修士論文

IoT ネットワークにおける通信挙動の複数エントロピーを用いた軽量型侵入検知手法

桂 祐成
奈良先端科学技術大学院大学
先端科学技術研究科
情報理工学プログラム

主指導教員: 藤川 和利 教授
情報基盤システム学研究室（情報科学領域）

令和 4 年 1 月 24 日提出

本論文は奈良先端科学技術大学院大学先端科学技術研究科に
修士 (工学) 授与の要件として提出した修士論文である。

桂 祐成

審査委員：

藤川 和利 教授	（主指導教員，情報科学領域）
飯田 元 教授	（副指導教員，情報科学領域）
門林 雄基 教授	（副指導教員，情報科学領域）
新井 イスマイル 准教授	（副指導教員，情報科学領域）

IoT ネットワークにおける通信挙動の複数エントロピーを用いた軽量型侵入検知手法*

桂 祐成

内容梗概

2016 年にマルウェア Mirai のソースコードが公開されて以降，様々な亜種が作られ，現在もそれらの感染活動が確認されている．このマルウェアは，家庭用のルータや Web カメラなどの IoT(Internet of Things) 機器に感染するマルウェアであり，それらの機器を悪用した DDoS 攻撃の規模は 600Gbps を超え現在も猛威を奮っている．このような背景から IoT ネットワークにおける機械学習を活用した侵入検知手法に関する研究が盛んに行われている．しかし，既存研究で活用されている手法では，要求されるリソースが高いため，リソースに制約のある IoT ゲートウェイなどでは，侵入検知システムを動作させることが困難である．本研究では，正常時に特定のサーバと定期的に通信するといった IoT 機器の通信挙動に着目し，ホスト毎の通信挙動を複数のエントロピー (宛先ポート番号，送信元ポート番号，送信時間間隔等) を用いて表すことを提案する．IoT 機器の正常時の通信挙動を捉えることで，既存手法よりも少ない特徴量で軽量な機械学習アルゴリズムを用いた場合でも同等の侵入検知性能を維持することが可能となる．評価の結果，提案手法を用いた場合，現状想定されている IoT ゲートウェイ上で本来の目的に影響を与えずに動作できることを確認した．

キーワード

IoT ネットワーク, 侵入検知システム, 機械学習, ネットワークセキュリティ, マルウェア

*奈良先端科学技術大学院大学 先端科学技術研究科 修士論文, 令和 4 年 1 月 24 日.

Lightweight Intrusion Detection Using Multiple Entropy Features of Traffic Behavior in IoT Networks*

Yusei Katsura

Abstract

Since the release of the source code of the Mirai malware in 2016, various variants have been created. Their infection activities are still being observed. This malware infects IoT (Internet of Things) devices such as home routers, webcams, etc. The scale of DDoS attacks using Mirai-infected IoT devices has exceeded 600 Gbps. There has been a lot of research on intrusion detection methods using machine learning in IoT networks. However, the resources required by existing methods are high. Therefore, it is challenging to run intrusion detection systems on resource-limited IoT gateways. In this research, we focus on the communication behavior of IoT devices, such as periodic communication with a specific server during benign operations. We propose to represent the communication behavior of each host using multiple entropy features (destination port number, source port number, transmission time interval, etc.). The proposed method can maintain the same intrusion detection performance even when using a lightweight machine learning algorithm with fewer features than existing methods by capturing the communication behavior of IoT devices during benign operations. The evaluation results show that the proposed method can operate on the assumed IoT gateway without affecting the actual operation.

*Master's Thesis, Graduate School of Science and Technology, Nara Institute of Science and Technology, January 24, 2021.

Keywords:

IoT Network, Intrusion Detection, Machine Learning, Network Security, Malware

目 次

図 目 次

表 目 次

1. はじめに

2016 年にマルウェア Mirai のソースコード[1]が公開されて以降，様々な亜種が作られ，現在も感染活動が確認されている [1]．このマルウェアは，家庭用のルータや Web カメラなどの IoT (Internet of Things) 機器を踏み台にして 600Gbps の規模を超える DDoS 攻撃を実施する [2] など，現在も猛威を奮っている．このような状況に対して，総務省，国立研究開発法人 情報通信研究機構，国内のインターネットサービス事業者が連携し，ID/Password に不備がある IoT 機器の調査及び当該機器の利用者に対する注意喚起を行う NOTICE (National Operation Towards IoT Clean Environment)[2]が 2019 年 2 月より実施されている．NOTICE プロジェクトが 2021 年 2 月から 5 月にかけて実施した調査によると平均約 65,000 の IoT 機器がポート待ち受け状態であり，これらのうち平均 8,400 の IoT 機器においてバナー収集が可能な状態であった．さらに，これらのバナー収集が可能な IoT 機器の内，約 2,000 の IoT 機器は，パスワード設定の不備でアクセスが可能な状態であった．ログイン可能な IoT 機器は，Mirai の実行に必要なプログラムがなかったために Mirai に感染することはなかったが，ログイン可能な IoT 機器の約 8 割で，任意のプログラム実行や任意の宛先に対する通信，設定項目の閲覧，変更が可能であった．パスワードの設定に不備がある IoT 機器は悪用されるリスクが高く，このような IoT 機器の脆弱性を狙った攻撃を早期に検知し，対処することは攻撃の被害を最小限に抑える上で重要となる．

しかし，IoT 機器は通信規格やプロトコルスタックが従来のネットワークと異なる，計算リソースが限られているなどの制約から，従来のセキュリティ対策をそのまま適用することは難しい．そのため，IoT 機器に特化したセキュリティ対策が不可欠である．IoT 機器のセキュリティを強化する方法として，データの機密性と認証を提供する方法や IoT ネットワーク内のアクセス制御などを行う方法が提案されている [3]．

また，攻撃者を検知するための仕組みとして従来のネットワークでも利用され

[1]jgamblin, Mirai-source-code, https://github.com/jgamblin/Mirai-Source-Code

[2]NOTICE サイバー攻撃に悪用されるおそれのある IoT 機器の調査、注意喚起を行うプロジェクト, https://notice.go.jp

る侵入検知システムを用いる方法が提案されている．侵入検知システムは，ホストやネットワークの動作を監視し，セキュリティポリシーに違反する挙動を検知するとシステム管理者に警告を発する．侵入検知システムが用いる侵入検知手法には，あらかじめ定義された攻撃のパターンを捕捉し，攻撃の検知を行うシグネチャベースの侵入検知手法と，正常なパターンをあらかじめ学習し，そのパターンから外れた挙動を異常として検知するアノマリベースの侵入検知手法が存在する．シグネチャベースの侵入検知は，既知の攻撃に対して高い精度で攻撃を検知できるが，未知の攻撃に対しては検知精度が低くなる．アノマリーベースの侵入検知は，正常なパターンをあらかじめ学習し，それと異なるパターンを異常として検知するため，未知の攻撃に対しても有効である．そのため，機械学習を用いたアノマリベースの侵入検知手法に関する研究が数多く行われている．

しかし，機械学習を活用した従来の侵入検知手法の場合，求められるリソースが多いため，IoT ゲートウェイ上で動作させることは困難である．IoT ゲートウェイには，一般的なゲートウェイと比較して CPU の性能が低く，メモリ容量が小さいといったような計算リソースの制約がある．このような IoT ゲートウェイの特徴により，計算リソースの制約を考慮しない従来の侵入検知手法を IoT ゲートウェイ上で動作させる場合，パケット転送などの IoT ゲートウェイ本来の処理に悪影響を与える．そこで，本研究では計算リソースに制約のある IoT ゲートウェイ上で動作させることを想定して侵入検知手法を軽量化することを目指す．具体的な目標として，侵入検知処理におけるメモリ使用量の削減と検知処理時間の短縮を行う．

2. 想定環境

本章では，まず，侵入検知と侵入検知システムの定義，種類について説明し，本研究で用いる侵入検知手法の種類について説明する．次に，本研究で対象とするIoTアーキテクチャについて述べる．

2.1 侵入検知手法と侵入検知システム

米国国立標準技術研究所 (National Institute of Standards and Technology: NIST) の定義 [4, 5] では，侵入検知を “コンピュータまたはネットワークに発生するイベントを監視し，それらを分析することによって，インシデントと考えられる兆候を検知するプロセス”，侵入検知システムを “侵入検知プロセスを自動化するソフトウェア” と定義している．本研究ではこれらの定義を採用する．

一般的な侵入検知手法は，シグネチャベースとアノマリーベースの侵入検知手法に分類できる．さらに，そうした侵入検知手法を利用する侵入検知システムは設置方法の観点から，ホスト型とネットワーク型の侵入検知システムに分類できる．以降では，侵入検知手法と侵入検知システムの設置方法について説明する．

- シグネチャベースの侵入検知手法
  シグネチャベースの検知方法は既存の攻撃パターンを記したシグネチャに基づいて，コンピュータ上のプロセスやネットワークトラフィックの挙動を検査し，シグネチャと一致したものを検知する方法である．シグネチャベースの検知方法は，既知の攻撃パターンに対しては有効であるが，未知の攻撃パターンや回避方法を駆使した攻撃を検知することが困難である．

- アノマリベースの侵入検知手法
  アノマリベースの検知方法は，正常な挙動としてあらかじめ記録したプロファイルに基づいて，コンピュータ上のプロセスやネットワークトラフィックの挙動を検査し，プロファイルと異なる挙動を検知する方法である．この検知方法で使用されるプロファイルは，コンピュータ上のプロセスやネットワーク接続，システムログ等をもとに作成される．アノマリベースの検

知方法は，シグネチャベースの検知方法と異なり，正常な挙動かどうかに注視して検知を行うため，未知の攻撃パターンに対しても有効である．そのため，アノマリーベースの侵入検知手法が盛んに研究されている．

- ホスト型侵入検知システム
  ホスト型侵入検知システムは，監視対象とするホスト上に直接設置するシステムである．ホスト型侵入検知システムでは，監視対象とするホストの実行プロセスやログ，ネットワークトラフィックを監視する．

- ネットワーク型侵入検知システム
  ネットワーク型侵入検知システムは，監視対象とするネットワーク上に設置するシステムである．ネットワーク型侵入検知システムでは，監視対象とするネットワーク上のパケットを収集し，解析することで侵入検知を行う．

2.2 侵入検知手法と侵入検知システムのまとめ

本節ではIoT機器の特性について説明し，IoT機器に適した侵入検知手法と侵入検知システムについて議論する．

シグネチャベースの侵入検知手法では既知の攻撃パターンを記したシグネチャに基づいて侵入検知を行うため，脆弱性の発見から修正パッチが適用されるまでの間に行われるゼロデイ攻撃は検知することが困難である．また，IoT機器はセンサーやIPカメラ等様々な種類のIoT機器が存在するため，全ての攻撃パターンを記録することは困難であるため，IoT機器の侵入検知手法ではシグネチャベースの侵入検知手法は不適切であると考える．

IoT機器によって取得したデータはインターネットを介してクラウド上のサーバに送信されるため，IoT機器が通信する宛先は限られている．また，IoT機器は実装されたプログラムに従って一定の時間間隔でデータの送信を行うため，IoT機器の正常時の通信挙動は様々なサーバと通信を行う機器と比較して正常時の挙動を捉えることが容易であるため，アノマリベースの侵入検知手法が最適であると考える．

ホスト型侵入検知システムは監視対象とするホスト上に侵入検知システムを設置するため，計算リソースに制約のあるIoT機器上では動作させることは困難であるため，IoT機器の侵入検知ではホスト型侵入検知システムは不適切であると考える．

ネットワーク型侵入検知システムは監視対象とするネットワーク上に侵入検知システムを設置するため，IoT機器の性能に依存せずに侵入検知処理を行うことができるため，IoT機器の侵入検知ではネットワーク型侵入検知システムは適切であると考える．

2.3 想定するIoTアーキテクチャの構成

本研究で想定するIoTアーキテクチャはIoT機器が収集したデータをネットワークを介してサーバーに送信するIoT層，収集したデータに基づいて迅速に応答を返す処理を実行するFog層，収集したデータを保管，分析し，アプリケーションの実行等を行うCloud層で構成されるIoTアーキテクチャを想定する．

IoT層

IoT層はセンサ等のIoT機器で構成されており，収集したデータはIoTゲートウェイを介してFog層またはCloud層のサーバに送信する．

Fog層

Fog層はIoT層から送信されたデータを受信し，そのデータを処理またはCloud層に送信する役割を担う．収集したデータに基づいた迅速な応答が求められる場合，この層で処理が実行される．

Cloud層

Cloud層はIoT層またはFog層から送信されたデータの保管と分析を行う役割を担う．

本研究ではIoT層で動作するIoTゲートウェイ上で侵入検知を実行することを想定する．IoT機器は工業製品として同じ製品が数多く市場に供給されている．そのため，特定のIoT機器に脆弱性が見つかりマルウェアの感染活動が行われ

ると同様の脆弱性を持つ IoT 機器が連鎖的にマルウェアに感染する．IoT ゲートウェイよりも上位の層で侵入検知を行う場合，IoT ゲートウェイに接続されている他の IoT 機器がマルウェアの感染リスクに晒される．IoT ゲートウェイ上で侵入検知を行い，異常な通信を行う IoT 機器を早期に検知し，対処することができればマルウェアの感染による二次被害を防ぐことが可能となる．

2.4 IoT ゲートウェイ

IoT ゲートウェイは IoT 層内の IoT 機器とクラウド上のサーバとの通信を中継するための機器である．IoT ゲートウェイを活用することで，インターネット接続機能を IoT 機器自体に保持する必要がなくなる．そのため，IoT ネットワークの構築にかかるコストの削減や接続管理等の運用面での利便性が向上する．その一方で，IoT ゲートウェイは比較的安価な組み込み機器を用いるため，計算リソースに限りがある．

表 1 に IoT ゲートウェイとして販売されている製品のスペックを示す．IoT ゲートウェイ上で侵入検知を行う場合，本来の IoT ゲートウェイとしての処理動作があるため，実際に使用可能な CPU やメモリといった計算リソースは表 1 の値よりも小さくなる．このようなスペックの IoT ゲートウェイ上で計算リソースの制約を考慮しない従来の侵入検知手法を IoT ゲートウェイ上で動作させる場合，パケット転送などの IoT ゲートウェイ本来の処理に悪影響を与える．そのため，IoT ゲートウェイ上での侵入検知を行うためには軽量な侵入検知手法が必要となる．

[3]屋外型 IoT ゲートウェイ, https://www.co-nss.co.jp/energy/gateway/
[4]OpenBlocks IoT VX2, https://www.plathome.co.jp/product/openblocks-iot/vx2/
[5]CONEXIOBlackBear, https://conexio-iot.jp/serviceproduct/blackbear.html

表 1 IoT ゲートウェイのスペック

	屋外型 IoT ゲートウェイ (日新システムズ)[3]	OpenBlocks IoT VX2[4]	CONEXIOBlack Bear[5]
CPU	ARM Cortex-A7 (528MHz)	Intel Atom E3805 Dual Core (1.33GHz)	ARM Cortex-A9 (1.0GHz)
メモリ	512MB	2GB	2GB
ストレージ	3.8GB	32GB	32GB
インターフェース	Wi-Fi, Ethernet, etc.	Wi-Fi, Ethernet, Bluetooth, etc.	Wi-Fi, Ethernet, Bluetooth, etc.

3. 関連研究

本章では，まず，IoT ネットワークにおける侵入検知に関する研究について紹介する．

3.1 IoT ネットワークにおける侵入検知手法

Koroniotis らは，深層学習に基づくネットワークフォレンジックフレームワークを提案している [6]．この手法では，深層学習のハイパーパラメータの値を選択するために粒子群最適化アルゴリズム (PSO: Particle Swarm Optimization) を用いており，生成された深層学習モデルが誤検知率を最小化しながら高い検知精度を達成することが示されている．また，この手法ではペイロードの暗号化やプライバシーの問題についても述べられており，深層学習に使用する特徴量はフロー単位の特徴量が用いられている．侵入検知に用いられている深層学習モデルは，入力層と出力層を含めた 7 層で構成されている．この手法で想定されている環境は，IoT 層，ネットワーク層，クラウド層の 3 つのグループに分けられている．IoT 層では，IoT 機器が Bluetooth や Wi-Fi などの Local Area Network (LAN) を介して相互に動作することで，収集されたデータを Fog 層に送信する．Fog 層では，

IoT 機器のシステムで用いられる MQTT (Message Queue Telemetry Transport) を用いてクラウド層の各サービスに IoT 機器が収集したデータを送信する．想定環境における侵入検知はネットワーク層で行うことを想定している．この手法での検知精度は 99.9%と示されている．

Ullah らは，IoT ネットワークのための新しいアノマリベースの侵入検知手法として，一次元畳み込みニューラルネットワーク，二次元畳み込みニューラルネットワーク，三次元畳み込みニューラルネットワークのそれぞれを用いた手法を提案している [7]．この手法では，パケットキャプチャデータからフローベースの特徴量を使用し，生成された特徴量を Recursive Feature Elimination (RFE) アルゴリズムを用いて，80 個の特徴量を 64 個にまで減らし，攻撃の検知に有効な特徴量の選択を行っている．選択された 64 個の特徴量は，畳み込み層，正規化層，平均プーリング層，ドロップアウト層からなるブロックを 4 つに重ねた深層畳み込みニューラルネットワークで学習され各カテゴリに検知される．この手法の検知精度は，一次元畳み込みニューラルネットワークを用いた手法の場合，最大値は 100.0%，最小値は 99.76%，二次元畳み込みニューラルネットワークを用いた手法の場合，最大値は 100.0%，最小値は 99.51%，三次元畳み込みニューラルネットワークを用いた手法の場合，最大値は 100.0%，最小値は 99.47%と示されている．

Ibitoye らは，IoT ネットワークにおける深層学習ベースの侵入検知システムとして，Feedforward Neural Network (FNN) と Self-Normalizing Neural Networks (SNN) を用いたアノマリベースの侵入検知システムを実装している [8]．Ibitoye らは，実装したアノマリーベースの侵入検知システムに対して敵対的サンプルの効果を検証している．実装した IDS の性能は，現実的なネットワーク環境を設計して作成されたデータセットを用いて評価している．この手法で使用されている機械学習アルゴリズムは，入力層，中間層，出力層からなる単純な深層学習モデルを使用している．検知精度については，FNN の検知精度が 95.1%，SNN の検知精度が 91%と示されている．

Alrashdi らは，IoT ネットワーク上で動作する侵入検知システムとして AD-IoT を提案している [9]．この手法では，元々のデータセットとして提供されている特徴量 49 個を ExtraTree を使用して 12 個に減らし，深層学習と比較して軽量な

機械学習アルゴリズムであるランダムフォレストを使用している．特徴量の削減と軽量な学習モデルを使用することで検知処理時間の削減と検知に要する計算リソースの軽量化を実現している．検知精度については 98%と示されている．

上記の通り，IoT ネットワークを想定した様々な侵入検知手法が提案されている．これらの既存研究のうち深層学習を活用する侵入検知の研究では，深層学習を活用したことによる高い検知精度が示されている．しかしながら，深層学習を活用した手法は，要求する計算リソースが多いため，2.4 節で述べた通り IoT ゲートウェイ上において動作が困難である．既存手法では侵入検知を行う単位が送信元 IP アドレス，宛先 IP アドレス，送信元ポート番号，宛先ポート番号，プロトコルの組み合わせから成るフロー毎に侵入検知を行う．そのため，フローのパターンが多く表れる DoS，DDoS 攻撃等の検知では入力データのレコード数が増加するため，侵入検知で使用するメモリ量が増加する．さらに，既存手法では使用する特徴量の数が 10 個以上であるが，使用する特徴量の数が多い場合，生成される学習モデルが大きくなり検知処理時間が遅くなる可能性がある．しかしながら，特徴量を減らしてしまうと検知精度が下がる可能性があるため，IoT 機器を対象とした場合，より良質な特徴量が必要となる．既存手法の比較を表 2 に示す．

表 2 既存研究の比較

	Koroniotis ら [6]	Ullah ら [7]	Ibitoye ら [8]	Alrashdi ら [9]
モデルサイズ	×	×	×	○
入力データ レコード数	×	×	×	×
特徴量の数	△ (13)	× (64)	△ (10)	△ (12)
侵入検知精度	99.9%	99.5% – 100%	91% – 95.1%	98%

3.2 エントロピーを利用した IoT 機器の識別手法

Hung らが提案する IoT デバイス分類手法 [10] は，通信トラフィックの複数の特徴から求めたエントロピーを基にして機械学習を行っている．この分類手法は，IoT 機器のパケットキャプチャデータを入力とした軽量な機械学習手法であり，侵入検知手法として応用できる可能性がある．この手法では，ランダムフォレス

トなどの軽量な機械学習アルゴリズムを用いてIoT機器が送信するパケットの特徴のエントロピー値に基づいたデバイスの状態の分類を行っている．多くのIoT機器は，Webサイトや SNS(Social Networking Service) 等の様々なサービスにアクセスする一般的なコンピュータと異なり，あらかじめ設定されたサーバーのみと通信を行う．この特徴を考慮して，IoTデバイスが通信するサーバーを検査することで，デバイスが破損していないか，非正規のサーバーに情報を送信している可能性があるかどうかを示すことができるとHungらは述べている．この点に関して，HungらはIoT機器が送信するパケットの特徴のエントロピー値に基づいたデバイス分類において，異常トラフィックが発生している状況下でのIoT機器の送信するトラフィックのエントロピーの値についても評価が行われている．この評価の中で，IoT機器が接続するネットワークが正常なトラフィックと異常なトラフィックの間でエントロピーの値に違いがあることが確認されている．

この手法は，IoT機器の状態の分類を，5分間隔で分割されたパケットキャプチャデータを使用して生成したエントロピーを用いて行っており，様々なIoT機器の通信を解析して，正常時の挙動，異常時の挙動についても識別することが可能であることを示している．この手法では6個の特徴量のみを使用し，94.74%の精度でIoT機器を識別できることから，各IoT機器の通信挙動を示すエントロピーの値をIoTゲートウェイ上で動作させる侵入検知システムに採用することができれば，侵入検知の精度を維持しつつ特徴量を削減することができる．

しかしながら，通信挙動を示すエントロピーを生成する際の時間間隔が5分であるため，この手法をそのまま侵入検知に用いる場合，侵入検知までにかかる時間に，パケット収集時間である5分間が加わる．さらに，この手法ではデータサイズのエントロピーが特徴量として使用されているが，データサイズのエントロピーを特徴量として使用する場合，パケットをセッション単位に分類し，各データサイズの値を求めた上でエントロピーを計算する必要がある．さらに，この手法では収集したトラフィックからホストを識別しているため，侵入検知システムとしてホスト毎のトラフィックから特徴量の生成を行う場合，送信元IPアドレスのエントロピーの値は変化しないため特徴量として使用することができない．そのため，この関連研究を侵入検知システムとして使用するためにホスト毎のト

ラフィックから生成することが可能な特徴量を検討する必要がある.

4. 複数エントロピーを用いた軽量型侵入検知手法

4.1 提案手法の概要

本手法では侵入検知処理時の入力データを削減することで，侵入検知の計算処理を軽量化する．検知処理をフロー単位からホスト単位に変更するとともに，ホスト毎の通信挙動を複数のエントロピー (宛先 IP アドレス，宛先ポート番号，パケットサイズ，等) を使用し，IoT 機器の状態を識別する手法を採用した侵入検知手法を提案する．

4.2 特徴量の検討

3.2 節の研究では IoT 機器の識別に使用されている特徴量として，送信元 IP アドレス，宛先 IP アドレス，送信元ポート番号，宛先ポート番号，パケットサイズ，データサイズのエントロピーを特徴量として使用している．この中で侵入検知システムとして送信元 IP アドレスをホストとして分類を行った際に生成することが容易な特徴量は宛先 IP アドレス，送信元ポート番号，宛先ポート番号，パケットサイズのエントロピーである．宛先 IP アドレスのエントロピーの値は正常な IoT 機器の場合，特定のホストとの通信を行うため，エントロピーを計算する際に表れる宛先 IP アドレスに一定の偏りがあるため，エントロピーの値が各 IoT 機器固有の値として一定の振れ幅で表れる．送信元ポート番号のエントロピーの値は正常時は実装されている OS に依存したエフェメラルポートが使用されるため，使用されている送信元ポート番号にある程度のばらつきがありエントロピーの値が高い値となる．しかし，攻撃を受けている異常状態のホストは攻撃者が対象としているサービスポートを送信元ポート番号として応答を返すため，送信元ポート番号のエントロピーの値は正常時よりも低い値として表れる．宛先ポート番号のエントロピーの値は正常時は一定の時間間隔でクラウド上のサーバーに対して特定のサービスを使用してデータを送信するため，エントロピーを計算する際に表れる宛先ポート番号に一定の偏りがあることから，エントロピーの値が各 IoT 機器固有の値として一定の振れ幅で表れる．パケットサイズのエントロピー

は正常なIoT機器では実装されているセンサー等で収集されたデータを一定の時間間隔でクラウド上のサーバーに送信するため，エントロピーの値が各IoT機器固有の値として一定の振れ幅で表れる．

また，侵入検知システムとしてホスト毎に生成可能な追加の特徴量としてパケットの送信時間間隔のエントロピーを使用することが可能である．正常なホストの場合，パケットの送信はIoT機器にあらかじめ実装されたプログラムに従って行われるため，時間間隔のエントロピーの値が一定のばらつきで表れる．しかし，異常なホストの場合，攻撃対象となるサーバー，ホストに対して短い時間間隔で大量にパケットを送信するため，エントロピーの値がとても小さい値として表れる．

さらに，ホスト毎に送信されるパケット数と送信されるパケットサイズを観測することで，送信されるパケットサイズの平均を特徴量として使用することが可能となる．正常なホストの場合，送信するパケットのサイズはIoT機器にあらかじめ実装されたプログラムに従って行われるため，各ホスト毎に特有のパケットサイズでパケットを送信している．異常なホストの場合，正常時とは異なるパケットサイズのパケットが送信されるため，この特徴量を使用することで正常なホストと異常なホストを容易に分類することが可能となる．

上記のことから提案手法として侵入検知を行う際の検知単位をホスト毎に変更し，通信挙動を表す特徴量として，宛先IPアドレス，送信元ポート番号，宛先ポート番号，パケットサイズ，送信時間間隔のエントロピーと送信パケットサイズの平均の6つを特徴量として使用する．

4.3 提案手法の実装

図1に提案手法を使用した侵入検知システムのワークフローを示す．提案手法を使用した侵入検知システムは，パケット収集，パケット分類，特徴抽出，侵入検知の4つのフェーズで構成される．パケット収集フェーズではIoTゲートウェイを通過するパケットを一定の収集時間で収集する．パケット分類フェーズではパケット収集フェーズで収集したパケットをホスト毎に分類する．特徴抽出フェーズではパケット分類フェーズで分類したホスト毎のパケットから特徴量を生成す

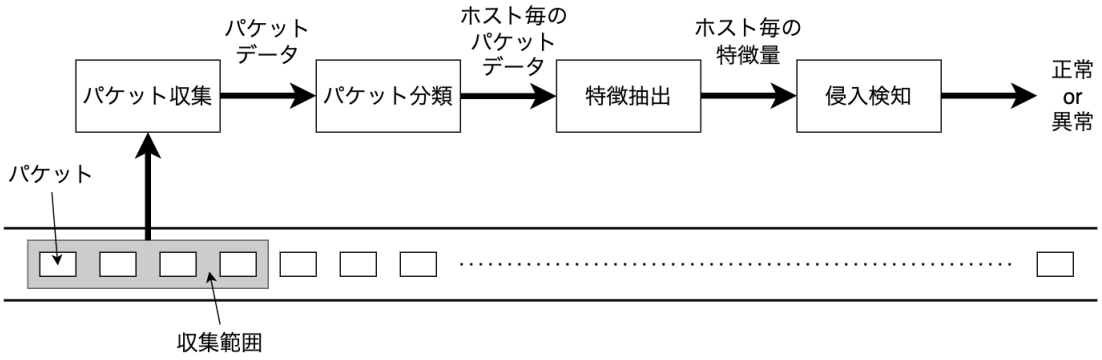

図 1　提案手法の概略図

る．侵入検知フェーズでは特徴抽出フェーズで生成した特徴量を用いて対応するホストの通信が攻撃であるかどうかを検知する．以降では，各フェーズの詳細について述べる．

- パケット収集フェーズ
  提案手法では，セッション単位に特徴量を生成する方法とは異なり，特徴量の生成をホスト単位に行う．そのため，セッション単位に特徴量を生成する方法の場合，セッションの終了またはタイムアウトを待ってから特徴量の生成を行うが，ホスト単位に特徴量を生成する方法では，任意の時間でパケットの収集を止める必要がある．提案手法では，パケットの収集時間を 10 秒とした．

- パケット分類フェーズ
  収集したパケットをホスト毎に分類する．提案手法では，プログラミング言語 Python を用いて，パケットの分類を行うプログラムを実装しており，実装したプログラム上で収集したパケットを読み込むために Python ライブラリモジュール Scapy を用いた．

- 特徴抽出フェーズ
  提案手法では，IoT 機器の通信挙動を複数のエントロピーと送信パケットサイズの平均を用いて表現し，これらのデータを特徴量として機械学習による侵入検知を実現する．特徴抽出フェーズでは，パケット分類フェーズ

でホスト単位に分類されたデータを用いて，ホスト毎の特徴量を生成する．特徴抽出フェーズで抽出する特徴量とデータ形式を表 3 に示す．エントロピーの計算には平均情報量の計算式である式 (1) を使用した．N は出現する値の種類の総数，p_i は出現する各値の出現確率である．

$$H(X) = -\sum_{i=1}^{N} p_i \log(p_i) \tag{1}$$

宛先 IP アドレスのエントロピーの計算は，式 (1) を使用する際に，出現する宛先 IP アドレスの種類が N となり，各宛先 IP アドレスの出現確率が p_i となる．出現確率 pi は 10 秒間隔で送受信したパケットの各宛先 IP アドレスの出現回数を 10 秒間隔で送受信したパケットの総数で割ることで求められる．

宛先ポート番号のエントロピーの計算は，式 (1) を使用する際に，出現する宛先ポート番号の種類が N となり，各宛先ポート番号の出現確率が p_i となる．出現確率 pi は 10 秒間隔で送受信したパケットの各宛先ポート番号の出現回数を 10 秒間隔で送受信したパケットの総数で割ることで求められる．

送信元ポート番号のエントロピーの計算は，式 (1) を使用する際に，出現する送信元ポート番号の種類が N となり，各送信元ポート番号の出現確率が p_i となる．出現確率 pi は 10 秒間隔で送受信したパケットの各送信元ポート番号の出現回数を 10 秒間隔で送受信したパケットの総数で割ることで求められる．

パケットサイズのエントロピーの計算は，式 (1) を使用する際に，出現するパケットサイズの種類が N となり，各パケットサイズの出現確率が p_i となる．出現確率 pi は 10 秒間隔で送受信したパケットの各パケットサイズの出現回数を 10 秒間隔で送受信したパケットの総数で割ることで求められる．

パケットの送信時間間隔のエントロピーの計算は，式 (1) を使用する際に，出現するパケットの送信時間間隔の種類が N となり，各パケットの送信時間間隔の出現確率が p_i となる．出現確率 pi は 10 秒間隔で送受信したパケッ

トの各パケットの送信時間間隔の出現回数を 10 秒間隔で送受信したパケットの総数から 1 を引いた値で割ることで求められる．

表 3 生成する特徴量

特徴量	データ形式
送信パケットサイズの平均	浮動小数点
送信時間間隔のエントロピー	浮動小数点
宛先 IP アドレスのエントロピー	浮動小数点
宛先ポート番号のエントロピー	浮動小数点
送信元ポート番号のエントロピー	浮動小数点
パケットサイズのエントロピー	浮動小数点

- 侵入検知フェーズ
  提案手法における侵入検知フェーズについて述べる．侵入検知フェーズでは，特徴抽出フェーズでホスト単位に生成された特徴量を用いて，接続されているホストが正常か異常を検知する．侵入検知のアルゴリズムには IoT ゲートウェイ上でも動作が可能で，軽量な侵入検知を実現するために，軽量かつ検知精度が高い機械学習アルゴリズムを用いる．本研究では，いくつかの軽量な機械学習アルゴリズムを用いて検知精度を比較し，検知精度の平均が最も高い機械学習アルゴリズムを侵入検知フェーズで用いることとした．侵入検知フェーズで用いる軽量な機械学習アルゴリズムの候補として，ランダムフォレスト，サポートベクタマシーン，k 近傍法を使用する．

5. 評価

本章では，評価に使用するデータセットと提案手法で用いる特徴量の生成，評価環境について述べる．次に，提案した特徴量の有効性を示し，侵入検知フェーズで用いる機械学習モデルを選定するために，提案した特徴量を用いた侵入検知の精度を既存手法と比較する．最後に，既存研究と提案手法の機械学習モデルの検知時間とメモリ使用量に関する評価を行う．

5.1 データセット

本評価では，既存研究の SNN[8] で用いられた Bot-IoT データセット [11] を使用した．Bot-IoT データセットは，UNSW Canberra で現実的なネットワーク環境を設計して作成されている．データセットは，Pcap ファイル，Argus ファイル，CSV ファイルの形式で提供されており，これらのファイルは，ラベリング処理を容易にするために，攻撃カテゴリとサブカテゴリに基づいて分離されている．データセットには，DDoS 攻撃，DoS 攻撃，OS フィンガープリント，ポートスキャン，キーロギング，データ流出のカテゴリが含まれており，DDoS 攻撃と DoS 攻撃は，使用されるプロトコルに基づいてさらに整理されている．本評価では提供されている Pcap ファイルと既存手法の SNN[8] で使用されている 10 best features の CSV ファイルを使用した．

5.2 特徴量の生成

本節では，評価で使用する特徴量を生成するための前処理と提案した特徴量の生成について述べる．本評価では，提案手法と既存手法を比較するために，提案手法に用いる特徴量の他に，既存手法に用いる特徴量を生成した．特徴量の生成では，まず，Bot-IoT データセットで提供されている Pcap ファイルを図 2 のように 10 秒間隔に分割した．Pcap ファイルの分割には，Wireshark の CLI ツールである editcap を使用した．既存手法の AD-IoT[9] で使用するフロー単位の特徴量はネットワーク分析フレームワーク Zeek を使用し，提案手法で使用するホスト

単位の特徴量はプログラミング言語 Python を用いて実装したプログラムにより生成した．評価に使用する特徴量の生成手順を図 3 に示す．既存手法の SNN で使用する特徴量は Ibitoye らが使用した Bot-IoT データセットで提供されている CSV ファイルを使用した．既存手法の AD-IoT と SNN で使用した特徴量を表 4 と表 5 に示す．

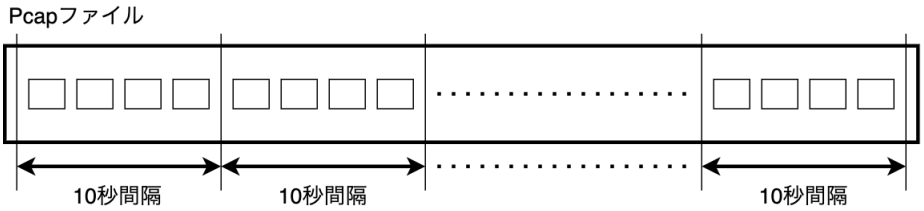

図 2 Pcap ファイルを 10 秒間隔に分割

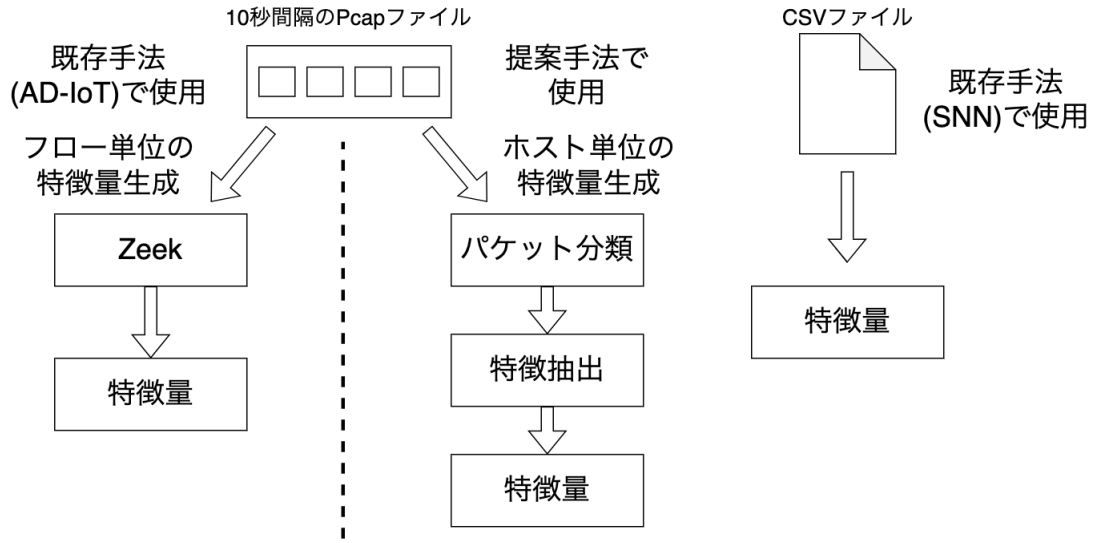

図 3 評価に使用する特徴量の生成手順

5.3 生成した特徴量の評価と侵入検知精度の評価

本節では，提案した特徴量の有効性を評価し，その後，侵入検知フェーズで用いる機械学習アルゴリズムを選定する．最後に，提案手法の侵入検知精度の比較

表 4 AD-IoT の評価で使用した特徴量

特徴量
プロトコル
サービス (http, ssh, telnet, 等)
セッションの継続時間
送信データサイズ
受信データサイズ
セッションの終了状態
コンテンツギャップの見逃しバイト数
コネクションの状態履歴
送信パケット数
同一送信元 IP アドレスのデータサイズ
受信パケット数
同一宛先 IP アドレスのデータサイズ

表 5 SNN の評価で使用した特徴量

特徴量
Argus シーケンス番号
集計したレコードの平均継続時間
集計したレコードの標準偏差
集計したレコードの最小セッション継続時間
集計したレコードの最大セッション継続時間
1 秒間あたりの同一送信元–宛先 IP アドレスのパケット数
1 秒間あたりの同一宛先–送信元 IP アドレスのパケット数
同一送信元 IP アドレスのパケット数
同一宛先 IP アドレスのパケット数
集計したレコードの平均継続時間

を行う．

5.3.1 評価方法

評価で使用する機械学習アルゴリズムは，scikit-learn のランダムフォレスト，サポートベクタマシーン，k 近傍法を選択した．評価方法は 10 回交差検証により出力される Accuracy の平均値を侵入検知精度とした．評価で使用する特徴量は，提案手法には 5.2 節で生成したホスト単位の特徴量を使用し，既存手法の AD-IoT には 5.2 節で生成したフロー単位の特徴量を使用した．既存手法の SNN には Ibitoye ら [8] の研究で使用されている Bot-IoT データセットの 10best-features を特徴量として使用した．本評価で使用した特徴量のレコード数を表 6 に示す．

表 6 特徴量のレコード数

	提案手法	AD-IoT	SNN
通常 レコード数	85,025	69,340	477
攻撃 レコード数	48,772	165,905,915	3,668,045
総レコード数	133,797	165,975,255	3,668,522

5.3.2 評価環境

本評価は，奈良先端科学技術大学院大学の小規模計算サーバーの超並列演算ノード上で実施した．使用した超並列演算ノードのスペックを表 7 に示す．

5.3.3 生成した特徴量の評価と侵入検知精度の評価結果

評価結果を表 8 に示す．評価の結果，提案手法の侵入検知精度の値が最も高いのはランダムフォレストの 99.8%であった．また，提案手法の侵入検知精度の値が最も低いのはサポートベクタマシーンの 97.1%であった．既存手法の AD-IoT では侵入検知精度が 99.9%であり，提案手法よりも 0.1%高い値となった．さらに，深層学習を使用した既存手法である SNN の侵入検知精度は 90.1%と最も低い値となった．

表 7 超並列演算ノードのスペック

	超並列演算ノード
model	Fujitsu PRIMERGY CX2570 M2
CPU	Intel Xeon E5-2650v4 ×2 (Broadwell-EP)
CPU clock	2.20 GHz
CPU cores	24 (12 ×2)
memory	256 GiB
GPU	NVIDIA Tesla P100 ×2
CUDA cores	7,168 (3,584 ×2)
GPU memory	32 GiB (16 GiB ×2)
storage	HDD 1.2 TB ×2

表 8 提案した特徴量を用いた場合の侵入検知精度の比較

機械学習アルゴリズム	侵入検知精度 Accuracy (%)
提案手法 ランダムフォレスト	99.8
提案手法 k 近傍法	99.0
提案手法 サポートベクタマシン	97.1
既存手法 (AD-IoT)	99.9
既存手法 (SNN)	90.1

5.4 検知処理時間の評価

本節では，既存手法の AD-IoT と深層学習ベースの SNN，提案した特徴量を学習させたランダムフォレストによる侵入検知処理速度の評価について述べる．

5.4.1 評価方法

本評価では 5.3 節で用いた学習モデルを使用して検知時間の評価を行う．学習モデルへのデータの入力は 1 レコード単位に行い，特徴量の入力から侵入検知の結果を出力するまでを検知処理時間として計測した．検知時間は Python の標準ライブラリである time モジュールを使用し，検知を行う関数の実行前の時刻と実行後の時刻を記録し，差分を検知処理時間とした．

5.4.2 評価環境

本評価では，侵入検知手法を IoT ゲートウェイ上で動作させることを考慮して，Raspberry Pi 3 Model B 上で提案手法の検知処理時間の評価を行った．評価に使用した Raspberry Pi 3 Model B のスペックを表 9 に示す．深層学習を使用した既存手法の SNN と提案手法の評価は，5.3.2 節で使用した超並列演算ノード上で評価を行った．

表 9 Raspberry Pi のスペック

	Raspberry Pi 3 Model B
CPU	Broadcom BCM2837 ARM Cortex-A53
CPU clock	1.2 GHz
CPU cores	4
memory	1 GB
storage	64 GB

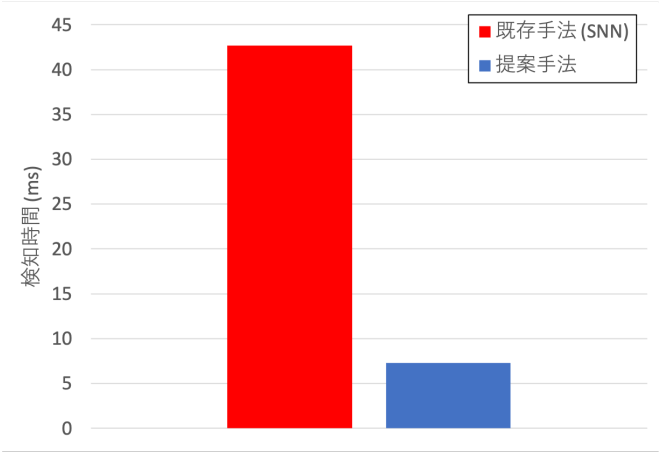

図 4　既存手法 (SNN) と提案手法の処理時間

5.4.3　評価結果

AD-IoT と提案手法の検知処理時間と SNN と提案手法の検知処理時間を表 10 に示す．図 4 に深層学習を使用した既存手法の SNN と提案手法の処理時間を示し，図 5 に IoT ゲートウェイ上での動作を想定し，Raspberrry Pi 3 Model B 上で評価した既存手法の AD-IoT と提案手法の処理時間を示す．評価の結果，既存手法の AD-IoT を使用した場合の検知処理時間は 65.6 ミリ秒であったのに対して提案手法では 36.9 ミリ秒と検知処理時間を 28.7 ミリ秒 (56%) 短縮されることを確認した．また，深層学習を使用した既存手法の SNN の検知処理時間は 42.7 ミリ秒であったのに対して，提案手法では 7.30 ミリ秒と検知処理時間が 35.4 ミリ秒 (83%) 短縮されたことを確認した．

表 10　検知時間

侵入検知手法	検知処理時間 (ms)
AD-IoT (Raspberrry Pi 3 Model B)	65.6
提案手法 (Raspberrry Pi 3 Model B)	36.9
SNN (超並列演算ノード)	42.7
提案手法 (超並列演算ノード)	7.30

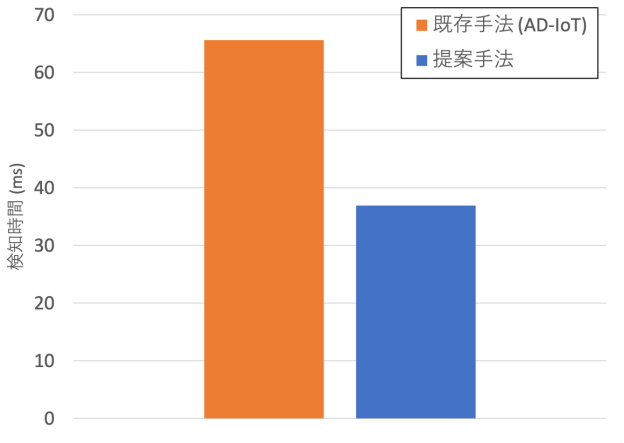

図 5　既存手法 (AD-IoT) と提案手法の処理時間

5.5　メモリ使用量の評価

本節では提案手法の検知処理時のメモリ使用量の評価について述べる．

5.5.1　評価方法

本評価では，10 秒間隔の Pcap ファイルから生成された特徴量を使用し，既存手法と提案手法の検知処理時のメモリ使用量を評価する．既存手法の AD-IoT では Zeek を使用してフロー単位に生成された特徴量 12 個を使用して機械学習を行い学習モデルを構築し，提案手法では自身が作成したプログラムを使用して生成した特徴量 6 個を使用して機械学習モデルを構築した．学習モデルへの入力は特徴量として生成されるデータ単位に行う．メモリ使用量のピークを検知処理毎に記録し，その平均を検知処理のメモリ使用量とする．メモリ使用量の計測には time コマンドの-v オプションを使用し，最大メモリ使用量を検知処理毎のメモリ使用量のピークとした．

5.5.2　評価環境

提案手法のメモリ使用量の評価は，奈良先端科学技術大学院大学の小規模計算サーバーのクラスタノード上で評価を行った．使用したクラスタノードのスペッ

クを表 11 に示す．

表 11 クラスタノードのスペック

	クラスタノード
model	Supermicro SYS-1028GR-TR
CPU	Intel Xeon E5-2650v4 ×2 (Broadwell-EP)
CPU clock	2.20 GHz
CPU cores	24 (12 ×2)
memory	256 GiB
GPU	NVDIA Quadro P4000 ×1
CUDA cores	1,792
GPU memory	8 GiB
storage	SSD 240 GB ×1

5.5.3 メモリ使用量の評価結果

各攻撃カテゴリごとのメモリ使用量を表 12 に示す．評価の結果，提案手法のメモリ使用量は最大で 105MiB となり，既存手法よりも 331MiB 少ないことを確認した．図 6 に 10 秒間隔に分割された Pcap ファイルから生成された特徴量を使用して侵入検知を行った際のメモリ使用量を示す．

表 12　メモリ使用量

	既存手法 AD-IoT (MiB)	提案手法 (MiB)
DDoS HTTP	118	105
DDoS TCP	436	103
DDoS UDP	348	104
DoS HTTP	131	104
DoS TCP	284	105
DoS UDP	371	104
Scan Service	118	104
Scan OS	118	104
Theft Keylogging	117	105
Theft Data_Exfiltration	116	105

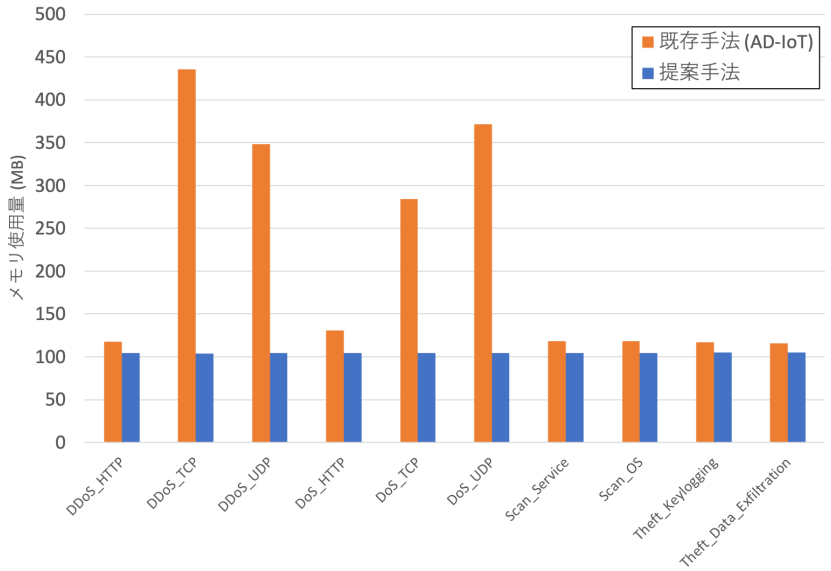

図 6　攻撃カテゴリごとのメモリ使用量

6. 考察

本章では，5 章の評価結果に基づいて，提案した特徴量の有効性について考察する．次に，既存手法の AD-IoT と深層学習を使用した既存手法の SNN，提案手法を比較した際の提案手法の有効性について考察する．

6.1 特徴量の有効性

検知精度の評価結果から，通信挙動を表す特徴量として複数のエントロピーと送信パケットサイズの平均を用いることで，正常なホストの通信挙動と異常なホストの通信挙動を検知することが可能であることを確認できる．提案した特徴量を用いて侵入検知を行った場合，使用する特徴量の数が 12 個から 6 個に減ったにも関わらず既存手法である AD-IoT と比較して検知精度の低下は 0.1%であった．

このように検知精度を維持できた要因について分析する．検知精度に影響を与えた特徴量について分析するためにランダムフォレストの特徴量の寄与率上位三件を表 13 に示す．図 7, 8, 9 にランダムフォレストの分類寄与率が高かった特徴量上位三件の 10 秒間隔に観測された送信元 IP アドレスの出現回数を示す．最も寄与率の高かった特徴量は送信パケットサイズの平均であった．図 7 の 10 秒間隔に観測されたホストの送信パケットサイズの平均のヒストグラムを確認すると，正常なホストの送信パケットサイズの平均は 0–50Byte，250–300Byte, 800–1000Byte の値になるのに対して，異常なホストでは送信パケットサイズの平均が 200Byte 以下の値に集中していることが確認できる．このことから送信パケットサイズの平均が正常なホストと異常なホストの分類に有効な特徴量として寄与したと考えられる．次に，図 8 の 10 秒間隔に観測されたホストのパケットの送信時間間隔のエントロピーのヒストグラムを確認すると，正常なホストではパケットの送信時間間隔のエントロピーが多くの場合，0.5 から 4.0 の値になるのに対して，異常なホストではパケットの送信時間間隔のエントロピーが多くの場合，0 もしくは 0 に近い値となる．これは異常なホストが短い時間間隔で大量のパケットを送信するため，同じ送信時間間隔のパケットが多く出現し，エントロピーの値を小さくしたためである．このことから，パケットの送信時間間隔のエントロピーが正

表 13 既存手法 (AD-IoT) と提案手法の平均レコード数

特徴量	寄与率
送信パケットサイズの平均	0.310
パケットの送信時間間隔のエントロピー	0.195
送信元ポート番号のエントロピー	0.144

常なホストと異常なホストの分類に寄与したと考えられる．最後に，図 9 の 10 秒間隔に観測されたホストの送信元ポート番号のヒストグラムを確認すると，正常なホストでは送信元ポート番号がエフェメラルポートを使用するため送信元ポート番号の値がばらつきエントロピーの値が高くなっているが，異常なホストでは攻撃を受けたホストの応答パケットによって同じ送信元ポート番号が多く表れるためエントロピーの値が正常時よりも低くなり，正常なホストと異常なホストの分類に寄与したと考えられる．これらのことから提案した特徴量は正常なホストと異常なホストの分類に有効な特徴量であると言える．

6.2 既存手法と提案手法の検知時間の比較

検知処理時間の評価結果から，提案手法は深層学習を使用した既存手法の SNN と軽量な機械学習モデルを使用している既存手法の AD-IoT よりも検知処理時間が短いことが分かる．既存手法の AD-IoT と提案手法を比較するとどちらの手法も機械学習モデルにランダムフォレストを使用しているが，既存手法の AD-IoT では特徴量の数が 12 個であるのに対して，提案手法では特徴量の数が 6 個と特徴量の数が少ない．このことが検知処理時間の短縮に貢献したと考える．実際に，ランダムフォレストで構築される最大の木の深さが既存手法の AD-IoT では平均で 22.8 であるのに対して提案手法では 18.4 となり提案手法の方が木の深さが浅くなった．このように，提案した特徴量を使用することで侵入検知における正常なホストと異常なホストの分類が容易となり，侵入検知処理の速度を短縮することが可能となった．

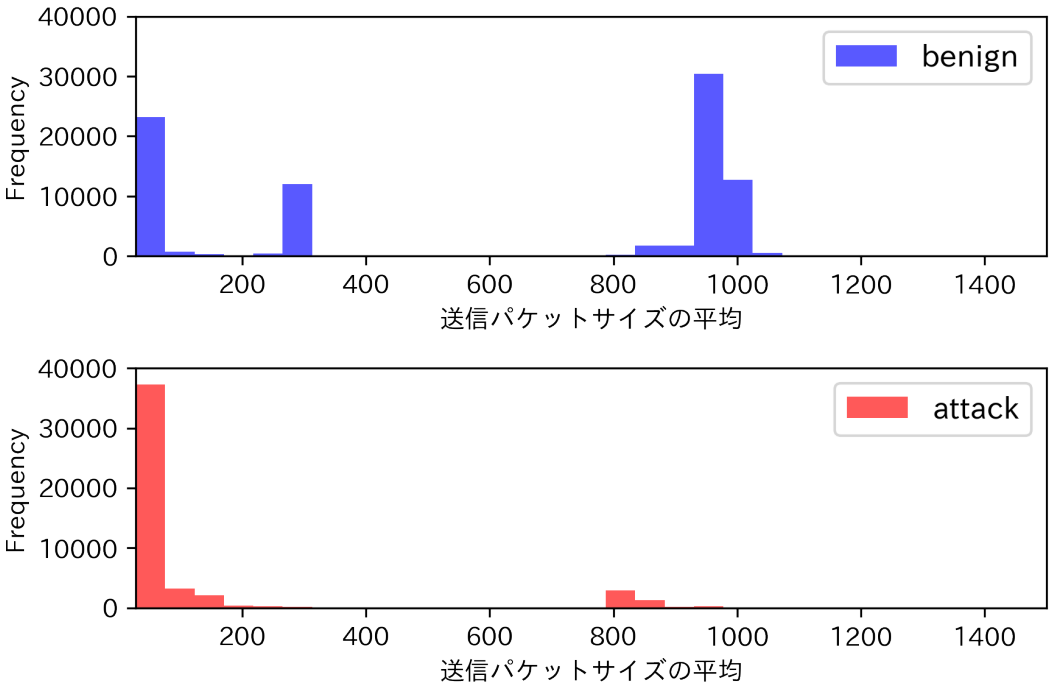

図 7　10 秒間隔に観測されたホストの送信パケットサイズの平均のヒストグラム

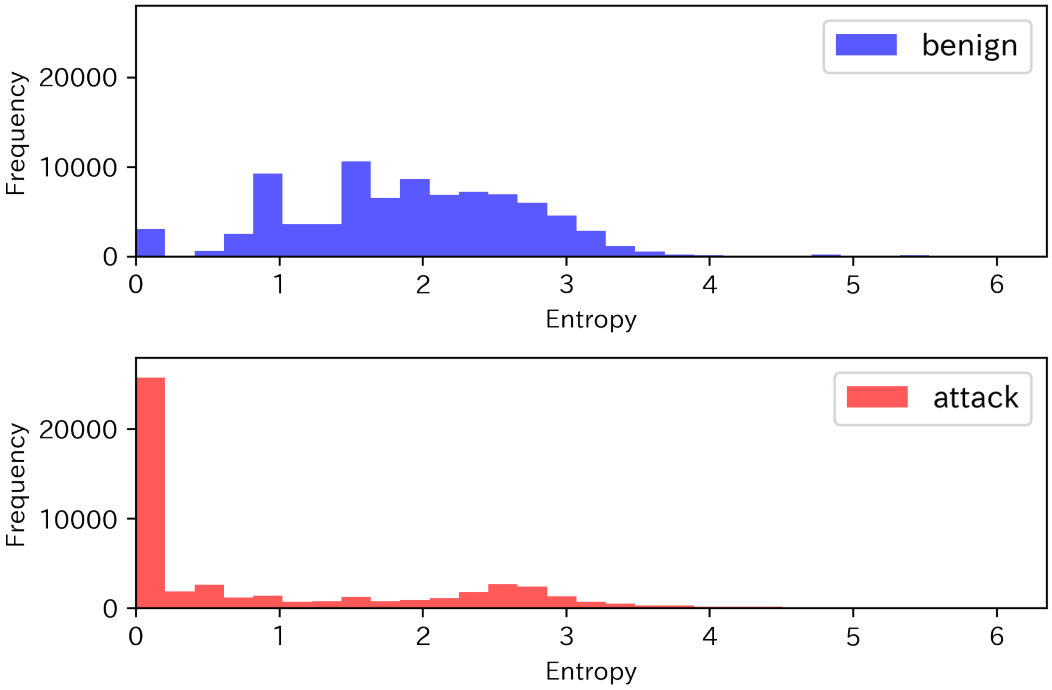

図 8 10 秒間隔に観測されたホストの送信時間間隔のエントロピーのヒストグラム

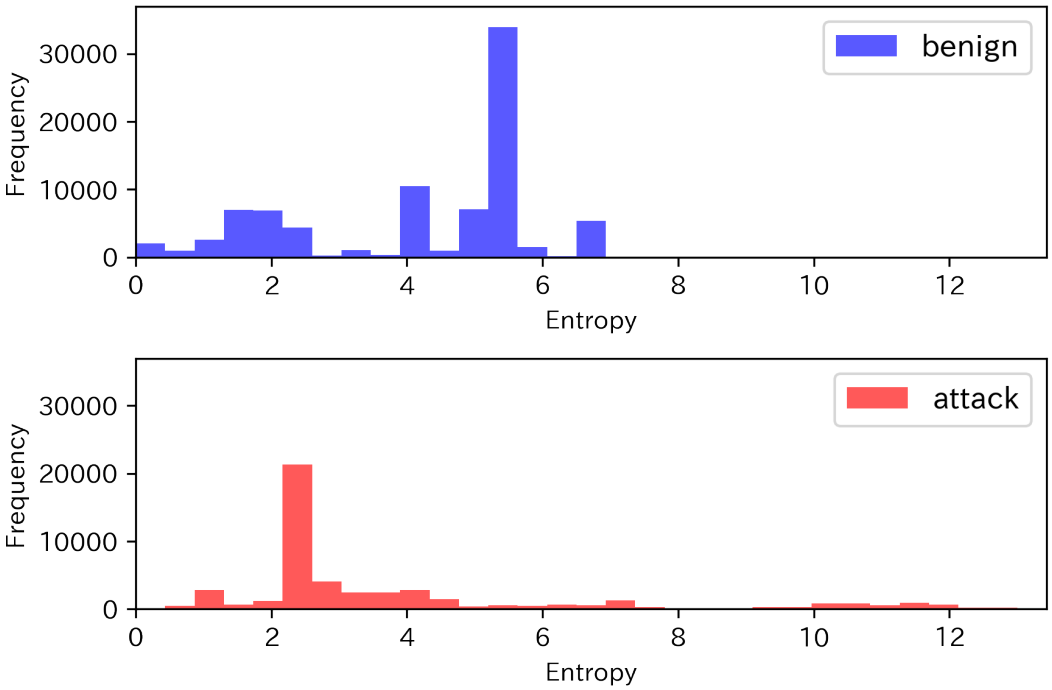

図 9　10 秒間隔に観測されたホストの送信元ポート番号のエントロピーのヒストグラム

表 14 既存手法 (AD-IoT) と提案手法の平均レコード数

	既存手法 (AD-IoT)	提案手法
DDoS HTTP	213	8
DDoS TCP	371,297	11
DDoS UDP	202,944	11
DoS HTTP	302	8
DoS TCP	149,703	11
DoS UDP	221,352	11
Scan Service	279	11
Scan OS	282	10
Theft Keylogging	27	10
Theft Data Exfiltration	13	8

6.3 既存手法と提案手法のメモリ使用量の比較

メモリ使用量の評価結果から，既存手法の AD-IoT ではフローのパターンが多く表れる TCP と UDP の DoS，DDoS 攻撃の侵入検知でメモリ使用量が高くなることが確認できる．対して，提案手法では侵入検知を行う単位をフロー単位からホスト単位に変更したことで，フローのパターンに影響を受けずにメモリの使用量が一定の値に保たれていることが確認できる．提案手法で侵入検知のメモリ使用量が最も大きい TCP の DDoS 攻撃は，既存手法ではメモリ使用量が 436MiB であったのに対して，提案手法では 103MiB に抑えられている．この結果は，侵入検知を行う単位をフロー単位からホスト単位に変更したことで，フローのパターン数の影響を受けなかったためと考えられる．実際に，AD-IoT と提案手法の攻撃カテゴリごとの平均レコード数は表 14 の通り，AD-IoT でメモリ使用量が大きかった TCP と UDP の DoS，DDoS 攻撃の平均レコードは全て 10 万レコードを超えているのに対して，提案手法では 11 レコードに抑えられている．このように，侵入検知処理をフロー単位からホスト単位に変更することで侵入検知処理に使用するメモリ量を減らすことが可能となる．

6.4 今後の課題

提案手法ではパケットを 10 秒間隔に収集し，各パケットを送信元 IP アドレス毎に分類して特徴量の生成を行う．そのため，提案手法を用いた場合の課題として，スロースキャン攻撃等の検知が困難となる．スロースキャン攻撃では，マルウェアに感染したホストが侵入検知システムによる侵入検知を回避するために，正常時の通信挙動と同じような送信時間間隔でパケットを送信する．提案手法では一定の時間間隔で収集したパケットから特徴量として複数のエントロピーと送信パケットサイズの平均を生成するため，収集する時間間隔に合わせたスキャン攻撃を実行した場合，エントロピーの値が大きな変動を示さないため，侵入検知が回避される可能性がある．また，提案手法ではパケットの収集時間間隔を 10 秒と設定している．したがって，パケットの分類，特徴の抽出，侵入検知を行うためには，各ホスト毎に 10 秒間のパケット収集時間を待つ必要がある．そのため，各 IoT 機器に合わせた最適なパケット収集時間を動的に調整する仕組みを実現することが必要がある．

また，本研究では侵入検知処理に着目して検知処理時間とメモリ使用量の削減を行ったが，パケットの分類と特徴量抽出の処理も軽量化する必要がある．本研究では，パケットの分類と特徴量の抽出にはプログラミング言語 Python を使用して実装を行った．しかし，Python はインタプリタ型の言語であるため，パケットの分類と特徴量の抽出の際に実行されるプログラムを逐次的に解釈し，実行するため，パケットの分類と特徴量の抽出処理の速度は遅くなる．

7. おわりに

IoT 機器が接続される IoT ゲートウェイは計算リソースが限られており，IoT ゲートウェイとしての本来の処理に加えて，侵入検知システムを IoT ゲートウェイ上で動作させる場合，パケットの転送などの IoT ゲートウェイとしての本来の処理に悪影響を与える．そのため，計算リソースに制約のある環境でも動作する軽量な侵入検知手法が求められている．本研究では，侵入検知処理をフロー単位からホスト単位に変更することで侵入検知を行う際に入力データのレコード数を削減し，また，ホスト単位の侵入検知において複数のエントロピーと送信パケットサイズの平均を特徴量として使用することで特徴量の数を削減した．これにより，侵入検知における検知精度を 99.8%と既存手法と同等の値を維持しながら，検知処理時間を 35.4 ミリ秒短縮し，メモリ使用量を 331MiB 削減した．今後の課題として，提案手法ではスロースキャン攻撃等の検知が困難である．今後は各ホスト毎の最適なパケット収集時間を動的に設定する仕組みを導入することで，短い収集時間であってもスロースキャン攻撃によるエントロピーの変化を補足し，検知を行うことを可能にする．

謝辞

主指導教員であり，適切な研究指導をしていただく等様々な面でサポートをして頂いた本学情報基盤システム学研究室の藤川和利教授に心から感謝致します．副指導教員であり，研究についてのご指導，論文の添削だけでなく，精神的に挫けそうになった時にサポートをして頂いた本学情報基盤システム学研究室の新井イスマイル准教授に心から感謝致します．研究についてのご指導とご助言だけでなく，学内システムで用いられている運用技術などについても教えてくださいました本学情報基盤システム学研究室の垣内正年助教に心から感謝致します．研究についてのご指導だけでなく，休日にも関わらず修士論文を添削してくださった遠藤新助教に心から感謝致します．また，研究活動を行うにあたり，研究についての議論やご助言を頂いた本学情報基盤システム学研究室の大平修慈氏に心から感謝致します．同一の研究分野の学生として研究の議論と研究活動でのご助言を頂いた小松聖矢氏に心から感謝致します．様々な事務手続きや研究室内でのコミュニケーションの機会を提供して下さった辻元理恵女史に心から感謝致します．そして，研究室の生活において様々な面で支えてくれた本学情報基盤システム学研究室の学生の皆様に心から感謝致します．最後に，博士前期課程への進学にあたり，私の意思を尊重し，経済面，生活面での援助と精神面で励ましの言葉を下さった家族に心から感謝します．

参考文献

[1] 国立研究開発法人情報通信研究機構 サイバーセキュリティ研究所サイバーセキュリティ研究室. NICTER 観測レポート 2020. https://www.nict.go.jp/cyber/report/NICTER_report_2020.pdf.

[2] Krebs on Security. DDoS on Dyn Impacts Twitter, Spotify, Reddit. https://krebsonsecurity.com/2016/10/ddos-on-dyn-impacts-twitter-spotify-reddit/, 2016.

[3] Bruno Bogaz Zarpelão, Rodrigo Sanches Miani, Cláudio Toshio Kawakani, and Sean Carlisto de Alvarenga. A survey of intrusion detection in internet of things. *Journal of Network and Computer Applications*, Vol. 84, pp. 25–37, 2017.

[4] Karen Scarfone and Peter Mell. Guide to intrusion detection and prevention systems (IDPS). Technical report, National Institute of Standards and Technology, 2007.

[5] NIST. 侵入検知および侵入防止システム（IDPS）に関するガイド. https://www.ipa.go.jp/files/000025364.pdf, 2007.

[6] Nickolaos Koroniotis, Nour Moustafa, and Elena Sitnikova. A new network forensic framework based on deep learning for internet of things networks: A particle deep framework. *Future Generation Computer Systems*, Vol. 110, pp. 91–106, 2020.

[7] Imtiaz Ullah and Qusay H. Mahmoud. Design and development of a deep learning-based model for anomaly detection in IoT networks. *IEEE Access*, Vol. 9, pp. 103906–103926, 2021.

[8] Olakunle Ibitoye, Omair Shafiq, and Ashraf Matrawy. Analyzing adversarial attacks against deep learning for intrusion detection in IoT networks. In *2019*

IEEE Global Communications Conference (GLOBECOM), pp. 1–6. IEEE, 2019.

[9] Ibrahim Alrashdi, Ali Alqazzaz, Esam Aloufi, Raed Alharthi, Mohamed Zohdy, and Hua Ming. AD-IoT: Anomaly Detection of IoT Cyberattacks in Smart City Using Machine Learning. In *the 9th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 0305–0310. IEEE, 2019.

[10] Hung Nguyen-An, Thomas Silverston, Taku Yamazaki, and Takumi Miyoshi. Entropy-based IoT devices identification. In *the 21st Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 73–78. IEEE, 2020.

[11] Nickolaos Koroniotis, Nour Moustafa, Elena Sitnikova, and Benjamin Turnbull. Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset. *Future Generation Computer Systems*, Vol. 100, pp. 779–796, 2019.