埋め込み表現の分布シフトに基づく バックドアニューロンの特定および除去手法

岩花 一輝[1,a)]　伊東 燦[1]　三浦 尭之[1]　山﨑 雄輔[1]　芝原 俊樹[1]

概要：推論時に, 攻撃者のみが知るトリガを入力に付与することで，特定のクラスへ分類されるようにモデルを汚染するバックドア攻撃の脅威が指摘されており, 様々な対策手法が研究されている．その中でもバックドアにとって重要なニューロン（バックドアニューロン）を特定し, それに基づきバックドアを除去する手法が注目されている．バックドアニューロンはトリガを用いることで特定できることが知られている．しかし, 防御者はトリガーを知らないため，これらの対策手法ではバックドアニューロンの高精度な特定に課題が存在する．本研究では, バックドアニューロンを正確に特定するために, モデルの埋め込み表現における分布シフトに着目する．バックドア攻撃では, 正常な入力の埋め込み表現に対してバックドアニューロンが強く活性化することで, 汚染クラスへ分布が移動すると考えられる．この考えに基づき, 正常な入力の埋め込み表現から特定のクラスへ分類されるような分布シフトを求める手法および, 得られた分布シフトから汚染クラスを特定する手法を提案する．実験評価を通じて, 従来手法よりも正確にバックドアニューロンを特定可能なことと，得られた分布シフトを用いてファインチューニングすることでバックドアが除去できることを確認した．

キーワード：バックドアニューロン, 分布シフト, バックドア攻撃, バックドア除去

Identifying and Erasing Backdoor Neurons Based on Embedding Shift

Kazuki Iwahana[1,a)]　Akira Ito[1]　Takayuki Miura[1]　Yusuke Yamasaki[1]　Toshiki Shibahara[1]

Abstract: Backdoor attacks compromise models so that inputs with hidden triggers are misclassified into a specific target class. While existing defenses attempt to identify and remove “backdoor neurons” using the trigger, defenders typically lack this knowledge, making detection difficult. We instead focus on distribution shifts in embedding representations: in backdoor attacks, embeddings of clean inputs drift toward the poisoned class due to backdoor neuron activations. Leveraging this insight, we propose a method to detect such shifts, identify the poisoned class, and locate backdoor neurons. Experiments show our method achieves more accurate neuron identification and enables effective backdoor removal through fine-tuning.

Keywords: Backdoor neurons, Embedding shift, Backdoor attack, Backdoor removal

1. はじめに

人工知能 (AI) が, 画像認識や自動運転の分野をはじめとして, 人々に広く浸透し始めてきている一方で, 性能の良い AI だけでなく安全な AI の研究開発も求められている．AI の安全性を脅かす攻撃として, バックドア攻撃 [1] が知られている．バックドア攻撃は, 正常なデータに対しては通常の振る舞いを行うが, 攻撃者のみが知る情報（トリガ）が付与されたデータに対しては攻撃者の意図したクラスのデータとして認識されるように, モデルを汚染する攻撃である．

バックドア攻撃への対策は, 異常な入力の検知 [2] やバックドア攻撃を受けたモデルかどうかの検知 [3] をはじめとして様々に研究されているが，安全な AI の運用という観

[1] NTT 社会情報研究所
NTT Social Informatics Laboratories
[a)] kazuki.iwahana@ntt.com

点から, 汚染されたモデルに対して, バックドア攻撃の影響を除去することが重要となる. 除去手法の多く [4–8] は, モデル内部のニューロンのうち, バックドア攻撃の成功に重要なニューロン (バックドアニューロン) を特定し, 除去している. バックドアニューロンの中でも, 正常なデータとトリガが付与されたデータ（汚染データ）を入力した際のニューロンの活性値の差 (Trigger-Activated Changes, TAC) [9] が大きいほど, バックドア攻撃の成功に重要度が高いニューロンとして考えられている [8,10]. しかし, TAC を計算するためには, 汚染データが必要となるが, 防御者は TAC を計算することはできないので, 汚染データを知らない状況下で TAC に基づくバックドアニューロンを特定する手法は未解決問題である. いくつかの既存手法 [4–8] でも, それぞれ独自の手法を用いて, バックドアニューロンの特定を試みているが, 依然として TAC に基づくバックドアニューロンの特定率が低く, バックドアの除去に失敗する場合がある.

TAC に基づくバックドアニューロンをより正確に特定し, 頑健なバックドア除去を実現するために, 本研究ではモデルの中間層の特徴量空間（埋め込み表現）における分布シフトに基づいてバックドアニューロンを特定し, 除去する手法を提案する. バックドア攻撃では, 正常なデータに対するニューロンの活性値に加えて, トリガに反応したニューロンが活性値することで, 汚染データが汚染クラスに分類される. すなわち, バックドアニューロンの影響を受けることで, 埋め込み表現において正常クラスから汚染クラスへの分布のシフトを引き起こすと考えられる.

本研究の貢献は以下のとおりである.

- 任意のクラスのデータを特定のクラスへ分類させるような埋め込み表現における分布シフトを求める手法を提案する. 計算された分布シフトは TAC に基づくバックドアニューロンと強い相関を示していることを実験的に確認し, 既存手法よりもバックドアニューロンの特定率が高いことを確認した.
- 汚染クラスにおける分布シフトは正常クラスに比べて, L2 ノルムが小さいという性質を実験的に明らかにし, その知見に基づいて, 計算された各クラスの分布シフトから汚染クラスを特定する手法を提案する. 実験的に汚染クラスを特定できていることを確認した.
- 最後に, 汚染クラスにおける分布シフトを用いてファインチューニングすることで, 精度を維持しつつもバックドアが除去できた. さらに最先端の防御手法と比較しても性能の良いバックドア除去手法であることを確認した.

2. 関連研究

2.1 バックドア攻撃

バックドア攻撃は, 正常なデータに対しては自然な振る舞いを行う一方で, 汚染データに対しては攻撃者が意図したクラスを返すようにモデルに細工する攻撃である. BadNets [1], Blend [11], Trojann [12] のような可視性のあるトリガを付与する攻撃は簡易に攻撃が成功する一方で, ステルス性が低く対策が容易となる. バックドア攻撃のステルス性を高めるために, 正常なデータと汚染データの違いを識別できないようなトリガが考えられてきた [13–15]. さらに, 入力するデータでのステルス性だけでなく, モデル内部におけるステルス性を高める攻撃 [16–18] も考えられてきた.

2.2 バックドア除去

モデルからバックドアを除去する手法として, ニューロンの枝刈りもしくは, ファインチューニングする手法 [4–8,10] が知られている. FP [4] は正常なデータに対して活性化しないニューロンがバックドアニューロンだという仮定に基づき, 不要なニューロンを枝刈りしファインチューニングを行う. 具体的に, CLP [5] は, 正常なデータと汚染データの活性値の差分 (TAC) が大きいニューロンをバックドアの攻撃成功に重要なニューロンとする指標を導入している. しかし, TAC の計算には汚染データが必要であり, 通常の防御者はトリガを知らないため, 計算できない. TSBD [8] では, 正常なデータを用いて忘却し, 忘却前後の重みの変化量が大きいニューロンをバックドアニューロンとみなす手法を提案した. また, 既存のバックドア除去手法 [4–7] よりも, TAC に基づくバックドアニューロンの特定率は高いことを示したものの, 十分ではないため, バックドア除去に失敗する可能性がある.

3. 問題設定

本節では, まず本研究が想定する攻撃者と防御者の脅威モデルについて述べる. さらに, バックドア攻撃の定式化に加えて, バックドアニューロンについて述べる.

3.1 脅威モデル

攻撃者と防御者のそれぞれの目的・能力を述べる.

攻撃者. 攻撃者の目的は, 汚染データに対しては汚染クラス, 正常なデータに関しては正解のクラスの出力をする汚染モデルパラメータを得ることである. このとき, 攻撃者は学習データセットもしくは, モデルの学習プロセスに全てアクセス可能であると仮定する.

防御者. 防御者の目的は, 与えられたモデルがバックドアに汚染されていることを検知し, モデルからバックドアの

除去を行うことである．このとき, 防御者は学習済みモデルにアクセス可能であり, モデルの学習データセットと同一分布の少量データセットを保有している．

3.2 バックドア攻撃

入力のチャネル数 c, 高さ h, 幅 w とし, 入力次元を $d_{\text{in}} = c \times h \times w$ とする．データの入力空間を $X \subset \mathbb{R}^{d_{\text{in}}}$, クラス数を N, クラス $i \in [N-1]$, に対して i 番目の要素が 1 となるようなワンホットベクトルを $y_i \in \{0,1\}^N$ とする．データを入力とし各クラスに属する確率を出力するニューラルネットワークを $f : X \to [0,1]^N$, モデルパラメータを $\boldsymbol{\theta}$, 損失関数を ℓ と表記する．このとき, d_{emb} 次元のベクトルである最終線形層の手前の層までの関数 $\phi_{\boldsymbol{\theta}} : \mathbb{R}^{d_{\text{in}}} \to \mathbb{R}^{d_{\text{emb}}}$ により出力された値を, 本研究では “埋め込み表現” と呼称する．データ $\boldsymbol{x} \in X$, 埋め込み表現, 最終 (第 L 層目) の線形層の重み $\boldsymbol{W}_L \in \mathbb{R}^{N \times d_{\text{emb}}}$, バイアス項 $\boldsymbol{b} \in \mathbb{R}^N$, Softmax 関数を用いると, ニューラルネットワークの出力は式 1 のように書ける．

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \text{Softmax}(\boldsymbol{W}_L \phi_{\boldsymbol{\theta}}(\boldsymbol{x}) + \boldsymbol{b}) \tag{1}$$

また, バックドア攻撃に必要なトリガを $\boldsymbol{\delta} \in \mathbb{R}^{d_c \times d_w \times d_h}$, 汚染クラスを $t \in [N-1]$ とする．このとき, バックドアに汚染されたパラメータ $\boldsymbol{\theta}_{\text{bd}}$ は, 式 2 で得られる．

$$\boldsymbol{\theta}_{\text{bd}} = \underset{\boldsymbol{\theta}}{\text{argmin}}\, \mathbb{E}_{(\boldsymbol{x},y)\in D}[\ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y) + \ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}), y_t)] \tag{2}$$

すなわち, モデルパラメータ $\boldsymbol{\theta}_{\text{bd}}$ は, ある学習データセット D の全ての $(\boldsymbol{x}, y)$ に対して, 正解のクラスの平均ロスができるだけ小さく，汚染データ $\boldsymbol{x}+\boldsymbol{\delta}$ に対しては汚染クラス t の平均ロスができるだけ小さいように選択される．

3.3 バックドアニューロンの特定

バックドアに汚染されたモデルに対して汚染データを入力した場合, 正常なニューロンの活性値に加えて, バックドアニューロンが強く活性化することで汚染クラスへ分類される．バックドアニューロンを特定することができれば, そのバックドアニューロンの影響を消すようにファインチューニングすることで, 汚染モデルからバックドア攻撃の影響を除去することが可能となる．

本研究では, 既存手法 [8, 10] と同様に, 正常なデータと汚染データのニューロンの活性値の差 (Trigger-Activated Changes, TAC) [9] が大きいほど, バックドアの攻撃成功にとって重要なニューロンであるとみなす．具体的に, ある $\boldsymbol{x}$ に対して L 層目における k 番目のニューロンの TAC は以下のように計算される．

$$\text{TAC}_k^{(\ell)}(\boldsymbol{x}) = ||f_{\boldsymbol{\theta}}^{(\ell,k)}(\boldsymbol{x}+\boldsymbol{\delta}) - f_{\boldsymbol{\theta}}^{(\ell,k)}(\boldsymbol{x})||_2 \tag{3}$$

しかし, TAC の計算には汚染データ $\boldsymbol{x}+\boldsymbol{\delta}$ が必要であり, 通常の防御者はトリガ $\boldsymbol{\delta}$ を知らないため, バックドアニューロンの正確な特定は困難である．

本研究の主たる問いは, 汚染データを知らない状況で, TAC に基づいたバックドアニューロンを正確に特定し, モデルからバックドアを除去できるかを明らかにすることである．

4. 提案手法

本研究では, 汚染モデルのバックドアニューロンの特定するために, 埋め込み表現における分布シフトに着目する．[19] によると, バックドア攻撃は正常なクラスのデータに対して, トリガの影響により中間層の分布シフトが起こることで, 汚染クラスへ分類されることが知られている．すなわち, 汚染クラスへ向かう埋め込み表現の分布シフトは, TAC に基づくバックドアニューロンの影響を受けていると考えられる．

上記の考えに基づき, 汚染クラスにおける埋め込み表現の分布シフトを計算する手法, および, 計算された分布シフトに基づきバックドアを除去する手法を提案する．図 1 に示すように, 提案手法は , (1) 各クラスにおける埋め込み表現の分布シフトの計算, (2) 計算された分布シフトを用いた汚染クラスの特定, (3) さらに汚染クラスの分布シフトを用いたバックドア除去の 3 段階から成る．以下, 各段階の詳細について説明する．

4.1 埋め込み表現の分布シフトの計算

あるクラス i における埋め込み表現の分布シフトを $\boldsymbol{s}_i^* \in \mathbb{R}^{d_{\text{emb}}}$, 損失の閾値 α とすると, 本手法の分布シフトは式 4 のように計算される．

$$\begin{aligned} \boldsymbol{s}_i^* = &\underset{\boldsymbol{s}_i}{\text{argmin}} \quad ||\boldsymbol{s}_i||_2 \\ &\text{s.t.} \quad \mathbb{E}_{(\boldsymbol{x},y)\in D}[\ell\big(\text{Softmax}(\boldsymbol{W}_L(\phi_{\boldsymbol{\theta}}(\boldsymbol{x}) + \boldsymbol{s}_i) + \boldsymbol{b}), y_i\big)] < \alpha \end{aligned} \tag{4}$$

分布シフトの条件としては, 式 1 における埋め込み表現の出力に $\boldsymbol{s}_i$ を加算することで, クラス i に分類される損失ができる限り小さくなる, すなわちクラス i に分類される確率が高くなる必要がある．一方, そのような条件を満たすような $\boldsymbol{s}_i$ は簡易に存在してしまう．例えば, $\boldsymbol{W_L}$ におけるクラス i の成分が正となるような $\boldsymbol{s}_i$ の要素に対して, 十分に大きな値を設定することでクラス i に分類されてしまう．このような $\boldsymbol{s}_i$ は求めるべき分布シフトとは異なる．そのため, $\boldsymbol{s}_i$ ができるだけ小さくなるように, L2 制約項を主目的とする．これは TAC が正常なデータの活性値に比べて, それほど大きな値を取らないという従来の知見 [9, 18] に基づくものである．

既存の分布シフトを求める手法 [19] も存在するが, 5.2

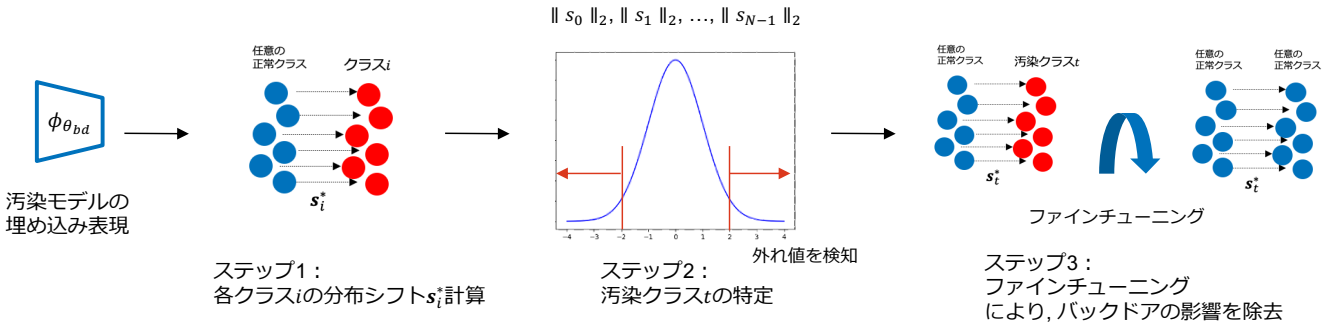


図 1: 提案手法の概要

節に，提案手法の方が TAC に基づくバックドアニューロンを復元できていることを示す．

4.2 汚染クラスの特定

各クラスの分布シフトを用いて，汚染クラスを特定する手法について述べる．4.1 節では各クラスの分布シフト $\boldsymbol{s}_0^*, \cdots, \boldsymbol{s}_{N-1}^*$ を求めた．ここから汚染クラスの分布シフトを用いてバックドアを除去することを試みたいが，防御者は汚染クラスを知らないが普通である．具体的には以下の処理を行う．

(1) $\boldsymbol{s}_0^*, \cdots, \boldsymbol{s}_{N-1}^*$ の L2 ノルムをそれぞれ，$c_0, \cdots, c_{N-1}$ とする．

(2) $c_0, \cdots, c_{N-1}$ に対して，平均 μ と標準偏差 σ を用いて標準化したものを $z_0, \cdots, z_{N-1}$ とする．すなわち，$\forall i \in Y : z_i = (c_i - \mu)/\sigma$ である．

(3) 外れ値の閾値を β として，$|z_i| > \beta$ となるようなクラス i を汚染クラス t と決定する．

本手法は，各クラスの分布シフトに対して，汚染クラスと正常クラスの L2 ノルムに違いがないか調べたところ，汚染クラスの分布シフトの L2 ノルムが正常クラスに比べて，小さくなることを実験的に確認し，その知見に基づくものである．これは学習時にトリガを強く記憶することにより，正常なクラスよりも最小のシフトで汚染クラスへ分類させることが起因すると考えられる．

4.3 分布シフトを用いたバックドア除去

既存手法 [19] に基づいて，4.2 節で求めた汚染クラスの分布シフトを用いたバックドア除去手法を式 5 に示す．

$$\theta_{\mathrm{ft}}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \quad \mathbb{E}_{(\boldsymbol{x},y)\in D}\Big[\ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y) + \ell\big(\mathrm{Softmax}(\boldsymbol{W}_L(\phi_{\boldsymbol{\theta}}(\boldsymbol{x}) + \boldsymbol{s}_p^*) + \boldsymbol{b}), y\big)\Big] \tag{5}$$

具体的には埋め込み表現に対して，$\boldsymbol{s}_t^*$ 方向に分布がずれたとしても元の正常クラスと認識するようにファインチューニングすることで，トリガ付きデータを入力したとしても正常なクラスへ分類される．さらに，元々の正常なタスクに対する性能も維持する必要があるため，正常なデータに対して正常なクラスへ分類されるような損失も加える．

5. 実験

本節では，4 節で述べた提案手法の有効性を確認するために，実験評価を行う．具体的には，(1) 4.1 によって計算された汚染クラスにおける分布シフトが，TAC に基づくバックドアニューロンをどれだけ正確に特定できているか，(2) 4.2 で述べた提案手法が，汚染クラスを特定できているか．(3) 4.3 節で述べた汚染クラスの分布シフトに基づいて，バックドア除去を行なった際の除去性能が良いかどうか，を明らかにする．

5.1 設定

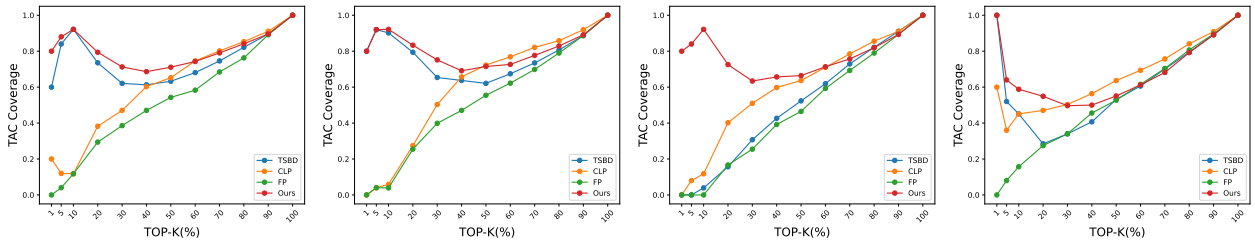
ここでは，本実験で用いた各設定について述べる．

データセットとニューラルネットワークのアーキテクチャ. 本実験で用いるデータセットは 10 クラスの画像分類データである CIFAR10 である．なお，ニューラルネットワークのアーキテクチャとして，ResNet18 を用いた．

攻撃手法. 本実験では，4 つのバックドア攻撃，Badnet [1], Trojan [12], Blend [11], WaNet [13] を対象に提案手法の有効性を評価した．汚染モデルの学習設定として，エポック数を 100，学習オプティマイザーを確率的勾配降下法 SGD，学習率を 0.1 とした．また，バックドア攻撃の設定として，汚染クラスは 0 で，基本的な汚染率は 0.1，汚染クラス以外の全てのクラスのデータに対してトリガ付きデータを仕込むような all-to-one 設定としている．その他の各手法のハイパーパラメータについては元論文に従う．

防御手法. 本実験では，バックドアニューロンを特定および，その影響を除去する 3 つの手法，FP [4], CLP [5], TSBD [8] を対象に提案手法の有効性を比較した．なお，各手法のハイパーパラメータについては元論文に従う．

提案手法. 本実験では，既存手法 [6, 8, 10] に従い，防御者は 5.0%の正常なデータとクラスの組を持っていると仮定している．各クラスの埋め込み表現の分布シフトを計算す



(a) BadNets (b) Trojan (c) Blend (d) WaNet

図 2: 各防御手法と TAC に基づくバックドアニューロンとの重複率

表 1: 各防御手法とのコサイン類似度の比較

	BadNets	Trojan	Blend	WaNet
FP	-0.0135	0.0044	-0.2168	-0.3478
CLP	-0.1019	-0.0820	-0.1104	-0.1395
TSBD	0.8026	0.7458	-0.1580	0.6447
Ours	0.9294	0.9211	0.9087	0.7170

る際, ステップ数を 200, 学習オプティマイザーを Adam, 学習率を 0.01, $\alpha = 0.1$ とし, 汚染クラスの特定に用いた β は 2.0 とした. また, バックドアを除去するためのファインチューニングでは, 学習オプティマイザーを SGD, 学習率 0.01, エポック数を 50 とした.

5.2 埋め込み表現の分布シフトとバックドアニューロンの特定率

まず, 汚染クラスにおける分布シフトがどれだけ TAC と類似しているかを確認する. 各ニューロンに対する TAC の値を横軸とし, 提案手法によって計算された分布シフトを縦軸として図 3 に示す. 結果として, 提案手法により計算された埋め込み表現における分布シフトはどの攻撃においても強い相関を示していることを確認した. すなわち, 提案した埋め込み表現の分布シフトにより, TAC に基づくバックドアニューロンを特定できることを示している.

ここで, 分布シフトの計算手法は既存手法 BEEAR [19] でも提案されているため, 提案手法と BEEAR の比較を図 4 に示す. 提案手法は TAC と相関のある分布シフトを復元できているのにも関わらず, BEEAR では TAC と相関がないような分布シフトであることを確認した. これは BEEAR の最適化問題では分布シフトに関する大きさの制約が無いため, 大きな値を取ってしまうことが原因であると考えられる.

最後に, 提案手法による埋め込み表現の分布シフトが, 既存手法と比べて TAC に基づくバックドアニューロンをどれほど正確に特定できているかを, コサイン類似度および, TOP-K% おける TAC に基づくバックドアニューロンとの重複率, 2 つの指標を用いて示す. 各手法によって計算されたニューロンの重要度を, コサイン類似度はベクトルの値を含めて類似性を比較できる. 重複率は求められたニューロンの重要度を降順に並び替えた際に, TOP-K% において TAC に基づくバックドアニューロンとどれほど一致しているかを比較できる. まず, コサイン類似度の結果を表 1 に示す. 既存のバックドアニューロンの特定手法, とくに TSBD は BadNets でコサイン類似度 0.802 と高い値を示したものの, 提案手法は 0.929 と上回っている. さらに, 既存の特定手法と比べて提案手法はどの攻撃に関しても安定して高い類似度を示している. また, 重複率の結果を図 2 に示す. 提案手法は低い K において高い重複率を示しており, これは TAC が高いニューロンをより正確に特定できていることを意味する. また, コサイン類似度と提案手法は同様にどの攻撃においても安定して高い重複率を示した.

5.3 汚染クラスの特定

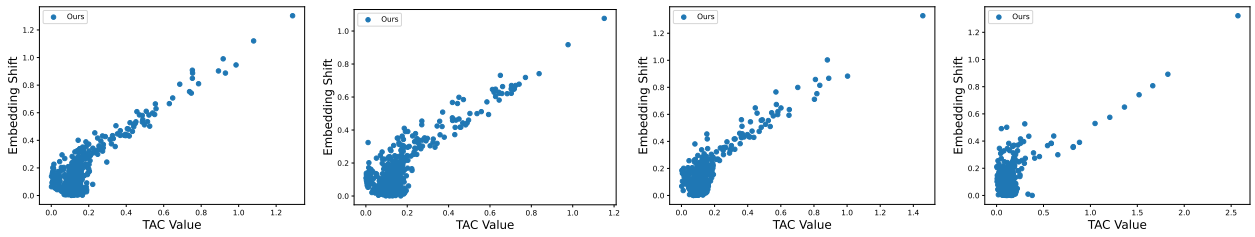
続いて, 4.2 節で述べた汚染クラスの特定手法により, 汚染クラスを特定できるかどうかを確認する. 各攻撃手法に対して, 各クラスの分布シフトの L2 ノルムをそれぞれ示した表を以下に示す. 表には L2 ノルムとその標準化後の値を示しており, 標準化後の値で閾値 2 を超えるものは太字にしている.

表の結果から, どの攻撃手法においても汚染クラスの分布シフトの L2 ノルムは他のクラスと比べての低くなっていることがわかる. 実際に, 汚染クラス 0 の標準化した後の値を右側に示しているが $|z_0| > 2$ を満たすため, 汚染クラスの検知ができている.

5.4 バックドア除去

提案手法によるバックドアの除去性能および, 既存の防御手法との性能差を実験的に確認する.

評価指標. まず, 既存研究 [8, 10] に従い, 3 つのバックドア攻撃における評価指標を導入する. 通常タスクの正解率である精度 (ACC). トリガ付きデータが汚染クラスに分類される割合を示す攻撃成功率 (ASR), さらに精度を維持しつつバックドアのみを除去できた指標である防御機能率 (DER) を用いる. DER は, $\mathrm{DER} = (\max(0, \Delta\mathrm{ACC}) - \max(0, \Delta\mathrm{ASR} + 1))/2$ で計算され, 0 から 1 の範囲を取り, 1 に近いほど効果的にバックド



(a) BadNets (b) Trojan (c) Blend (d) WaNet

図 3: ResNet18: 各攻撃手法に対する埋め込み表現の分布シフトと TAC との関係

表 2: ResNet18: 各クラスの分布シフトの L2 ノルムと標準化後の値

		Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
L2 ノルム	BadNets	4.5902	5.7068	5.0467	5.1318	5.2074	5.2113	5.2538	5.2919	5.3592	5.4181
	Trojan	4.4713	5.7013	5.0536	5.1178	5.1718	5.2152	5.3006	5.3549	5.4100	5.5185
	Blend	4.3044	5.6376	5.0498	5.1227	5.1316	5.1859	5.2904	5.3484	5.4945	5.4183
	WaNet	3.3077	5.2566	4.6121	4.7036	5.0406	5.0018	5.0899	4.9878	4.8985	5.1066
標準化	BadNets	**-2.3270**	1.7873	-0.6448	-0.3312	-0.0529	-0.0383	0.1182	0.2585	0.5065	0.7236
	Trojan	**-2.4268**	1.4998	-0.5679	-0.3630	-0.1907	-0.0519	0.2205	0.3938	0.5699	0.9163
	Blend	**-2.5897**	1.2725	-0.4304	-0.2191	-0.1934	-0.0362	0.2667	0.4346	0.8578	0.6372
	WaNet	**-2.8203**	0.8617	-0.3560	-0.1830	0.4535	0.3802	0.5467	0.3538	0.1851	0.5782

表 3: ResNet50: 各クラスの分布シフトの L2 ノルムと標準化後の値

		Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
L2 ノルム	BadNets	4.1445	6.0710	5.3332	5.6283	5.5600	5.6164	5.7831	5.6715	5.6829	5.8469
	Trojan	4.6884	6.5206	6.0624	6.1341	6.0718	6.0586	6.3775	6.3162	6.1675	6.4331
	Blend	4.1723	6.0552	5.7315	5.7608	5.7854	5.8249	5.9527	5.8044	5.7338	5.9459
	WaNet	3.5280	6.4481	6.4772	6.3523	6.5849	6.6053	6.8495	6.4669	6.6952	6.8567
標準化	BadNets	**-2.7915**	1.0795	-0.4031	0.1900	0.0526	0.1659	0.5011	0.2768	0.2997	0.6291
	Trojan	**-2.8403**	0.8911	-0.0421	0.1040	-0.0229	-0.0497	0.5998	0.4748	0.1823	0.5295
	Blend	**-2.8226**	0.9767	0.2145	0.1467	0.1992	0.2967	0.6129	0.4186	0.1411	0.5478
	WaNet	**-2.8567**	0.8002	0.3877	0.1302	0.2580	0.2912	0.7876	0.4413	0.1991	0.6712

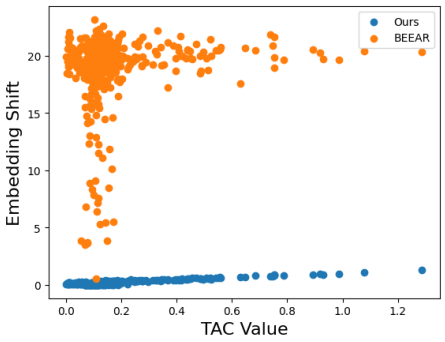


図 4: BadNets に対する提案手法と BEEAR [19] の比較

アのみを除去できていることを示している.

実験結果. 表 4 に, 各攻撃手法と防御手法の実験結果を示す. Average は各指標の平均を示している. また, 防御していない場合に DER は存在しないため – と記載している.

提案手法は, 実際に攻撃成功率を平均約 2.0% まで, 下げることができ, 効果的にバックドアを取り除くことができている. 一方, 精度も防御する前に比べると, 平均 1.3% ほど低下しているものの高く維持できている. また, 防御機能率は, 既存のどの防御手法よりも上回っているため, 既存の防御手法よりも優位な手法であると言える.

5.5 アブレーションスタディ

本節では, 実験条件を変更した場合の提案手法の頑健性を示す.

汚染クラスの変更. 汚染クラスを特定する手法の頑健性を確かめるため, 汚染クラスが 0 の場合に限らず, それ以外の場合においても提案手法により汚染クラスを検知できるかを確認する. その結果を図 5 に示す. 横軸は汚染クラス, 縦軸は各クラスを示し, 表示されている数値は標準化後の値である. 基本的に, 正常クラスに比べて汚染クラスの分布シフトの L2 ノルムが小さく, 標準化された値が-2 を下回ることを確認した. ただし, BadNets においてクラス 1 のみを汚染した場合には, 標準化後の値が-2 を上回った. 図 5 の Class1 の行を見ると, クラス 1 が全体的に他のクラスに比べて高い L2 ノルムを示している. すなわち, クラス

表 4: ResNet18: 各攻撃手法と防御手法の精度, 攻撃成功率, 防御機能率

	No defense			FP			CLP			TSBD			Ours		
	ACC↑	ASR↓	DER↑	ACC↑	ASR↓	DER↑	ACC↑	ASR↓	DER↑	ACC↑	ASR↓	DER↑	ACC↑	ASR↓	DER↑
BadNets	0.9381	1.0000	-	0.9364	1.0000	0.4992	0.9130	0.3310	0.8220	0.9069	0.1253	0.9217	0.9255	0.0573	0.9650
Trojan	0.9400	1.0000	-	0.9346	0.0204	0.9871	0.8397	0.0119	0.9439	0.9012	0.0357	0.9628	0.9244	0.0067	0.9889
Blend	0.9329	0.9991	-	0.9313	0.1296	0.9340	0.9009	0.3046	0.8313	0.8876	0.0366	0.9586	0.9182	0.0148	0.9848
WaNet	0.9341	0.9959	-	0.9337	0.0081	0.9937	0.1023	1.0000	0.0841	0.8785	0.7928	0.5738	0.9246	0.0046	0.9909
Average	0.9363	0.9988	-	0.9340	0.2895	0.8535	0.6890	0.4119	0.6703	0.8936	0.2476	0.8542	0.9232	0.0208	**0.9824**

表 5: ResNet50: 各攻撃手法と防御手法の精度, 攻撃成功率, 防御機能率

	No defense			FP			CLP			TSBD			Ours		
	ACC↑	ASR↓	DER↑	ACC↑	ASR↓	DER↑	ACC↑	ASR↓	DER↑	ACC↑	ASR↓	DER↑	ACC↑	ASR↓	DER↑
BadNets	0.9148	1.0000	-	0.9099	0.5843	0.7054	0.6383	0.0933	0.8151	0.8141	0.0474	0.9259	0.8878	0.0453	0.9638
Trojan	0.9269	1.0000	-	0.9189	0.0266	0.9827	0.5080	0.0324	0.7743	0.8727	0.0224	0.9617	0.8992	0.0101	0.9811
Blend	0.9211	0.9959	-	0.9116	0.1733	0.9065	0.4762	0.0568	0.7471	0.8342	0.0507	0.9292	0.8930	0.0248	0.9715
WaNet	0.9283	0.9892	-	0.9191	0.0088	0.9856	0.5954	0.0014	0.8274	0.8380	0.7151	0.5919	0.9065	0.0074	0.9800
Average	0.9228	0.9963	-	0.9149	0.1982	0.8951	0.5545	0.0460	0.7910	0.8398	0.2089	0.8522	0.8966	0.0219	**0.9741**

1 に分類されるためには, L2 ノルムが大きいような分布シフトが必要である. ゆえに, 汚染した場合には正常な場合と比べて L2 ノルムが小さくなるものの, 他の汚染クラスと比べると大きくなってしまい, 特定が失敗してしまうと考えられる. このように, データセットが持つ各クラスの特徴に左右されないような汚染クラスの特定手法は, 今後の課題である.

アーキテクチャの変更. アーキテクチャを変更した場合の提案手法の有効性を確かめる. まず, ResNet50 を用いた場合の, 分布シフトの L2 ノルムおよび標準化後の値を表 3 に示す. ResNet18 が, 512 次元の埋め込み表現のベクトルに対して, ResNet50 では 2048 次元と高次元にも関わらず, 表 2 に示した結果と同様に, 汚染クラスの分布シフトの L2 ノルムが正常クラスよりも小さく, 汚染クラスを特定できることを確認した. さらに, バックドア攻撃の影響を除去した結果を表 5 に示す. ResNet18 と同様に ResNet50 でも, 提案手法が既存の防御手法よりも性能が良いことを確認した.

汚染率の変更. バックドア攻撃では, 汚染率が高いほどトリガを認識しやすく低いほどトリガを認識しづらいことが知られている. そこで, 汚染率を 0.01, 0.05 とそれぞれ変更した場合に, 提案手法により, バックドアを除去できるかどうかを確認する. 実験結果を表 6 に示す. なお, WaNet [13] は汚染率に依存しない攻撃のため, 除いている. BadNets に関して, 汚染率が 0.1 の時と比べて, 0.179 と攻撃成功率が高くなってしまっているものの, 汚染率が小さい場合でも, 精度を維持しつつ頑健に除去できていることを確認した.

表 6: それぞれの汚染率による提案手法の性能評価

		No Defense			Ours		
		ACC↑	ASR↓	DER↑	ACC↑	ASR↓	DER↑
0.01	BadNets	0.9379	1.0000	-	0.9207	0.1791	0.9018
	Trojan	0.9394	0.9997	-	0.9215	0.0207	0.9806
	Blend	0.9376	0.9682	-	0.9158	0.0030	0.9717
	Average	0.9383	0.9893	-	0.9193	0.0676	0.9514
0.05	BadNets	0.9428	1.0000	-	0.9147	0.0249	0.9735
	Trojan	0.9402	1.0000	-	0.9205	0.0109	0.9847
	Blend	0.9378	0.9973	-	0.9166	0.0134	0.9813
	Average	0.9403	0.9991	-	0.9173	0.0164	0.9799

6. 制約

6.1 汚染クラスが複数存在する場合

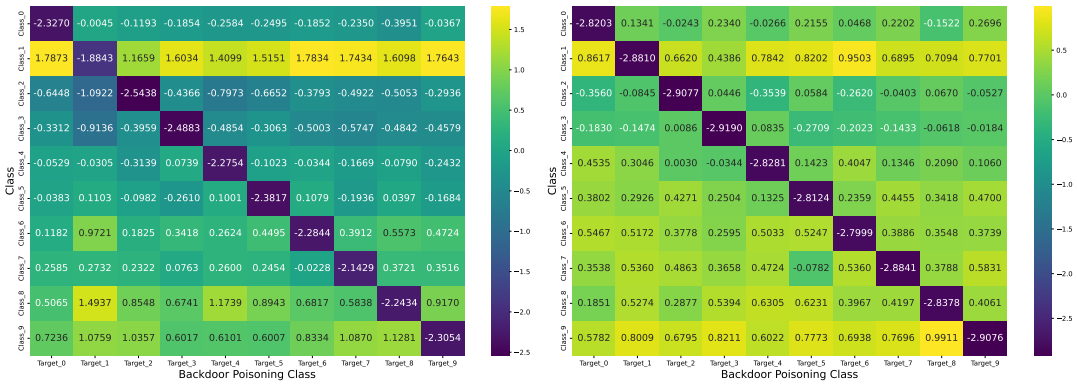
モデルの複数のクラスが汚染されている場合には, 汚染クラスの特定手法がうまく機能しないと考えられる. 4.2 節で述べた手法は, 正常クラスに対して相対的に汚染クラスの分布シフトの L2 ノルムが小さくなっているという経験的な結果に基づいて, L2 ノルムが外れ値であるクラスを汚染クラスとしている. すなわち, 汚染クラスが正常クラスに対して多くなってくると, たとえ, 汚染クラスだとしても外れ値にはなり得ない. 汚染クラスが複数想定する場合の検知方法については今後の解決すべき課題である.

6.2 One-to-One 設定でのバックドア攻撃の場合

本研究では, 汚染クラス以外のデータに対してトリガを付与することで汚染クラスに分類されるようにする All-to-One 設定を想定している. そのため, ある特定のクラスのデータに対してトリガを付与することで汚染クラスに分類される One-to-One 設定では, 埋め込み表現の分布シフトによりバックドアニューロンを正確に求められない可能性が高い. しかし, 式 4 の分布シフトの計算時に, 全てのデータを利用するのではなく特定のクラスのデータを利用して分布シフトを求められれば, One-to-One 設定でも提案手法がうまく機能すると考えられる. この実験的な検証は今後の課題とする.

7. おわりに

本研究では, 汚染クラスにおける埋め込み表現における



(a) BadNets (b) WaNet

図 5: 各汚染クラスと正常クラスの分布シフトの L2 ノルム

分布シフトに基づいてバックドアニューロンを特定し, 除去する手法を提案した. 実際に, 提案手法により計算された埋め込み表現の分布シフトが, 既存手法よりも, TAC に基づくバックドアニューロンを高精度に特定できていること, さらには, 精度を維持しつつバックドア除去もできていることを確認した.

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] Gao, Y., Xu, C., Wang, D., Chen, S., Ranasinghe, D. C. and Nepal, S.: Strip: A defence against trojan attacks on deep neural networks, *Proceedings of the 35th annual computer security applications conference*, pp. 113–125 (2019).

[3] Shen, G., Liu, Y., Tao, G., An, S., Xu, Q., Cheng, S., Ma, S. and Zhang, X.: Backdoor scanning for deep neural networks through k-arm optimization, *International Conference on Machine Learning*, PMLR, pp. 9525–9536 (2021).

[4] Liu, K., Dolan-Gavitt, B. and Garg, S.: Fine-pruning: Defending against backdooring attacks on deep neural networks, *International symposium on research in attacks, intrusions, and defenses*, Springer, pp. 273–294 (2018).

[5] Zheng, R., Tang, R., Li, J. and Liu, L.: Data-free backdoor removal based on channel lipschitzness, *European Conference on Computer Vision*, Springer, pp. 175–191 (2022).

[6] Li, Y., Lyu, X., Ma, X., Koren, N., Lyu, L., Li, B. and Jiang, Y.-G.: Reconstructive neuron pruning for backdoor defense, *International Conference on Machine Learning*, PMLR, pp. 19837–19854 (2023).

[7] Wu, D. and Wang, Y.: Adversarial neuron pruning purifies backdoored deep models, *Advances in Neural Information Processing Systems*, Vol. 34, pp. 16913–16925 (2021).

[8] Lin, W., Liu, L., Wei, S., Li, J. and Xiong, H.: Unveiling and mitigating backdoor vulnerabilities based on unlearning weight changes and backdoor activeness, *Advances in Neural Information Processing Systems*, Vol. 37, pp. 42097–42122 (2024).

[9] Zheng, R., Tang, R., Li, J. and Liu, L.: Data-free backdoor removal based on channel lipschitzness, *European Conference on Computer Vision*, Springer, pp. 175–191 (2022).

[10] Zhu, M., Wei, S., Shen, L., Fan, Y. and Wu, B.: Enhancing fine-tuning based backdoor defense with sharpness-aware minimization, *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4466–4477 (2023).

[11] Chen, X., Liu, C., Li, B., Lu, K. and Song, D.: Targeted backdoor attacks on deep learning systems using data poisoning, *arXiv preprint arXiv:1712.05526* (2017).

[12] Liu, Y., Ma, S., Aafer, Y., Lee, W.-C., Zhai, J., Wang, W. and Zhang, X.: Trojaning attack on neural networks, *25th Annual Network And Distributed System Security Symposium (NDSS 2018)*, Internet Soc (2018).

[13] Nguyen, T. A. and Tran, A. T.: WaNet-Imperceptible Warping-based Backdoor Attack, *International Conference on Learning Representations* (2021).

[14] Nguyen, T. A. and Tran, A.: Input-aware dynamic backdoor attack, *Advances in Neural Information Processing Systems*, Vol. 33, pp. 3454–3464 (2020).

[15] Doan, K., Lao, Y., Zhao, W. and Li, P.: Lira: Learnable, imperceptible and robust backdoor attacks, *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11966–11976 (2021).

[16] Shokri, R. et al.: Bypassing backdoor detection algorithms in deep learning, *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, IEEE, pp. 175–183 (2020).

[17] Zhong, N., Qian, Z. and Zhang, X.: Imperceptible Backdoor Attack: From Input Space to Feature Representation, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, International Joint Conferences on Artificial Intelligence Organization, pp. 1736–1742 (2022).

[18] Xu, X., Liu, Z., Koffas, S. and Picek, S.: Towards Backdoor Stealthiness in Model Parameter Space, *arXiv preprint arXiv:2501.05928* (2025).

[19] Zeng, Y., Sun, W., Huynh, T., Song, D., Li, B. and Jia, R.: BEEAR: Embedding-based Adversarial Removal of Safety Backdoors in Instruction-tuned Language Models, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 13189–13215 (2024).