversary\_scale$ を決定し, 8 行目で被害者モデルの推論結果と組み合わせることで抽出モデルの学習データセットを作成する. このデータセットを用いて, 抽出モデルを学習する.

表 1: 本実験で用いるデータセットと精度

データセット名	特徴量の次元数	インスタンス数	クラス数	精度
Adult	14	32600	2	81.7%
Cancer	32	569	2	98.9%
Diabetes	9	768	2	76.1%
Arrhythmia	279	452	16	64.3%

5. 実験

本節では TEMPEST に関する実験を述べる. 実験の目的と設定を述べたのち, 実験結果を示す. また, それらを踏まえて TEMPEST について考察する.

本稿の実験目的は, TEMPEST の有効性として, 高い精度を持つ抽出モデルが得られるか明らかにすることである. 既存のモデル抽出攻撃 [3], [5], [7] と比べて, 精度が改善されていることを確認する. とくにデータフリー攻撃 [5] と比べて, 3.2 節で述べた問題点を改善しているにもかかわらず, 同等以上の精度が達成できるか確認する.

5.1 実験設定

5.1.1 データセットとアーキテクチャ

本実験で用いるデータセットを 表 1 に示す. Adult[*2], Cancer[*3], Diabetes[*4] は 2 クラス分類を行う表形式データセットである. より複雑なタスクとなる多クラス分類を行う表形式データセットとしては Arrhythmia[*5] を用いる. なお, Adult では数値的意味を持たない名義変数 (カテゴリカル変数) が利用されている特徴が存在する. 名義変数の特徴に対しては, 統計情報は意味を持たないため, その

[*2] https://archive.ics.uci.edu/ml/datasets/adult
[*3] https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)
[*4] https://www.kaggle.com/uciml/pima-indians-diabetes-database
[*5] https://archive.ics.uci.edu/ml/datasets/arrhythmia

特徴量を構成する要素の中からランダムサンプリングしてデータを生成する．各データセットは被害者モデルの学習用, 抽出モデルの性能の検証用, 抽出モデルが用いる公開情報用として, 1:3:1 に分割する．

モデルのアーキテクチャは, 全結合層三層のニューラルネットワークを用いており, 中間層のノード数は 90 個である．本実験では, 被害者モデルと抽出モデルのアーキテクチャは同一のものを用いる．ハイパーパラメータは, 被害者モデルと抽出モデルの学習率を 0.01, エポック数を 30 とする．また, 被害者モデルのデータの正規化処理に関しては, 平均と分散を用いた正規化処理を行う．

5.1.2 ベースライン

ベースラインとして各データセットにおける被害者モデルの精度を示す．Adult は 81.7%, Breast Cancer は 98.9%, Diabetes は 76.1%, Arrhythmia は 64.3%である．このベースラインの下で, 代表的なモデル抽出攻撃である PRADA [3] による抽出モデルと TEMPEST の精度及び忠実度を比較する．PRADA [3] は攻撃開始時の初期値として被害者モデルの学習データを数個は持つことを前提としており, 本実験では各データセットの初期サンプルとして各クラスにつき 10 個持っているとする．ただし, Arrhythmia ではそれぞれのクラスで偏りがあり, 初期サンプルは 76 個とする．また, データフリー攻撃 [5] とも精度と忠実度を比較する．

なお, 本実験では実在する公開資料を統計情報として利用する事例調査は行わない．直観として実在する統計情報を用いた現実世界の事例調査を行うほうが望ましいが, 著者らが調査した範囲では前述したデータセット群に適した公開情報は発見できなかった．一方, 上述したデータセットの分割を公開されている統計情報の代わりとして用いることで, 実験の主旨である TEMPEST の有効性や性質などは検討が可能である．このため, 前述した各データセットの分割により, 抽出モデルの性能を評価する．

5.2 実験結果

本実験の結果を以下に示す．PRADA [3], TEMPEST , およびランダムクエリで得た抽出モデルの精度と忠実度を図 2, 図 3 にそれぞれ示す．ここで”Random”とはデータフリー攻撃 [5] の処理で生成モデルを用いず, 0 から 1 の間で生成したクエリのことを指し, 被害者モデルの正規化処理 $Victim\_scale$ を考慮しない状況設定となっている．

TEMPEST のデータ生成方法: データ生成方法において Gen_std を用いた場合に, どのデータセットにおいても Gen_min より高い精度, 忠実度を示した．この理由は, 公開されている平均と分散から生成されるデータの方が, 一様分布から生成されるデータよりも本来のデータ分布の傾向を捉えているためと考えられる．このため, 可能な限り, 平均と分散を用いるほうが望ましい．

PRADA との比較: 図 2 によると, クエリが初期サンプルに該当する場合, TEMPEST の精度は PRADA [3] より低かった．一方, 精度の差は高々 5 ポイント以内であり, データ生成が効果的にできていると言える．また, クエリ数が増加するにつれて Diabetes, Arrhythmia では TEMPEST の方が PRADA [3] より高い精度を示した．なお, Adult と Cancer でクエリ数が 2500 のときに PRADA の方が高くなる理由は, TEMPEST により得られるデータを用いた学習が収束してしまったことが考えられる．

また, 図 3 によると, Adult, Diabetes のような特徴量の次元数が少ないデータセットでは, TEMPEST よりも PRADA [3] の方が忠実度の点で優れている．一方, 特徴量の次元数が多い Cancer と Arrhythmia では, 忠実度は TEMPEST の方が高くなった．

以上を踏まえると, インスタンス数が限られるようなデータに対しては TEMPEST は初期サンプルを用いる攻撃よりも精度と忠実度の改善が期待できる．

Random との比較: Adult, Cancer, Diabetes のような低次元のデータセットに対しては, クエリ数が少ない場合でも TEMPEST と同等の精度であった．これは特徴量の次元数が少ないことに起因して, データの生成空間が小さいためが考えられる．すなわち, ランダム生成でも本来のデータ分布を捉える可能性が高くなっている．一方, データの次元数が多い Arrhythmia では, Random は PRADA, TEMPEST いずれに対しても精度, 忠実度ともに大きな差がある．これは次元数が多いことに起因して, ランダム生成では十分なデータ分布を反映できず, 抽出モデルの決定領域が不安定になっているためと考えられる．

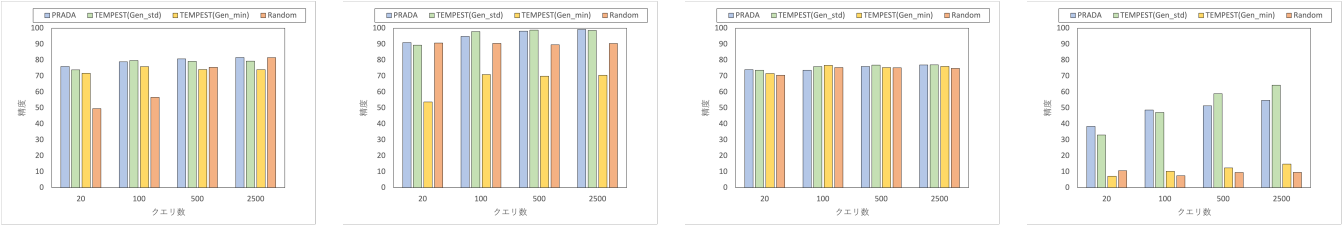
6. 考察

前節で述べた実験結果を踏まえて, TEMPEST の性質を考察する．とくに正規化処理の影響とデータセットにおける次元数の影響を考察する．また, 初期データとして若干のデータサンプルを攻撃者が持っていた場合, 精度を改善できるかについても検討する．

6.1 被害者モデルにおける正規化処理の影響

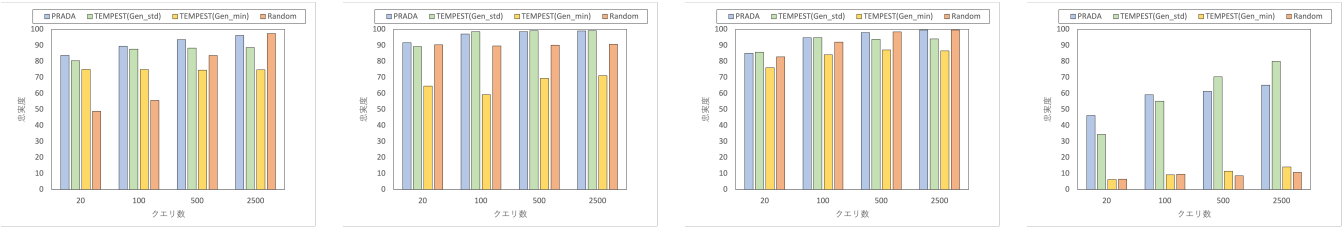
被害者モデルにおいて正規化処理の違いが与える影響について述べる．具体的に, 被害者モデルの学習において, 平均・分散を用いて正規化処理した場合 ($Standard$ と表記) と, 最小値・最大値を用いて正規化処理した場合 ($MinMax$ と表記) で, 被害者モデルの精度がどの程度変化するか各データセットで評価した．その結果を表 2 に示す．

表によると, Cancer と Diabetes のような次元数が小さく, かつ, 次元数に対してインスタンス数が 10 倍以上あるようなデータセットは, 平均・分散を用いて正規化した方が精度が高い．これは各特徴量の平均と分散が安定することで, 被害者モデルの学習も安定するためと考えられる．これに対して, Arrhythmia のように次元数とインスタンス



(a) Adult (b) Breast Cancer (c) Diabetes (d) Arrhythmia

図 2: 各手法における抽出モデルの精度比較

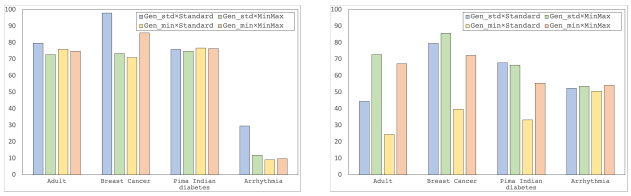


(a) Adult (b) Breast Cancer (c) Pima Indian Diabetes (d) Arrhythmia

図 3: 各手法における抽出モデルの忠実度比較

表 2: 正規化処理の違いによる被害者モデルの精度

データセット名	*Standard*	*MinMax*
Adult	81.69%	81.69%
Cancer	99.31%	91.55%
Diabetes	75.89%	70.32%
Arrhythmia	58.44%	62.14%



(a) 正規化処理: Standard (b) 正規化処理: MinMax

図 4: 被害者モデルの正規化処理の違いによる精度

数に差がない場合は, 最小値・最大値を用いるほうがより幅広いデータ分布を考慮できる. なお, Adult のようにインスタンス数が多い場合, 正規化処理の違いにかかわらず, 学習が収束しやすい. このため, Adult では精度が変化しなかったと考えられる.

以上を踏まえると, 次元数とインスタンス数の差が小さい場合, すなわち, 次元数に対して十分なインスタンス数がない場合を除き, 平均・分散を用いた正規化処理の方が望ましいといえる.

6.2 抽出モデルにおける正規化処理の影響

抽出モデルにおいて正規化処理の違いが与える影響について述べる. 具体的に, TEMPEST を行う抽出モデルにおいて, それぞれの正規化処理における精度の変化を各データセットで評価した. 被害者モデルの正規化処理が *Standard* の場合の結果を図 4a に, また, 被害者モデルの学習時の正規化処理が $MinMax$ の場合の結果を図 4b にそれぞれ示す. なお, 各グラフにおいてクエリデータの生成方法も平均・分散を用いる場合 (Gen_std と表記) と最小値・最大値を用いる場合 (Gen_min と表記) で計測している.

図 4a においては平均・分散を用いてデータ生成し, かつ, 平均・分散を用いて正規化処理を行った場合 (Gen_std×Standard に相当) が, どのデータセットにおいても精度が高い傾向がみられた. 同様に, 図 4b においては平均・分散を用いてデータ生成し, かつ, 最小値・最大値を用いて正規化処理を行った場合 (Gen_std×MinMax に相当) が, 精度が高い傾向がみられた.

以上のことから, TEMPEST におけるデータ生成は平均・分散を用いること, また, 抽出モデルの正規化処理については被害者モデルと同じものを用いることで, より高い精度が安定して得られるようになる.

6.3 初期サンプルを持っていた場合

攻撃者が PRADA [3] のように初期サンプルを持っている場合, TEMPEST より得られる抽出モデルの精度が向上するか確認する. 具体的に, 5.1 節と同じ回数のクエリを被害者モデルにするとして, 攻撃者がクエリするサンプルのうち 1 割を初期サンプルとして所有し, 残りの 9 割を TEMPEST で生成するものとする. このとき, 初期サンプルとして持つサンプルは各クラスごとに等分割する. 例えば, クエリ数が 20 個の場合, 初期サンプルは各クラスにつき 1 個ずつ (2 クラスなので, 計 2 個の初期サンプル) 持つ. 同様に, クエリ数 100 個の場合は各クラスにつき 5 個

ずつ, クエリ数 500 個の場合, 各クラスにつき 25 個ずつ持つものととする. なお, Arrhythmia ではクエリ数は 76 個, 380 個, 1900 個とし, 初期サンプルはクラスごとの偏りを考慮して, それぞれ 11 個, 48 個, 113 個とする. 上述した設定において, 初期サンプルを持っていない場合の抽出モデルの精度や忠実度をベースラインとし, 初期サンプルを持つ場合の精度と忠実度の実験結果を表 3 に示す.

表 3 によると, Diabetes を除き, いずれのデータセットにおいてもクエリ数が少ない場合は精度と忠実度が改善される傾向がみられた. とくに Cancer と Arrhythmia では両方とも約 5 ポイント改善されている. 一方, ベースラインからの差はクエリ回数が増えるにつれ, Arrhythmia を除き, いずれのデータセットにおいても小さくなっている.

以上の結果を踏まえ, TEMPEST ではクエリ回数が少ないほど, 初期サンプルを持つことで精度と忠実度は大きく改善される. 直観的には, クエリ数が明らかに少ない場合は初期サンプルによる学習の効果が高い. 一方, クエリ回数が増えることで抽出モデルの学習が収束に近づくため, 初期サンプルによる利点が相対的に弱くなったと考えられる. なお, Arrhythmia のような次元数に対してインスタンス数が少ないデータに対しては, 初期サンプルが多いほど性能が改善しやすい傾向にある.

6.4 推論クラス数の影響

被害者モデルの推論結果におけるクラス数が TEMPEST に与える影響を評価する. 表形式の多くは 2 クラス分類であったため, 代表的な多クラス分類として, 画像形式ではあるが MNIST を用いて検討する. 5.1 節と同様に, PRADA [3] と比べて, 抽出モデルの精度と忠実度をそれぞれ比較する. その結果を表 4 に示す.

表 4 に示す通り, TEMPEST 攻撃は MNIST において PRADA よりも精度および忠実度が低くなった. これは画像に対しては統計情報が意味をなさないことが原因と考えれる. TEMPEST 攻撃によって生成されたデータは図 5 に示す. 図 5 を見る限り, 公開情報から生成されたデータは 0-9 いずれのラベルとも大きく隔たりがある. つまり, 画像に対して統計情報を用いてデータ生成したとしても, 学習に効果的なデータになるとは限らない.

関連して, MNIST における多クラス分類も MNIST の精度と忠実度が低下した要因と考えられる. 多クラス分類では, 各クラスに対する統計情報がより必要になると予想される. 実際に, 2 分類である Adult, Cancer, Diabetes と比較して, 16 分類である Arrhythmia は図 3 に示すように忠実度が低下しやすい傾向もみられている. このため, 多クラス分類のモデルに対しては TEMPEST の改善を考える必要がある.

図 5: TEMPEST 攻撃により生成された図

6.5 制約条件と潜在的な対策

TEMPEST の制約条件と潜在的な対策について議論する. TEMPEST における最大の制約条件は公開された統計情報が必要なことである. このため, 例えばユーザにとって要配慮情報であり非公開が多いデータを用いたモデルに対しては, そもそも攻撃の実行が容易でない. 関連して, MNIST の結果に示されるように, 統計情報が意味をなさない場合も攻撃が難しい.

同様に, 6.4 節に述べたように, 特徴量の次元数や推論クラス数が多い場合は, 統計情報が各特徴量やクラスにおいて十分に確保されている必要がある. MNIST や Arrhythmia では 2 クラスよりも多いクラス数であったため, 統計情報が不足しており, 精度が下がったと考えられる.

TEMPEST に対する潜在的な対策としては, 学習データに対する差分プライバシ [23] の適用が有効と考えられる. 既存のモデル抽出攻撃対策としては差分プライバシを被害者モデルの推論に適用する BDPL [24] があるが, TEMPEST の対策では学習に適用する点が異なる. 一般に, 差分プライバシは学習データの分布を曖昧にする効果があり, 例えば平均や分散, 最大値・最小値が本来のデータ分布と異なってくる. これにより, 公開された統計情報と被害者モデルの学習データに乖離を生じさせることで, アルゴリズム 1 のデータ生成が機能しないことが考えられる. 上述した対策の詳細な検討は, 今後の課題である.

7. 結論

本稿では表形式データにおいてデータの正規化処理を考慮した新たなモデル抽出攻撃 TEMPEST を示した. TEMPEST では公開された統計情報を前提とすることで, 攻撃者が学習データそのものを持つことがないデータフリー設定でのモデル抽出攻撃を現実的な仮定の下で行うことができる. 表形式のデータセットを用い, 特徴量の次元数およびインスタンス数に着目して検討を行ったところ, TEMPEST のデータ生成では平均と分散を統計情報として用いることが望ましく, また, 抽出モデルの正規化処理は可能な限り被害者モデルと同じものを用いることが望ましいことが分かった. また, インスタンス数が少ないときは, 初期サンプルを必要とするような従来の攻撃 [3] と同等程度の精度と忠実度が, データフリー設定というより強い条件においても満たせることも確認した.

今後の課題はより現実的な脅威の模索である. 今回の実験ではデータセットを等分割して行っており, 現実の公開情報を用いて実験するような現実の事例研究は行えていな

表 3: 初期サンプルを持つことによるによる TEMPEST の精度と忠実度の評価

	Adult		Cancer		Diabetes		Arrhythmia	
クエリ数	精度 (%)	忠実度 (%)	精度 (%)	忠実度 (%)	精度 (%)	忠実度 (%)	精度 (%)	忠実度 (%)
20(76)	74.43(+0.59)	81.41(+1.03)	94.25(+4.91)	93.74(+4.52)	72.51(-1.13)	86.42(+0.72)	38.23(+5.25)	40.47(+6.06)
100(380)	78.90(-0.82)	88.33(+0.72)	98.21(+0.33)	98.21(-0.32)	76.53(+0.63)	94.03(-0.76)	52.74(+5.59)	65.00(+9.88)
500(1900)	79.16(-0.03)	88.30(+0.02)	98.84(+0.01)	99.15(+0.06)	76.69(-0.13)	94.09(+0.31)	60.58(+1.15)	77.03(+6.67)

表 4: MNIST における被害者モデルの精度と忠実度

	PRADA 初期サンプルあり		PRADA 初期サンプルなし		TEMPEST	
クエリ数	精度	忠実度	精度	忠実度	精度	忠実度
300	33.49%	33.91%	10.32%	10.42%	10.02%	9.95%
1500	41.44%	72.80%	50.39%	13.23%	10.06%	9.95%
7500	78.48%	89.82%	52.79%	17.86%	16.62%	16.76%
37500	82.38%	95.83%	54.77%	24.88%	35.40%	36.20%

い. 事例調査を細心の倫理的配慮を踏まえて行うことで, モデル抽出攻撃の脅威を洗い出す狙いがある. また, もう一つの今後の課題は差分プライバシを学習データに適用することで, 被害者モデルの精度を下げることなく TEMPEST の防御が可能か検討することにある.

参考文献

[1] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R. and Le, Q. V.: XLNet: Generalized Autoregressive Pretraining for Language Understanding, *Proc. of NeurIPS 2019*, Curran Associates, Inc., pp. 5753–5763 (2019).

[2] Tramér, F., Zhang, F. and Juels, A.: Stealing Machine Learning Models via Prediction APIs, *Proc. of USENIX Security 2016*, USENIX Association, pp. 601–618 (2016).

[3] Juuti, M., Szyller, S., Marchal, S. and Asokan, N.: PRADA: Protecting against DNN Model Stealing Attacks, *Proc. of EuroS&P 2019*, IEEE, pp. 512–527 (2019).

[4] Orekondy, T., Schiele, B. and Fritz, M.: Knockoff Nets: Stealing Functionality of Black-Box Models, *Proc. of CVPR 2019*, IEEE, pp. 4954–4963 (2019).

[5] Truong, J.-B., Maini, P., Walls, R. J. and Papernot, N.: Data-Free Model Extraction, *Proc. of CVPR 2021*, pp. 4771–4780 (2021).

[6] Pal, S., Gupta, Y., Shukla, A., Kanade, A., Shevade, S. K. and Ganapathy, V.: ActiveThief: Model Extraction Using Active Learning and Unannotated Public Data, *Proc. of AAAI 2020*, Vol. 34, No. 01, AAAI, pp. 865–872 (2020).

[7] Kariyappa, S., Prakash, A. and Qureshi, M. K.: MAZE: Data-Free Model Stealing Attack Using Zeroth-Order Gradient Estimation, *Proc. of CVPR 2021*, pp. 13814–13823 (2021).

[8] Krishna, K., Tomar, G. S., Parikh, A. P., Papernot, N. and Iyyer, M.: Thieves on Sesame Street! Model Extraction of BERT-based APIs, *CoRR*, Vol. abs/1910.12366, pp. 1–18 (2019).

[9] Keskar, N. S., McCann, B., Xiong, C. and Socher, R.: The Thieves on Sesame Street are Polyglots - Extracting Multilingual Models from Monolingual APIs, *Proc. of EMNLP 2020*, ACL, pp. 6203–6207 (2020).

[10] Szyller, S., Duddu, V., Gröndahl, T. and Asokan, N.: Good Artists Copy, Great Artists Steal: Model Extraction Attacks Against Image Translation Generative Adversarial Networks, *CoRR*, Vol. abs/2104.12623 (2021).

[11] Arik, S. O. and Pfister, T.: TabNet: Attentive Interpretable Tabular Learning, *Proc. of AAAI 2021*, Vol. 35, No. 08, AAAI, pp. 6679–6687 (2021).

[12] Yoon, J., Zhang, Y., Jordon, J. and van der Schaar, M.: VIME: Extending the Success of Self- and Semi-supervised Learning to Tabular Domain, *Proc. of NeurIPS 2020*, Vol. 33, Curran Associates, Inc., pp. 11033–11043 (2020).

[13] Yu, H., Yang, K., Zhang, T., Tsai, Y.-Y., Ho, T.-Y. and Jin, Y.: Cloudleak: Large-scale deep learning models stealing through adversarial examples, *Proc. of NDSS 2020*, Internet Society, pp. 1–16 (2020).

[14] Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Berkay Celik, Z. and Swami, A.: Practical Black-Box Attacks Against Machine Learning, *Proc. of ASIACCS 2017*, ACM, pp. 506–519 (2017).

[15] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J. and Fergus, R.: Intriguing properties of neural networks, *Proc. of ICLR 2014* (2014).

[16] Hu, X., Liang, L., Li, S., Deng, L., Zuo, P., Ji, Y., Xie, X., Ding, Y., Liu, C., Sherwood, T. and Xie, Y.: DeepSniffer: A DNN Model Extraction Framework Based on Learning Architectural Hints, *Proc. of ASPLOS 2020*, ACM, p. 385–399 (2020).

[17] Reith, R. N., Schneider, T. and Tkachenko, O.: Efficiently Stealing Your Machine Learning Models, *Proc. of WPES 2019*, ACM, p. 198–210 (2019).

[18] Chen, K., Guo, S., Zhang, T., Xie, X. and Liu, Y.: Stealing Deep Reinforcement Learning Models for Fun and Profit, *Proc. of ASIACCS 2021*, ACM, p. 307–319 (2021).

[19] Chandrasekaran, V., Chaudhuri, K., Giacomelli, I., Jha, S. and Yan, S.: Exploring Connections Between Active Learning and Model Extraction, *Proc. of USENIX Security 2020*, USENIX Association, pp. 1309–1326 (2020).

[20] Wang, B. and Zhenqiang Gong, N.: Stealing Hyperparameters in Machine Learning, *Proc. of IEEE S&P 2018*, IEEE, pp. 36–52 (2018).

[21] Zhu, Y., Cheng, Y., Zhou, H. and Lu, Y.: Hermes Attack: Steal DNN Models with Lossless Inference Accuracy, *Proc. of USENIX Security 2021*, USENIX Association (2021).

[22] Hua, W., Zhang, Z. and Suh, G. E.: Reverse Engineering Convolutional Neural Networks Through Side-channel Information Leaks, *Proc. of DAC 2018*, IEEE, pp. 1–6 (2018).

[23] Dwork, C.: Differential Privacy, *Proc. of ICALP*, LNCS, Vol. 4052, Springer, pp. 1–12 (2006).

[24] Zheng, H., Ye, Q., Hu, H., Fang, C. and Shi, J.: BDPL: A Boundary Differentially Private Layer Against Machine Learning Model Extraction Attacks, *Proc. of ESORICS 2019*, LNCS, Vol. 11735, Springer, pp. 66–83 (2019).