

アンサンブル代理識別器を用いた バイレベル最適化中毒攻撃の実用性評価

木場 翔太^{1,a)} 長谷川 皓一² 山口 由紀子³ 嶋田 創³

概要：機械学習/深層学習を用いた識別器の利用はサイバーセキュリティの分野においても多用されている。一方で、機械学習/深層学習を利用したシステム自体に対する攻撃も多数提案されており、機械学習/深層学習は完全なセキュリティを保証するものではない。我々の研究室では、以前から、機械学習/深層学習を用いた識別器に対する標的型中毒攻撃の脅威について着目し、中毒攻撃用データの生成手法を研究してきた。以前に考案・評価を行った敵対的生成ネットワークと強化学習を併用した手法は、生成器と敵対的に学習された代理識別器から得られる報酬に依存していた。本研究の提案する手法では、生成器とは独立して学習した複数の異なるアーキテクチャの代理識別器を用いたバイレベル最適化によって、単一の学習プロセスで生成器を最適化し、最終的な攻撃対象であるマルウェア識別器へと中毒攻撃を転移させることを試みた。提案手法を用いて LightGBM ベースの識別器に対する攻撃実験を行った結果、検知精度を悪化させることは困難であり、提案手法は実用上の課題を抱えることが示唆された。

キーワード：マルウェア検知、標的型中毒攻撃、アンサンブル代理識別器、バイレベル最適化

Practical Evaluation of Bilevel Optimized Poisoning Attacks Using Ensemble Surrogate Classifiers

SHOTA KIBA^{1,a)} HIROKAZU HASEGAWA² YUKIKO YAMAGUCHI³ HAJIME SHIMADA³

Abstract: The use of classifiers based on Machine Learning and Deep Learning are widely employed in the field of cybersecurity. However, there are numerous proposals that attack Machine Learning and Deep Learning based system so that the use of these technologies do not guarantee complete security. In our laboratory, we have focused on the threat of poisoning attacks and promoted researches on poisoning attack data generation. Previously, we devised and evaluated a method that combined a Generative Adversarial Network and Reinforcement Learning, but this method relied on rewards obtained from a surrogate classifier that was trained with the generator. In this study, we propose a method that optimizes the generator in a single learning process through bilevel optimization using surrogate classifiers with multiple different architectures trained independently of the generator. The proposed method attempts to transfer the poisoning attack to the malware classifier which is the final target of the attack. The results of an attack experiment using the proposed method against a LightGBM-based classifier showed that it was difficult to degrade detection accuracy, suggesting that the proposed method faces practical challenges.

Keywords: Malware Detection, Targeted Poisoning Attack, Ensemble Surrogate Classifiers, Bilevel Optimization

¹ 名古屋大学大学院情報学研究科

Graduate School of Informatics, Nagoya University

² 国立情報学研究所ストラテジックサイバーレジリエンス研究開発センター

Center for Strategic Cyber Resilience Research and Devel-

opment, National Institute of Informatics

³ 名古屋大学情報基盤センター

Information Technology Center, Nagoya University

a) kiba@net.itc.nagoya-u.ac.jp

1. はじめに

サイバーセキュリティの領域では、機械学習や深層学習が防御技術の要となりつつある。膨大なデータからマルウェア特有のパターンや通信の異常を自動で学習する能力は、未知の脅威を検知し、攻撃を未然に防ぐ上で絶大な力を発揮する。実際に、最新のマルウェア対策製品や侵入検知システム、フィッシングサイト識別などに機械学習や深層学習は広く組み込まれている。

しかし、そのような防御システムが、新たな攻撃の標的となっている。入力データに微細な加工を施して検知を回避する回避攻撃や、学習データに汚染された情報を混入させてモデルの識別精度を悪化させる中毒攻撃などが挙げられる。こうした攻撃の中でも、将来的に特に深刻な脅威となり得るのが、マルウェア識別システムに対する標的型の中毒攻撃である。攻撃者は、標的とする組織が用いるマルウェア識別システムに対して、特定の種類のマルウェアのみを良性プログラムと誤判定させるための偽学習データを事前に拡散する。拡散された偽学習データを、マルウェア識別システムの運用者が意図せず学習データに取り込むことで、本番環境での特定マルウェアへの防御機能が無力化されてしまう。

このような標的型中毒攻撃の脅威の検証のためには、中毒攻撃用の検体の作成手法を考案し、作成された検体が及ぼす影響やその軽減方法の研究を推進する必要がある。以前に考案し、評価を行った敵対的生成ネットワーク (GAN: Generative Adversarial Network) と強化学習を併用した中毒攻撃用の検体の作成手法は、GAN の学習過程で得られた単一の代理検知器から得られる報酬に依存していた [1]。そのため、本研究の提案手法では、中毒攻撃用の検体を出力する生成器とは独立して、複数の異なるアーキテクチャの代理検知器を学習する。そして、学習した代理検知器を用いたバイレベル最適化によって生成器を学習することで、マルウェア検知器による検知を回避するという目的と、特定のマルウェアのみの検知精度を悪化させるという目的を、単一の学習プロセス内で統合的に最適化することを試みる。最終的に、学習した生成器から生成した中毒攻撃用検体の混入前後で、マルウェア識別器の判定精度を比較し、提案手法の有効性を評価した。

2. 関連研究

2.1 攻撃の転移性と代理モデル

機械学習/深層学習モデルに対する攻撃の転移性とは、ある特定のモデルを騙すために生成された敵対的サンプルが、アーキテクチャや学習データが異なる別のモデルにも有効である現象を指す。攻撃の転移性が発生する原因や、アーキテクチャやパラメータが不明な識別器に対する攻撃

転移を発生させるための代理モデルの学習手法について、様々な研究が行われている。Suciu らは、機械学習モデルに対する回避攻撃や中毒攻撃の転移性を、以下の 4 つの攻撃者の能力の指標から体系的に評価することを試みた [3]。

- F: 攻撃者が、標的モデルが使用する特徴量空間についてどの程度の知識を持っているか
 - A: 攻撃者が、標的モデルの学習アルゴリズムやハイパラメータをどの程度把握しているか
 - I: 攻撃者が、標的モデルの学習に用いられたデータにどの程度アクセスできるか
 - L: 攻撃者が、偽学習データを作成する際に、実際にどの特徴量を操作・変更することが許可されているか
- 評価の結果、攻撃者の “F” と “L” を制限することが、攻撃の成功率を低下させるために特に有効であり、堅牢なシステムを構築するためには、各データの来歴の追跡や特徴量の秘匿などのデータ中心のセキュリティ対策を優先して行うことが重要であると主張された。Liu らは、それまで小規模なデータセットでしか議論されてこなかった回避攻撃の転移性について、大規模なデータセットと当時の最先端のモデルを用いて詳細に分析した [4]。具体的には、敵対的サンプルが単にモデルを誤分類させる非標的型攻撃であれば、異なるモデル間でも比較的容易に転移が発生するが、特定のクラスに誤分類させる標的型攻撃は、既存の手法では、その標的を維持したまま他のモデルに転移が発生する確率が非常に低いことを示した。また、この課題に対して、アーキテクチャの異なる複数の代理モデルを同時に騙すように敵対的サンプルを生成する手法を提案し、その有効性を示した。Ilyas らは、回避攻撃に対する脆弱性はモデルの偶発的な欠陥ではなく、人間には知覚できず、微小な擾動によって容易に変化してしまう「非堅牢な特徴」に起因すると主張した [5]。非堅牢な特徴はデータセットに内在するため、この特徴を悪用するように作られた攻撃は、アーキテクチャが異なるモデルの間でも転移する可能性が高いと考えられる。Demontis らは、攻撃の転移の成否の条件について、回避攻撃と中毒攻撃の両方に適用可能な理論を提示した [6]。彼らは、攻撃の転移が成功する主要な要因として、入力に関する損失関数の勾配が大きい脆弱な標的モデルが対象であること、代理モデルと標的モデルとの勾配の方向の類似、そして代理モデルの適度な単純さを挙げた。

2.2 代理モデルのアーキテクチャ

代理モデルのアーキテクチャ選定は、転移攻撃の成功率を左右する重要な要素である。一般的に代理モデルには深層ニューラルネットワーク (DNN) が用いられるが、DNN はその高い表現力と引き換えに、意思決定プロセスが不透明であるという課題を抱えている。この解釈性の低さという課題に対し、モデルの挙動を理解しやすい決定木を代理モデルとして利用するアプローチが考えられる。しかし、

従来の決定木は微分不可能であるため、勾配ベースの攻撃手法を直接適用できないという欠点があった。この問題に対し、各ノードでの分岐を確率的に行うことによってモデル全体の微分可能性を確保するモデルの研究が行われている。Irsoy らは、決定木の各分岐ノードにおいて入力特徴量の線形結合にシグモイド関数を適用することで、入力が左右の子ノードに確率的にルーティングされる Soft Decision Tree (SDT) を提案した [7]。この構造により、ツリー全体が微分可能になり、勾配法によるモデル全体の学習を実現した。また、Kontschieder らは、DNN による特徴学習と、微分可能な決定木のアンサンブルモデルである deep Neural Decision Forest (dNDF) による分類を統合し、全体を一つのモデルとして最適化する手法を提案した。このモデルでは、まず入力データが DNN によって高レベルな特徴量表現に変換される。この特徴量表現が、dNDF を構成する全ての決定木に渡される。各決定木は、SDT と同様に確率的なルーティングを経て、葉ノードで定義されたクラスの事後確率分布を出力する。最終的な予測結果は、全ての決定木が出力した確率分布の平均として得られる。これは、深層学習モデルの出力層を、従来の全結合層とソフトマックス関数から、微分可能な決定木のアンサンブルに置き換えた、より表現力の高いアーキテクチャであると考えられる。

2.3 バイレベル最適化問題としての中毒攻撃

バイレベル最適化問題とは、ある最適化問題の制約条件の中に、さらにもう一つの最適化問題を入れ子状に含まれる構造を持つ数理計画問題である [9]。バイレベル最適化問題は、未知のタスクに対する汎化・適応能力を獲得する学習パラダイムであるメタ学習の基本的な枠組みそのものであり、実際に、Finn らの研究 [10] や Ravi らの研究 [11] に代表される多くのメタ学習手法は、バイレベル最適化問題として定式化されている。Huang らは、このメタ学習の枠組みを応用したクリーンラベル中毒攻撃手法である MetaPoison を提案した [12]。この手法では、まず中毒攻撃用のベースサンプルを、識別精度を悪化させるターゲットサンプルとは異なるラベルから選択する。そして、ベースサンプルに摂動を加えた中毒攻撃用データで代理識別器を仮想的に更新し、代理識別器のターゲットサンプルに対する検証損失を最大化するように摂動を最適化する。この手法の強力な点は、従来の攻撃が識別器の現在の状態を騙すことに留まるのに対し、学習後の性能悪化という未来の結果を直接最適化する点にある。

3. 偽学習データの生成手法

本節では、本研究が提案する、特定のマルウェアファミリに含まれるサンプルのみの検知精度を下げる偽学習データの生成手法について説明する。3.1 節では、旧手法の概

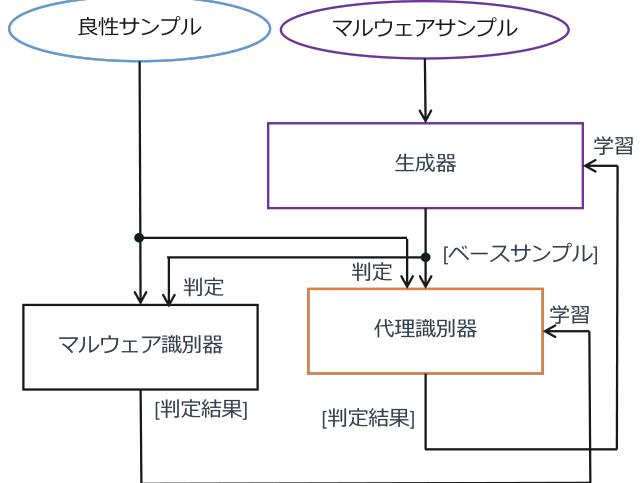


図 1 MalGAN の全体像

Fig. 1 Overview of MalGAN in the previously proposed method

要を説明する。3.2 節では、生成器の学習を利用する代理識別器の学習手法について説明する。3.3 節では、偽学習データを生成する生成器の学習手法について説明する。

3.1 旧手法の概要

旧手法 [1] において、最終的な偽学習データを生成するために学習させるコンポーネントを図 1、図 2 に示す。まず、MalGAN [2] によって生成器と代替識別器を学習させる。代替識別器は、生成器が生成する偽学習データと良性サンプルについて、攻撃対象のマルウェア識別器の判定を模倣するように学習させる。生成器は、自身が生成した偽学習データを代替識別器が良性と判定するように学習させる。MalGAN の学習が終了した時点では、生成器が产出する偽学習データはマルウェア識別器による検知は回避できるが、特定のマルウェアファミリに含まれるサンプルのみの検知精度を悪化させることはできない。

そして、特定のマルウェアファミリのサンプル（図 2 ターゲットサンプル）を入力として学習済みの生成器が产出するベースサンプルを初期状態とし、学習済みの代替識別器と相互作用させながらエージェントを強化学習によって学習させる。具体的には、1つ前のステップの状態に摂動を加えたベクトルを新しい状態とし、その状態を偽学習データとして用いて代替識別器を汚染し、各種サンプルに対する識別精度の悪化・維持を報酬として計算する。エージェントの役割は、検知回避が可能になった偽学習データに摂動を加えることで、選択的な中毒攻撃を可能にすることである。

学習させた生成器とエージェントを用いて偽学習データを生成して SVM に対する中毒攻撃の検証を行い、良性サンプルの良性率を 99.1%、ターゲット以外のマルウェアファミリの悪性率を 46.1% に維持しつつ、ターゲットマル

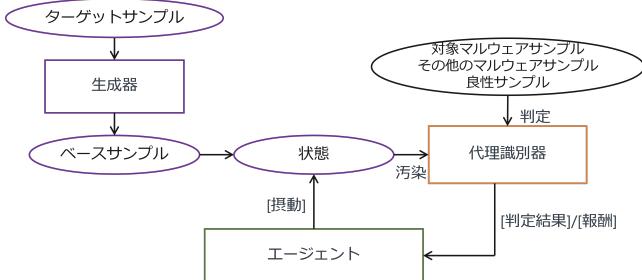


図 2 強化学習の全体像

Fig. 2 Overview of Reinforcement Learning in the previously proposed method

ウェアファミリの悪性率を 2.5%まで減少させるという一定の成果を示した。しかし、LightGBM[13]に対しては、特定のマルウェアファミリに含まれるサンプルのみの検知精度を悪化させると目的を達成することができなかった。LightGBM を含む他の機械学習モデルにも有効な偽学習データを生成するために、この手法の問題点が 2 点考えられる。1 点目は、代理識別器が、最終的な攻撃対象であるマルウェア識別器の挙動を十分に模倣できていなかった点である。MalGAN では、代理識別器は、生成器が生成する偽学習データと良性サンプルについてのみ、マルウェア識別器の挙動を学習する。しかし、マルウェア識別器は、全てのマルウェアファミリのサンプルそのものも学習するため、両者が学習する決定境界は大きく異なる。強化学習において、この代理識別器から得られる報酬を利用してエージェントを学習することは、攻撃対象のマルウェア識別器へと攻撃を転移させるために適切ではないと考えた。2 点目は、検知回避と選択的な中毒能力の獲得のために、学習プロセスが分断されている点である。この手法では、MalGAN によって学習させた生成器が検知回避に特化した偽学習データを生成し、強化学習によって学習したエージェントはその偽学習データの中毒能力を高める挙動を探索する。MalGAN で学習させた生成器は、代理識別器の決定境界を越えるための局所的な最適解を見つけることができるが、それが中毒能力の獲得という目的のための最適解の出発点として適している保証はない。そのため、学習プロセス全体が非効率になり、両方の目的を高いレベルで達成する偽学習データの生成が困難となると考えられる。本研究ではこれらの問題を解決するため、後述するアンサンブル代理識別器とバイレベル最適化の適用を試みた。

3.2 代理識別器の学習手法

本研究で提案する中毒攻撃手法の第 1 段階は、最終的な攻撃対象であるマルウェア識別器への攻撃転移を成功させるため、その挙動を模倣する代理識別器を構築することである。この代理識別器の構成や学習戦略が、生成される中毒データの質、ひいては攻撃の転移性を左右する。

単一の代理識別器を学習に利用すると、生成器はその特定のアーキテクチャのモデルの弱点や特性を突くように過剰に最適化されてしまう可能性がある。このような特化型の攻撃は、攻撃者視点でアーキテクチャやパラメータが不明であるブラックボックス識別器に対して転移しない可能性が高い。この問題に対し、より汎用的で転移性の高い攻撃を実現するため、本手法では、Multilayer perceptron (MLP) と dNDF[8]を組み合わせたアンサンブル代理識別器を採用する。アーキテクチャの異なる複数のモデルを同時に騙すことを生成器に学習させることで、特定のモデルの弱点に依存しない、より普遍的な脆弱性を探索させることを目指した。

3.1 節で述べた代理識別器の問題を解決するため、本手法では、代理識別器を生成器の学習プロセスから独立させて学習を行う。まず、代理識別器を学習するためのデータセット $D_{sc} = \{x_i\}_{i=1}^N$ を用いて、サンプル x_i に対するマルウェア識別器の判定ラベル y'_i を教師データとしたデータセット $D'_{sc} = \{(x_i, y'_i)\}_{i=1}^N$ を作成する。ここで、 y'_i は 0 が良性、1 がマルウェアを示す。本手法で用いる 2 つの代理識別器 f_{MLP} と f_{dNDF} は、データセット D'_{sc} を用いて、それぞれ独立に学習される。各モデルのパラメータをそれぞれ θ_{MLP} , θ_{dNDF} とおき、標準的な分類問題で用いられる交差エントロピー損失を \mathcal{L}_{CE} とすると、各代理識別器の学習は、以下の最適化問題を解くことに相当する。

$$\arg \min_{\theta_{MLP}} \sum_{(x_i, y'_i) \in D'_{sc}} \mathcal{L}_{CE}(f_{MLP}(x_i; \theta_{MLP}), y'_i)$$

$$\arg \min_{\theta_{dNDF}} \sum_{(x_i, y'_i) \in D'_{sc}} \mathcal{L}_{CE}(f_{dNDF}(x_i; \theta_{dNDF}), y'_i)$$

3.3 生成器の学習手法

本手法の第 2 段階は、学習したアンサンブル代理識別器を用いて、偽学習データを生成する生成器を学習することである。

3.1 節で述べた学習プロセスの分断の問題を解決するために、本手法ではバイレベル最適化を利用し、検知回避と、特定のマルウェアファミリに含まれるサンプルのみの検知精度の悪化を、单一の学習プロセス内で統合的に最適化することを目指した。機械学習/深層学習モデルに対する中毒攻撃をバイレベル最適化問題として考えることができるのは、以下の 2 つの階層的な問題として定式化できるからである。

- 上位問題：攻撃対象モデルの再学習後の性能悪化を最大化するように、偽学習データを生成するモデルのパラメータを更新する
- 下位問題：偽学習データを含むデータセットに対する損失を最小化するように、攻撃対象モデルのパラメータを更新する

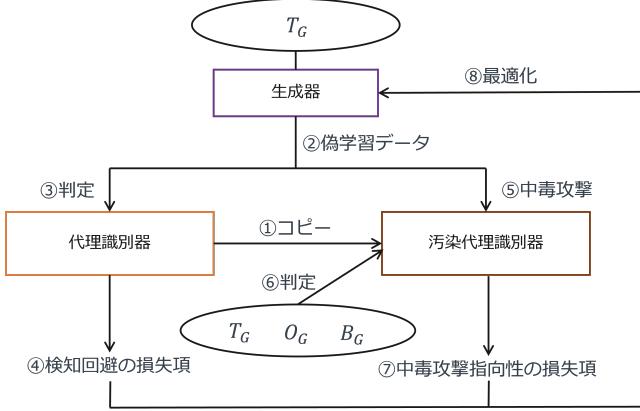


図 3 生成器の学習手法

Fig. 3 Overview of the generator training method

本手法における生成器の学習手法について説明する。生成器の学習手法の全体像を図 3 に示す。まず、生成器学習用のデータセット $D_G = \{x_i\}_{i=1}^M$ を、識別精度を悪化させるマルウェアファミリ（ターゲットファミリ）に属するサンプルの集合 T_G 、その他のマルウェアファミリに属するサンプルの集合 O_G 、そして良性サンプルの集合 B_G に分割する。生成器 G は、サンプル $x \in T_G$ とノイズ z から運動 $\delta = G(x, z; \theta_G)$ を生成し、偽学習データ $x' = x + \delta$ を生成する（図 3 ②）。 \mathcal{L}_{evade} は、生成された偽学習データ x' が現在の代理識別器に良性と誤分類されることを促す検知回避損失であり、各代理識別器の損失の平均として計算される（図 3 ③）、（図 3 ④）。

$$\mathcal{L}_{evade} = \mathbb{E}_{x \in T_G} \left[\frac{1}{2} \sum_{f \in \{f_{MLP}, f_{dNDF}\}} \mathcal{L}_{CE}(f(x'; \theta), 0) \right]$$

\mathcal{L}_{poison} は、生成された偽学習データ x' によって更新された代理識別器の性能悪化を促す中毒損失である。この損失は、 T_G に属するサンプルの検知精度の悪化を促す項と、 O, B に属するサンプルの検知精度の維持を促す項の重み付き和である（図 3 ⑤）、（図 3 ⑥）、（図 3 ⑦）。

$$\mathcal{L}_{poison} = \lambda_1 \mathcal{L}_{T_G} + \lambda_2 \mathcal{L}_{O_G} + \lambda_3 \mathcal{L}_{B_G}$$

各項は、更新後の代理識別器のパラメータ θ' を用いて以下のように計算される。

$$\begin{aligned} \mathcal{L}_{T_G} &= \mathbb{E}_{x \in T_G} \left[\frac{1}{2} \sum_{f \in \{f_{MLP}, f_{dNDF}\}} \mathcal{L}_{CE}(f(x; \theta'), 0) \right] \\ \mathcal{L}_{O_G} &= \mathbb{E}_{x \in O_G} \left[\frac{1}{2} \sum_{f \in \{f_{MLP}, f_{dNDF}\}} \mathcal{L}_{CE}(f(x; \theta'), 1) \right] \\ \mathcal{L}_{B_G} &= \mathbb{E}_{x \in B_G} \left[\frac{1}{2} \sum_{f \in \{f_{MLP}, f_{dNDF}\}} \mathcal{L}_{CE}(f(x; \theta'), 0) \right] \end{aligned}$$

以上の構成要素を用いて、生成器 G の学習は、以下の最適

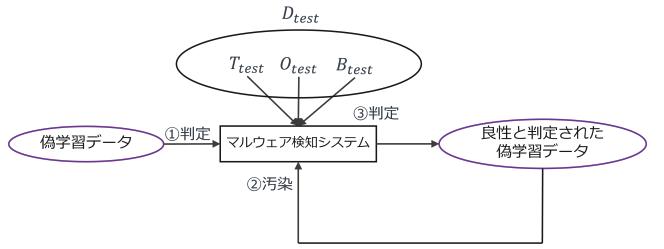


図 4 中毒攻撃の評価手順

Fig. 4 Overview of the evaluation procedure of the proposed poisoning attack method

化問題として定式化される（図 3 ⑧）。

$$\arg \min_{\theta_G} (\mathcal{L}_{evade} + \mathcal{L}_{poison})$$

$$\text{subject to } \theta_{MLP}^* = \arg \min_{\theta_{MLP}} \mathcal{L}_{CE}(f_{MLP}(x'; \theta_{MLP}), 0)$$

$$\theta_{dNDF}^* = \arg \min_{\theta_{dNDF}} \mathcal{L}_{CE}(f_{dNDF}(x'; \theta_{dNDF}), 0)$$

4. 評価

本節では、前節で提案した手法によって学習した生成器を用いた中毒攻撃の有効性を評価した結果を示す。提案手法の評価は、代理識別器の学習の経緯の確認、マルウェア識別器に対する代理識別器の忠実度の確認、生成器の学習の推進の確認、マルウェア識別器への中毒攻撃の評価、という順番で行った。

4.1 評価手順

評価手順の全体像を図 4 に示す。まず、 T_G に属するサンプルをベースとして、3.2 節で示した学習過程において検証損失が最小となった時点の生成器を用いて N 個の偽学習データを生成する。生成した N 個の偽学習データを攻撃対象の LightGBM ベースのマルウェア識別器に入力し、判定結果を取得する（図 4 ①）。この中から、良性と判定されたデータのみを、実際に使用する偽学習データとして選別する。選別した偽学習データに良性ラベルを付与し、マルウェア識別器の学習データに追加して、マルウェア識別器を再学習する（図 4 ②）。この汚染後のマルウェア識別器の性能を、これまでの学習プロセスで一度も使用されていない独立したデータセット D_{test} を用いて評価する。具体的には、 D_{test} をターゲットファミリに属するサンプルの集合 T_{test} 、その他のマルウェアファミリに属するサンプルの集合 O_{test} 、そして良性サンプルの集合 B_{test} に分割し、各集合に属するサンプルに対する識別精度を確認する（図 4 ③）。

4.2 評価結果と考察

代理識別器の学習の経過を図 5、図 6 に示す。図 5、図 6 の横軸は、エポック数（学習データ全体を学習した回数）であり、縦軸は損失回数の値である。図 5 の MLP 代理

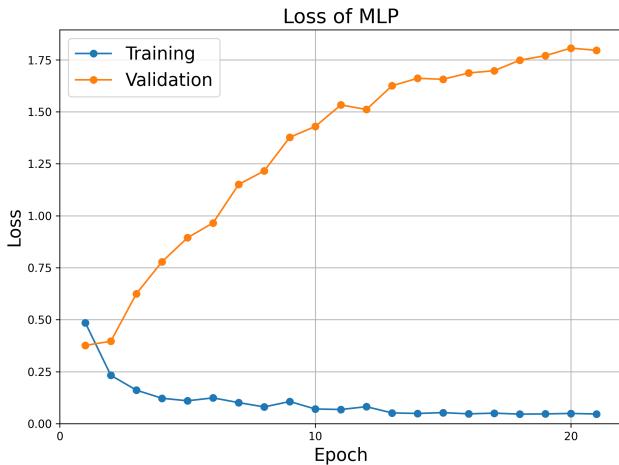


図 5 MLP 代理識別器の学習損失・検証損失の評価

Fig. 5 Evaluation of MLP surrogate classifier training loss and validation loss

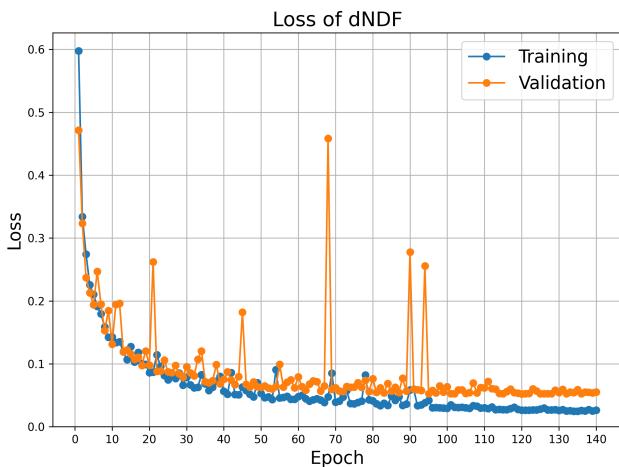


図 6 dNDF 代理識別器の学習損失・検証損失の評価

Fig. 6 Evaluation of dNDF surrogate classifier training loss and validation loss

識別器は、訓練損失がエポックを経るごとに減少しているのに対し、検証損失は最初のエポック以降、上昇し続けており、過学習が起こっていると考えられる。一方、図 6 の dNDF 代理識別器は、MLP と比較して訓練損失と検証損失の乖離が小さく、汎化性能を維持したまま学習が進行していると考えられる。学習後、テストデータを用いて各代理識別器とマルウェア識別器との判定一致率（忠実度）を測定した結果、MLP は 86.31%、dNDF は 98.85% となった。この結果は、学習曲線の傾向を裏付けるものであると考えられる。MLP 代理識別器は未知のデータに対する汎用的な模倣性能の向上が限定期であった一方で、dNDF 代理識別器は安定した学習によって高い汎化性能を獲得し、マルウェア識別器の挙動を忠実に模倣することに成功したと考えられる。dNDF 代理識別器が MLP 代理識別器と比較して高い忠実度を達成した要因は、LightGBM と dNDF とのアーキテクチャの類似性であると考えられる。LightGBM

は決定木をベースとしたアンサンブルモデルであり、その決定境界は、特微量空間における階層的かつ軸に平行な分割の組み合わせによって形成される。dNDF は微分可能な決定木そのものをアーキテクチャに組み込んでいるため、MLP と比較すると、決定境界が LightGBM に類似すると考えられる。

次に、生成器の学習の経過を図 7 に示す。図 5、図 6 と同様に、図 7 の各グラフの横軸はエポック数であり、縦軸は損失関数の値である。 \mathcal{L}_{evade} と \mathcal{L}_B は、学習の初期に低い値へと移行し、その後も安定した推移が見られる。この 2 つの値が低いままで推移したことからは、偽学習データの回避性能と良性サンプルの識別精度に大きな影響を与えるに、ターゲットファミリのサンプルの識別精度の悪化が実現可能である可能性が示される。一方で、 \mathcal{L}_T と \mathcal{L}_O の推移からは、4 つの目的を全て満たす最適解を見つけるように生成器を学習させることができなかったことが示唆される。 \mathcal{L}_T は、急減するエポックもありながら、安定して低い値を維持することはなかった。そして、 \mathcal{L}_O は、 \mathcal{L}_T が最も急増したエポックに一度だけ急減したが、学習全体を通じて高い値を維持している。これは、生成器がターゲットファミリのサンプルの識別精度悪化に有効な振動を生成する状態へと一時的に収束するものの、それが他のマルウェアファミリのサンプルの識別精度の悪化を伴い、目的全体の最適解として維持されなかつた結果であると考えられる。

最後に、汚染前のマルウェア識別器の判定結果と汚染後のマルウェア識別器の判定結果を表 1 に示す。なお、表 1 の通過率は、マルウェア識別器に良性と判定された偽学習データの割合を示す。表 1 からは、まず、生成された偽学習データの通過率が 99% を超えており、生成器はマルウェア識別器に対する回避性能を学習の過程で獲得することができたと考えられる。一方で、表 1 は、図 7 から考察した複数の目的を同時に最適化する困難さ以上の課題を示している。4.1 節で記載したとおり、マルウェア識別器の汚染に用いた生成器は学習過程で検証損失が最小であったエポックのモデルである。図 7 に示すとおり、このエポックでは \mathcal{L}_T も急減しており、この時点の生成器を用いたマルウェア識別器の汚染によって、ターゲットファミリのサンプルの識別精度が減少することが期待された。しかし、実際は、偽学習データの数を 200,000 まで増加させても、ターゲットファミリのサンプルの識別精度は約 0.1% しか低下しなかった。また、 \mathcal{L}_O は該当エポックにおいて高い値であったため、その他のマルウェアファミリのサンプルの識別精度も同様に悪化されることが予想されたが、実際にはほぼ変化が見られず、総じて攻撃が転移していないことが確認できる。この原因は、Demontis らの研究 [6] で示された、代理モデルと標的モデルとの勾配の方向の類似が達成されていなかったことにあると考えられる。2.1 節で引用した研究を含む、攻撃の転移性に関する多くの研究は、モデル

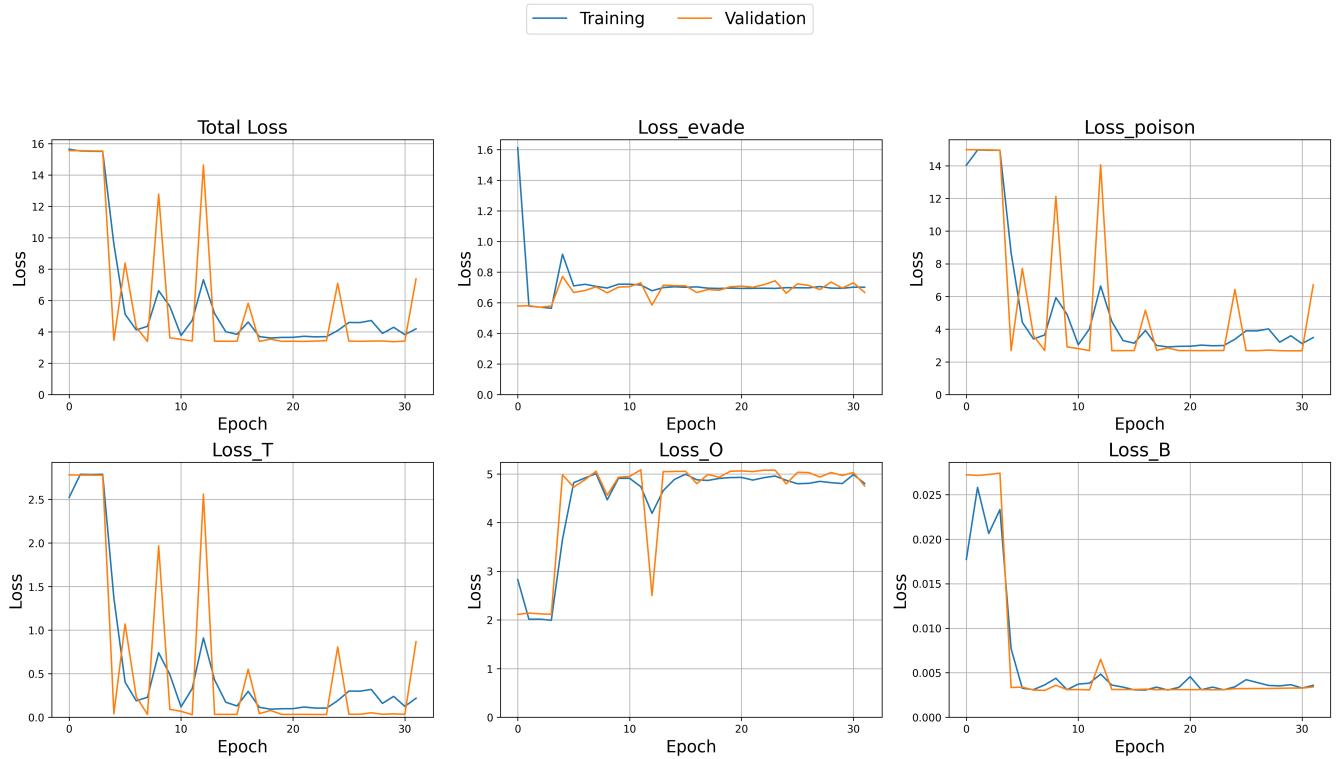


図 7 生成器の学習損失・検証損失の評価

Fig. 7 Evaluation of generator training loss and validation loss

表 1 マルウェア識別器に対する中毒攻撃の指向性の評価

Table 1 Evaluation of the directionality of the poisoning attack against the malware classifier

N	通過率 (%)	T_{test} の悪性率 (%)	O_{test} の悪性率 (%)	B_{test} の良性率 (%)
0	-	100.0	99.38	99.67
10,000	100.0	99.88	99.33	99.65
50,000	99.52	99.88	99.35	99.70
100,000	99.52	99.88	99.35	99.65
200,000	99.52	99.88	99.34	99.61

全体が单一の微分可能な関数として表現されるニューラルネットワーク間の現象を分析したものである一方で、本研究は、ニューラルネットワーク代理モデルから LightGBM へと攻撃を転移させようと試みた。dNDF は決定木の分岐構造を模倣した優れたモデルであるが、あくまでニューラルネットワークである。MLP や dNDF が学習する決定境界は本質的に滑らかな形状を持つのに対し、LightGBM が学習する決定境界は、多数の決定木の組み合わせによる、複雑で不連続な形状を持つ。本研究の提案手法は、Ilyas らが示したデータセットレベルの普遍的な脆弱性を突く手法 [5] とは対照的に、モデルの内部的な学習メカニズムそのものを標的としており、より局所的でモデル依存であると分類できる。そのため、この決定境界の幾何学特性の不一致は、攻撃の転移を実現するためには致命的であったと考えられる。

5. おわりに

本研究では、LightGBM ベースのマルウェア識別器に対する新たな標的型中毒攻撃手法の提案・評価を行った。以前に考案・評価を行った GAN と強化学習を併用した手法が、生成器に脆弱な代理識別器の利用や分断された学習プロセスなどの課題を抱えていたためである。具体的には、攻撃の転移性を高めるため、LightGBM とアーキテクチャが類似する dNDF を含むアンサンブル代理識別器を、生成器とは独立して学習させた。そして、学習したアンサンブル代理識別器を用いて、検知回避と中毒効果の二つの目的を統合された単一のプロセスで最適化するために、バイレベル最適化の枠組みに基づき、偽学習データを生成する生成器を学習させた。評価実験の結果、提案手法では、ターゲットファミリのサンプルの識別精度悪化と、その他のマルウェアファミリのサンプルの識別精度の維持を両立

することが困難であることが示された。また、提案手法のような局所的でモデル依存な中毒攻撃の転移を実現する上では、代理識別器と攻撃対象の識別器との学習原理の違いが根本的な問題になるという知見を得た。LightGBM マルウェア識別器への転移を実現するためには、微分不可能ではあるが、より決定境界が類似するランダムフォレストや LightGBM 自体を代理モデルとして利用し、摂動を生成するモデルを強化学習によって学習する手法が有効である可能性が考えられる。

謝辞 本研究は JSPS 科研費 23K28086, 24K14959 の助成を受けたものである。

参考文献

- [1] 木場翔太, 長谷川皓一, 山口由紀子, 嶋田創, “高次元特微量と連続行動空間対応強化学習による特定マルウェア通過用偽学習データ生成,” 電子情報通信学会技術報告, vol.124, no.422, ICSS2024-115, pp.359–366, March 2025.
- [2] Weiwei Hu and Ying Tan, “Generating Adversarial Malware Examples for Black-Box Attacks Based on GAN,” arXiv preprint arXiv:1702.05983 [cs.LG], February 2017.
- [3] Octavian Suciu, Radu Marginean, Yigitcan Kaya, Hal Daume III and Tudor Dumitras, “When Does Machine Learning FAIL? Generalized Transferability for Evasion and Poisoning Attacks,” In Proceedings of the 27th USENIX Security Symposium (USENIX Security ’18), pp.1299–1316, August 2018.
- [4] Yanpei Liu, Xinyun Chen, Chang Liu and Dawn Song, “Delving into Transferable Adversarial Examples and Black-box Attacks,” In Proceedings of the 5th International Conference on Learning Representations (ICLR 2017), pp.2235–2248, April 2017.
- [5] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Mądry, “Adversarial Examples are not Bugs, they are Features,” In Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), pp.125–136, December 2019.
- [6] Ambra Demontis, Marco Melis, Maura Pintor, Matthew Jagielski, Battista Biggio, Alina Oprea, Cristina Nită-Rotaru, Fabio Roli, “Why Do Adversarial Attacks Transfer? Explaining Transferability of Evasion and Poisoning Attacks,” In Proceedings of the 28th USENIX Security Symposium (USENIX Security ’19), pp.321–338, August 2019.
- [7] Ozan Irsøy, Olcay Taner Yıldız, and Ethem Alpaydin, “Soft decision trees,” In Proceedings of the 21st International Conference on Pattern Recognition (ICPR 2012), pp.1819–1822, November 2012.
- [8] Peter Kotschieder, Madalina Fiterau, Antonio Criminisi, and Samuel Rota Bulò, “Deep Neural Decision Forests,” In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV 2015), pp.1467–1475, December 2015.
- [9] Benoît Colson, Patrice Marcotte, and Gilles Savard, “An overview of bilevel optimization,” Ann Oper Res, vol.153, pp.235–256, September 2007. <https://doi.org/10.1007/s10479-007-0176-2>
- [10] Chelsea Finn, Pieter Abbeel, and Sergey Levine, “Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks,” In Proceedings of the 34th International Conference on Machine Learning (ICML 2017), pp.1126–1135, August 2017.
- [11] Sachin Ravi, and Hugo Larochelle, “Optimization as a Model for Few-Shot Learning,” In Proceedings of the 5th International Conference on Learning Representations (ICLR 2017), pp.2936–2946, April 2017.
- [12] W. Ronny Huang, Jonas Geiping, Liam Fowl, Gavin Taylor, and Tom Goldstein, “MetaPoison: Practical General-purpose Clean-label Data Poisoning,” In Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020), pp.12080–12091, December 2020.
- [13] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu, “LightGBM: A Highly Efficient Gradient Boosting Decision Tree,” In Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS 2017), pp.3146–3154, December 2017.