# スマートホームシステムにおける Dempster-Shafer Theory に基づく異常検知手法の実装と評価

M2017SC010 竹田 拓哉
指導教員：河野 浩之

## 1 はじめに

現在，IoT デバイスの増加に伴い，不正アクセスや乗っ取りといった事件が発生し，2016 年 8 月に行われた「DEF CON」では 23 製品から 47 の脆弱性が発見された．

本研究では，スマートホームシステムに対し不正アクセスが発生した場合に Dempster-Shafer Theory (以下，DST) に基づいた Intrusion Detection System (以下, IDS) の検知手法を提案する．本研究での不正アクセスは，スクリプトを実行し IoT 機器を遠隔から不正に操作する攻撃を指す．不正アクセスに該当する不正スクリプトを用いたとき，平均パケット到着間隔，要求パケットバイト数，応答パケットバイト数の 3 つのパラメータが異常な値になることが Nobakht らによって報告されている [3].

そこで，スマートホーム内の IoT 機器とユーザーの持つスマートフォン等の情報端末を OpenFlow コントローラ (以下，コントローラ) と OpenFlow スイッチ (以下，スイッチ) を介して通信する環境を構築する．IDS では，パラメータ毎の危険度を DST の組み合わせ規則で統合し総合的に危険度判定する異常検知手法である Risk Parameters-Dempster-Shafer Theory(RP-DST) 手法 [5] を用いる．通常の通信と判別した際は本来の宛先に転送し，不正アクセスと判別した際はパケットを破棄するシステムを構築する．

2 章では，IoT 機器の脆弱性，IDS の異常検知アルゴリズムについて述べる．3 章で OpenFlow を用いた異常検知システムの概要について，4 章でデータセットによる RP-DST 手法の性能評価について説明する．5 章では，OpenFlow での RP-DST 手法の性能評価について説明する．6 章ではまとめを行う．

## 2 IoT 機器と異常検知アルゴリズム

2.1 節では，IoT 機器の脆弱性について述べる．2.2 節では，IDS の異常検知アルゴリズムについて述べる．

### 2.1 IoT 機器の脆弱性

IPA の報告によると，スマートハウスにおける各機器の情報を不正に取得された場合，個人のプライベートデータの漏えいという脅威に加えて取得したデータ（消費電力や施錠状況）を悪用される可能性がある [4]．利用者に不利益が被る可能性があるとし，こうした攻撃を防ぐ必要がある．

### 2.2 異常検知手法

Nobakht らは，スマートホームシステムへの不正アクセスの事例を紹介し，IoT-IDM と呼ばれる異常検知，軽減フレームワークを提案している [3].

また IDS の異常検知アルゴリズムの Data Fusion を用いた研究は Li らがまとめている [1]．Data Fusion 技術の 1 つである DST を IDS に用いた研究も進められている，Madalina らはクラウド環境で DST を適用しフォルトツリーで原因を分析し，偽陰性率と偽陽性率を低下させ精度の高い IDS を実現している [2].

RP-DST 手法は，DST に基づく異常検知手法であり，通信ログから得られるパラーメータ毎の危険度を統合し，危険度測定している．NSL-KDD Dataset を用いて性能評価をしており，低い偽陽性率と偽陰性率を実現している [5].

## 3 OpenFlow を用いた異常検知システムの概要

3.1 節では提案するスマートホームシステムについて，3.2 節で OpenFlow メッセージについて述べる．3.3 節では，異常検知手法である RP-DST 手法について述べる．

### 3.1 提案するスマートホームシステムの概要

本研究では，携帯端末からスマートホームシステムへの接続時に OpenFlow を介して通信するネットワークトポロジを提案する．OpenFlow を実装したホストで通信の安全性を判別し，危険と判別した場合には通信を遮断する．IDS を含むスマートホームシステムのシステム概要を図 1 に示す．このシステムでは，外部ネットワークからスマートホームシステム内の IoT 機器を操作する通信のみを対象とし危険度を判定する事を想定している．スイッチに到着したパケットの処理フローを以下に示す．

1. スイッチ内のフローテーブルを参照し，到着したパケットの処理が決められたルールがあれば 9. を実行する．
2. スイッチは Packet In メッセージをコントローラに送信する．
3. スイッチに届くパケットを IDS にミラーリングする．
4. RP-DST 手法を採用した IDS によって危険度を測定する．
5. IDS はコントローラに判別結果を出力する．
6. コントローラは判別結果を読み込み，新たなルールを作成する
7. Flow Mod メッセージを基にスイッチのフローテーブルにルールを書き込む．
8. Packet Out メッセージよりパケットの情報をスイッチに送り返す．

9. スイッチは到着したパケットに対しフローテーブルの条件に合ったアクション (Forward，Drop) を実行する．

スイッチは Packet In メッセージ送信後は，スイッチからヘッダフィールドを含む情報を送信し，その後スイッチはコントローラからアクションを指定する Flow Mod メッセージを受信するまで待機する．

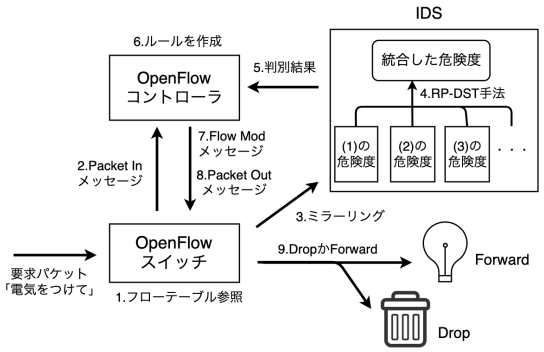


図 1 スマートホームシステムの概要

Flow Mod メッセージを作成するために，スイッチに到着したパケットを IDS を実装したマシンにミラーリングすることによって，異常検知に必要な通信ログのパラメータを測定している．IDS では，RP-DST 手法を採用し，先述した通信ログから得られるパラメータの危険度を統合し，判別結果を出力する．

### 3.2 OpenFlow メッセージ

スイッチはフローテーブルを保持しており，アクションが決められたルールが書き込まれている．到着したパケットに対して，ヘッダフィールド (IP アドレスなどの識別子) が一致するルールで指定されたアクションを実行する．フローテーブルを更新するための主要なメッセージは次の 3 つである．

Packet In メッセージ
: 到着したパケットに一致するルールがフローテーブルに存在しなかった場合，スイッチから到着パケットのヘッダーフィールドなどの情報をコントローラへと送るために用いられる．

Flow Mod メッセージ
: コントローラが作ったルールをスイッチのフローテーブルに書き込むために用いられる．ルールを設定してから一定時間後に消すためのハードタイムアウトと，フローが参照されない時間が一定時間経過した後に消すためのアイドルタイムアウトを設定できる．

Packet Out メッセージ
: Packet In メッセージでコントローラ に送られてきたパケットを本来の宛先に送るために，スイッチへと送り返す際に用いられる．

### 3.3 RP-DST 手法

DST は不確実な要素に確率を割り当て，2 つの独立した情報を統合できるという特徴を持つ定理である．DST の組み合わせ規則は，パラメータが 2 つの場合の統合アルゴリズムであり，$m_1$ と $m_2$ の統合結果を $m_{12}$ のように表わす．パラメータの数が n(3 以上) の組み合わせの場合，$m_{12}$ に対し $m_3$，$m_4$，$\cdots$，$m_n$ を順に DST の組み合わせ規則を用いて統合していくことで求められる．パラメータの数が n のとき，統合に用いる式を以下の (1)〜(6) 式を示す．まず予備割当関数 $q_{12\cdots n}$ は (1)〜(3) 式で計算する．

$$q_{12\cdots n}(\emptyset) = m_{12}(T)\prod_{i=3}^{n} m_i(F) + m_{12}(F)\prod_{i=3}^{n} m_i(T) \tag{1}$$

$$\begin{aligned} q_{12\cdots n}(T) = {} & m_{12\cdots n-1}(T)m_n(T) + \\ & m_{12\cdots n-1}(T)m_n(T,F) + m_{12\cdots n-1}(T,F)m_n(T) \end{aligned} \tag{2}$$

$$\begin{aligned} q_{12\cdots n}(F) = {} & m_{12\cdots n-1}(F)m_n(F) + \\ & m_{12\cdots n-1}(F)m_n(T,F) + m_{12\cdots n-1}(T,F)m_n(F) \end{aligned} \tag{3}$$

求めた $q_{12\cdots n}$ を $m_{12\cdots n}$ に変換する計算方法を (4)〜(6) 式に示す．パラメータが 2 つの場合と同様に $q_{12\cdots n}$ を $q_{12\cdots n}(\emptyset)$ で正規化し，$m_{12\cdots n}$ を計算する．

$$m_{12\cdots n}(T) = \frac{q_{12\cdots n}(T)}{1 - q_n(\emptyset)} \tag{4}$$

$$m_{12\cdots n}(T,F) = \frac{m_{12}\prod_{i=3}^{n} m_i(T,F)}{1 - q_{12\cdots n}(\emptyset)} \tag{5}$$

$$m_{12\cdots n}(F) = \frac{q_{12\cdots n}(F)}{1 - q_{12\cdots n}(\emptyset)} \tag{6}$$

提案手法である RP-DST 手法は，このように DST の組み合わせ規則に基づきパラメータ毎の危険度の統合する手法である．RP-DST 手法により求めた $m_{12\cdots n}$ は危険度判定に用いる．

## 4 データセットによる RP-DST 手法の性能評価

4.1 節では RP-DST 手法の性能評価に用いるデータセットと前処理について，4.2 節でデータセットを用いた性能評価について述べる．

### 4.1 IoT ネットワークにまつわるデータセットと前処理

UCI Machine Learning Repository からインストールした detection_of_IoT_botnet_attacks_N_BaIoT Data Set を用いて RP-DST 手法の性能評価を行う．また同じデータセットを使用した SVM で性能評価を行い，RP-DST 手法と SVM との性能比較を行う．UCI Machine Learning Repository はカリフォルニア大学アーバイン校が運営しており，機械学習やデータマイニングで扱うサンプルデータを集めた配布サイトである．また detection_of_IoT_botnet_attacks_N_BaIoT Data Set は IoT 機器をターゲットにしたマルウェアである gafgyt，mirai を任意のデバイスに感染させ，そのときの通信の特徴を記録したデータデットである．データセット内には 9 種類のデバイスが保存されており，Danmini_Doorbell のようにメーカー名と製品名がセットとなりデータセットとして保存されている．

複数あるデータセットの中から，データ数が最も多い Philips_B120N10_Baby_Monitor を選択した．このデータセット内のアノマリデータは gafgyt の 5 種類 (combo:58152 インスタンス，junk:28349 インスタンス，scan:27859 インスタンス，tcp:92581 インスタンス，udp:105782 インスタンス) と，mirai の 5 種類 (ack:91123 インスタンス，scan:103621 インスタンス，syn:118128 インスタンス，udp:217034 インスタンス，udpplain:80808 インスタンス) がある．ノーマルデータの数は 160138 インスタンスである．

ノーマルデータと各攻撃 5 種類ずつの計 10 種類のアノマリデータを 10 分の 1 スケールにランダムサンプリングした．アノマリデータとノーマルデータの半分をトレーニングデータ，もう半分をテストデータとした．

### 4.2 データセットによる性能評価

性能評価する際に使用する特徴量は weight，mean，variance を選択した．weight は平均パケット到着数，mean は平均パケットバイト数，variance はパケットバイト数の分散を表している．4.1 節で説明したトレーニングデータの中から特徴量を SVM と RP-DST 手法に学習させ，テストデータによる性能を行った．性能比較を表 1，表 2 に示す．

表 1 RP-DST 手法と SVM の性能比較

| | 正解率 | 偽陽性率 | 偽陰性率 |
|---|---|---|---|
| SVM | 99.97% | 0.03% | 0.00% |
| RP-DST 手法 | 99.82% | 1.13% | 0.02% |

表 1 より，RP-DST 手法に比べ，SVM が正解率，偽陽性率，偽陰性率において優れた結果となった．しかし，RP-DST 手法の性能として正解率は 99.82% となり，偽陰性率は 0.02% に抑えることができた．

表 2 は学習時間と応答時間をまとめた．RP-DST 手法では，学習時間が 0.09(sec) であり，応答時間が 0.14(sec) であった．これら 2 つの指標より，RP-DST 手法の方が SVM より高速に処理を行えることがわかった．

表 2 RP-DST 手法と SVM の学習時間と応答時間

| | 学習時間 | 応答時間 |
|---|---|---|
| SVM | 7.56($sec$) | 2.52($sec$) |
| RP-DST 手法 | 0.09($sec$) | 0.14($sec$) |

## 5 OpenFlow 環境での RP-DST 手法の性能評価

5.1 節で IDS を含むスマートホームシステムの概要について，5.2 節で実験環境について説明する．5.3 節で RP-DST 手法による異常検知，5.4 節でトレーニングデータの作成について説明する．5.5 節でスマートホームシステムでの RP-DST 手法の性能評価について説明する．

### 5.1 IDS を含むスマートホームシステムの概要

OpenFlow を含む実験環境を図 2 に示す．OpenFlow スイッチからミラーリングされた通信は，C 言語プログラムでパケットキャプチャしている．パケットキャプチャして得られたデータは SQLite3 で実装したデータベースに保存する．IDS で保存されたデータを抽出し，危険度測定する．

また RaspberryPi 上に WebIOPi を実装している．WebIOPi は RaspberryPi 上で動作する web サーバであり，RaspberryPi を IoT として動作させるためのソフトウェアの 1 つである．ブラウザからリモートで GPIO の ON/OFF を操作するような web ページを作成し，操作用端末でアクセスしている．

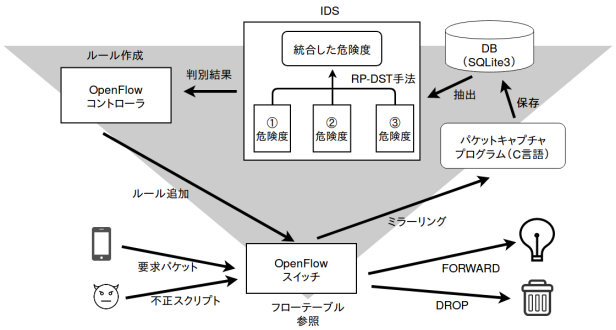


図 2 OpenFlow を含む実験環境

### 5.2 実験環境

OpenFlow の実装環境は次の通りである．コントローラのフレームワークには Trema を選択した．開発，テスト，デバッグツールを備えており，Ruby と C 言語に対応している．スイッチのフレームワークには Open vSwitch を選択した．VMware によって開発されており，Apatch2.0 ライセンスで提供している．

表 3 OpenFlow の実装環境

| | |
|---|---|
| OS | Ubuntu14.04 |
| プロセッサ | AMD FX-8350 Eight-Core |
| メモリ | 16GB |
| コントローラ | Trema v.0.4.6 |
| スイッチ | Open vSwitch v.2.5.0 |

### 5.3 RP-DST 手法による異常検知

学習に使用する特徴量は，平均パケット到着数，要求パケットバイト数，転送速度 (bps) である．これらの特徴量は先行研究 [3] を考慮し，選択している．

まず平均パケット到着間隔における危険度 ($m_a$) を測定する．通常の通信である確率を $m_a(normal)$，異常な通信である確率を $m_a(anomaly)$，どちらか判別できない確率を $m_a(normal, anomaly)$ と表す．$m_a(normal)$ と $m_a(anomaly)$ の値が 0.5 に近い値ほど，$m_a(normal, anomaly)$ が高くなるよう設定しており合計は 1 である．要求パケットバイト数の危険度 ($m_b$)，転送速度の危険度 ($m_c$) も同様に算出する．$m_a$，$m_b$，$m_c$ を RP-DST 手法で統合し $m_{abc}$ を算出し，$m_{abc}(normal)$，$m_{abc}(anomaly)$ の値を比較し，通常か異常かの判別を行う．

また RP-DST 手法の実装は，java でコーディングしており約 400 行程度のプログラムである．

### 5.4 トレーニングデータの作成

4 章で構築した OpenFlow 環境での RP-DST 手法を実装した異常検知システムの性能を評価する．

図 2 の環境でスイッチで素通しの設定を行い，1 時間にわたり RestAPI への接続を含む通信ログのデータをパケットキャプチャで収集した．このデータの中には不正アクセスを想定したスクリプトを 600 回実行したデータも含まれている．この不正アクセスは hue_blackout.bash と呼ばれるマルウェアの挙動を模擬した Python スクリプトを作成し実行している．不正アクセスのデータはアノマリデータとしてラベル付けを行い，その他のデータはノーマルデータとしてラベル付けをした．

### 5.5 スマートホームシステムでの RP-DST 手法の性能評価

アノマリデータ数が 600 インスタンス，500 インスタンス，400 インスタンスのトレーニングデータを作成し，実験を行った．それぞれのトレーニングデータを RP-DST 手法で学習し，通常な通信や不正スクリプトを実行した際の異常検知に関する性能を表 4 に示す．

アノマリデータ数が 600 に近くにつれ，正解率は向上した．アノマリデータ数が 500 インスタンスの時，95.34% の正解率であった．アノマリデータ数を 600 インスタンスに増やしたとき，偽陰性率が 7.75% から 2.58% に改善され，正解率が 95.70% に向上した．

表 4 スマートホームシステムでの性能評価

| アノマリデータの数 | 正解率 | 偽陽性率 | 偽陰性率 |
|---|---|---|---|
| 400 | 94.57% | 4.40% | 19.19% |
| 500 | 95.34% | 4.43% | 7.75% |
| 600 | 95.70% | 4.43% | 2.58% |

## 6 終わりに

本研究では，DST に基づく異常検知手法である RP-DST 手法を提案した．実験では IoT ネットワークにまつわるデータセットを使用した．正解率の比較を行うと SVM は 99.97% であり，RP-DST 手法は 99.82% であった．SVM の方が高い精度を示したが RP-DST 手法も高い正解率となった．また学習時間，応答時間を測定した結果，SVM より RP-DST 手法の方が高速に処理できる．

OpenFlow 環境で RP-DST 手法を実装した IDS を構築した．ネットワーク上を流れる通信をキャプチャし，不正スクリプトによるスマートホーム IoT の不正操作を検出する実験を行った．トレーニングデータに含まれるアノマリデータが 500 インスタンス，600 インスタンスのとき正解率が 95% 以上になった．

RP-DST 手法を IDS に採用することで，400 行程度のコード数で高い正解率となることに加え，パラメータチューニングにかかる時間的コストが減るメリットが挙げられる．さらに応答時間の速さにより大量の通信が発生する場合でも対応できる可能性がある．

## 参考文献


[1] Li, G. et al.: Data Fusion for Network Intrusion Detection: A Review, *Security and Communication Networks*, pp.1–16 (2018).

[2] Lonea, A. et al.: Detecting DDoS Attacks in Cloud Computing Environment, *International Journal of Computers Communications Control*, Vol.8, pp.70–78 (2012).

[3] Nobakht, M. et al.: A Host-based Intrusion Detection and Mitigation Framework for Smart Home IoT using OpenFlow, *2016 11th International Conference on Availability, Reliability and Security*, pp.147–156 (2016).

[4] 技術本部セキュリティセンター: IoT 開発におけるセキュリティ設計の手引き，技術報告，独立行政法人情報処理推進機構 (accessed Jun. 2018). https://www.ipa.go.jp/security/iot/iotguide.html.

[5] 竹田拓哉ほか: スマートホームシステムにおける Dempster-Shafer Theory に基づく異常検知手法の提案, 情報科学技術フォーラム講演論文集，Vol.4, No.17, pp.153–156 (2018).