トリガとモデル両方の識別不可能性を持つバックドア攻撃

岩花 一輝[1,a)]　矢内 直人[1]　藤原 融[1]

概要：機械学習のバックドア攻撃は攻撃対象のモデルに対し, トリガと呼ばれるある特定の入力においてのみ攻撃者の意図した不正な出力が得られるような, 隠れた領域を埋め込む攻撃である. 従来の攻撃手法ではトリガとモデルの挙動からバックドアの存在が検知される問題がある. 本稿ではトリガとモデルの挙動双方の観点においても, モデルにバックドアが存在するか識別不可能な新たなバックドア攻撃 IBDF (Indistinguishable Backdoor in Dual Form) を検討する. 大まかには, 通常の入力と見た目が一致するトリガ付き入力を生成するモデルと, そのトリガを入力する被害者モデルの両方において, 中間層の値も識別ができないように競合学習する. 実験を MNIST, GTSRB で行ったところ, IBDF は精度と攻撃成功率を損なうことなく, トリガとモデルの識別不可能性を満たすことを示した. 関連して, トリガとモデル双方の識別不可能性を満たすことで, トリガの復元やバックドアの除去がより困難になることも期待される.

キーワード：機械学習, バックドア攻撃, トリガの識別不可能性, モデルの識別不可能性

Indistinguishable Backdoor Attacks for Triggers and Models

Kazuki Iwahana[1,a)]　Naoto Yanai[1]　Toru Fujiwara[1]

Abstract: Backdoor Attacks on machine learning are attacks where an adversary obtains the expected output for a particular input called a trigger. In this paper, we present a novel backdoor attack whereby backdoors in a victim model are indistinguishable for both triggers and models. Loosely speaking, a generator generates triggers indistinguishable from benign inputs while a victim model takes the resulting triggers. They then learn adversarially so that values in their hidden layers are indistinguishable between benign inputs and triggers. We demonstrate that our attack provides indistinguishability for triggers and models as well as high accuracy and attack success rate on the MNIST and GTSRB datasets. We also indicate that recovering triggers and removing backdoors become more challenging by providing the indistinguishability.

Keywords: machine learning, backdoor attack, trigger indistinguishability, model indistinguishability

1. 序論

機械学習が顔認識や言語処理など様々な分野で利用が進んでいる一方, 機械学習モデルへのバックドア攻撃 [1] も高い関心を受けている. 大まかには, バックドア攻撃では攻撃者がトリガと呼ばれる入力が与えられたときだけ誤った推論を行うようにモデルの学習を行う. これにより, 実際にそのモデル（被害者モデルと呼ぶ）がサービスとして利用された際に, 攻撃者により何らかの脆弱性（バックドア）が埋め込まれることになる. 例えば, 顔認証で高い権限のユーザへのログイン [2] や, 機械翻訳を通じたフィッシング詐欺への誘導 [3] も指摘されている.

従来のバックドア攻撃では, 推論時にトリガそのものか被害者モデルの挙動を観測することで, モデルにバックドアが埋め込まれているか明らかにすることが可能となる. つまり, 被害者モデルのユーザの観点からは, バックドア攻撃の有無が識別できる. 一方, バックドア攻撃は攻撃者が自ら脆弱性を埋め込むという動機から, 攻撃者の観点において, バックドアの存在は可能な限り被害者モデルの管理者には気づかれないことが望ましい. このような背景において, バックドア攻撃の識別不可能性 [3], [4], [5], [6], [7], [8], [9]

[1] 大阪大学, Osaka University
[a)] k-iwahana@ist.osaka-u.ac.jp

が近年注目されている．著者の知る限り, バックドア攻撃の識別不可能性は二つの観点がある: トリガとなる入力がそれ以外の通常の入力と区別できないトリガの識別不可能性 [3]; トリガとなる入力を与えられた際の被害者モデルの中間層の値が, 通常の入力におけるモデルのものと区別できないモデルの識別不可能性 [8] である．

本稿では, バックドア攻撃に関する上記の二つの識別不可能性を同時に満たす攻撃が可能か明らかにする．実は, バックドア攻撃の攻撃成功率を維持したまま, 上記の二つの識別不可能性を同時に満たせるかは非自明である．直観として, トリガの識別不可能性の下ではトリガと通常の入力が被害者モデルにとっても判別しがたく, 攻撃者の観点からはバックドアの埋め込みが難しくなる．一方, バックドアが確実に埋め込めるように, トリガが判別しやすくなるような学習をした場合, 一個一個の入力に対するモデルの中間層の値が変化しやすくなるため, モデルの識別不可能性が達成しにくくなる．

本稿では, 上述した問いに向けた新たな攻撃 IBDF (Invisible Backdoors in Dual Forms) を示す．IBDF では被害者モデルへのバックドアの埋め込みに際し, トリガを生成するモデルを用意する．このとき, トリガの生成モデルと被害者モデル双方でモデルの識別不可能性を満たすように競合学習させる．本稿の主な知見は, 競合学習において生成モデルと被害者モデル両方で中間層の値を考慮することで, モデルの識別不可能性とトリガの識別不可能性を両立させるバックドア攻撃に成功することが分かったことである．生成モデルと被害者モデル両方での中間層の値の考慮が重要であることを実験的にも示している．関連して, トリガの識別不可能性とモデルの識別不可能性を両立させることで, 攻撃者の観点からはより強いバックドア攻撃を実現できる可能性も示している．具体的に, モデルからのトリガの正確な復元がより困難になること, また, 既存方式 [1], [8] と比べて除去されにくいようなバックドア攻撃が実現できることを示唆している（詳細は 6 節を参照されたい．）．

2. 関連研究

本節では関連研究として識別不可能性を持つバックドア攻撃と従来のバックドア攻撃の対策について述べる．

2.1 識別不可能性を持つバックドア攻撃

識別不可能性を持つバックドア攻撃は, トリガの識別不可能性が主に議論されている [2], [3], [4], [5], [6], [7]．モデルに与える学習データに摂動を加えることで, 人間の感覚として違和感がないようなトリガ付きデータを生成する手法である．近年では自然言語処理 [3] や顔認証 [2] などアプリケーションを対象とした攻撃も検討されている．摂動以外の手法としては画像に特化した方式としてステガノグラフィを応用した方法 [4] もある．

一方, 著者の知る限り, モデルの識別不可能性の研究は数が少ない [8], [9]．代表的な手法は競合学習を用いる Adversarial Embedding [8] である．最新の成果である Blind Code Poisoning [9] は学習処理に関与せずバックドアを埋め込むが, モデルの識別不可能性は付加的に検討されている．つまり, モデルの識別不可能性の議論は, Adversarial Embedding が代表的といえる．このため, 本稿では Adversarial Embedding と比較することで, トリガの識別不可能性とモデルの識別不可能性の両立を検討する．

2.2 バックドア攻撃への対策

バックドア攻撃の対策はトリガの検知 [10], [11] とモデルの異常検知 [12], [13] に分けられる．トリガの検知はバックドアを発火させるトリガそのものを検知する手法であり, 近年の手法 [10] では GAN を用いることで, バックドアを埋め込まれたモデルからトリガを復元する．次に, モデルの異常検知はモデルの中間層の値を見ることで, 通常の入力とトリガ付き入力におけるモデルの挙動の違いに着目する [12], [13]．文献 [14] によるとモデルの異常検知は, バックドアの埋め込まれたモデルは常にバックドアに関する特徴量に注目する性質を利用している．上述したトリガの検知とモデルの異常検知はバックドアを埋め込まれたモデルが公開利用されたあとでも適用できる汎用的な枠組みといえる．このため, 本稿では IBDF の評価として, それぞれの代表的な手法 GangSweep [10] と Fine-Pruning [13] についても考察する．

3. 問題設定

本節では本稿の問題設定としてバックドア攻撃, および, トリガの識別不可能性とモデルの識別不可能性を定式化する．次に, バックドア攻撃における評価指標を述べる．

3.1 機械学習とバックドア攻撃

データの入力空間を X, データが持つラベルの空間を Y とする．機械学習では, 学習データ $(x, y) \in X \times Y$ から, 未知のデータ $x' \in X$ に対し正解ラベル $y' \in Y$ を計算するようなモデル $M : X \to Y$ を得ることを目的とする．

この機械学習におけるバックドア攻撃は, 特定の入力に対してのみ被害者モデル M の推論結果が異なるような学習を行うことである．具体的に攻撃者は, ソースクラス s に属する入力 x_s に対し, トリガ Tr を埋め込むことで $x^* \in X^*$ を得る．このとき攻撃者が分類したいターゲットクラス t のラベル, $y_t = M(x^*)$ となるように学習させることで, バックドア付きモデル M^* を得る．

このとき, 以下の両方の条件を満たすなら, 被害者モデル $M : X \to Y$ はバックドアを埋め込まれたという：(1) 任意の通常の入力とラベル $(x, y) \in X \backslash X^* \times Y$ において, $M(x) = M^*(x)$ となる; (2) 任意のトリガ付き入力とラベ

ル $(x^*, y_t) \in X^* \times Y$ において, $M^*(x^*) = y_t$ となる.

また, 本稿では攻撃者がモデル M の学習を被害者から依頼される際にバックドアが埋め込まれた被害者モデル M^* を構築する外部委託設定 [8] を検討する. この設定では攻撃者はモデル M^* の学習アルゴリズムを, 本来のモデル M のものとすり替えることができる. 一般的な機械学習の実装では学習の結果としてモデルのパラメータのみが出力されることから, モデル M の学習を依頼したユーザの観点からは, そのパラメータがどのように得られたか知ることができない. このため, 外部委託設定におけるアルゴリズムのすり替えは現実的といえる.

3.2 バックドア攻撃の評価指標

本稿ではバックドア攻撃の評価指標として攻撃の性能, トリガの識別不可能性, モデルの識別不可能性を議論する. 以下にそれぞれ詳細を述べる.

3.2.1 攻撃の性能

バックドア攻撃の性能を表す指標であり, 識別不可能性の有無にかかわらず議論される. 被害者モデル M^* における推論精度と攻撃性能の二点からなる.

推論精度: 任意の通常の入力とラベル $(x, y) \in X \backslash X^* \times Y$ において, $y = M^*(x)$ となる割合として表す. 攻撃者はモデルの価値を落とさないよう, 推論精度が維持できるように被害者モデル M^* にバックドアを埋め込む必要がある.

攻撃成功率: トリガ付き入力 $x^* \in X^*$ において, 推論結果 $M^*(x^*)$ がターゲットラベル y_t に等しくなる割合として表す. 攻撃者は, 埋め込んだバックドアを確実に利用できることを表す指標であり, M^* がトリガを正確に認識できているほど攻撃成功率は高くなる.

3.2.2 トリガの識別不可能性

トリガの識別不可能性とは, トリガ付き入力 $x^* \in X^*$ と通常の入力 $x \in X \backslash X^*$ が識別できないことを意味する. この評価指標は機械学習モデルそのものではなく, モデルへの入力が持つ性質として評価される. 本稿ではトリガの識別不可能性の評価指標として, 機械学習モデルを通じて二つの画像の類似度を数値化する LPIPS (Learned Perceptual Image Patch Similarity) [15] を用いる. これは画像分類に限定した議論であり, 任意のドメインへの応用は難しい. 一方, LPIPS の利用は以下の利点もある. まず, LPIPS は既存のトリガの識別不可能性 [4] で利用されている指標であり, 既存研究と性能を公平に比較できる. また, 画像の類似度を数値化することで, モデルの識別不可能性による影響含めたトリガの識別不可能性を定量的に評価できるようになる. 計算方法は文献 [15] を参照されたい.

3.2.3 モデルの識別不可能性

モデルの識別不可能性は, トリガ付き入力と通常の入力それぞれにおける中間層の分布に違いがみられないことを意味する. 本稿ではモデルの識別不可能性の指標として, バックドアに対してモデルが異常検知される割合を用いる. 2.2 節で述べた通り, トリガ付き入力の中間層と通常の入力の中間層の値が離れているほど, モデルからバックドアの検知が容易になるためである.

具体的に, モデルの識別不可能性の指標として式 (1) を導入する. 式 (1) において Z の値が 0 に近いほど, モデルの識別不可能性は強い. 直観的には, バックドアの検知が高々 5 割でしかされない場合, ランダムな検知とみなせるためである. なお, $correct\_score$ の算出に何らかの異常検知手法を用いる.

$$Z = \frac{|correct\_score - 0.5|}{0.5} \tag{1}$$

既存研究 [8] ではモデルの識別不可能性は厳密に定義しておらず, 上式は本稿による定義である. これはモデルの異常検知に依存した定義ではあるが, 以下の利点がある. まず, 既存のモデルの識別不可能性 [8] では既存の異常検知手法 [12] の結果のみ示している一方, 本稿では上式を通じてより一般的な異常検知含めた定量評価を行えるようになる. とくに, 性能を数値化することで, トリガの識別不可能性による影響含めたモデルの識別不可能性に関する性質を定量的に評価できる点が大きい.

3.3 本稿の主たる問い

本稿の問いは, 精度と攻撃成功確率を下げることなく, 前述したトリガとモデル二つの識別不可能性を両立できるか明らかにすることである. 大まかには, それぞれの識別不可能性を満たす手法 [4], [8] を個々に用いたとしても, 両方の識別不可能性が同時に達成されるとは限らない. 例えば, トリガの識別不可能性 [4] を満たす方式では, モデル M^* の中間層がトリガ付きの入力 $x^* \in X^*$ か通常の入力 $x \in X$ か正確に推論ができるように, M^* のパラメータを慎重に更新する必要がある. 直観として, トリガの識別不可能性を考慮したまま攻撃成功率と推論精度を維持できるよう, モデル M の重み W を最適化する. このとき, M^* のパラメータ W は通常のモデル M はもちろん, 従来のバックドア攻撃 [1] のものとも大きく異なることがありえる. このような状況では, 中間層の状態の分類 [12] や不要なパラメータ消去によるバックドアの除去手法 [13] により, 対策がむしろ容易になる. つまり, (1) 式の Z の値が大きくなることで, モデルの識別不可能性を満たさなくなる.

4. IBDF: 識別不可能性を持つバックドア

本節ではトリガとモデル両方の識別不可能性を持つバックドア攻撃 IBDF を示す. まず攻撃の全体像を述べたのち, バックドアの学習アルゴリズムについて述べる.

4.1 攻撃の全体像

IBDF の着想は被害者モデルの学習を行う際に, トリガの生成モデルと被害者モデル両方において, トリガと通常

アルゴリズム 1 バックドア攻撃

入力: 通常の学習データ $(x, y) \subset D_{train}$, 被害者モデル M, M の中間層 H, 生成モデル G, 交差エントロピー誤差 L_{ce}, 学習回数 $epochs$, バッチサイズ B, ソースクラス y_s, ターゲットクラス y_t, ハイパーパラメータ α, β.

出力: 被害者モデル M, 生成モデル G.

procedure EMBEDDING TRIGGERS(x_s, G)
  $z \leftarrow Gaussian(0, 1)$
  $Tr = G(z)$ //トリガ生成
  $x_p = Clip(x_s + Tr)$
  return x_p, Tr
end procedure
for $range(epochs)$ **do**
  for $range(0, |D_{train}|/B)$ **do**
    $x_B, y_B \leftarrow (x, y)$ を B で分割
    x_s, x_t ($\leftarrow x \in y_s$ と $x \in y_t$ から B 個ランダム抽出)
    $x_p, Tr \leftarrow$ EMBEDDING TRIGGERS(x_s, G)
    $L_G = ||Tr||_2 + \alpha||H(x_p) - H(x_t)||_1$
    $L_M = L_{ce}(x_B, y_B) + \beta||H(x_p) - H(x_t)||_1$
    L_G, L_M から G, M を更新
  end for
end for

の入力に関する中間層の距離を狭めるように競合学習させることである. これにより, 精度と攻撃成功率を下げることなく, トリガとモデル両方の識別不可能性を達成する.

具体的に, まずトリガの生成モデルにおいて, 通常の入力とトリガ付き入力の距離が小さくなるようにトリガを生成する. これにより, トリガの識別不可能性が満たされる. このとき, 生成したトリガに対してモデルの中間層が通常の入力のものと距離が近くなるように生成モデルを学習させることで, モデルの識別不可能性も強まるようなトリガを生成できるようになる.

一方, 生成モデルだけで中間層の距離を考慮した場合, 被害者モデルの方で中間層の値が異なってしまい, バックドアの埋め込みに失敗する可能性もある. 実際に生成モデル側だけで競合学習をした際, 攻撃成功率が下がる状況が確認された. このため, 被害者モデルの学習においてもトリガ付き入力と通常の入力における中間層の距離を考慮することで, モデルの識別不可能性を満たしたまま攻撃成功率を改善させる. 以上の観点を踏まえ, トリガとモデル両方の識別不可能性を持つ攻撃アルゴリズムを構成する.

4.2 学習アルゴリズム

IBDF の処理をアルゴリズム 1 に示す. アルゴリズムではトリガの生成モデル G と被害者モデル M を 13 行目と 14 行目でそれぞれ同時に更新させるような競合学習を行っている. 以下にそれぞれの動作を述べる.

まず生成モデル G では, 標準正規分布に従う潜在変数 z を入力とし, トリガを出力する. このとき, 生成モデルはトリガの識別不可能性と, 被害者モデルに関してモデルの識別不可能性を強めるトリガを生成するように学習する. 具体的に, まずトリガの識別不可能性を満たすために, 13 行目の処理でトリガ Tr の L_2 ノルムが小さくなるよう学習する. 直観的に, トリガの識別不可能性は, 通常の入力に対して生成されるトリガの大きさが小さいことと等価であるため, トリガの L_2 ノルムの値で評価している. また, 並行してモデルの識別不可能性を満たすため, 13 行目の処理でソースクラスのトリガ付き入力に関する中間層の値 $H(x_t)$ と, ターゲットクラスの通常の入力に関する中間層の値 $H(x_p)$ が一致するように, L_1 ノルムを計算する.

一方, 被害者モデル M では通常のデータを正しく分類するよう, また, モデルの識別不可能性を満たすように, 14 行目で学習を行う. まず交差エントロピー誤差関数 L_{ce} を用いて, 通常のデータ x_B と正しいクラス y_B の誤差を計算する. また, 被害者モデル M の学習においても, ソースクラスのトリガ付き入力に関する中間層の値 $H(x_t)$ が, ターゲットクラスの通常の入力に関する中間層の値 $H(x_p)$ と近づくように学習させる. これにより, トリガの識別不可能性とモデルの識別不可能性を両立させる.

なお, 13 行目と 14 行目のハイパーパラメータは, 学習の収束を促すために導入した. 予備実験によりハイパーパラメータを持たない状態で学習したところ, 学習が収束しなかった. これらのハイパーパラメータを導入・調整することで, バックドアを効率的に埋め込めるようになる.

5. 実験

本節では IBDF を実験評価する. 実験の目的と設定をまず述べ, 次に結果を示す.

5.1 実験目的

本実験では 3.2 節の評価指標で IBDF を評価する. まず 3.2 節の評価指標において, IBDF を既存研究 [1], [4], [8] と比較することで, トリガの識別不可能性とモデルの識別不可能性による精度と攻撃成功率への影響を評価する. これにより, 本稿の着想である生成モデルと被害者モデル両方で中間層の値を考慮する競合学習の有効性を確認する. 関連して, IBDF をモデルの識別不可能性を持つ既存手法 [8] と比較することで, トリガの識別不可能性によるモデルの識別不可能性への影響も明らかにする. なお, 既存研究 [8] ではモデルの識別不可能性を明確に定義しておらず, 本稿の指標とは厳密には異なる可能性がある. 一方, モデルの識別不可能性のみを扱う唯一の研究 [8] と本稿の指標で比較することにより, トリガの識別不可能性によるモデルの識別不可能性への影響のみを議論する狙いがある.

5.2 実験設定

本実験の設定について述べる. ライブラリ Tensorflow を用いて実装した. また, 実験環境として OS が Ubuntu 18.04, CPU は Intel(R) Xeon(R) Gold 6140 CPU @ 2.30GH で,

表 1: データセットに関する情報

データセット	ラベル数 (ソース, ターゲット)	入力サイズ	学習, 検証データ数
MNIST	10(7,9)	$28 \times 28 \times 1$	60000,10000
GTSRB	43(7,20)	$48 \times 48 \times 3$	39209,12630

GPU は NVIDIA GV100GL を用いた.

5.2.1 データセットとアーキテクチャ

本実験で用いたデータセットとアーキテクチャの設定について述べる. 詳細は表 1 に示す.

MNIST: 0 から 9 までの数字の白黒手書き画像データセットである. 本稿のアーキテクチャはチャネル数 32 の畳み込み層, チャネル数 64 の畳み込み層, 2×2 のマックスプーリング, ニューロン数 128 の全結合層, ニューロン数 10 の全結合層から成る. また, IBDF におけるハイパーパラメータは $\alpha = 0.1, \beta = 0.001$ で, $epochs = 300$ とする.

GTSRB: 速度制限など 43 種類の道路標識の画像データセットである. 本稿のアーキテクチャは既存研究 [16] と同じ構成を用い, 2 つの畳み込み層とマックスプーリングの組み合わせを 3 層重ね, 2 つの全結合層からなる. また, IBDF におけるハイパーパラメータは $\alpha = 0.01, \beta = 1.0^{-6}$ で, 55 エポックから $\alpha = 0.001, \beta = 1.0^{-5}$ に設定した. この理由は 5.3.1 節に記載する.

なお, 生成モデルのアーキテクチャはチャネル数を除いて MNIST と GTSRB で同じものを用いる. u 個のチャネル数の逆畳み込み演算, バッチ正規化, learkyReLu を CD_u とすると, $CD_{128} - CD_{64} - CD_1$ の構成となる.

5.2.2 ベースライン

評価用のベースラインとして, 前節に述べたアーキテクチャでバックドアを埋め込まれていない通常のモデルを用意した. 推論精度は MNIST で 99.30%, GTSRB で 96.97% であり, これらの数値をベースラインとする.

また, IBDF との比較手法として, 従来のバックドア攻撃 BadNets [1] と既存のモデルの識別不可能性を持つ手法 Adversarial Embedding [8] の公開ライブラリ[*1]を利用した. なお, ライブラリでは BadNets と Adversarial Embedding のトリガの生成方法は共通の手法を用いている.

一方, トリガの識別不可能性については今回の環境に適した公開ライブラリが著者の調査範囲では見つけられなかった. このため, IBDF のアルゴリズムを応用し, トリガの識別不可能性のみの方式を構成した. 具体的には, アルゴリズム 1 に示した 13,14 行目の $||H(x_p) - H(x_t)||_1$ の項を, トリガ付き画像がターゲットクラス y_t に分類される $L_{ce}(x_p, x_t)$ の項に置き換えることで, トリガの識別不可能性のみを考慮した学習を行った. この手法を Naive Invisible と呼称する. 併せて, 参考数値として既存のトリガの識別不可能性の文献 [4] に記載された数値とも比較する.

[*1] https://github.com/Trusted-AI/adversarial-robustness-toolbox

表 2: データセットごとの精度と攻撃成功率

	MNIST		GTSRB	
	精度	攻撃成功率	精度	攻撃成功率
通常	99.30	-	96.97	-
BadNets [1]	99.25	100	97	99.33
AE [8]	98.92	100	96.44	99.11
既存研究 [4]	99	99	97.29	98.44
Naive	99.11	100	96.25	98.89
IBDF	**98.78**	**91.15**	**97.28**	**99.78**

5.2.3 識別不可能性

トリガの識別不可能性とモデルの識別不可能性の評価指標の算出には, それぞれ以下の要素技術を用いた.

トリガ: LPIPS の計算に利用するモデルとして, ImageNet で学習した VGG モデルを用いる[*2]. 既存研究 [4], [15] によると ImageNet で訓練された VGG モデルは人間の視覚を模倣している. つまり, トリガの識別不可能性を人間に近い感覚で定量的に評価できる.

モデル: 式 (1) の $correct\_score$ の計算として, 既存手法の Activation Clustering [12] を用いる. 具体的に, トリガ付き入力と通常の入力それぞれに関する最終特徴抽出層の値を k-means 法でクラスタリングする. その際のバックドアの分類成功率を $correct\_score$ とする. また, クラスタリングの際に, トリガ付き入力と通常の入力の比率により $correct\_score$ の変動が考えられる. このため, 通常の入力の数を 1000 で固定する一方, トリガ付き入力の数を 200, 400, 600, 800, 1000 と変化させ, それぞれ計測する.

5.3 実験結果

5.3.1 精度と攻撃成功率

精度と攻撃成功率の結果を表 2 に示す. ここで, AE は Adversarial Embedding, Naive は Naive Invisible をそれぞれ表す. まず, IBDF の精度は MNIST で 98.78%, GTSRB で 97.28%であった. これは通常のモデルの精度と比較して, MNIST で約 0.5%の劣化が見られた. 一方, GTSRB では 97.28%と通常のモデルの精度と比較しても 0.3%の向上を確認した. 次に, IBDF の攻撃成功率は MNIST で 91.15%, GTSRB で 99.78%であった. とくに MNIST において, 既存研究 [1], [4], [8] と比較して大きく低下した.

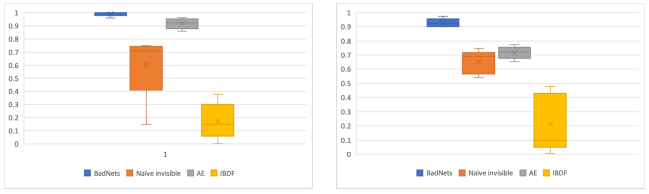
攻撃成功率の低下の詳細としてエポックごとの精度と攻撃成功率の変化を調べたところ, 図 2 に示すように徐々に低下していた. また, GTSRB では 70 エポックまでは安定した高い精度と攻撃成功率が見られたが, $||Tr||_2$ が大きな値で収束してしまっていた. ハイパーパラメータ α, β を変更することで $||Tr||_2$ の値を下げることはできたが, これにより攻撃成功率が不安定になったと考えられる.

以上を踏まえると, MNIST, GTSRB ともに $||Tr||_2$ の値

[*2] LPIPS は機械学習モデルによる数値表現であり, このモデルは被害者モデル, および, 生成モデルとは別に用意される.

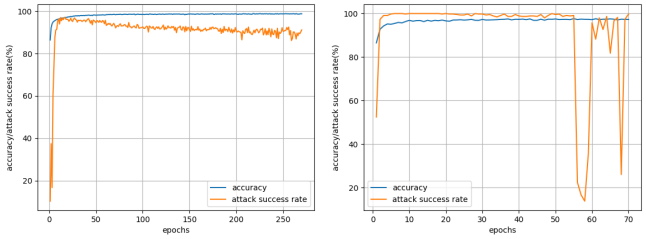
表 3: トリガの識別不可能性の評価

	Origin	BadNets [1] AE [8]	既存研究 [4]	Naive invisible	IBDF
MNIST			-		
LPIPS	0	0.043	2.7e-5	0.001	0.019
GTSRB			-		
LPIPS	0	0.021	1.2e-4	0.001	0.014



(a) MNIST (b) GTSRB

図 1: モデルの識別不可能性



(a) MNIST (b) GTSRB

図 2: エポックごとの精度と攻撃成功率

を下げることで, 攻撃成功率の低下が見られた. これは, トリガの変化によるニューロンの発火パターンの変化の影響を受けたためと考えられる. すなわち, IBDF において攻撃成功率はトリガの識別不可能性の影響を大きく受ける.

5.3.2 トリガの識別不可能性

トリガの識別不可能性の実験結果を表 3 に示す. なお, 既存研究 [4] の LPIPS の値は文献参照による数値であり, 画像は表記していない. BadNets と Adversarial Embedding と比較して LPIPS は下がっているため, IBDF によるトリガの識別不可能性は強くなっている. しかし, 既存研究 [4] や Naive Invisible と比較して, LPIPS の値は高くなった. これはモデルの識別不可能性を強めた結果, トリガの識別不可能性とのトレードオフが発生したとみなせる.

5.3.3 モデルの識別不可能性

モデルの識別不可能性の実験結果を図 1 に示す. (1) 式の Z の平均値は, MNIST において 0.174, GTSRB において 0.211 である. BadNets [1], Adversarial Embedding [8], Naive Invisible と比較しても値は低いことから, IBDF のモデルの識別不可能性は強いと言える. その理由を, Naive Invisible と Adversarial Embedding との比較から述べる.

まず, Naive Invisible では LPIPS の数値が低いため, トリガ付き入力の分布は BadNets よりも通常の入力に近い. このことに起因して, Naive Invisible における中間層の値の分布も通常の入力に近いところにあるとみなせる. 一方, 実際には中間層の値の分布もある程度の距離がないと, バックドアが正確に認識できない. つまり, Naive Invisible では攻撃成功率を高くするために中間層の分布に距離が生じるような学習が行われた. そのため, Naive Invisible のモデルの識別不可能性は弱くなった.

次に Adversarial Embedding では, Naive Invisible に近いモデルの識別不可能性の強さであった. これは, Adversarial Embedding の構成ではトリガ付き入力における中間層を元の画像に近い中間層に近づけることから, トリガの識別不可能性に近い効果が得られたためと考えられる. なお, Adversarial Embedding については評価指標を変えた際に異なる結果が得られる可能性はある.

6. 考察

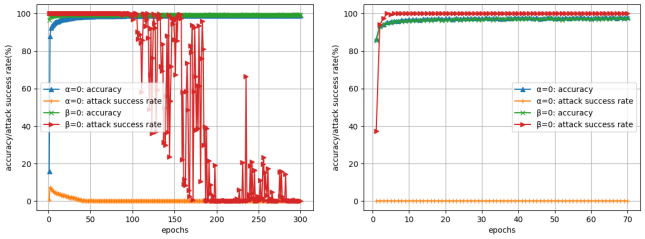
IBDF の性質としてアルゴリズム 1 の有効性, トリガの識別不可能性の応用, モデルの識別不可能性の応用を考察する. IBDF の制約条件と潜在的な対策も議論する.

6.1 アルゴリズムの有効性

IBDF のアルゴリズムの有効性について, 生成モデルと被害者モデルの損失関数 L_M, L_G に用いられている中間層に関する項 $||H(x_p) - H(x_t)||_1$ の影響から議論する. 具体的に, ハイパーパラメータ α, β をそれぞれ 0 にした場合の精度と攻撃成功率の変化を計測する. このとき, $\alpha = 0$ の場合は生成モデルの学習で, $\beta = 0$ の場合は被害者モデルで, それぞれモデルの識別不可能性を考慮しない学習を行っていることを意味する. それぞれについて計測したところ, 図 3 の結果が得られた.

図 3 によると, MNIST では $\alpha = 0, \beta = 0$ いずれの場合においても, 攻撃成功率が 0%まで収束しており, バックドアが埋め込めていないことが分かる. 一方, GTSRB では $\alpha = 0$ の場合は攻撃成功率が 0%に収束しているが, $\beta = 0$ の場合は攻撃成功率が 100%に収束している. $\beta = 0$ で攻撃成功率が高い理由は, 生成モデルによるトリガ単体で被害者モデルの中間層の値が各クラス間で互いに近くなるよう, 学習が進むためである. このときに生じる作用としては, 生成モデルが出力する画像がターゲットクラスに指定した画像そのものとなり, 結果としてトリガの識別不可能性が弱くなる. 実際に生成されたトリガは 9 に近い画像であり, その際の LPIPS は 0.293 だった. すなわち, トリガの識別不可能性を満たさない.

以上から, 中間層に関する項 $||H(x_p) - H(x_t)||_1$ を生成モデル, 被害者モデル両方に埋め込むことが, モデルの識別不可能性とトリガの識別不可能性の両立には重要である.



(a) MNIST (b) GTSRB

図 3: ハイパーパラメータに関する精度と攻撃成功率

表 4: Gangsweep [10] により, 復元された画像

	BadNets [1]	AE [8]	Naive	IBDF
トリガ				
クラス 9				
クラス 0				
クラス 1				

6.2 トリガの識別不可能性の応用

IBDF でトリガの識別不可能性を達成したことによる既存のトリガ検知手法への耐性を議論する. 具体的に, 既存のトリガ検知手法 Gangsweep [10] により, 被害者モデルからトリガが復元されるか検証する. 検証では, MNIST データセットにおいて攻撃者が目標とするターゲットクラスを 9 としてバックドアを埋め込んだときに, 9 および通常のクラスとして 0, 1 に対し GangSweep をそれぞれ実行する. その結果を表 4 に示す.

まず, BadNets [1], Adversarial Embedding [8] は単純なトリガのため, クラス 9 においてトリガに近い画像が復元された. また, Naive Invisible と IBDF ではトリガとしてランダムノイズのような画像が出力された. Naive Invisible ではクラス 9 と通常の入力 0, 1 の結果を比較したとき, 各画像が大きく異なっており, クラス 9 の出力がトリガに類似しているとみなせる. 一方, IBDF ではいずれのクラスにおいても同様に, ノイズのような画像が出力された. 直観的に, IBDF はいずれのクラスにおいても復元された情報が類似しており, トリガでどのクラスに推論されるかは判断が難しい. すなわち, トリガの識別不可能性とモデルの識別不可能性両方を達成することで, トリガの正確な復元を回避できる可能性がある.

6.3 モデルの識別不可能性の応用

IBDF で Activation Clustering [12] を基にモデルの識別不可能性を満たした際に, 他のモデルの異常検知手法に耐性があるか議論する. 具体的に, 不要なニューロンを枝刈りすることで攻撃成功率のみ低下させる Fine-Pruning [13] により, モデルの精度と攻撃成功率の変化を確認する.

MNIST と GTSRB において Fine-Pruning を実行した結果を図 4, 図 5 に示す. ここで, 横軸はニューロンの枝刈り率を示す. IBDF は MNIST, GTSRB ともに枝刈り率が 80%まで精度と攻撃成功率を維持できており, 90%以降は精度と攻撃成功率で 10%程の差があるもののどちらも低下した. これらの精度と攻撃成功率の相関は, IBDF では攻撃成功率のみ下げることが難しいことを意味する.

一方, 図 4 によると BadNets, Adversarial Embedding, Naive Invisible は 30%のニューロンを枝刈りすることで, 攻撃成功率のみ大幅に低下した. 同様に, 図 5 によると, Adversarial Embedding と Naive Invisible は枝刈り率が 80%になると攻撃成功率が大幅に低下した. 一般にバックドアは少量のパラメータ変更で除去可能であり [17], 上述した結果は Adversarial Embedding と Naive Invisible でバックドアが除去されたことを意味する.

上述した結果を踏まえると, IBDF ではトリガの識別不可能性とモデルの識別不可能性を両立させたことで, 従来の識別不可能性を持つ手法 [4], [8] と比べて, バックドアが除去されにくいように強く埋め込めたことが示唆される.

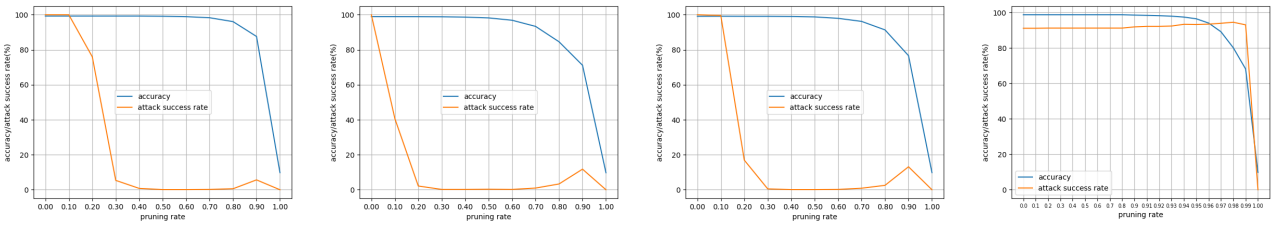
6.4 制約条件と潜在的な対策

IBDF の制約条件と潜在的な対策について議論する. まず, 学習アルゴリズムの制約として図 2 に示すように攻撃成功率が不安定になる. この対策として, アルゴリズム 1 の 13, 14 行目における $H(x_s)$ と $H(x_t)$ の誤差の評価方法の変更があげられる. 本稿では L_2 ノルムで評価しているが, L_1, L_∞ などで評価することで, 攻撃成功率が安定する可能性がある. この検討は, 今後の課題である.

また, 6.2 節で示した通り, IBDF では, トリガの検知によりクラス間で類似した画像が復元される. そのため, バックドア攻撃を受けた被害者の観点からは, トリガ復元手法により復元された画像の類似性を考慮することで, バックドア攻撃に対して対策できる可能性がある. この対策の検討は, 今後の課題である.

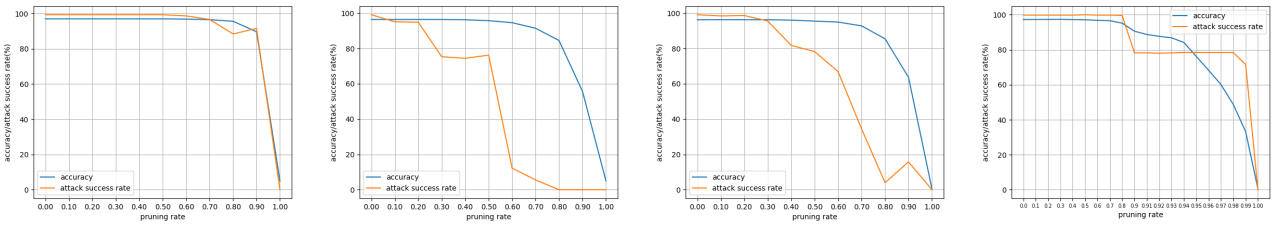
7. 結論

本稿ではトリガの識別不可能性とモデルの識別不可能性を両立させたバックドア攻撃 IBDF を提案した. IBDF では, トリガの生成モデルと被害者モデルの両方でトリガ付き入力と通常の入力における中間層の距離を考慮する競合学習を行うことで, 精度と攻撃成功率を損なうことなく, トリガの識別不可能性とモデルの識別不可能性を満たせることを示した. また, トリガの識別不可能性とモデルの識別不可能性を両立させることで, 攻撃者の観点からはトリガの正確な復元を防げること, また, 従来技術よりも除去されにくいようにバックドアを埋め込むことが期待できる. 今



(a) BadNets (b) Adversarial Embedding (c) Naive Invisible (d) IBDF

図 4: Pruning Defense(MNIST)



(a) BadNets (b) Adversarial Embedding (c) Naive Invisible (d) IBDF

図 5: Pruning Defense(GTSRB)

後の課題は, $L1, L_\infty$ を利用して IBDF の攻撃成功率を安定させること, また, モデルから復元されるトリガの類似度を利用した対策の提案である.

謝辞 本研究の一部は, 戦略的イノベーション創造プログラム（SIP）第 2 期「IoT 社会に対応したサイバー・フィジカル・セキュリティ」（管理法人：NEDO）にて実施されている.

参考文献

[1] Gu, T., Liu, K., Dolan-Gavitt, B. and Garg, S.: BadNets: Evaluating Backdooring Attacks on Deep Neural Networks, *IEEE Access*, Vol. 7, pp. 47230–47244 (2019).

[2] Xue, M., He, C., Wang, J. and Liu, W.: Backdoors hidden in facial features: a novel invisible backdoor attack against face recognition systems, *Peer-to-Peer Networking and Applications*, Vol. 14, No. 3, pp. 1458–1474 (2021).

[3] Li, S., Liu, H., Dong, T., Zhao, B. Z. H., Xue, M., Zhu, H. and Lu, J.: Hidden Backdoors in Human-Centric Language Models, *CoRR*, Vol. abs/2105.00164 (2021).

[4] Li, S., Xue, M., Zhao, B., Zhu, H. and Zhang, X.: Invisible Backdoor Attacks on Deep Neural Networks via Steganography and Regularization, *IEEE Transactions on Dependable and Secure Computing*, pp. 1–1 (2020).

[5] Zhong, H., Liao, C., Squicciarini, A. C., Zhu, S. and Miller, D.: Backdoor Embedding in Convolutional Neural Network Models via Invisible Perturbation, *Proc. of CODASPY 2020*, ACM, pp. 97–108 (2020).

[6] Ning, R., Li, J., Xin, C. and Wu, H.: Invisible Poison: A Blackbox Clean Label Backdoor Attack to Deep Neural Networks, *Proc. of INFOCOM 2021*, IEEE (2021).

[7] Saha, A., Subramanya, A. and Pirsiavash, H.: Hidden Trigger Backdoor Attacks, *Proc. of AAAI 2020*, Vol. 34, No. 07, AAAI, pp. 11957–11965 (2020).

[8] Tan, T. J. L. and Shokri, R.: Bypassing Backdoor Detection Algorithms in Deep Learning, *Proc. of EuroS&P*, IEEE, pp. 175–183 (2020).

[9] Bagdasaryan, E. and Shmatikov, V.: Blind Backdoors in Deep Learning Models, *Proc. of USENIX Security 2021*, USENIX Association (2021).

[10] Zhu, L., Ning, R., Wang, C., Xin, C. and Wu, H.: GangSweep: Sweep out Neural Backdoors by GAN, *Proc. of MM 2020*, ACM, pp. 3173–3181 (2020).

[11] Wang, B., Yao, Y., Shan, S., Li, H., Viswanath, B., Zheng, H. and Zhao, B. Y.: Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks, *IEEE S&P 2019*, IEEE, pp. 707–723 (2019).

[12] Chen, B., Carvalho, W., Baracaldo, N., Ludwig, H., Edwards, B., Lee, T., Molloy, I. M. and Srivastava, B.: Detecting Backdoor Attacks on Deep Neural Networks by Activation Clustering, *Proc. of SafeAI 2019* (2019).

[13] Liu, K., Dolan-Gavitt, B. and Garg, S.: Fine-Pruning: Defending Against Backdooring Attacks on Deep Neural Networks, *Proc. of RAID 2018*, LNCS, Vol. 11050, Springer, pp. 273–294 (2018).

[14] Chou, E., Tramèr, F. and Pellegrino, G.: SentiNet: Detecting Localized Universal Attacks Against Deep Learning Systems, *Proc. of IEEE SPW 2020*, IEEE, pp. 48–54 (2020).

[15] Zhang, R., Isola, P., Efros, A. A., Shechtman, E. and Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric, *Proc. of CVPR 2018*, IEEE, pp. 586–595 (2018).

[16] Huang, X., Alzantot, M. and Srivastava, M. B.: NeuronInspect: Detecting Backdoors in Neural Networks via Output Explanations, *CoRR*, Vol. abs/1911.07399 (2019).

[17] Chen, X., Wang, W., Bender, C., Ding, Y., Jia, R., Li, B. and Song, D.: REFIT: A Unified Watermark Removal Framework For Deep Learning Systems With Limited Data, *Proc. of AsiaCCS 2021*, ACM, pp. 321–335 (2021).