Nexmon による CSI ベースのスマートデバイス状態推定

原田　海斗†　　寺本　京祐††　　野村裕一郎†††　　峰野　博史††††

† 静岡大学情報学部　〒 432–8011 静岡県浜松市中区城北 3 丁目 5-1
†† 静岡大学大学院総合科学技術研究科情報学専攻　〒 432–8011 静岡県浜松市中区城北 3 丁目 5-1
††† 静岡大学学術院情報学領域　〒 432–8011 静岡県浜松市中区城北 3 丁目 5-1
†††† 静岡大学グリーン科学技術研究所　〒 432–8011 静岡県浜松市中区城北 3 丁目 5-1
E-mail: †{harada.kaito.20,teramoto.kyosuke.18}@shizuoka.ac.jp,
††{nomura.yuhichiro,mineno}@inf.shizuoka.ac.jp

あらまし　現在のスマートデバイスは，利用者自身でも動作状態の把握が難しく，不正な通信や動作が行われている場合に気づく術がない．そこで，スマートデバイスの状態 (アプリケーションの動作状況，操作内容など) を推定するシステムの実現を目指している．従来の研究では，スマートデバイスの状態推定手法として様々なアプローチが検討されているが，汎用性や導入コストの面で懸念が残る．本研究では，Wi-Fi 電波の通信媒体波及時におけるチャネル状態情報 (Channel State Information; CSI) を，CSI 収集用ファームウェアパッチである Nexmon を用いて収集・分析し，スマートデバイス状態推定を行う手法を提案する．基礎評価として，6 種類のアプリケーションに対して 8 種類の時系列モデルでの試行の結果，最大 87.6%の精度でアプリケーション推定が可能なことを確認した．また，分類アプリケーションの組み合わせによる精度変化を検証し，最大 100%で推定できることを明らかにした．

キーワード　Wi-Fi，CSI，Nexmon，時系列データ，スマートデバイス，アプリケーション推定

1　はじめに

近年，スマートフォンやパソコンなど多機能 IoT デバイス (スマートデバイス) が広く普及し，その活用シーンは多岐にわたる．世界のスマートデバイスの普及台数は年々増加しており，2025 年には約 440 億台に達すると予測されている．スマートデバイスの増加に伴って，攻撃者によるスマートデバイスに対しての不正操作や情報漏洩が懸念される．しかし，現在のスマートデバイスは，利用者自身でも動作状態の把握が難しく，不正な通信や動作が行われている場合に気づく術がない．

そこで，スマートデバイスにおけるアプリケーションの動作状況を推定するシステムの実現を目的としている．このシステムを活用することで，利用者自身がスマートデバイスの動作状況を把握し，第三者による不正な操作や情報漏洩を未然に防ぐことができる．また，スマートデバイスの使用状況を管理するシステムに組み込むことによって，より頑健なスマートデバイス管理システムの実現も期待できる．

従来の研究では，スマートデバイスの通信トラヒックから得られる統計情報を用いる手法 [1] や，状態遷移モデルと IoT センサ情報を用いたエッジコンピューティングによる手法 [2] が検討されている．しかし，通信トラヒックを用いた手法では，情報量に限界がありスマートデバイスの状態を詳細に捉えることが難しい．また，状態遷移モデルと IoT センサ情報を用いた手法では，状態遷移モデルの汎用性，導入コストの高さが懸念される．

本研究では，Wi-Fi 電波の通信媒体波及時における CSI を，CSI 収集用ファームウェアパッチである Nexmon [3] を用いて収集・分析し，スマートデバイス状態推定を行う手法を提案する．

2　関連研究

2.1　通信トラヒック分析による複数のスマートデバイスにおける機能推定手法の評価 [1]

ここでは，複数のスマートデバイスが有線・無線でルータに接続される環境を想定している．接続されるデバイス全てが，単一のエッジルータを介してインターネット接続を行う環境で通信トラヒックを収集している．また，通信トラヒックから 32 個の特徴量を算出し，RandomForest へ入力することで推定を行う手法が提案されている．評価データとして，8 機種のスマートデバイスで 8 種類の機能を実行した際の通信トラヒックを収集している．結果として，スマートデバイスの機種と機能の 16 通りに対して，88%の精度で分類できることを明らかにした．

しかし，通信トラヒックを用いた手法では，機種や機能による通信トラヒックの差異が小さく，スマートデバイスの状態を詳細に捉えることが難しい．これは，通信トラヒックによる特徴量が持つ情報量が少ないことが起因している．そのため，より詳細なスマートデバイスの状態推定を行うためには，通信トラヒック以外の情報を用いる必要がある．

2.2　状態遷移モデルと IoT センサによるエッジコンピューティングを用いたスマートデバイスの異常検知 [2]

実際の家庭環境に設置されているスマートデバイスの使用状

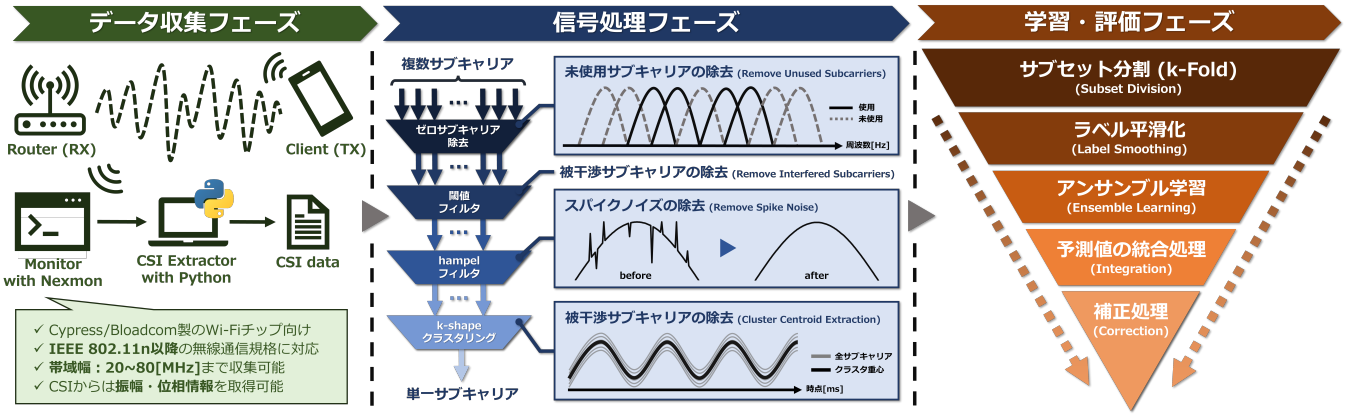

図 1　提案手法：概要図

況を，各デバイスに対応する IoT センサ値とエッジコンピュータを用いて収集している．また，家庭の活動状況を考慮した状態遷移モデルを，IoT センサ値を用いて構築し，不正操作を検出する手法が提案されている．

結果として，誤検出率：20.1%未満，検出率：72.3%でスマートデバイスの状態推定が可能なことを明らかにした．

しかし，この手法では各スマートデバイスに対して状態遷移モデルを構築する必要があり，汎用性の観点で懸念が残る．また，スマートデバイスの操作状況を収集するために，各デバイスそれぞれに IoT センサと，エッジコンピュータが必要であるため，導入コストの高さも懸念される．そのため，より汎用性が高く，導入コストの低いスマートデバイスの状態推定手法が必要である．

3　提 案 手 法

Wi-Fi 電波の通信媒体波及時における CSI を，CSI 収集用ファームウェアパッチである Nexmon を用いて収集・分析し，スマートデバイス状態推定を行う手法を提案する．CSI は，通信技術分野において広く使用される通信路の状態情報である．具体的には，Wi-Fi 電波が送受信デバイス間で伝播する際の電波散乱，フェージング，到来距離・角度による電力減衰などの複合的な要因を考慮したチャネル状態を表し，振幅・位相情報を持つ．CSI は，本来は通信品質の評価や通信路推定などに使用されるが，本研究では，スマートデバイスの動作状況を推定するための特徴量として利用する．CSI は通信トラヒックや RSSI(Received Signal Strength Indicator) と比較して取得できる情報が多く，スマートデバイスの状態を詳細に捉えることができる．CSI を収集するソフトウェアとして Nexmon を使用する．Nexmon を使用することで，CSI 収集不可能な低コストデバイスでも CSI 収集可能となる．

図 1 に提案手法の概要図を示す．提案手法は，「データ収集フェーズ」「信号処理フェーズ」「学習・評価フェーズ」の 3 つのフェーズで構成される．

3.1　データ収集フェーズ

「データ収集フェーズ」では，CSI を Nexmon を適用したデバイスを用いて取得し，Python ベースの CSI 解析プログラムを用いて，各サブキャリアの振幅・位相情報を取得する．このとき，スマートデバイスが Wi-Fi ルータのクライアントとして IEEE802.11n 規格以降の通信方式 (OFDM や MIMO などのマルチキャリア・アンテナ通信) かつ，帯域：2.4/5[GHz] で帯域幅：20/40/80[MHz] で通信を行う状況を想定する．

Nexmon 適用デバイスは，指定されたチャネルと帯域幅に該当する通信を観測するため，スマートデバイスが接続しているネットワークに接続する必要はない．そのため，多様なタスクに対して適用可能なデータ収集手法であり，システム全体として汎用性が高くなる．また，新たに導入するデバイスは Nexmon を適用するデバイスのみであるため，従来手法と比較して導入コストが低い．

3.2　信号処理フェーズ

「信号処理フェーズ」では，データ収集フェーズで取得した CSI データの前処理を行う．前処理として，ゼロサブキャリア除去，閾値フィルタ，hampel フィルタ，k-shape クラスタリングを適用する．

ゼロサブキャリア除去は，CSI データの中で振幅が 0 のサブキャリアを除去する処理である．IEEE802.11n 規格以降で使用されるマルチキャリア通信では他チャネルとの干渉を抑制するため，一部のサブキャリアをデータ通信に使用しない工夫がされている [4]．そのため，それらのサブキャリア情報の除去をすることで通信に使用されたサブキャリア情報のみを抽出する．

閾値フィルタでは，振幅情報が一定値を超えるサブキャリアを除去するフィルタである．ゼロサブキャリアに隣接するサブキャリアは他チャネルからの干渉を受けやすく，ノイズが多く含まれるため，学習には不向きである．具体的には，ゼロサブキャリアを除く k 個のサブキャリア $S_k \ni s_k$ に対して式 1 を適用し，閾値 α を超えるサブキャリアを除去する．

$$threshold(S_k) = \begin{cases} S_k & if\ s_k < \alpha \\ None & otherwise \end{cases} \quad (\alpha > 0) \qquad (1)$$

hampel フィルタ [5] では，選定されたサブキャリア集合の各要素に対してノイズ処理を行う．Nexmon によって収取された

CSI データにはスパイクノイズが多く含まれている．そのため，外れ値検知アルゴリズムである hampel フィルタはノイズ除去として有効に作用する．具体的には，単一サブキャリアに対して，中央値 $\widetilde{X_i}$，標準偏差 σ_i のスライディングウィンドウ $X_i \ni x_i$ を作成する．その後，全スライディングウィンドウに式 2 を適用し，$\beta\sigma_i$ を超える要素 x_i を中央値 $\widetilde{X_i}$ に置換する．

$$hampel(X_i) = \begin{cases} \widetilde{X_i} & if\ |x_i - \widetilde{X_i}| > \beta\sigma_i \\ x_i & otherwise \end{cases} \quad (\beta > 0) \quad (2)$$

k-shape クラスタリングでは，形状ベースの時系列クラスタリング手法である k-shape [6] を適用し，全サブキャリアクラスタの重心信号を抽出する．ここで，サブキャリアは変動幅が異なる程度の差異しか見られず，相関係数が非常に高い．そのため，全サブキャリア情報をすべて使用すると学習コストが高くなり，共線性による過学習に陥る可能性がある．時系列クラスタリングによって得られる重心信号を代表値として学習に使用することで，学習コストの低減を図る．

3.3 学習・評価フェーズ

「学習・評価フェーズ」では，前処理を行った CSI データを入力として時系列モデルによる学習・評価を行う．具体的には，ラベル平滑化 [7] を施した学習データに対して，k-Fold アンサンブル学習モデルを構築し，予測値の統合・補正を行うことで最終予測値を得る．ラベル平滑化は，K クラスの One-Hot エンコーディングされた目的変数 $y_k = [y_1, ..., y_K]$ に対して，ノイズ γ を加える手法 (式 3) である．これにより，モデルの過学習を抑制し，類似した特徴量に対する分類精度，汎化性能が向上する．

$$smoothing(y_i) = y_i(1-\gamma) + \frac{1}{K}\gamma \quad (0 < \gamma < 1.0) \quad (3)$$

k-Fold アンサンブル学習時において使用するサブセットはスライディングウィンドウ形式で作成され，目的変数の分布はランダムである．また，スライディングウィンドウは，ウィンドウサイズ (入力時点数) が可変かつ，複数の目的変数が含まれないように作成する．分割された k 個のサブセットそれぞれに対して，同一パラメータ・アーキテクチャの学習モデルを構築する．得られる k 個の予測値は，相加平均によって統合する．また，統合された予測値に対して，過去の予測値から妥当性を評価し，適切な値に補正することで最終予測値を得る．

4 基礎評価

4.1 データ収集フェーズ

図 2 に，基礎評価における CSI データの収集環境を示す．スマートデバイスとして iPhone12，Wi-Fi ルータとして WSR-1800AX4P-WH を使用した．また，スマートデバイス上で動作するアプリケーションは，TikTok，LINE，パズル&ドラゴンズ (Puzzle)，雀魂 (Mahjong)，YouTube，コミックシーモア (Comic) の 6 種類とした．Nexmon を適用する CSI 収集デバイスとして，安価かつ入手が容易なことから RaspberryPi4B を使用した．通信条件は，スマートデバイスと Wi-Fi ルータ間の距離を 1.5[m] で配置し，通信規格：IEEE802.11n の帯域：2.4[GHz]，帯域幅：20[MHz] として通信を行った．収集条件は，サンプリングレート：5.0[Packet/s] として収集を行った．

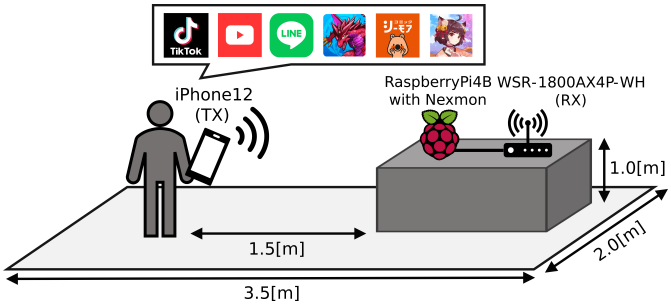

図 2　基礎評価：CSI データ収集環境

表 1　実 行 環 境

CPU	Intel Core i9-13900KF
GPU	GeForce RTX 3090
OS	Ubuntu 22.04.2 LTS
Software	Python3.9, tensorflow2.12, CUDA11.8

4.2 信号処理フェーズ

収集した CSI は，64 個のサブキャリア情報で構成される．IEEE802.11n 規格では，8 つのサブキャリアはデータ搬送には使用されないため，ゼロサブキャリアとして除去を行った．また，式 1 の閾値フィルタ ($\alpha = 3000$) を適用し，振幅情報が一定値を超えるサブキャリアの除去を行った．また，抽出された各サブキャリアに対して，式 2 の hampel フィルタ ($\beta = 2.0$) によるスパイクノイズの除去を行い，k-shape を用いて全サブキャリアの代表値を抽出した．

4.3 学習・評価フェーズ

学習・評価フェーズでは，表 1 に示す実行環境下で学習を行った．目的変数に対して式 3 のラベル平滑化処理 ($\gamma = 0.1$) を施した．また，k-Fold アンサンブル学習モデルに入力するサブセットの分割数は 3 とし，相加平均によって予測値の統合を行った．補正処理として，分類タスクにおいて誤分類した箇所がスパイク状に出現することから，式 2 の hampel フィルタ ($\beta = 1.5$) を使用した．

4.3.1 時系列モデルと入力時点数の選定

k-Fold アンサンブル学習モデルとして，8 種類の時系列モデル (RandomForest，TCN，1D-CNN，LSTM-FCN，Transformer，1D-ResNet，1D-DenseNet) を試行した．

表 2 に，各時系列モデルによる精度比較を示す．各時系列モデルに対して，入力時点数を 200，250，300，350，400，450[Packet] と変化させ，各学習モデルで得られる最大精度を記録した．各時系列モデルにおける準最大精度を下線，最大精度を太字で表記した．ラベル数に極端な偏りがないことから，評価指標は Accuracy[%] とした．最良結果として，学習

表 2　時系列モデルと入力時点数による精度比較 (Accuracy[%])

入力時点数	時系列モデル							
	RandomForest	TCN	1D-CNN	LSTM-FCN	Transformer	1D-ResNet	1D-DenseNet	LSTM
200	**67.9**	65.9	66.7	63.4	56.7	**72.1**	57.5	36.7
250	64.0	72.3	70.9	62.3	60.9	67.4	75.4	**51.6**
300	63.5	84.0	70.2	68.4	59.7	71.1	76.5	41.2
350	60.8	**87.6**	71.1	**68.5**	61.6	63.9	73.9	40.9
400	57.1	71.7	72.0	63.1	61.8	68.6	72.1	47.1
450	58.5	60.4	**76.0**	68.0	**62.4**	66.3	**80.9**	39.3

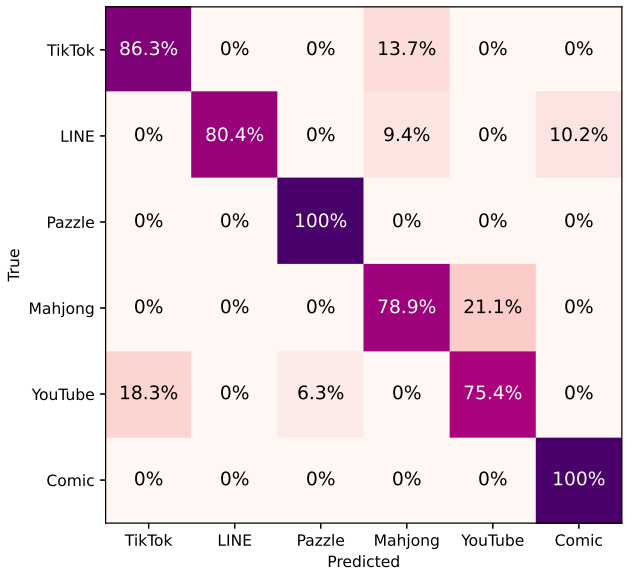

図 3　最良結果における分類結果 (混同行列)

モデル：TCN(Temporal Convolutional Network)，入力時点数：350[Packet](≃ 70[s]) とした時の Accuracy：87.6[%](F 値：0.871) を確認した．

図 3 に，表 2 で最良結果を確認した条件下での分類結果 (混同行列) を示す．分類難易度がアプリケーションによって異なり，誤分類しやすい組み合わせが存在することが確認できる．

4. 3. 2　入力時点数による精度比較

本節の第 1 項で最良結果を確認した学習モデル：TCN に対して，入力時点数を変化させた場合の精度比較を行った．入力時点数を 100〜500[Packet]，hampel フィルタのパラメータを $\beta = 1.0$〜3.0 として試行した．

図 4 に，各入力時点数ごとに得られた最大精度の推移を示す．

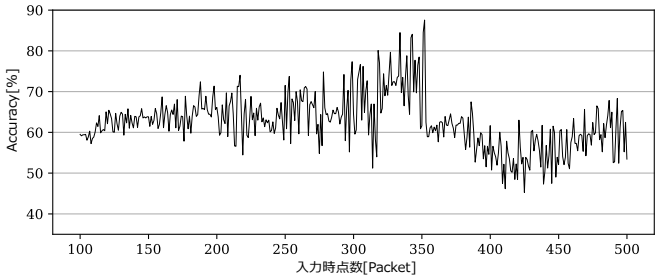

図 4　入力時点数による精度比較

表 3　アプリケーションによる精度比較 (● 分類対象，○ 非分類対象)

アプリケーションの組み合わせ						Acc.[%]
TikTok	LINE	Puzzle	Mahjong	YouTube	Comic	
●	●	●	●	●	○	94.4
●	●	●	●	○	●	71.9
●	●	●	○	●	●	90.7
●	●	○	●	●	●	89.1
●	○	●	●	●	●	94.1
○	●	●	●	●	●	89.8
●	●	●	●	○	○	66.5
●	●	●	○	○	●	66.4
●	●	○	○	●	●	91.8
●	○	○	●	●	●	90.3
○	○	●	●	●	●	95.4
○	●	●	●	●	○	**100.0**

入力時点数が増加するに伴い，分類精度の向上する傾向が確認できる．しかし，それと同時に入力時点数が 350[Packet] を超えたあたりで急激な精度低下がみられる．

4. 3. 3　アプリケーションによる精度比較

本節の第 1 項で最良結果を確認した学習モデル：TCN に対して，アプリケーションの組み合わせによる精度比較を行った．

表 3 に，分類対象とするアプリケーションの組み合わせに対する精度比較を示す．分類対象とするアプリケーションの組み合わせによって，分類精度に差異が生じることが確認できる．

5　考　　察

はじめに，図 3 における分類結果について考察する．“Puzzle” と “Comic” はどのアプリケーションとも誤分類することなく，適切に分類ができている．しかし，“Mahjong” と “YouTube” は分類精度が 80%を下回っており，基礎評価で対象としたアプリケーションの中で分類難易度が高いと言える．“Mahjong” は “YouTube” との誤分類が目立っているが，これは両者の操作方法 (横持ち，操作時の指の動きなど) が類似することに起因していると考えられる．また，“YouTube” は “TikTok” との誤分類が目立っており，これは両者の内部特性 (ストリーミング，バッファリングなど) が類似することに起因していると考えられる．

次に，図 4 における入力時点数による精度比較ついて考察する．一般的に，入力時点数が増加すると入力データの情報量が増加し，学習モデルの分類精度は増加する．しかし，特定の入力時点数 (=350[Packet]) を超えると，各アプリケーションが

持つ特徴量の差異が小さくなり，誤分類を引き起こしている可能性がある．

最後に，表 3 におけるアプリケーションによる精度比較について考察する．一般的に，分類クラス数の減少に伴って，分類精度は増加する．しかし，分類対象とするアプリケーション数が減少した場合に，分類精度が低下する組み合わせが存在している．提案手法における学習・評価フェーズでは，モデルの汎化性能の向上を目的としたラベル平滑化処理行っている．ラベル平滑化処理は，分類クラス内に類似する特徴量を持つ組み合わせが存在する場合に有効な手法である．しかし，分類アプリケーション内に特徴量が類似する組み合わせが存在しない場合，ラベル平滑化はモデルの性能を低下させてしまう可能性がある．このことから，分類対象とするアプリケーションの組み合わせによっては類似した特徴量を持つアプリケーションが少ないため，分類精度が低下していると考えられる．

6 議　　論

はじめに，スマートデバイス状態推定において使用する CSI の特徴量に影響を与える要因について議論する．スマートデバイス操作によって発生するトラヒックやトランザクション，操作方法・位置・環境などが挙げられる．それらを明かにするためには，トラヒックやトランザクションを可視化し，CSI 特徴量との関連性を検証する必要がある．また，非操作時と操作時の場合と操作位置に動的な変化を加えた場合の CSI 特徴量をそれぞれ分析する必要がある．

次に，基礎評価で対象としたスマートデバイス，およびアプリケーション以外に対する有効性について議論する．スマートデバイスに関しては，通信を行う無線通信規格，帯域幅などの通信条件を一致させた場合は，他のスマートデバイスにも適用可能であると考えられる．しかし，アプリケーションに関しては，同様のアプリケーションであってもバージョンや設定による差異が考えられる．それらを解明するには，異なるスマートデバイスと，異なるバージョンや設定のアプリケーションの組み合わせによる CSI 特徴量の変化を検証する必要がある．

最後に，複数人のスマートデバイスを対象とした場合の有効性について議論する．複数人が同時に，通信条件の異なるスマートデバイスを操作する場合，単一の Nexmon 適用デバイスで複数の CSI を取得することはできない．そのため，通信条件が異なる場合は，複数の Nexmon 適用デバイスを用いて，複数人のスマートデバイス状態推定を行う必要がある．通信条件が同一であっても，複数人を対象とする場合は，CSI 特徴量を分離する必要がある．

7 おわりに

Nexmon による CSI を用いたスマートデバイス状態推定手法を提案し，基礎評価を行った．8 種類の時系列モデルを試行した結果，学習モデル：TCN によって最大分類精度 87.6%を確認した．また，入力時点数による精度比較を行うことで，スマートデバイス状態推定における最適な入力時点数を明らかにした．さらには，アプリケーションの組み合わせによる精度比較を行うことで，アプリケーションによる分類難易度を検証した．

本研究のスマートデバイス状態推定手法は CSI データを用いるため，通信トラヒックを使用する従来手法 [1] と比較して，取得できる情報が多く，スマートデバイスの状態を詳細に捉えることができる．また，複数の IoT センサとエッジコンピュータを用いる従来手法 [2] と比較して，新たに導入するデバイスが少なく，導入コストが低いといえる．加えて，ドメイン知識に基づいた状態遷移モデルを構築する必要がないため，汎用性が高いといえる．

今後の展望として，スマートデバイス状態推定モデルの精度向上，結果に起因する内外的要因 (個人差や環境，動作アプリケーションの数・種類など) についての検証を進める．また，時系列向けの半教師あり学習 [8] や，データ拡張手法 [9] と組み合わせることで，学習コストが低く，汎用性の高い学習を検討する．最終的には，本研究の成果を活用し，スマートデバイスの動作状態を推定するシステムの実現を目指す．

謝　　辞

本研究の一部は，静岡大学グリーン科学研究所プロジェクト研究支援 (23205) を受けたものである．

文　　献

[1] 祐一服部, 豊荒川, 創造井上. 通信トラヒック分析による複数の iot デバイスにおける機能推定手法の評価. マルチメディア，分散，協調とモバイルシンポジウム 2022 論文集, Vol. 2022, pp. 655–661, 2022.

[2] 田中雅弘, 山内雅明, 大下裕一, 村田正幸, 上田健介, 加藤嘉明. 家庭活動の状況推定を用いたスマートホームネットワークの異常検出手法. IEICE Technical Report; 信学技報, Vol. 119, No. 461, pp. 219–224, 2020.

[3] Francesco Gringoli, Matthias Schulz, Jakob Link, and Matthias Hollick. Free your csi: A channel state information extraction platform for modern wi-fi chipsets. pp. 21–28, 2019.

[4] Defeng David Huang and Khaled Ben Letaief. Carrier frequency offset estimation for ofdm systems using null subcarriers. *IEEE Transactions on Communications*, Vol. 54, pp. 813–823, 2006.

[5] Ronald K. Pearson, Yrjö Neuvo, Jaakko Astola, and Moncef Gabbouj. The class of generalized hampel filters. In *EUSIPCO*, pp. 2501–2505, 2015.

[6] John Paparrizos and Luis Gravano. k-shape: Efficient and accurate clustering of time series. *SIGMOD Rec.*, Vol. 45, pp. 69–76, 2016.

[7] Rafael Müller, Simon Kornblith, and Geoffrey Hinton. When does label smoothing help?, 2020.

[8] Haoyi Fan, Fengbin Zhang, Ruidong Wang, Xunhua Huang, and Zuoyong Li. Semi-supervised time series classification by temporal relation prediction. In *ICASSP*, pp. 3545–3549, 2021.

[9] Arthur Le Guennec, Simon Malinowski, and Romain Tavenard. Data augmentation for time series classification using convolutional neural networks, 2016.