多出力LSTMによる定常状態電力波形推定を利用した消費電力解析にもとづくデバイスの異常動作検知手法

高崎 和成[1,a)]　木田 良一[2,b)]　金子 博一[2,c)]　戸川 望[1,d)]

概要：集積回路 (IC) の複雑化・高機能化に伴う設計・製造の外注化によりハードウェアデバイスのセキュリティ課題が顕在化している．ハードウェアデバイスの異常動作検知手法の一つに電力解析があるが，IoT (Internet of Things) デバイスの中には OS (Operating System) 上で複数のアプリケーションが動作し複雑な消費電力波形を持つものも多く，これまでの電力解析手法をそのまま適用するだけでは異常動作を検知できない．そこで OS などによる定常的な電力波形を推定し測定した電力波形から除去することで，アプリケーションプログラム実行による電力変化を抽出し異常動作検知する手法が提案されている．これまで時間軸逆方向に推定する処理を必要としたため過去の異常動作しか検知できなかった．本稿では多出力 LSTM (long short-term memory) を利用して時間軸方向の推定のみで定常状態電力波形を推定する手法を提案する．提案手法により従来手法より早い段階で異常動作を検知できる．実験の結果，提案手法により IoT デバイス上での異常動作を検知することに成功した．

キーワード：異常検知，ハードウェアトロイ，電力解析，シングルボードコンピュータ，LSTM

Power-Analysis Based Anomalous Behavior Detection Utilizing Steady State Power Waveform Generated by LSTM with Many Output Dimentions

Kazunari Takasaki[1,a)]　Ryoichi Kida[2,b)]　Hirokazu Kaneko[2,c)]　Nozomu Togawa[1,d)]

Abstract: Hardware security issues have emerged in recent years since more third parties are involved in various stages in designing and manufacturing integrated circuits (ICs). Power analysis is one of the methods to detect anomalous behaviors in IC products, but it is hard to apply conventional methods to those where an operating system (OS) and various software programs are running, because they have complex power waveforms. In this paper, we propose an anomalous behavior detection method which can apply to such a complex hardware device. In the proposed method, steady-state power waveforms caused by an OS is firstly obtained using LSTM (long short-term memory), one of the recurrent neural network architectures. By using LSTM only in the forward direction, we can obtain the steady-state power waveforms in a real-time manner. Hence we can detect an anomalous behavior in a real-time manner. Experimental results demonstrate the effectiveness of the proposed method.

Keywords: anomalous behavior detection, hardware Trojan, power analysis, single board computer, LSTM

[1] 早稲田大学
Waseda University
[2] 株式会社ラック
LAC Co., Ltd.
[a)] kazunari.takasaki@togawa.cs.waseda.ac.jp
[b)] ryoichi.kida@lac.co.jp
[c)] hirokazu.kaneko@lac.co.jp
[d)] ntogawa@waseda.jp

1. はじめに

近年の IC (integrated circuit) の高機能化・省電力化によって，IoT (Internet of Things) デバイスにおいても OS (operating system) を搭載したデバイスが利用され，OS

上で様々なアプリケーションが動作するものがある [1], [2]. IoT デバイスに対する攻撃は近年増加しており，通常外部に漏れることのない情報に不正にアクセスするなどセキュリティ上の危険性が指摘されている [3]．IoT デバイスにおけるセキュリティ上のリスクには悪意のあるソフトウェアの他に，悪意ある動作をするハードウェアであるハードウェアトロイがある．ハードウェアトロイとはハードウェアデバイス上に直接組み込まれた不正な動作をする回路である．ハードウェアトロイによる不正な動作の例として，暗号化通信に用いる鍵情報の漏洩やデバイスに大量の情報を処理させ負荷を与えることで機能不全を引き起こす Denial of Service (DoS) 攻撃がある [4]．このようなハードウェアトロイが攻撃者によって IC の設計段階や製造段階などで挿入される危険性が指摘されている [5]．より複雑化し高機能化した IoT デバイスの設計・製造には様々なサードパーティが関わっており，ハードウェアトロイが挿入されるリスクが高まっている [6]．ハードウェアトロイをいかに検知するかは重要な課題となっている．

ハードウェアトロイを検知する手法にはハードウェア設計段階で検知する手法とハードウェア製造後に検知する手法がある [7], [8]．ここでハードウェア製造段階でハードウェアトロイを検知することに焦点を当てる．ハードウェア製造後に検知する手法に論理検証とサイドチャネル解析がある [8]．論理検証は入力に対する出力値と期待される出力値を比較することでハードウェアトロイを検知する．しかし近年の大規模化し複雑な回路を対象にした論理検証では入力値の組み合わせが膨大となり，実回路のハードウェアトロイを検知することは容易ではない [9]．サイドチャネル解析はデバイスから意図せず漏出する電磁波や電力などのサイドチャネル情報を解析し，ハードウェアトロイを検知する．サイドチャネル解析は論理検証のようにテストデータを用意する必要がないため，大規模な回路でのハードウェアトロイ検知に適している．

一方，前述のような OS が動作する IoT をデバイスの消費電力波形は OS による電力やアプリケーションプログラム実行による電力などが同時に消費され，複雑な波形になる．そのため，既存の電力解析にもとづくハードウェアトロイ検知手法をそのまま適用して異常動作を検知するのは難しい．OS が動作する IoT デバイス上の異常動作を検知するには，OS とアプリケーションプログラムを含むデバイス全体の電力振る舞いをモデル化し，アプリケーションと消費電力の関係を解析する必要がある．

OS などが実行される IoT デバイスにおける電力解析にもとづく異常動作検知手法として，アプリケーションプログラム実行による電力変化のみを抽出し異常動作検知を行う手法が提案されている [10]，[11]．この手法では IoT デバイスの消費電力をアプリケーションプログラムを実行していない定常状態の電力とアプリケーションプログラム実行による電力変化 (アプリケーション電力) にモデル化し，測定したデバイス全体の消費電力から定常状態の電力波形を差し引くことでアプリケーション電力を抽出・解析し，デバイスの異常動作を検知する．とくに [11] は平均的な定常状態電力波形を推定するのに，再帰的ニューラルネットワークの一つである LSTM (long short-term memory) を用いる．定常状態の消費電力波形を LSTM を用いて学習し，アプリケーションプログラムの実行と非実行を繰り返すデバイスの非実行区間の消費電力波形を初期入力として定常状態電力波形を推定する．[11] では時間軸方向と時間軸逆方向に定常状態電力波形を推定し，重み付けを変更しながら平均を取り，最終的な定常状態電力波形を得る．時間軸逆方向へ定常状態電力波形を推定する過程があるため，過去の定常状態電力波形しか推定することができず，現在実行中あるいは実行直後のアプリケーションについて異常動作検知するのは難しい．

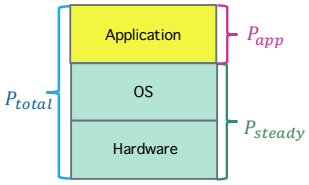
本稿では時間軸順方向のみによる定常状態電力波形推定に基づく IoT デバイス異常動作検知手法を提案する．提案手法により現在の消費電力波形の入力から未来の定常状態電力波形を推定することが可能になり，[11] より早い段階での異常動作の検知が可能になる．実験結果から，従来手法 [11] より早い段階で異常動作の検知に成功した．

本稿は以下のように構成される．2 章で [11] や本稿で対象とするデバイスモデルを定義する．3 章で，LSTM で時間軸方向と時間軸逆方向に定常状態電力波形を推定することでアプリケーション電力を抽出し異常動作を検知する手法 [11] を説明する．4 章で，多出力の LSTM モデルを用いて時間軸方向のみに依存して定常状態電力波形を推定する手法を提案する．5 章で，アプリケーションを実装したデバイスの消費電力波形を計測し，提案手法を適用する実験と実験結果を示す．6 章で本稿をまとめる．

2. デバイスモデル

本章では提案手法で対象とするデバイスモデルを説明する．対象モデルの消費電力に着目する．対象デバイスモデルは OS 上でアプリケーションプログラムを実行する．図 1(a) に対象モデルの電力モデルを示す．[12] に示される通り，電力モデルはハードウェア部分，OS 部分，アプリケーション部分の三つに分かれる．ハードウェア部分には周辺機器や発光ダイオード，冷却装置などの消費電力が含まれる．OS 部分には OS 実行による消費電力を含む．アプリケーション部分ではアプリケーションプログラム実行による消費電力を含む．

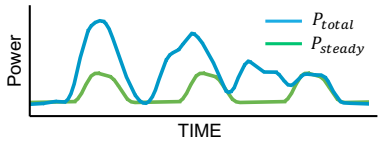
IoT デバイスは消費するエネルギーを抑えるため，センサで周りの状況を監視しながら条件を満たすときのみアプリケーションプログラムを実行するものが多い．対象モデ



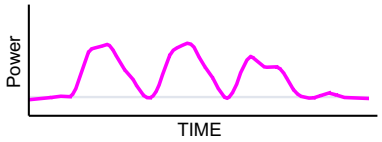
(a) 電力モデル.

(b) アプリケーションモデル.

図 1: 対象モデル.



(a) P_{steady} と P_{total}.



(b) 抽出されたアプリケーション電力 P_{app}.

図 2: [10] の対象モデルの電力波形.

ルは図 1(b) に示す通り，複数のアプリケーションプログラムが連続して実行されているアプリケーション実行区間とアプリケーションプログラムを実行しない定常状態がある．定常状態における消費電力は，図 1(a) におけるハードウェアと OS による消費電力であり，定常状態電力波形を P_{steady} とする．電力波形 P_{steady} は時系列の電力値によって $P_{\text{steady}} = \{p_i\}$ で表される．ここで p_i は時刻 t_i の電力値を表す．以降，電力波形は同様に表わされるものとする．

次に，アプリケーション電力 P_{app} に着目する．アプリケーション電力とはデバイスがアプリケーションプログラムを実行することによる電力変化を指す．デバイス全体の消費電力波形を P_{total} とすると，次式が成り立つ．

$$P_{\text{app}} = P_{\text{total}} - P_{\text{steady}}. \tag{1}$$

式 (1) に従い，アプリケーション電力 P_{app} を抽出する．

異常動作を検知する上で次の三点を想定する．

(1) IoT デバイス上で実行されるアプリケーションは既知である．

(2) 各アプリケーションはアクティブ状態とスリープ状態を持つ．

(3) 各アプリケーションは同時に動作しない．

上記のアプリケーションが実装された IoT デバイスを対象モデルとしてアプリケーション電力の抽出をする．抽出されたアプリケーション電力に基づき，IoT デバイスの異常動作を検知する．

3. LSTM による定常状態電力波形推定を用いた異常動作検知手法 [11]

本章では LSTM を用いて時間軸方向と時間軸逆方向に定常状態電力波形を推定することでアプリケーション電力を抽出し異常動作を検知する手法 [11] を説明する．

3.1 異常検知の流れ

[11] は以下の 5 つのステップで異常動作を検知する．

ステップ 1 ：アプリケーションプログラムを実行していないときのデバイスの電力波形 P'_{steady} とアプリケーション実行状態のデバイス全体の消費電力波形 P'_{total} の測定.

ステップ 2 ：測定した波形の平滑化．平滑化した定常状態の電力波形を $\overline{P}_{\text{steady}}$，平滑化したアプリケーション実行状態のデバイス全体の電力波形を P_{total} とする．

ステップ 3 ：LSTM を用いた平均的な定常状態電力波形 P_{steady} の推定.

ステップ 4 ：アプリケーション電力 P_{app} の抽出.

ステップ 5 ：Local outlier factor (LOF) 法を用いた異常動作検知手法 [13] の適用.

ステップ 1 では，アプリケーションプログラムを実行していない状態のデバイス全体の消費電力波形 P'_{steady} と，アプリケーションプログラムが実行されている状態のデバイス全体の消費電力波形 P'_{total} を測定する．**ステップ 2** では P'_{steady} と P'_{total} を移動平均を用いて平滑化する．平滑化された消費電力波形をそれぞれ $\overline{P}_{\text{steady}}$ と P_{total} とする．**ステップ 3** は LSTM を用いて定常状態の電力波形 $\overline{P}_{\text{steady}}$ を学習し，平滑化したアプリケーション実行状態の消費電力波形 P_{total} から定常状態電力波形を推定する．**ステップ 3** の詳細は 3.2 節で説明する．**ステップ 4** では式 (1) によりアプリケーション電力 P_{app} を抽出する．**ステップ 5** では**ステップ 4** で抽出した P_{app} を用いて LOF 法を用いた異常動作検知手法 [13] を適用する．[13] はデバイスの消費電力波形をアクティブ状態と非アクティブ状態に分割し，アクティブ状態の継続時間と消費エネルギーを特徴量として LOF を適用し異常なアクティブ状態を検知する．図 2(a) に対象モデルの平均的な定常状態電力波形 P_{steady} と平滑化したデバイス全体の電力波形 P_{total} の例を示す．図 2(b) に対象モデルの抽出したアプリケーション電力 P_{app} を示す．

3.2 ステップ 3: LSTM による定常状態電力波形推定手法

本節では [11] の**ステップ 3** を説明する．**ステップ 3** はさらに以下の 3 つのステップで成り立つ．

ステップ 3.1 $\overline{P}_{\text{steady}}$ の順方向および逆方向の学習.

ステップ 3.2　順方向および逆方向へ定常状態電力波形を推定.

ステップ 3.3　**ステップ 3.2** で推定した順方向および逆方向の波形の重複部分を重み付けを変えて平均を取ることによる P_{steady} の取得.

ステップ 3 では，**ステップ 2** で得た，定常状態の測定波形を平滑化した $\overline{P}_{\mathrm{steady}}$ とデバイス全体の測定波形を平滑化した P_{total} をもとに，LSTM を利用することでアプリケーション実行中の定常状態電力波形 P_{steady} を推定する．P_{steady} を予測できれば式 1 によりアプリケーション電力波形 P_{app} を得ることができる．[11] は定常状態電力波形の推定を，時間軸方向である順方向と時間軸と逆の逆方向に行い，重複部分について重み付けを変えて平均を取ることで行う．[11] は n 入力 1 出力の LSTM を用いた機械学習モデルを使用する．

ステップ 3.1 では，順方向・逆方向それぞれ，n 個の連続した入力値と 1 つの電力値を正解ラベルとして学習する．順方向の学習では n 個の連続した定常状態電力値を入力とし，続く 1 つの定常状態電力値を正解ラベルとして学習する．逆方向の学習も同様に，逆方向の n 個の連続値を入力とし，逆方向に連続する 1 つの定常状態電力値を正解ラベルとして学習する．

ステップ 3.2 では，P_{total} 中のアプリケーション非実行区間の波形を初期入力として，**ステップ 3.1** で学習したモデルを用いて順方向・逆方向に定常状態電力波形を推定する．順方向を例に取ると，P_{total} 中のアプリケーション非実行区間のうち n 個の連続した電力値を入力として，続く 1 つの定常状態電力値を推定する．次に推定した電力値を含む連続した n 個の電力値を入力として新たに続く 1 つの電力値を推定する．以上を繰り返すことで順方向に定常状態電力波形を推定する．逆方向についても同様に推定し，推定する．

ステップ 3.3 では**ステップ 3.2** で推定した順方向・逆方向の波形を重み付けを変更しながら平均をとり，P_{steady} を得る (詳細は [11] を参照).

4. 多出力 LSTM を利用した定常状態電力波形推定手法

3 章で示した手法 [11] のステップ 3 は，定常状態電力波形の精度を十分にするため，順方向と逆方向の双方から定常状態電力波形を推定しそれらの平均を取ることで P_{steady} を推定する．時間軸逆方向への推定処理があるため過去の定常状態電力波形しか推定することができない．このため現在実行中や実行直後のアプリケーションの異常動作を検知するのは難しい．そこで本稿では時間軸順方向の推定のみでアプリケーション電力を抽出可能な十分な精度の定常状態電力波形を推定する手法を提案する．順方向のみで定常状態電力波形を推定できることで未来の定常状態電力波形を推定可能になり，実行直後のアプリケーションのアプリケーション電力を抽出できる．これにより [11] より早い段階での異常動作検知が可能になる．

従来手法 [11] で用いられる LSTM は入力値 n 個に対し出力値は 1 個で，1 点ずつ電力値を推定する．提案手法では入力値 n 個に対し出力値 m 個の LSTM を用いて m 点ずつ電力値を推定することで定常状態電力波形を推定する．一方，1 点ずつではなく m 点ずつ電力値を推定すると，誤差の蓄積を軽減することができ，精度良く定常状態電力波形を推定できる可能性がある．その場合，順方向だけの推定でも十分な精度の定常状態電力波形を推定できる．例えば 1000 点の定常状態電力値を推定することを考えると，従来手法 [11] は順方向への推定だけで LSTM に対する入力・出力のサイクルを 1000 回繰り返す．これに対し m 点ずつ推定する方法では，例えば $m = 200$ とすると，LSTM に対する入力・出力のサイクルは 5 回のみで 1000 点の定常状態電力値を推定することができる．LSTM の入出力サイクル回数が減ることで推定による誤差の蓄積が最小限に抑えられるため，m 点ずつ推定する方がより精度の高い定常状態電力波形を推定することができる．予備実験の結果，$m = 200$ とし LSTM により順方向だけで定常状態を推定した結果，従来手法 [11] と同等以上の精度で定常状態電力波形の推定に成功した．

提案手法の概要を以下に示す.

ステップ 3.1'　$\overline{P}_{\mathrm{steady}}$ の学習.

ステップ 3.2'　P_{total} 中のアプリケーション非実行区間の波形を初期入力として定常状態電力波形を推定.

本稿では LSTM を用いたニューラルネットワークの入出力データの特性として以下を考える.

f_s **[Hz]**　：電力波形のサンプリング周波数.

$n_{\mathbf{in}}$　：ニューラルネットワークの入力数.

$n_{\mathbf{out}}$　：ニューラルネットワークの出力数.

$t_{\mathbf{in}}$ **[s]**　：n_{in} に対応する時間の長さ.

$t_{\mathbf{out}}$ **[s]**　：n_{out} に対応する時間の長さ.

上記の各値について式 (2)，(3) が成り立つ.

$$n_{\mathrm{in}} = f_s t_{\mathrm{in}}. \tag{2}$$

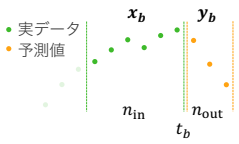
$$n_{\mathrm{out}} = f_s t_{\mathrm{out}}. \tag{3}$$

4.1 使用する機械学習モデル

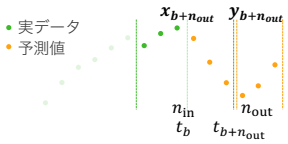
本節では使用する機械学習モデルを説明する．提案手法では全結合ニューラルネットワークを用いる．表 1 に使用するニューラルネットワークモデルを示す．入力層，中間層，出力層の 3 層の構造である．ユニット数は入力層が n_{in}，中間層の LSTM が 100，出力層が n_{out} である．活性化関数には ReLU を使用する．

表 1: 使用するニューラルネットワークモデル．各層の数字はユニット数．

入力層	中間層 (LSTM)	出力層	活性化関数
n_{in}	100	n_{out}	ReLU



(a) 時刻 t_b のときの入力ベクトル $\boldsymbol{x_b}$ と出力ベクトル $\boldsymbol{y_b}$.



(b) 時刻 $t_{b+n_{\text{out}}}$ のときの入力ベクトル $\boldsymbol{x_{b+n_{\text{out}}}}$ と出力ベクトル $\boldsymbol{y_{b+n_{\text{out}}}}$.

図 3: 多出力 LSTM による定常状態電力波形推定の模式図．初期入力以外は直前に推定した電力値を入力ベクトルに含み，次の定常状態電力値を推定する処理を繰り返すことでアプリケーション実行区間の定常状態定常状態電力波形を推定する．

4.2 ステップ 3.1'：LSTM による定常状態電力波形の学習

本節では LSTM の学習方法を説明する．**ステップ 2** で平滑化した定常状態電力波形 $\overline{P}_{\text{steady}}$ の時刻 t_i における電力値を p_i とする．$\boldsymbol{x_i} = (p_{i-n_{\text{in}}}, \ \cdots, \ p_{i-1})$ を入力ベクトル，$\boldsymbol{y_i} = (p_i, \ \cdots, \ p_{i+n_{\text{out}}-1})$ を正解ラベルベクトルとして i を 1 ずつ増加させ，学習する．

4.3 ステップ 3.2'：LSTM による定常状態電力波形の推定

4.2 節で学習した学習器を用いて P_{steady} を推定する．q_i を時刻 t_i における P_{total} の電力値とし，p_i^s を LSTM により推定される定常状態電力波形の時刻 t_i における電力値とする．

まず学習器に最初に入力する初期入力ベクトルを定義する．図 3 に提案手法の定常状態電力波形推定の模式図を示す．アプリケーションプログラムを実行しているときのデバイス全体の電力波形を平滑化した P_{total} のうち，アプリケーションプログラムを実行していない区間 (アプリケーション非実行区間) の t_{in} [s] の電力波形を初期入力ベクトルとする．図 3(a) に示すように，時刻 t_b を基準として，$\boldsymbol{x_b} = (q_{b-n_{\text{in}}}, \ \cdots, \ q_{b-1})$ を初期入力とし，学習済みの LSTM により $\boldsymbol{y_b} = (p_b^s, \ \cdots, \ p_{b+n_{\text{out}}-1}^s)$ を推定する．同様に図 3(b) に示すように，時刻 $t_b + t_{\text{out}}$ を基準として，その前の n_{in} 個のデータから学習済みの LSTM によって時刻

表 2: 被測定デバイス．

デバイス名	型番	使用用途
Jetson Nano	Jetson Nano Developer Kits	被測定デバイス
カメラモジュール	Raspberry Pi Camera Module V2	カメラ

表 3: 測定デバイス．

デバイス名	型番	使用用途
オシロスコープ	Tektronix MDO3104	電流・電圧測定
電流プローブ	Tektronix TCP0030A	電流測定
電源装置	KEITHLEY 2280S-32-6	電源

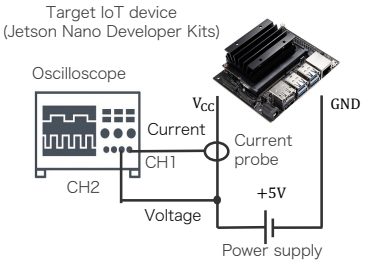


図 4: 測定回路図.

$t_b + t_{\text{out}}$ 以降の n_{out} 個データを推定する．なお，アプリケーション実行区間の入力データは LSTM によって推定されたデータを用いることにする．こうすることで P_{total} のうち，アプリケーション非実行区間の電力波形，つまり定常状態電力波形をもとに，アプリケーション実行区間の定常状態電力波形を推定することができ，その結果 P_{total} 中のすべての定常状態電力波形 P_{steady} を得ることができる．

5. 実験

本章では，多出力 LSTM による定常状態電力波形推定手法を利用したデバイスの異常動作検知実験を説明する．5.1 節では実験に用いるデバイスや実装するアプリケーションを説明し，5.2 節では実験結果を示す．

5.1 実験環境

本節では実験に用いる被測定デバイスと測定器，およびデバイスに実装するアプリケーションと異常動作を説明する．

5.1.1 使用するハードウェア

表 2 に本実験で使用する IoT デバイスを示す．デバイスには，近年 IoT デバイスのプロトタイプ開発に用いられるシングルコンピュータである Jetson Nano Developer Kits (Jetson Nano) [14] を用いる．Jetson Nano にはカメラモジュール (Raspberry Pi Camera Module V2) を接続する．表 3 に測定に利用する測定器を示す．IoT デバイス全体の消費電力波形を Tektronix 社のオシロスコープ (MDO3104) を利用してデバイス全体の電圧と電流を測定することで得る．電流の測定には電流プローブ (TCP0030A) を利用する．図 4 に測定回路図を示す．

表 4: 実装するアプリケーションと異常動作.

ラベル	内容	備考
R	データ受信	5MB のデータを受信.
E	暗号化	AES 暗号化 (鍵長 256bit).
S	データ送信	暗号化したデータを送信.
A	異常動作	暗号化したデータと異なるデータ (500kB) を送信.

5.1.2 実装するアプリケーションと異常動作

Mohammed らは Raspberry Pi 3 [1] を IoT ゲートウェイデバイスとしたホームエリアネットワークのテストベッドを提案している [15]．本実験では Jetson Nano を IoT ゲートウェイデバイスに見立て実験する．[15] に従い，表 4 に示すように 3 つの正常なアプリケーションと 1 つの異常動作を Jetson Nano に実装する．データ受信 (R) は同一ネットワーク内の別デバイスから 5MB のデータを受信する．暗号化 (E) は鍵長 256bit の AES 暗号化で受信したデータを暗号化する．データ送信 (S) は暗号化したデータを同一ネットワーク内の別のコンピュータへ送信する．異常動作 (A) は暗号化したデータと異なるデータ (500kB) を同一ネットワーク内の別のコンピュータへ送信する．各アプリケーションは Python 3.6.9 で実装，実行する．

正常動作では R, E, S の 3 つのアプリケーションプログラムを連続で実行する．異常動作はアプリケーション S の代わりに異常動作 A を実行する．異常動作 A は暗号化したデータが秘密鍵などの別のファイルに置き換えられる悪意ある動作を想定している．実験では，消費電力波形上で各アプリケーションプログラムを実行している区間を判別するため，各アプリケーションプログラムを実行中に GPIO(General Purpose Input Output) ピンの出力を HIGH(5V) に，実行していない区間で LOW(0V) にし，出力電圧を計測する．アプリケーション R は，デバイス側はいつデータが送信されてくるかがわからないため，GPIO は HIGH にならない．正常なアプリケーションの E, S を実行する間は GPIO1 が HIGH になり，異常なアプリケーション A を実行する間は GPIO2 が HIGH になる．出力される GPIO 値は 0 と 1 に二値化する．

5.2 実験結果

まず 3 章の**ステップ 1** に従い，Jetson Nano の定常状態の消費電力波形とアプリケーション実行状態の消費電力波形を計測する．定常状態の消費電力波形は 400s，アプリケーション実行状態は 200s 計測する．次に**ステップ 2** に従い，それぞれの波形を移動平均法により平滑化した $\overline{P}_{\mathrm{steady}}$ と P_{total} を得る．図 5 に $\overline{P}_{\mathrm{steady}}$ の一部を示す．

続いて 4 章で説明した提案手法の**ステップ 3** にもとづき定常状態電力波形を学習する．本実験では $f_s = 100$ [Hz], $t_{\mathrm{in}} = 1$ [s], $t_{\mathrm{out}} = 3$ [s] とする．したがって，本実験で用いるニューラルネットワークは入力数 100，LSTM のユニット数 100，出力数 300 となる．平滑化した定常状態電力波形 $\overline{P}_{\mathrm{steady}}$ から学習用データセットを作成し，LSTM を用いたニューラルネットワークで学習を行う．

表 5: LOF による異常検知結果.

番号	継続時間 [s]	消費エネルギー [W · s]	LOF 値	ラベル
1	1.32	1.5139	0.5653	1
2	2.89	5.5227	0.3513	1
3	2.47	1.2741	0.5695	1
4	1.49	1.6367	0.5445	1
5	2.84	5.3395	0.5830	1
6	2.31	1.2972	0.5695	1
7	1.39	1.6555	0.5705	1
8	2.79	5.1444	0.5697	1
9	2.29	1.2168	0.5695	1
10	1.38	1.6004	0.6397	1
11	2.83	5.0544	0.5697	1
12	2.33	1.3828	0.5695	1
13	1.37	1.5275	0.5653	1
14	2.80	5.2036	0.7391	1
15	2.09	1.0108	-0.1317	-1

次に学習したモデルを用いて定常状態電力波形を推定する．正常なアプリケーション R, E, S が実行されていない区間の連続した $t_{\mathrm{in}} = 1$ [s] の電力値を初期入力として，4 章の提案手法にもとづき推定する．図 6 に平滑化したアプリケーション実行状態の消費電力波形 P_{total} と，提案手法にもとづき推定した P_{steady} を示す．

ステップ 4 としてアプリケーション電力 P_{app} を抽出する．さらに P_{app} について，デバイスがアプリケーションプログラムや異常動作を実行している状態を明確化するため，さらに移動平均を適用した波形を P_{app} (SMA) とする．図 7 に P_{app} と P_{app} (SMA) を示す．

最後に**ステップ 5** として LOF による異常動作検知をする．**ステップ 4** で得た P_{app} (SMA) についてアクティブ状態の継続時間と消費エネルギーを特徴量として LOF により異常検知を行う．表 5 に LOF による異常検知結果を示す．番号はアクティブ状態の通し番号で，LOF 値は LOF が内部で異常度合いを判断する決定関数から得られる値である．ラベルは LOF が最終的に正常値か異常値かを判断した結果で，1 が正常値で -1 が異常値である．表 5 から，番号 15 のアクティブ状態が異常値と判断された．番号 15 のアクティブ状態は実際に異常動作 A が実行されているため，提案手法により異常動作を検知できた．図 8 に LOF による異常検知の結果を図示する．黄色のプロットはラベルが 1 の正常値で，赤色のプロットはラベルが -1 の異常値である．

以上の実験結果から，本稿で提案した**ステップ 3** の定常状態電力波形推定手法によりアプリケーション電力 P_{app} を抽出に成功し，さらに異常動作の検知にも成功した．

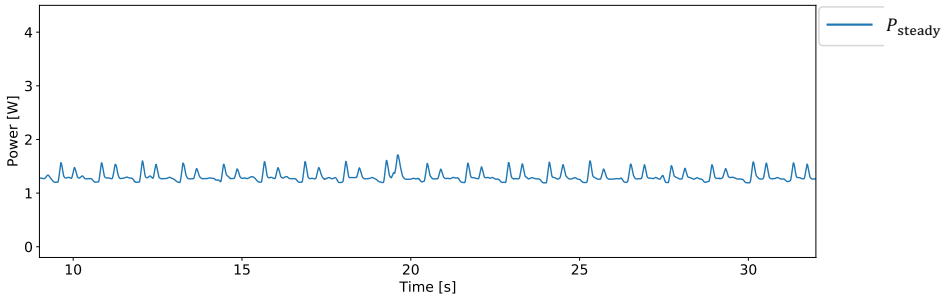
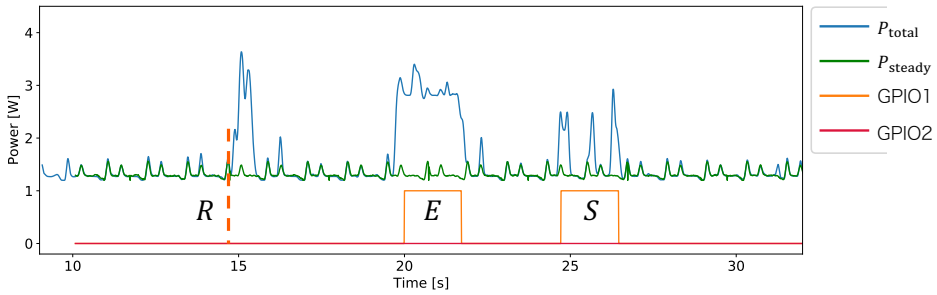
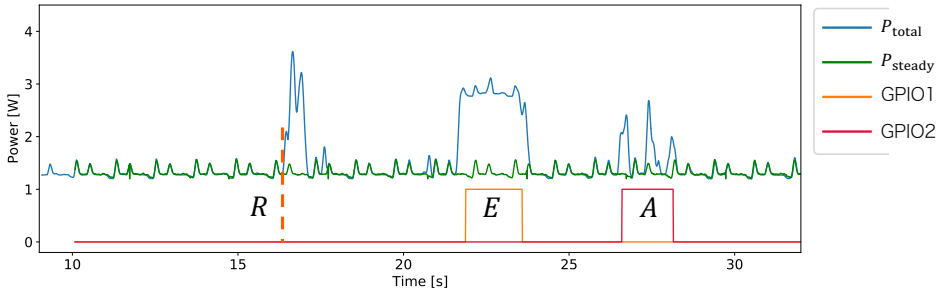


図 5: 学習に用いる平滑化した定常状態電力波形 $\overline{P}_{\text{steady}}$.



(a) 正常な 3 つのアプリケーション R, E, S を連続して実行 (初期入力ベクトルは 9s–10s の電力値).



(b) 正常なアプリケーション R, E と異常動作 A を連続して実行 (初期入力ベクトルは 9s–10s の電力値).

図 6: 各アプリケーションプログラム実行時の消費電力波形を平滑化した P_{total} および LSTM により推定した P_{steady}. 正常なアプリケーション E, S 実行時に GPIO1 が 1 になり，異常なアプリケーション A 実行時に GPIO2 が 1 になる．さらに正常なアプリケーション R の実行開始時間を示す.

6. おわりに

本稿では LSTM による定常状態電力波形推定手法を用いた異常動作検知手法 [11] のうち，定常状態電力波形推定手法について，出力数を複数にして学習・推定する手法を用いることで時間軸逆方向への学習・推定処理をせずに定常状態電力波形を推定する手法を提案した．時間軸逆方向への処理をしない結果，未来の定常状態電力波形を推定することが可能になり，[11] より早い段階での検知が可能になった．Jetson Nano を用いた実験の結果，提案手法により Jetson Nano の定常状態電力波形の推定に成功し，LOF による異常動作検知にも成功した．今後の課題として，本手法をリアルタイム検知に用いる場合の有効性の検証や定常状態の推定精度の向上が挙げられる.

謝辞 本報告は，総務省の「設計・製造におけるチップの脆弱性検知手法の研究開発」および総務省の令和 2 年度・令和 3 年度「ハードウェアチップの脆弱性検知手法の調査・検討等の請負」の成果の一部である.

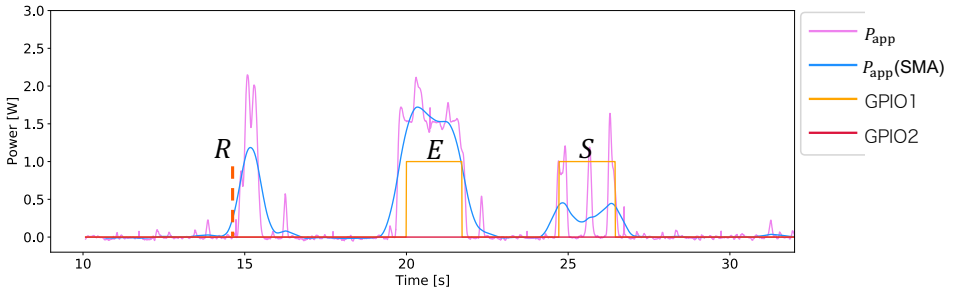
参考文献

[1] “Raspberry Pi.” [Online]. Available: https://www.raspberrypi.org/

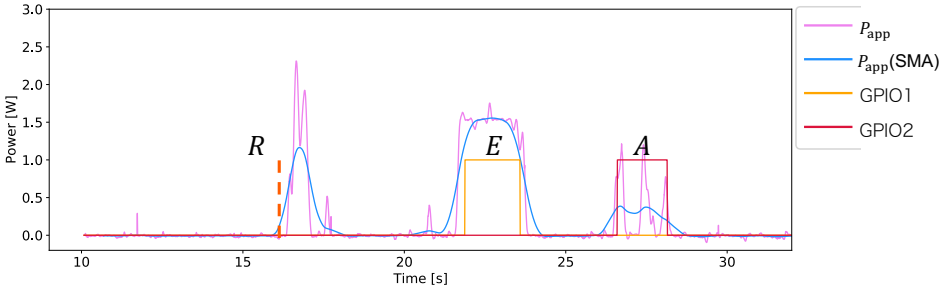
[2] “NVIDIA Jetson.” [Online]. Available: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/

[3] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, and A. Zanella, “Iot: Internet of threats? a survey of practical security vulnerabilities in real iot devices,” *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8182–8201, 2019.

[4] S. Bhunia, M. S. Hsiao, M. Banga, and S. Narasimhan,



(a) 正常な 3 つのアプリケーション R, E, S を連続して実行.



(b) 正常なアプリケーション R, E と異常動作 A を連続して実行.

図 7: **ステップ 4** により抽出した P_{app} と移動平均を用いてさらに平滑化した P_{app} (SMA).

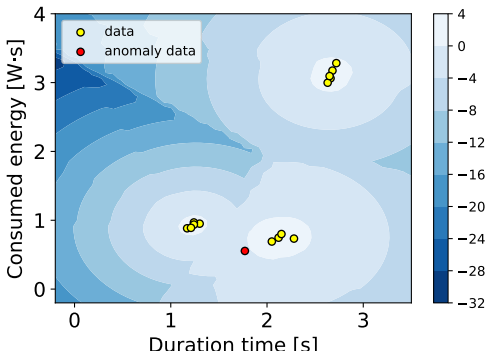


図 8: 抽出し平滑化した P_{app} (SMA) から得たプロットと LOF による異常動作検知の結果.

"Hardware trojan attacks: Threat analysis and countermeasures," *in Proc. of the IEEE*, vol. 102, no. 8, pp. 1229–1247, Aug 2014.

[5] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar, "Trojan detection using IC fingerprinting," *in Proc. IEEE Symposium on Security and Privacy*, pp. 296–310, 2007.

[6] H. Li, Q. Liu, J. Zhang, and Y. Lyu, "A survey of hardware trojan detection, diagnosis and prevention," in *Proc. 2015 14th International Conference on Computer-Aided Design and Computer Graphics (CAD/Graphics)*, Aug 2015, pp. 173–180.

[7] K. Hasegawa, M. Yanagisawa, and N. Togawa, "Hardware trojans classification for gate-level netlists using multi-layer neural networks," in *2017 IEEE 23rd International Symposium on On-Line Testing and Robust System Design (IOLTS)*, 2017, pp. 227–232.

[8] J. Francq and F. Frick, "Introduction to hardware Trojan detection methods," *in Proc. Design, Automation and Test in Europe, DATE*, pp. 770–775, 2015.

[9] S. Bhasin and F. Regazzoni, "A survey on hardware trojan detection techniques," in *Proc. 2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2015, pp. 2021–2024.

[10] K. Takasaki, K. Hasegawa, R. Kida, and N. Togawa, "An anomalous behavior detection method for iot devices by extracting application-specific power behaviors," in *Proc. 2020 IEEE 26th International Symposium on On-Line Testing and Robust System Design (IOLTS)*, 2020, pp. 1–4.

[11] K. Takasaki, R. Kida, and N. Togawa, "An anomalous behavior detection method based on power analysis utilizing steady state power waveform predicted by lstm," in *2021 IEEE 27th International Symposium on On-Line Testing and Robust System Design (IOLTS)*, 2021, pp. 1–7.

[12] B. Martinez, M. Montón, I. Vilajosana, and J. D. Prades, "The power of models: Modeling power consumption for iot devices," *IEEE Sensors Journal*, vol. 15, no. 10, pp. 5777–5789, Oct 2015.

[13] K. Hasegawa, K. Chikamatsu, and N. Togawa, "Empirical evaluation on anomaly behavior detection for low-cost micro-controllers utilizing accurate power analysis," in *Proc. 2019 IEEE 25th International Symposium on On-Line Testing and Robust System Design (IOLTS)*, July 2019, pp. 54–57.

[14] "NVIDIA Jetson Nano." [Online]. Available: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-nano/

[15] H. Mohammed, J. Howell, S. R. Hasan, N. Guo, F. Khalid, and O. Elkeelany, "Hardware trojan based security issues in home area network: A testbed setup," in *2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2018, pp. 972–975.