Anomalous IoT Behavior Detection by LSTM-Based Power Waveform Prediction

Ryusei Eda and Nozomu Togawa[a]
Department of Computer Science and Communications Engineering, Waseda University, Japan

Keywords: Hardware Trojan, Power Analysis, Anomalous Behavior Detection, LSTM.

Abstract: Internet of Things (IoT) devices have very rapidly spread out in recent years. In IoT devices where applications run on operating system (OS), the power consumption of the OS and the power consumption of the applications overlap, resulting in complex power waveform. Previous methods need to explicitly extract the application power waveform from the multiple signal sources in the measured power waveform, which often fail to detect anomalous behaviors. In this paper, we propose a method to detect anomalous behaviors by using LSTM (Long Short Term Memory). The proposed method learns power waveform containing multiple signal sources and compares the predicted waveform and the actual one. Then, we can successfully detect anomalous behaviors, even though the measured power waveform is composed of multiple signal sources. Experimental results show that anomalous behavior can be successfully detected from an IoT device built with Raspberry Pi4.

1 INTRODUCTION

In recent years, the rapid spread of IoT devices has increased the demand for integrated circuits (ICs) with more advanced and complex functions. As a result, IC design and manufacturing processes are separated, and the manufacturing process is often being outsourced in order to produce hardware devices inexpensively and efficiently. This has increased the risk of a malicious third party in the supply chain inserting hardware Trojans (Bhunia et al., 2014).

Hardware Trojans are circuits added to a hardware device against the intention of the designer, causing anomalous behavior such as information leakage and/or performance degradation (Bhunia et al., 2014). The use of components manufactured by third-party vendors enables inexpensive and efficient device manufacturing, but third-party components are at risk of having hardware Trojans. The hardware design and manufacturing can be divided into several processes, including specification, design, and manufacturing, but it is pointed out that there is a risk of hardware Trojans being inserted in all processes involving third-party vendors (Francq and Frick, 2015). It is an important issue how to detect hardware Trojans. There are two ways for detecting hardware Trojans: one is to detect them at the design stage (Jin and Makris, 2008; Islam et al., 2013; Oya et al., 2015; HaddadPajouh et al., 2018), and the other is to detect them after manufacturing (Agrawal et al., 2007; Chakraborty et al., 2009; Wang et al., 2013; Bhasin et al., 2013; Zaza et al., 2020). However, not all hardware Trojans can be detected at the design stage because it is not always possible to obtain design information. Detecting anomalous behaviors of IoT devices after manufacturing is important.

In order to detect anomalous behaviors of IoT devices after manufacturing, there is one effective method: side-channel analysis. Anomalous behaviors are detected by analyzing side-channel information, such as power consumption, based on the assumption that anomalous behaviors like hardware Trojans can affect side-channel information. There have been proposed several methods to detect anomalous behavior by measuring the power consumption as side-channel information and based on the duration time of an application program running and the amount of power consumption (Hasegawa et al., 2018; Takasaki et al., 2021a; Takasaki et al., 2021b). Based on the method in (Hasegawa et al., 2018), the extended version is proposed in (Takasaki et al., 2021b) to detect anomalous behaviors by applying it to IoT devices with steady-state power waveform.

The previous method (Takasaki et al., 2021a) subtracts the power consumed regularly by the operating system (OS) and hardware (called steady-state

[a] https://orcid.org/0000-0003-3400-3587

Eda, R. and Togawa, N.
Anomalous IoT Behavior Detection by LSTM-Based Power Waveform Prediction.
DOI: 10.5220/0013368900003944
In *Proceedings of the 10th International Conference on Internet of Things, Big Data and Security (IoTBDS 2025)*, pages 345-352
ISBN: 978-989-758-750-4; ISSN: 2184-4976

power waveform) from the measured power waveform, and extracts and analyzes only the power consumed by the application program (called application power waveform), and detects anomalous behavior. However, LSTM cannot well predict the steady-state power waveform, included in the total power waveform and thus we cannot efficiently extract the application power waveform sometimes. In addition, this method partitions the application power waveform into sections, each of which corresponds to a small operation, and does not consider a sequence of these sections. Hence, inter-section anomalous behavior cannot be detected.

In this paper, we propose a method to detect anomalous behavior from power waveforms including the steady-state power waveform based on the difference between the waveform predicted by LSTM and actual one. The proposed method firstly smooths and normalizes the power waveforms measured from IoT device. Then, it extracts fixed-length power waveforms in a sliding window manner and trains them using LSTM. The trained LSTM predicts the power waveform of the IoT device. By taking the difference between the predicted power waveform and the actual one, we detect anomalous behaviors by extracting the outliers of the difference using the Hotelling theory (Hotelling, 1992). The proposed method was applied to the IoT device built with Raspberry Pi4. As a result, we can successfully detect the anomalous behaviors.

The rest of this paper is organized as follows: Section 2 summarizes several related works on IoT device anomalous behavior detection and defines an IoT device model; Section 3 describes the proposed anomalous behavior detection method by LSTM-based power waveform prediction; Section 4 explains the experimented IoT device and Section 5 demonstrates the experimental results; Section 6 concludes the paper with several concluding remarks.

2 RELATED WORKS AND IoT DEVICE MODEL

In this section, we introduce related researches on IoT device anomalous behavior detection and define the power consumption of the hardware device targeted in this paper. Then, we discuss the challenges for detecting anomalous behavior for the IoT device with steady-state power waveform.

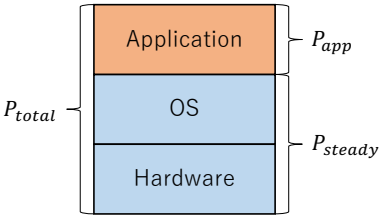

Figure 1: Power consumption model of the IoT devices.

2.1 Related Works

Side-channel analysis is one of the effective methods for detecting anomalous behavior of IoT devices. In side-channel analysis, all components are assumed to affect side-channel information such as power consumption, delay, and electromagnetic waves, and anomalous behavior of IoT devices is detected by analyzing the changes of them. Side-channel analysis is effective in detecting anomalous behavior in that it can detect anomalous behavior without affecting the output of the circuit and can be applied to microcontrollers and the software running on them (Wang et al., 2013; Hasegawa et al., 2018; Takasaki et al., 2021b).

Wang et al. detect anomalous behavior of IoT devices by comparing the power consumption of both the model without anomalies (golden model) and the target model (Wang et al., 2013). While their method has good detection accuracy, it costs much money and time to prepare the golden model. Hasegawa et al. do not use a golden model and detect anomalous behaviors based on the duration time of an application program and power consumption (Hasegawa et al., 2018). However, its detection accuracy cannot become high enough sometimes.

Takasaki et al. (Takasaki et al., 2021a) subtracts the steady-state power waveform from the measured power and extracts only the application power waveform. LSTM is used to estimate the steady-state power waveform. Since the input to the LSTM includes the data estimated previously, errors accumulate as the LSTM prediction proceeds. In addition, the application power waveform is partitioned into sections, each of which corresponds to a small operation, and the operation sequence is not taken into account. Hence, inter-section anomalous behavior cannot be detected.

2.2 IoT Device Model

In this paper, we assume a Raspberry Pi4 as an IoT device and target it for anomalous behavior detection. Figure 1 shows the power consumption model. As stated in (Martinez et al., 2015), the power consumption of the Raspberry Pi4 can be divided into three

parts: application part, OS part, and hardware part.

We assume that the effect of anomalous behavior appears in the application power waveform. Let P_{app} denote the power consumed by the application program. The OS and hardware periodically perform relatively small operations, and the power consumed by them is called the steady-state power and is denoted by P_{steady}. Then, the total power due to P_{app} and P_{steady} is denoted by P_{total}. P_{total} can be represented by is a time-series power data, which is defined $P_{\text{total}} = \{p_i\}$. where p_i represents the power value at time t_i. Based on the above, the application power waveform P_{app} can be expressed as below:

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

2.3 Challenges for Anomalous Behavior Detection for IoT Devices with Steady-State Power

In detecting anomalous behavior of IoT devices including steady-state power waveform, anomalous behaviors have been so far detected by extracting the application power waveform from a complex total power waveform.

For example, the previous methods remove the steady-state power waveform from the total power waveform by averaging the measured steady-state power waveform or by estimating it using LSTM (Hasegawa et al., 2018; Takasaki et al., 2021b). However, the steady-state power waveform may not be removed accurately, in which case correct anomalous behavior detection results cannot be obtained. In addition, no method has been proposed to remove effective steady-state power waveform.

The proposed method, on the other hand, does not remove the steady-state power waveform, but instead employs the strategy to effectively use LSTM to learn the total power waveform to detect anomalous behavior.

3 DETECTION OF ANOMALOUS BEHAVIOR BY LSTM-BASED POWER WAVEFORM PREDICTION

In this section, we propose a method for detecting anomalous behavior from power waveforms of IoT devices by LSTM-based power waveform prediction. The detailed flow is shown below:

Step 1 (Measure the Power Waveforms):

As a first step, we measure the power waveform running the target application programs on the IoT device. The measured power waveform P'_{total} is time-series data. When the measured power value at time t_i is p'_i, $P'_{\text{total}} = \{p'_i\}$.

Step 2 (Smooth the Power Waveforms):

Measured power waveform is affected by noise and contains fluctuations. Smoothing removes such noise and fluctuations and makes it easier to capture the shape characteristics of the measured power waveforms. The proposed method uses the KZ filter (Koopmans, 1995) as a smoothing method and applies a simple moving average twice. In this way, the power waveform P'_{total} obtained in **Step 1** is smoothed and normalized. P_{total} shows the smoothed and normalized power waveform obtained in **Step 2**.

Step 3 (Training and Waveform Prediction for LSTM):

In **Step 3**, we explain a method for predicting power waveform using LSTM.

In LSTM, the input/output data are set as follows:

f [**Hz**]: Sampling frequency of the power data.

n_{in} : The number of LSTM inputs.

n_{out} : The number of LSTM outputs.

t_{in} [**s**]: The length of time corresponding to n_{in}.

t_{out} [**s**]: The length of time corresponding to n_{out}.

n_{in} and n_{out} can be re-written by

$$n_{\text{in}} = f \times t_{\text{in}} \tag{2}$$

$$n_{\text{out}} = f \times t_{\text{out}} \tag{3}$$

Step 3.1 (Training of LSTM):

In **Step 3.1**, total power waveform P_{total} is learned by LSTM.

LSTM is a type of recurrent neural network (RNN) that can learn long term dependencies such as time-series data (Al-Selwi et al., 2024). Let p_i be the power value at time t_i of P_{total}. $x_i = (p_i, p_{i+1}, ..., p_{i+n_{\text{in}}-1})$ of length n_{in} is the input and $y_i = (p_{i+n_{\text{in}}}, p_{i+n_{\text{in}}+1}, ..., p_{i+n_{\text{in}}+n_{\text{out}}-1})$ of length n_{out} is the training data. The training data is prepared by incrementing i by 1, and learned by LSTM. We use MSE (Mean Squared Error) (Wang and Bovik, 2009) for the loss function and train the LSTM so that the output of the LSTM is the same as the training data. In this way, when x_i is input, the LSTM outputs a power

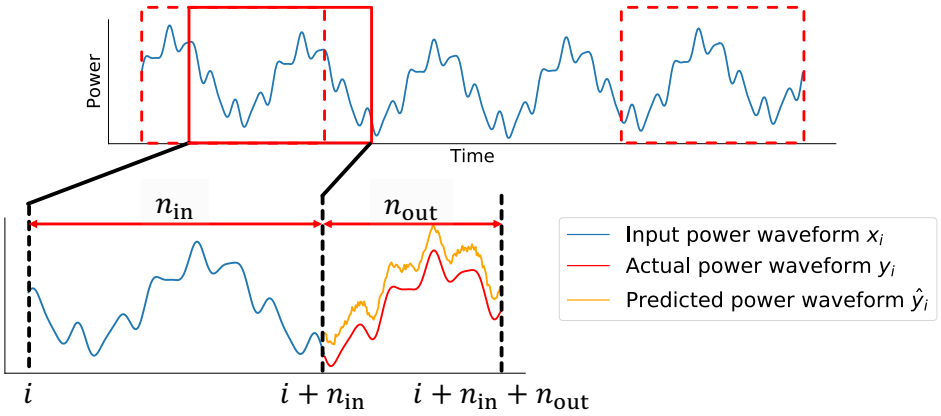

Figure 2: Waveform prediction by LSTM.

waveform close to y_i. n_{in} is the period for the dominant frequency of the power waveform (Ermshaus et al., 2023). n_{out} is set to $n_{\text{in}}/2$.

Since IoT devices usually repeat the same behavior, LSTM is expected to learn their behavior patterns. Since the proposed method does not use the golden model, there is a possibility that anomalous behaviors are contained in the training data, but it is difficult to learn such patterns since hardware Trojans are only activated under rare conditions (Salmani et al., 2012).

Step 3.2 (Waveform Prediction by LSTM):

In **Step 3.2**, the proposed method predicts the total power waveform in a sliding window manner of the target device using LSTM, which is depicted in Figure 2.

The partial waveform of length n_{in} is extracted from total power waveform and used as the input of LSTM. Then, LSTM takes this partial waveform $x_i = (p_i, p_{i+1}, ..., p_{i+n_{\text{in}}-1})$ of length n_{in} and predicts the subsequent partial waveform $\hat{y}_i = (\hat{p}_{i.i+n_{\text{in}}}, ...\hat{p}_{i.i+n_{\text{in}}+n_{\text{out}}-1})$ of length n_{out}, which must be similar to the training data y_i.

Step 4 (Normalization of Waveforms and Calculation of Differences):

In **Step 3**, LSTM learns and predicts the total power waveforms of the target device. In **Step 4**, the predicted power waveform $\hat{y}_i$ is compared with the actual power waveform y_i to detect anomalous behavior.

Let $\tilde{p}_i$ be the normalized power value at time i for p_i and $\tilde{\hat{p}}_{i.j}$ be the normalized power value at time j for $\hat{p}_{i.j}$. Then they are calculated by:

$$\tilde{p}_i = \frac{p_i - \text{norm\_min}_i}{\text{norm\_max}_i - \text{norm\_min}_i} \tag{4}$$

$$\tilde{\hat{p}}_{i.j} = \frac{\hat{p}_{i.j} - \text{norm\_min}_i}{\text{norm\_max}_i - \text{norm\_min}_i} \tag{5}$$

where

$$\text{norm\_min}_i = \min(\min(y_i), \min(\hat{y}_i)) \tag{6}$$

$$\text{norm\_max}_i = \max(\max(y_i), \max(\hat{y}_i)) \tag{7}$$

After normalizing y_i and $\hat{y}_i$ to $\tilde{y}_i = (\tilde{p}_{i+n_{\text{in}}}, ...\tilde{p}_{i+n_{\text{in}}+n_{\text{out}}-1})$ and $\tilde{\hat{y}}_i = (\tilde{\hat{p}}_{i.i+n_{\text{in}}}, ...\tilde{\hat{p}}_{i.i+n_{\text{in}}+n_{\text{out}}-1})$, respectively, the differences between them for all partial waveforms extracted from total power waveforms are calculated to obtain $Diff = \{d_1, d_2, ..., d_k\}$ where k is the total number of partial power waveforms. d_i is calculated using Equation (8).

$$d_i = \sum_{l=n_{\text{in}}}^{n_{\text{in}}+n_{\text{out}}-1} \left| \tilde{\hat{p}}_{i.i+l} - \tilde{p}_{i+l} \right| \tag{8}$$

If extracted partial waveforms that do not contain any anomalous behavior, the predicted waveform $\hat{y}_i$ and the actual power waveform y_i must be almost the same and d_i be small. On the other hand, it contains anomalous behavior, d_i becomes larger because LSTM has not learned the waveform pattern of anomalous behavior.

Step 5 (Detecting Anomalous Behavior)

In **Step 5**, anomaly scores for partial waveforms are calculated using $Diff$ obtained in **Step 4**, and anomalous behaviors are detected by the Hotelling theory.

Hotelling theory is a method to detect outliers in samples by calculating the anomaly score from the sample mean and variance. To detect outliers using the Hotelling theory, it is necessary to calculate the anomaly score using d_i for all partial waveforms. Let a_i be the anomaly score. Then, it is calculated below:

$$a_i = \left(\frac{d_i - \mu}{\sigma} \right)^2 \tag{9}$$

μ is the mean of $Diff$ and σ^2 is the variance of $Diff$. Hotelling theory assumes that the distribution of data

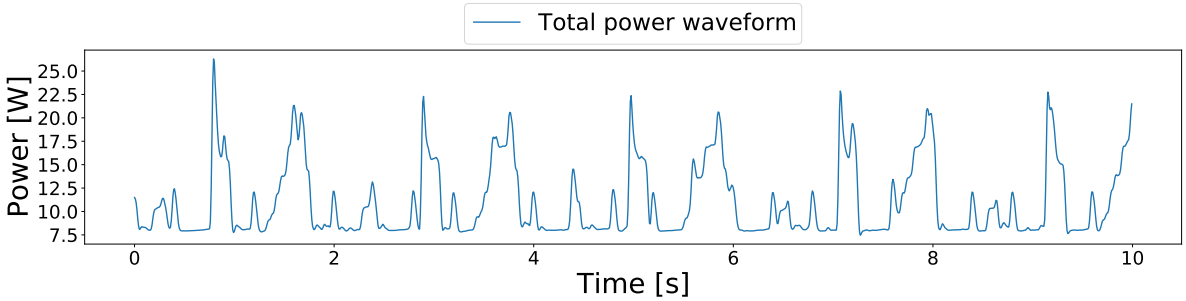

Figure 3: Total power waveform.

Table 1: Python libraries.

Library	Role	Version
tensorflow	Machine learning	2.3.0
Numpy	Data processing	1.18.1
Pandas	Data processing	1.1.5
scikit-learn	Anomaly detection	0.24.2
scipy	Statistical processing	1.4.1

follows a normal distribution, and the anomaly score calculated by Equation (9) follows a chi-square distribution with one degree of freedom.

Outliers are detected by determining the probability of occurrence of each anomaly. In the proposed method, the significance level is set to 1%, the threshold is the anomaly score that occurs with a probability of 1%, and an anomalous behavior is considered to exist in the partial waveform where the anomaly score larger than the threshold.

4 EXPERIMENTED IoT DEVICE

In this paper, we implemented an application on a Raspberry Pi4 model B as the target IoT device that performs two different operations below:

Normal Operation: Takes a picture and encrypts the image. Then, it turns the motor.

Anomalous Operation: Takes a picture and turns the motor without encrypting the image.

The application program running on Raspberry Pi4 repeats the operations above. In the normal operation, it takes a picture and encrypts a picture. Then, it turns the motor connected to it. The anomalous operation skips encryption process. The anomalous operation may be rarely run on the device, instead of the normal operation.

Table 2: Measuring devices.

Devices	Type	Role
Oscilloscope	Tektronix MSO64B	Measure a current and voltage
Current probe	Tektronix TCP0030A	Measure a current
Power supply	KEITHLEY 2280S-32-6	Supply power to IoT device

Table 3: Input/output shapes of LSTM.

Layer	Input shape	Output shape
LSTM	$1 \times n_{\text{in}}$	200
Dense	200	$1 \times n_{\text{out}}$

5 EVALUATION RESULTS

In this section, we measure the power waveform of Raspberry Pi4 during its operation, and evaluate its effectiveness in detecting anomalous behavior using the flow shown in Section 3.

5.1 Experimental Environment

The proposed method was implemented in Python 3.6.9 on the computer with a CPU of Intel Xeon Gold 6230R and the 1.5TB memory and evaluated through experimental evaluations. Table 1 and Table 2 summarize the Python libraries and the measuring devices that we used, respectively.

5.2 Extraction of Power Waveform and Training of LSTM

First, we measured the power waveform of the entire IoT device (**Step 1**). The sampling frequency f was 125kHz for 10 seconds, and 76 power waveforms were obtained in total. The measured data are extracted every 1000 samples and thus every power waveform is assumed to have 1250 power data (length 1250). Out of 76 measured power waveforms, one power waveform contains the anomalous behavior. Next, we smoothed the power waveforms (**Step 2**). A KZ filter was used for smoothing. The window size

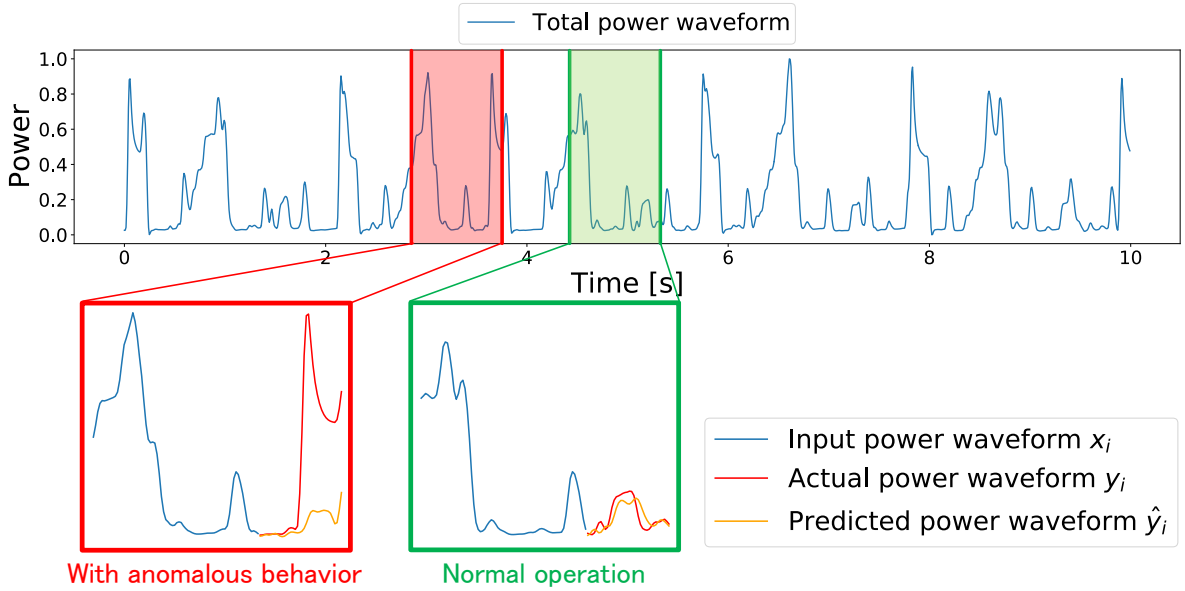

Figure 4: Total power waveform containing anomalous behavior and the results of waveform prediction by LSTM.

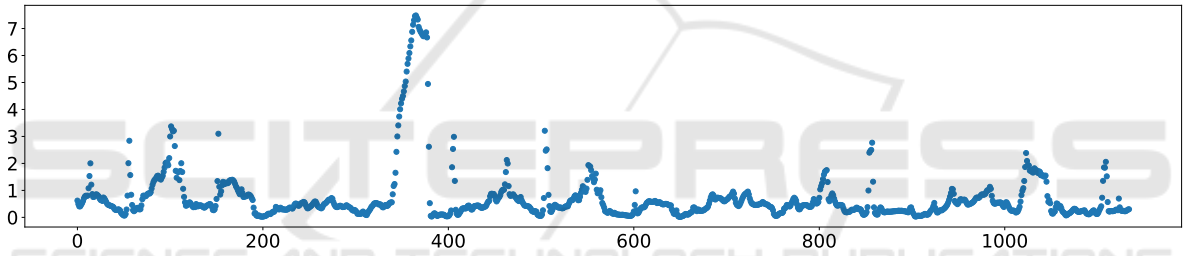

Figure 5: Difference between the partial waveform predicted by LSTM and the actual waveform in $Diff$.

of the KZ filter used for smoothing was $W = 3000$ for the first time and $W = 3000$ for the second time. Figure 3 shows an example of the total power waveform which is not contains anomalous behavior.

Of the 76 trained power waveform data, 59 were used for training and the remaining 19 were used for validation, and the power waveforms were learned by LSTM (**Step 3.1**). Table 3 shows a construction of LSTM and the parameters were $n_{\text{in}} = 76$ and $n_{\text{out}} = 38$. The input/output shape of LSTM is shown in Table 3. The number of epoch was set to 50. With the above parameters, LSTM learned the power waveforms of the target device.

5.3 Results of Waveform Prediction by LSTM

In this section, we show the results of prediction by LSTM. Figure 4 shows the total power waveform which contains anomalous behavior (red section) and partial waveforms with the result of LSTM prediction.

The trained LSTM predicted the waveforms following the input partial waveform (**Step 3.2**). Firstly, we pick up the green section in the total power waveform of Figure 4. The green section includes no anomalous behavior. The blue line in the green section shows the input partial power waveform x_i and the red line shows the actual power waveform y_i, following the input x_i. The orange line shows the waveform predicted by LSTM. In the green section, the red line and the orange line are nearly the same. On the other hand, we pick up the red section in Figure 4, which includes the anomalous behavior. In the same way, the red line shows the actual power waveform while the orange line shows the predicted on by LSTM. These two lines do not match well, indicating that the anomalous behavior exists. The LSTM predicted that small power consumption due to encryption should follow the input x_i, but the anomalous behavior skips it, resulting in a larger difference between the two waveforms.

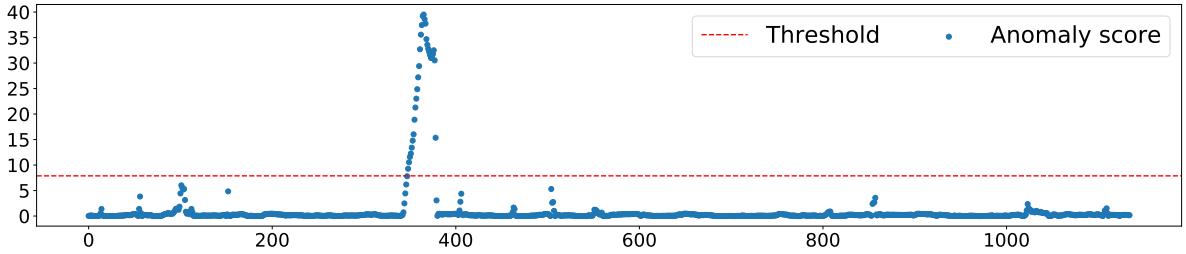

Figure 6: Anomaly score based on the prediction of waveforms for every partial power waveform.

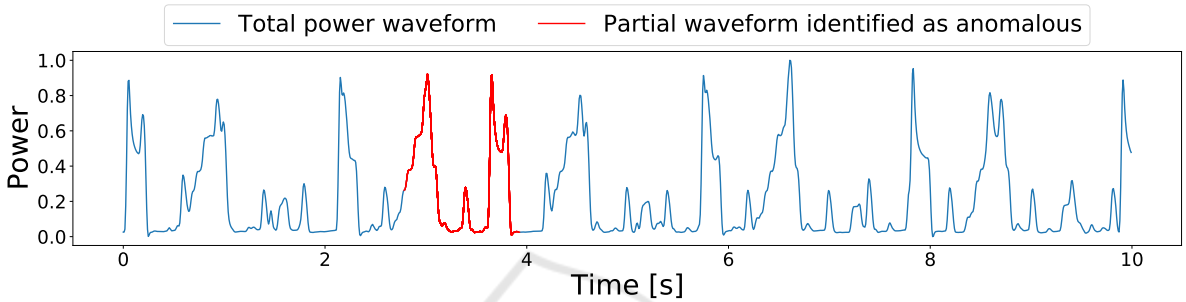

Figure 7: Result of anomaly detection.

5.4 Normalization of Waveforms and Results of Anomalous Behavior Detection

After prediction by LSTM, we calculate the difference between the actual waveform and the waveform predicted by LSTM for each partial power waveform.

In order to avoid underestimation or overestimation of the difference, we normalize partial waveform y_i and $\hat{y}_i$ using Equation (4) and Equation (5). Then, we calculate the difference between the waveform predicted by LSTM $\hat{y}_i$ and the actual waveform y_i using Equation (8) (**Step 4**). d_i in each partial power waveform is shown in Figure 5. From Figure 5, we can see that d_i is locally larger in certain areas. In order to objectively evaluate whether the partial waveforms which has larger d_i values contain anomalous behaviors or not, outliers of d_i are detected using Hotelling theory.

Anomaly score is calculated to use $Diff$ and Equation (9) (**Step 5**). The result is shown in Figure 6. The anomaly score of 7.9 is obtained, where the anomaly occurs with a probability of 1%. The red dotted in Figure 6 shows the anomaly score of 7.9.

The result of detecting anomalous behavior is shown in Figure 7. It can be seen that the section indicated as anomalous behavior in Figure 4 is identified as anomalous behavior. Therefore, the proposed method succeeded in detecting anomalous behavior.

6 CONCLUSION

In this paper, we proposed an anomalous behavior detection method for IoT devices based on the difference between LSTM prediction and total power waveforms and evaluated the effectiveness by applying the method to an IoT device built with Raspberry Pi4. Experimental results showed that the proposed method successfully detected an anomalous behavior on IoT device and indicates that the proposed method is effective as an anomaly detection method.

In the future, we will modify the machine learning structure and optimize hyper-parameters (hyper-parameter tuning) to improve accuracy.

ACKNOWLEDGEMENT

The results of this research were obtained through a contract research project (08101) sponsored by the National Institute of Information and Communications Technology (NICT).

REFERENCES

Agrawal, D., Baktir, S., Karakoyunlu, D., Rohatgi, P., and Sunar, B. (2007). Trojan detection using ic fingerprinting. In *Proc. 2007 IEEE Symposium on Security and Privacy (SP'07)*, pages 296–310.

Al-Selwi, S. M., Hassan, M. F., Abdulkadir, S. J., Muneer, A., Sumiea, E. H., Alqushaibi, A., and Ragab, M. G. (2024). Rnn-lstm: From applications to modeling techniques and beyond—systematic review. *Journal of King Saud University-Computer and Information Sciences*, page 102068.

Bhasin, S., Danger, J.-L., Guilley, S., Ngo, X. T., and Sauvage, L. (2013). Hardware trojan horses in cryptographic ip cores. In *Proc 2013 Workshop on Fault Diagnosis and Tolerance in Cryptography*, pages 15–29.

Bhunia, S., Hsiao, M. S., Banga, M., and Narasimhan, S. (2014). Hardware trojan attacks: Threat analysis and countermeasures. *Proceedings of the IEEE*, 102(8):1229–1247.

Chakraborty, R. S., Wolff, F., Paul, S., Papachristou, C., and Bhunia, S. (2009). Mero: A statistical approach for hardware trojan detection. In *Proc. International Workshop on Cryptographic Hardware and Embedded Systems*, pages 396–410.

Ermshaus, A., Schäfer, P., and Leser, U. (2023). Window size selection in unsupervised time series analytics: A review and benchmark. In *Proceedings of International Workshop on Advanced Analytics and Learning on Temporal Data*, pages 83–101. Springer.

Francq, J. and Frick, F. (2015). Introduction to hardware trojan detection methods. In *Proc. 2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 770–775.

HaddadPajouh, H., Dehghantanha, A., Khayami, R., and Choo, K.-K. R. (2018). A deep recurrent neural network based approach for internet of things malware threat hunting. *Future Generation Computer Systems*, 85:88–96.

Hasegawa, K., Yanagisawa, M., and Togawa, N. (2018). Detecting the existence of malfunctions in microcontrollers utilizing power analysis. In *Proc. 2018 IEEE 24th International Symposium on On-Line Testing And Robust System Design (IOLTS)*, pages 97–102.

Hotelling, H. (1992). *The generalization of Student's ratio*. Springer.

Islam, R., Tian, R., Batten, L. M., and Versteeg, S. (2013). Classification of malware based on integrated static and dynamic features. *Journal of Network and Computer Applications*, 36(2):646–656.

Jin, Y. and Makris, Y. (2008). Hardware trojan detection using path delay fingerprint. In *Proc. 2008 IEEE International workshop on hardware-oriented security and trust*, pages 51–57.

Koopmans, L. H. (1995). The spectral analysis of time series.

Martinez, B., Monton, M., Vilajosana, I., and Prades, J. D. (2015). The power of models: Modeling power consumption for iot devices. *IEEE Sensors Journal*, 15(10):5777–5789.

Oya, M., Shi, Y., Yanagisawa, M., and Togawa, N. (2015). A score-based classification method for identifying hardware-trojans at gate-level netlists. In *Proc. 2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 465–470.

Salmani, H., Tehranipoor, M., and Plusquellic, J. (2012). A novel technique for improving hardware trojan detection and reducing trojan activation time. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 20(1):112–125.

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

Takasaki, K., Kida, R., and Togawa, N. (2021b). An anomalous behavior detection method utilizing extracted application-specific power behaviors. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 104(11):1555–1565.

Wang, L., Xie, H., and Luo, H. (2013). Malicious circuitry detection using transient power analysis for ic security. In *Proc. 2013 International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering (QR2MSE)*, pages 1164–1167.

Wang, Z. and Bovik, A. C. (2009). Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE signal processing magazine*, 26(1):98–117.

Zaza, A. M., Kharroub, S. K., and Abualsaud, K. (2020). Lightweight iot malware detection solution using cnn classification. In *2020 IEEE 3rd 5G World Forum (5GWF)*, pages 212–217. IEEE.