グラフ学習を用いたハードウェアトロイ識別に対するトロイクラスタ法を用いた補正処理

吉見 尚[1,a)]　池上 裕香[1]　戸川 望[1]

概要：現在，日常生活に IoT 機器は広く浸透しており，IoT 機器に使用する IC の需要は年々増大している．IC の需要拡大に伴い，IC を安価で作成するために，IC の設計・製造段階で外部委託が行われることがあり，サードパーティがサプライチェーンに加わることが多い．その際，悪意のあるサードパーティによってハードウェアトロイ (HT) が回路に挿入されるリスクが高まっていることが報告されている．HT を設計段階で検出する手法の一つとして，グラフ学習による HT 識別手法が提案されており，高い有効性を示している．本稿では，HT 識別精度のさらなる向上のために，トロイクラスタ法による補正処理手法を提案する．提案手法では，トロイと識別されているノードをもとにトロイクラスタを作成し，トロイクラスタに含まれないトロイと識別されたノードをノーマルノードに補正する．提案手法により，GAT (Graph Attention Network) を使用した HT 識別に対して補正処理を行うことで，F-score 0.8848 を達成した．また，他補正手法と組み合わせることで，F-score が 0.8996，TPR が 87.35%，TNR が 99.87%，Accuracy が 99.84%，Precision が 97.78%を達成した．

キーワード：ハードウェアトロイ，機械学習，グラフ学習，IoT セキュリティ

A Correction Method Using Trojan Clustering for Graph-Learning-Based Hardware-Trojan Detection

Sho Yoshimi[1,a)]　Yuka Ikegami[1]　Nozomu Togawa[1]

Abstract: In this paper, we propose a correction method using Trojan clustering for graph-learning-based hardware-Trojan detection. By applying the proposed method to correct the HT detection results obtained by GAT, the detection accuracy of HTs was improved, demonstrating the effectiveness of the proposed method.

Keywords: hardware-Trojan, machine learning, graph-learning, IoT security

1. はじめに

近年，IoT 機器は日常生活に広く浸透している．IoT 機器には IC が使用され，IoT 機器の需要の増大とともに IC の需要は増加しており，IC を安価で作成するために，IC の設計・製造段階を外部委託する過程で，サプライチェーンにサードパーティが加わることが多い．その際，悪意のあるサードパーティによって，ハードウェアトロイ (HT) が回路に挿入されるリスクが高まっていることが報告されている [1]．2007 年 9 月のシリアによるイスラエル爆撃では，防空システムの監視レーダーが HT によって停止された疑いが報告されている [2]．

HT とは，IC の機能の改ざん・劣化や IC 自体の破壊，暗号化された情報の漏洩など，目的に合わせて様々な動作を行う悪意のある回路を指す [3]．HT は，小規模かつ特定の条件を満たしたときにはじめて悪意のある動作を引き起こすため，簡単に検出することはできない．

HT を検出する手法の 1 つとして，IC の設計段階におけ

[1] 早稲田大学大学院基幹理工学研究科情報理工・情報通信専攻
Dept. Computer Science and Communications Engineering, Waseda University
[a)] sho.yoshimi@togawa.cs.waseda.ac.jp

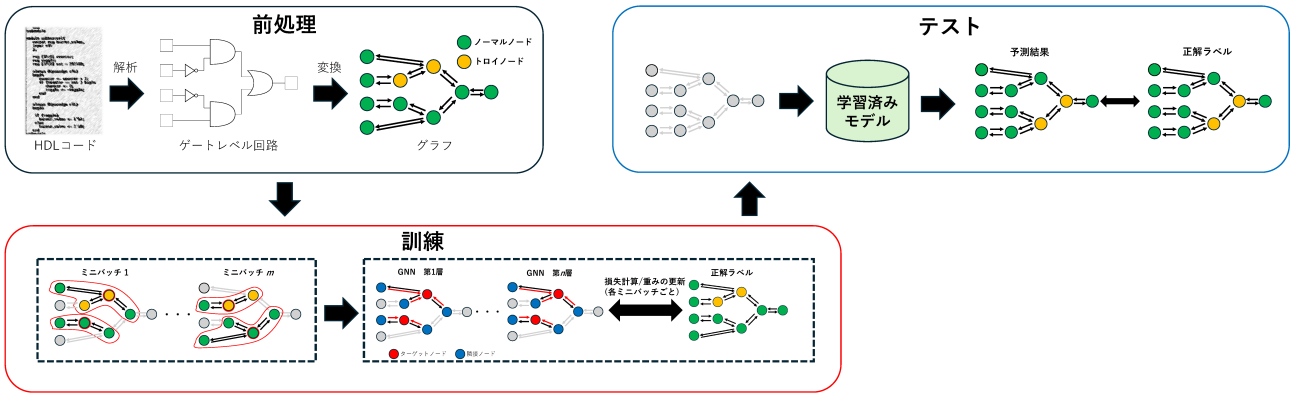

図 1 グラフ学習を用いた HT 識別手法.

る機械学習を用いた HT 検出手法が研究されている [4], [5]. 特に，グラフ学習による HT 識別手法 [6], [7] は，高い有効性を示している．設計段階での HT 識別では，ゲートレベルネットリストを対象に HT 識別を行う．グラフ学習による HT 識別においては，ゲートをノード，ネットをエッジとするグラフでネットリストを表現して学習し，各ノードがノーマルノードかトロイノードのどちらであるかを識別する．

HT を識別した結果を補正する手法として，[8] ではトロイクラスタ法を用いて各ネットがトロイかどうかを識別した結果を補正する手法を提案している．また，グラフ学習を用いて各ゲートがトロイであるかどうかを識別した結果に対して，識別結果を補正する手法として，我々は複数の学習済みモデルを用いた補正手法 [9] を提案している．本手法は補正対象となるノードを複数の学習済みモデルの多数決によって再識別し，評価実験より識別精度の向上が確認できている．

本稿では，別の補正処理手法として，設計段階におけるグラフ学習モデルの 1 つである GAT (Graph Attention Network) [10] による HT 識別に対して，トロイクラスタ法を用いた補正処理手法を提案する．提案手法では，トロイと識別されているノードをもとにトロイクラスタを作成し，トロイクラスタに含まれないトロイと識別されたノードをノーマルノードに補正することで，FP (False Positive) を TN (True Negative) に修正し，さらなる HT 識別精度の向上を目指す．

評価実験の結果，GAT による HT 識別に対してトロイクラスタ法を用いた補正処理手法を適用することで，識別精度の向上が確認できた．また，複数の学習済みモデルを用いた補正手法 [9] とトロイクラスタ法を用いた補正処理手法を組み合わせて補正することで，F-score が 0.8996，TPR が 87.35%，TNR が 99.87%，Accuracy が 99.84%，Precision が 97.78%を達成した．

本稿の構成を以下に示す．2 章で，グラフ学習を用いた HT 識別手法を説明する．3 章で，提案手法であるトロイクラスタ法を用いた HT 識別手法を説明する．4 章で，評価実験を行う．5 章で，本稿をまとめる．

表 1 実験に使用するネットリストのデータ.

#	ネットリスト名	ノーマルゲート数	トロイゲート数
1	B19-T100	63170	83
2	B19-T200	63170	83
3	RS232-T1000	289	13
4	RS232-T1100	293	11
5	RS232-T1200	296	10
6	RS232-T1300	290	9
7	RS232-T1400	290	12
8	RS232-T1500	291	13
9	RS232-T1600	293	10
10	s15850-T100	2397	27
11	s35932-T100	5967	15
12	s35932-T200	5962	15
13	s35932-T300	5965	36
14	s38417-T100	5656	12
15	s38417-T200	5656	15
16	s38417-T300	5688	15
17	s38584-T100	7064	9
18	s38584-T200	7064	83
19	s38584-T300	7064	731
20	wb-conmax_100	23194	15
21	B19_free	63253	0
22	RS232_free	284	0
23	s15850_free	2396	0
24	s35932_free	5965	0
25	s38417_free	5656	0
26	s38584_free	7064	0
27	wb-conmax_free	23194	0

2. グラフ学習を用いた HT 識別

本章では，グラフ学習による HT 識別を説明する．図 1 に，グラフ学習を用いた HT 識別手法の一例を概要として示す．グラフ学習を用いた HT 識別では，前処理段階，訓練段階，テスト段階の 3 段階に分けることができる．

前処理段階では，グラフ学習を用いた HT 識別は IC の設計段階における HT 検出を目的としているため，ゲートレベルネットリストが識別の対象となる．したがって，ゲートレベルネットリストをグラフ学習で学習できるように，ゲートをノード，ネットをエッジとするグラフに変換

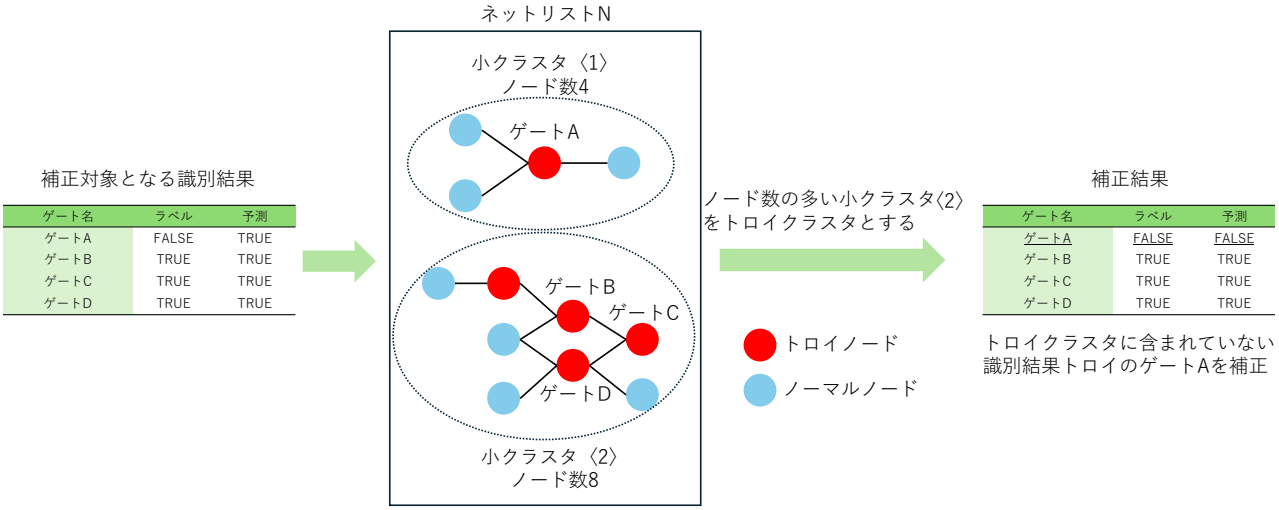

図 2 トロイクラスタ法を使用した補正処理手法.

する．HT を検出したいゲートレベルネットリストを変換したグラフをテストデータとし，それとは別に訓練に使用する複数のゲートレベルネットリストを訓練データとしてグラフに変換する．

訓練段階では，訓練データを用いてグラフ学習モデルの学習を行う．本稿では，ノーマルノードとトロイノードの不均衡を解消するために，ミニバッチによるサンプリングを行い，グラフ学習モデルとして GAT を使用する．

テスト段階では，訓練段階で作成した学習済みモデルを使用してテストデータのグラフにおける各ノードがトロイであるかどうかを識別する．本稿では，テスト段階で得られた HT 識別結果を初期 HT 識別結果とし，補正対象とする．

3. トロイクラスタ法を用いた補正処理手法

本章では，提案手法であるトロイクラスタ法を用いた補正処理手法を説明する．図 2 に提案手法の概要を示す．提案手法の流れを以下に示す．

第 1 段階 (初期 HT 識別)：

GAT を使用して，訓練データの集合 L に含まれるデータセットで学習し，テストデータ T に含まれるデータセットに対して HT 識別を行う．

第 2 段階 (HT 補正処理)：

(1) 初期 HT 識別でトロイと識別されたノードに対して，近傍 2 以内のノードとともに小クラスタを形成する．

(2) 小クラスタごとに同じノードが存在すれば，小クラスタを合体する．

(3) 最もノードを多く含んでいる小クラスタをトロイクラスタとする．トロイクラスタに含まれていない，初期 HT 識別でトロイと識別されたノードをノーマルノードに補正する．トロイクラスタに含まれていない初期 HT 識別でトロイと識別されたノードは，孤立していると判断できるため，FP であると判断し補正する．

表 2 特徴ベクトルの構成.

#	説明
1	プライマリ入力までの距離
2	プライマリ出力までの距離
3	入力側の信号線数
4	出力側の信号線数
5	定数
6	フリップフロップ
7	マルチプレクサ
8	インバータ，バッファ
9	加算器
10	論理ゲート (0 出力確率)
11	論理ゲート (1 出力確率)

第 1 段階 (初期 HT 識別) では，訓練データを用いて GAT モデルを学習する．そして学習済みの GAT モデルによって，テストデータの HT 識別を行う．テストデータに含まれる各ノードが，トロイノードかノーマルノードに初期識別される．

第 2 段階 (HT 補正処理) は，以下の考えに基づく．一般に HT は検知されないようにするために，回路内に局所的に挿入されている可能性が高い．したがって，初期識別でトロイノードと識別されたもののうち，孤立しているものは FP である可能性が高い．そのため，トロイと識別されたノードごとに小クラスタを作成し，ノードを共有する小クラスタを結合することで，局所的に存在しているトロイノードをまとめてトロイクラスタとすることができる．反対に，初期 HT 識別においてトロイクラスタに含まれないトロイノードは孤立していると分かるため，ノーマルノードに補正する．

例 1. 図 2 において，あるネットリスト N を初期識別した

表 3 初期 HT 識別結果.

ネットリスト名	TN	FP	FN	TP	F-score	TPR	TNR	Accuracy	Precision
B19-T100	63167	3	3	80	0.9639	0.9630	1.000	0.999	0.9639
B19-T200	63168	2	3	80	0.9697	0.9639	1.000	0.9999	0.9756
RS232-T1000	288	1	0	13	0.9630	1.000	0.9965	0.9967	0.9286
RS232-T1100	289	4	0	11	0.8800	1.000	0.9898	0.9901	0.7857
RS232-T1200	294	2	0	10	0.9091	1.000	0.9932	0.9935	0.8333
RS232-T1300	288	2	0	9	0.9000	1.000	0.9931	0.9933	0.8182
RS232-T1400	288	2	0	12	0.9231	1.000	0.9931	0.9934	0.8571
RS232-T1500	289	2	0	13	0.9286	1.000	0.9931	0.9934	0.8667
RS232-T1600	291	2	1	9	0.8571	0.9000	0.9932	0.9988	0.9286
s15850-T100	2395	2	1	26	0.9455	0.9630	0.9992	0.9988	0.9286
s35932-T100	5967	0	2	13	0.9286	0.8667	1.000	0.9997	1.000
s35932-T200	5961	0	5	11	0.8148	0.6875	1.000	0.9992	1.000
s35932-T300	5973	0	6	22	0.8800	0.7857	1.000	0.9990	1.000
s38417-T100	5655	1	3	9	0.8182	0.7500	0.9998	0.9993	0.9000
s38417-T200	5655	1	0	15	0.9677	1.000	0.9998	0.9998	0.9375
s38417-T300	5676	11	0	16	0.7443	1.000	0.9981	0.9981	0.5926
s38584-T100	7063	1	9	0	0.0000	0.0000	0.9999	0.9986	0.0000
s38584-T200	7064	0	0	83	1.000	1.000	1.000	1.000	1.000
s38584-T300	7063	1	0	731	0.9993	1.000	0.9999	0.9999	0.9986
wb-conmax-T100	23194	0	4	11	0.8462	0.7333	1.000	0.9998	1.000
B19_free	63169	1	0	0	-	-	1.000	1.000	-
RS232_free	254	25	0	0	-	-	0.9677	0.9677	-
s15850_free	2394	2	0	0	-	-	1.000	1.000	-
s35932_free	5965	0	0	0	-	-	1.000	1.000	-
s38417_free	5655	1	0	0	-	-	1.000	1.000	-
s38584_free	7062	1	0	0	-	-	1.000	1.000	-
wb-conmax_free	23194	0	0	0	-	-	1.000	1.000	-
平均	-	-	-	-	0.8602	0.8807	0.9946	0.9944	0.8576

結果，青色のノードはノーマルノード，赤色のノードはトロイノードと分類されたとする．実際には，ゲート A はノーマルノードであるが，トロイノードと誤識別されている．ゲート B，ゲート C，ゲート D は正しくトロイノードと識別されている．

4. 評価実験

本章では，提案手法の評価実験を行う．4.1 節で，実験条件を説明する．4.2 節で，GAT を使用した初期 HT 識別を説明する．4.3 節で，提案手法であるトロイクラスタ法を用いた補正処理手法の評価を行う．

4.1 実験条件

本節では，実験条件として計算機環境と，評価指標，実験に使用するデータセットを説明する．

使用する計算機環境は，メモリ 1.5TB で CPU に Xeon Gold 6230R，GPU に Tesla を搭載するコンピュータ，Python 3.8.10，torch 2.4.1，torch-geometric 2.6.1 である．

HT 識別は 2 値分類問題であるため，TN (True Negative，正しくノーマルゲートとして分類されたノーマルゲートの数)，TP (True Positive，正しくトロイゲートとして分類されたトロイゲートの数)，FN (False Negative，誤ってノーマルゲートとして分類されたトロイゲートの数)，FP (False Positive，誤ってトロイゲートとして分類されたノーマルゲートの数) に識別結果を分類できる．また，正しくトロイゲートとして分類されたトロイゲートの割合を示す $\mathrm{TPR} = \mathrm{TP}/(\mathrm{TP}+\mathrm{FN})$，正しくノーマルゲートとして分類されたノーマルゲートの割合を示す $\mathrm{TNR} = \mathrm{TN}/(\mathrm{TN}+\mathrm{FP})$，トロイゲートと判定されたノードのうち，真にトロイゲートであるノードの割合を示す $\mathrm{Precision} = \mathrm{TP}/(\mathrm{TP}+\mathrm{FP})$，TPR と precision の調和平均である $\text{F-score} = 2/(1/\mathrm{TPR}+1/\mathrm{Precision})$，すべてのゲートのうち正しく分類されたゲートの割合を示す $\mathrm{Accuracy} = (\mathrm{TP}+\mathrm{TN})/(\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN})$ を計算できる．上記 9 つの指標に基づき，HT 識別精度を評価する．提案手法では，FP に識別されたゲートを補正することを目的としているため，FP と TN の数の増減によって補正効果を評価する．また，F-score を総合指標とし，全体の HT 識別精度を評価する．

表 1 に実験に使用するゲートレベルネットリストを示す．データセットは，Trust-Hub [11] に公開されている 27 種類のネットリストであり，20 種類の HT 回路と，7 種類のトロイが含まれないフリー回路で構成される．27 種類のネットリストに関して 1 個抜き交差検証を行う．つまりある 1 つのネットリスト N に注目したとき，N を除く，26 種類のネットリストを用いて GAT モデルを学習し，その後，学習済みの GAT モデルを用いてネットリスト N をテストデータとして，N に含まれるノードをトロイノードとノーマルノードに分類する．

4.2 GAT を使用した初期 HT 識別

本節では，補正前の初期 HT 識別として，表 1 に示す 27

表 4 GAT による HT 識別に対するトロイクラスタ法を用いた補正結果.

ネットリスト名	FP に対する補正数	TN に対する補正数	TP に対する補正数	FN に対する補正数	TN	FP	FN	TP	F-score	TPR	TNR	Accuracy	Precision
B19-T100	3	0	0	0	63170	0	3	80	0.9816	0.9639	1.0000	1.0000	1.0000
B19-T200	2	0	0	0	63170	0	3	80	0.9816	0.9639	1.0000	1.0000	1.0000
RS232-T1000	1	0	0	0	289	0	0	13	1.0000	1.0000	1.0000	1.0000	1.0000
RS232-T1100	4	0	0	0	293	0	0	11	1.0000	1.0000	1.0000	1.0000	1.0000
RS232-T1200	2	0	0	0	296	0	0	10	1.0000	1.0000	1.0000	1.0000	1.0000
RS232-T1300	2	0	0	0	290	0	0	9	1.0000	1.0000	1.0000	1.0000	1.0000
RS232-T1400	2	0	0	0	290	0	0	12	1.0000	1.0000	1.0000	1.0000	1.0000
RS232-T1500	2	0	0	0	291	0	0	13	1.0000	1.0000	1.0000	1.0000	1.0000
RS232-T1600	2	0	0	0	293	0	1	9	0.9474	0.9000	1.0000	0.9967	1.0000
s15850-T100	0	0	0	0	2395	2	1	26	0.9455	0.9630	0.9992	0.9988	0.9286
s35932-T100	0	0	1	0	5967	0	3	12	0.8889	0.8000	1.0000	0.9995	1.0000
s35932-T200	0	0	1	0	5962	0	5	10	0.8000	0.6667	1.0000	0.9992	1.0000
s35932-T300	0	0	1	0	5973	2	7	19	0.8085	0.7308	0.9997	0.9985	0.9048
s38417-T100	1	0	1	0	5656	0	4	8	0.8000	0.6667	1.0000	0.9993	1.0000
s38417-T200	1	0	1	0	5656	0	1	14	0.9655	0.9333	1.0000	0.9998	1.0000
s38417-T300	1	0	1	0	5677	10	1	15	0.7317	0.9375	0.9982	0.9981	0.6000
s38584-T100	1	0	0	0	7064	0	9	0	0.0000	0.0000	1.0000	0.9987	0.0000
s38584-T200	0	0	0	0	7064	0	0	83	1.0000	1.0000	1.0000	1.0000	1.0000
s38584-T300	1	0	0	0	7064	0	0	731	1.0000	1.0000	1.0000	1.0000	1.0000
wb_conmax-T100	0	0	0	0	23194	0	4	11	0.8462	0.7333	1.0000	0.9998	1.0000
B19_free	0	0	0	0	63169	1	0	0	-	-	1.0000	1.0000	-
RS232_free	16	0	0	0	270	9	0	0	-	-	0.9677	0.9677	-
s15850_free	0	0	0	0	2394	2	0	0	-	-	0.9992	0.9992	-
s35932_free	0	0	0	0	5965	0	0	0	-	-	1.0000	1.0000	-
s38417_free	0	0	0	0	5655	1	0	0	-	-	0.9998	0.9998	-
s38584_free	0	0	0	0	7062	1	0	0	-	-	0.9999	0.9999	-
wb_conmax_free	0	0	0	0	23194	0	0	0	-	-	1.0000	1.0000	-
平均	-	-	-	-	-	-	-	-	0.8848	0.8629	0.9987	0.9983	0.9217

種類のデータセットに対して GAT を使用した HT 識別 [7] を行う.

GAT による HT 識別では，ネットリストを表現するグラフに対して，表 2 に示すように，文献 [12] に基づいた 11 種類の特徴ベクトルを付与する．GAT による HT 識別で使用するパラメータは，文献 [7] に基づいて，サンプリングのためのミニバッチ数が 5，GAT 層を 2 層，マルチヘッドのヘッド数が 5，モデルの最適化アルゴリズムに Adam を使用し，学習率を 0.01，エポック数を 1000，損失関数にバイナリクロスエントロピーを使用する．27 種類のデータセットに対して，識別対象のデータセットを除く 26 種類のデータセットを学習データとして HT 識別を行う.

表 3 に初期 HT 識別結果を示す．初期 HT 識別結果に対して， 3 章で提案したトロイクラスタ法に基づく補正処理を適用し，補正後の HT 識別精度を評価する.

4.3 トロイクラスタ法を用いた補正処理の評価

表 4 に 4.2 節で得られた初期 HT 識別結果に対して，トロイクラスタ法を用いた補正処理を適用したときの，HT 識別結果を示す．11 個のデータセットで TP を誤補正することなく，すべての FP を補正することができているため，FP を補正するのにトロイクラスタ法を用いた HT 補正処理は有効であると言える.

次に，トロイクラスタ法を用いた補正処理手法と，他補正手法を組み合わせたときの評価を行う．グラフ学習を用いた HT 識別に対する補正処理手法として複数の学習済みモデルを用いた補正処理手法 [9] が挙げられる．本手法では，複数の学習済みモデルを使用して，補正対象となる識別においてトロイノードと識別されたノードと近傍 2 以内のノードを各モデルの多数決によって再識別することで補正処理を行う．初期 HT 識別に対して，はじめに複数の学習済みモデルを用いた補正処理を行い，次にトロイクラスタ法を用いた補正処理を適用する．上記のように，複数の補正手法を段階的に適用することで，それぞれの手法が持つ補正特性を相補的に活用することが可能となる．具体的には，複数モデルによる多数決補正によって識別の安定性を高めた後，トロイクラスタ法により局所的な構造情報を利用した補正を行うことで，異なる観点からの補正を実現し，より高精度な HT 識別が期待できる.

表 5 に，初期 HT 識別結果に対して，複数の学習済みモデルを用いた補正処理を行い，次にトロイクラスタ法を用いた補正処理を行ったときの，各データセットごとの HT 識別結果を示す．24 個のデータセットですべてのノーマルノードを正しく識別できており，ノーマルノードに対する補正の有効性を示している．ノーマルノードの誤補正が減ったことにより，Precision が 85.76%から 97.78%に向上している.

表 6 に，初期 HT 識別結果と複数の学習済みモデルを使用した補正処理 [9]，提案したトロイクラスタ法による補正処理，2 つの補正処理手法を組み合わせた補正処理の 4 つの HT 識別精度を示す．トロイクラスタ法を用いた補正処理によって，初期 HT 識別結果から TNR が 0.14，Accuracy

表 5 2 つの補正処理手法を組み合わせた補正を行ったときの HT 識別結果.

ネットリスト名	FP に対する補正数	TN に対する補正数	TP に対する補正数	FN に対する補正数	TN	FP	FN	TP	F-score	TPR	TNR	Accuracy	Precision
B19-T100	3	0	0	2	63170	0	1	82	0.9939	0.9880	1.0000	1.0000	1.0000
B19-T200	2	0	0	3	63170	0	0	83	1.0000	1.0000	1.0000	1.0000	1.0000
RS232-T1000	1	0	0	0	289	0	0	13	1.0000	1.0000	1.0000	1.0000	1.0000
RS232-T1100	4	0	0	0	293	0	0	11	1.0000	1.0000	1.0000	1.0000	1.0000
RS232-T1200	2	0	0	0	296	0	0	10	1.0000	1.0000	1.0000	1.0000	1.0000
RS232-T1300	2	0	0	0	290	0	0	9	1.0000	1.0000	1.0000	1.0000	1.0000
RS232-T1400	2	0	0	0	290	0	0	12	1.0000	1.0000	1.0000	1.0000	1.0000
RS232-T1500	2	0	0	0	291	0	0	13	1.0000	1.0000	1.0000	1.0000	1.0000
RS232-T1600	2	0	0	0	293	0	1	9	0.9474	0.9000	1.0000	0.9967	1.0000
s15850-T100	0	0	1	0	2397	0	1	26	0.9811	0.9630	1.0000	0.9996	1.0000
s35932-T100	0	0	1	1	5967	0	3	12	0.8889	0.8000	1.0000	0.9995	1.0000
s35932-T200	0	0	1	0	5961	0	5	11	0.8148	0.6875	1.0000	0.9992	1.0000
s35932-T300	0	0	1	0	5973	0	7	21	0.8571	0.7500	1.0000	0.9988	1.0000
s38417-T100	1	0	2	1	5656	0	4	8	0.8000	0.6667	1.0000	0.9993	1.0000
s38417-T200	1	0	1	0	5656	0	1	14	0.9655	0.9333	1.0000	0.9998	1.0000
s38417-T300	1	2	1	0	5675	12	1	15	0.6977	0.9375	0.9979	0.9977	0.5556
s38584-T100	1	0	0	1	7064	0	8	1	0.2000	0.1111	1.0000	0.9989	1.0000
s38584-T200	0	0	0	0	7064	0	0	83	1.0000	1.0000	1.0000	1.0000	1.0000
s38584-T300	1	0	0	0	7064	0	0	731	1.0000	1.0000	1.0000	1.0000	1.0000
wb_conmax-T100	0	0	0	0	23194	0	4	11	0.8462	0.7333	1.0000	0.9998	1.0000
B19_free	0	0	0	0	63169	1	0	0	-	-	1.0000	1.0000	-
RS232_free	16	0	0	0	270	9	0	0	-	-	0.9677	0.9677	-
s15850_free	2	0	0	0	2396	0	0	0	-	-	1.0000	1.0000	-
s35932_free	0	0	0	0	5965	0	0	0	-	-	1.0000	1.0000	-
s38417_free	1	0	0	0	5656	0	0	0	-	-	1.0000	1.0000	-
s38584_free	1	0	0	0	7063	0	0	0	-	-	1.0000	1.0000	-
wb_conmax_free	0	0	0	0	23194	0	0	0	-	-	1.0000	1.0000	-
平均	-	-	-	-	-	-	-	-	0.8996	0.8735	0.9987	0.9984	0.9778

表 6 補正前後の HT 識別精度の比較.

手法	F-score	TPR (%)	TNR (%)	Accuracy (%)	Precision (%)
複数モデル [9]	0.8861	89.24	99.72	99.70	93.31
提案手法 (トロイクラスタ)	0.8848	86.29	99.87	99.83	92.17
複数モデル [9] ＋提案手法 (トロイクラスタ)	0.8996	87.35	99.87	99.84	97.78
初期 HT 識別結果	0.8602	88.07	99.73	99.70	85.76

が 0.13，Precision が 6.41 向上し，F-score は 0.8848 を達成した．TNR の向上から，トロイクラスタ法を補正に使用することは，FP を TN に補正することに有効であることが分かる．また，FP が減少したことで，F-score，Accuracy が向上し，特に Precision は大幅な改善が見られた．

また，複数の学習済みモデルを用いた補正処理手法とトロイクラスタ法を用いた補正処理手法を組み合わせることでより有効的な補正ができることが分かり，総合指標である F-score は最大 0.8996 を達成し，TPR が 87.35%，TNR が 99.87%，Accuracy が 99.84%，Precision が 97.78%を達成した．

5. おわりに

本稿では，7 個のフリー回路と 20 個のトロイ回路の計 27 種類のベンチマークを対象に，GAT を使用したグラフ学習による HT 識別後の補正処理手法を提案した．提案手法は，トロイクラスタ法を用いた補正処理であり，トロイクラスタを作成してトロイクラスタに含まれないトロイノードをノーマルノードに補正することで，識別精度の向上を試みる．

評価実験の結果，トロイクラスタ法を用いた補正処理によって効果的に FP を TN に補正できた．トロイクラスタ法を用いた補正処理によって，初期 HT 識別結果から TNR が 0.14，Accuracy が 0.13，Precision が 6.41 向上し，F-score は 0.8848 を達成した．また，総合指標である F-score は最大 0.8996 を達成し，TPR が 87.35%，TNR が 99.87%，Accuracy が 99.84%，Precision が 97.78%を達成した．

今後の課題として，トロイクラスタ法を用いて FN から TP への補正手法の提案が挙げられる．現状，トロイクラスタ法を用いた補正処理の際に，TP から FN への誤補正による TPR の低下が問題である．トロイクラスタ法を用いて FN から TP への効果的な補正を考案することで，さらなる HT 識別精度の向上が期待できる．

謝辞

本研究成果は，一部，国立研究開発法人情報通信研究機構 (NICT) の委託研究 (JPJ012368C08101) により得た．

参考文献

[1] B. Liu, and G. Qu, “Vlsi supply chain security risks and mitigation techniques: A survey,” *Integration*, vol. 55,

pp. 438–448, 2016.
[2] S. Adee, “The hunt for the kill switch,” *iEEE SpEctrum*, vol. 45, no. 5, pp. 34–39, 2008.
[3] M. Tehranipoor, and F. Koushanfar, “A survey of hardware trojan taxonomy and detection,” *IEEE design & test of computers*, vol. 27, no. 1, pp. 10–25, 2010.
[4] K. G. Liakos, G. K. Georgakilas, S. Moustakidis, P. Karlsson, and F. C. Plessas, “Machine learning for hardware trojan detection: A review,” in *2019 Panhellenic Conference on Electronics & Telecommunications (PACET)*. IEEE, 2019, pp. 1–6.
[5] Z. Huang, Q. Wang, Y. Chen, and X. Jiang, “A survey on machine learning against hardware trojan attacks: Recent advances and challenges,” *IEEE Access*, vol. 8, pp. 10 796–10 826, 2020.
[6] K. Hasegawa, K. Yamashita, S. Hidano, K. Fukushima, K. Hashimoto, and N. Togawa, “Node-wise hardware trojan detection based on graph learning,” *IEEE Transactions on Computers*, 2023.
[7] 池上 裕香，山下 一樹，長谷川 健人，福島 和英，清本 晋作 戸川 望，“実環境回路に挿入されたハードウェアトロイを対象としたグラフ学習によるゲートレベルハードウェアトロイ識別”，電子情報通信学会技術研究報告 (Web)，vol. 122，no. 402 (VLD2022 73-122)，pp. 191–196，2023.
[8] 根岸良太郎, 戸川望, “ 機械学習を用いたハードウェアトロイ識別におけるトロイクラスタ法による補正処理”，2024 Symposium on Cryptography and Information Security (SCIS)，2024.
[9] 吉見尚，池上裕香，戸川望，“グラフ学習を用いたハードウェアトロイ識別に対する複数の学習済みモデルを用いた補正処理”，2025 Symposium on Cryptography and Information Security (SCIS)，2025.
[10] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, “Graph attention networks,” *arXiv preprint arXiv:1710.10903*, 2017.
[11] Trust-HUB, https://www.trust-hub.org/.
[12] K. Hasegawa, M. Yanagisawa, and N. Togawa, “Trojan-feature extraction at gate-level netlists and its application to hardware-trojan detection using random forest classifier,” in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2017, pp. 1–4.