IoT機器の不正機能検知におけるLLMの有効性検証

伊東 実聖[1,a)]　上園 大智[2,b)]　九鬼 琉[2,c)]　宮本 岩麒[2,d)]　佐々木 貴之[3,e)]　吉岡 克成[4,f)]

概要：近年，IoT 機器への不正機能埋め込みによるセキュリティ脅威が深刻化している．従来のシグネチャベース検知手法では，多様な実装形態を持つ不正機能の検知が困難であり，大規模言語モデル（LLM）を用いたソースコード解析による検知手法が注目されているが，IoT ファームウェアにおける有効性は十分に検証されていない．本研究では，LLM が IoT ファームウェアの不正機能検知において実用的な有効性を持つかを検証する．評価実験では，OpenWrt に不正機能を埋め込んだデータセット DarkWrt とバックドア検知ベンチマーク ROSARUM のソースコードを Claude Opus 4 に入力し，不正機能の有無を判定させた．その結果，DarkWrt では検知率 99%(9.9/10 ファイル)，誤検知率 0%(0/84 ファイル)，ROSARUM では検知率 85.7%(18/21 ファイル)，誤検知率 0.0086%(7/81753 ファイル) を達成した．これらの結果は，提案手法が高い検知精度を有していることを示している．さらに，3 つのルーター製品のファームウェアの解析では，Claude Opus 4 は解析対象ファイル全体のうち平均 0.47%を不正機能の疑いがありと判定し，その中には既知の不正機能が含まれていることを確認した．さらに，手動解析により Claude Opus 4 が検出したファイルを調査したところ，未知の不正機能 1 件を発見した．これらの実機での結果は，実環境における提案手法の有効性を示唆している．加えて，GPT-4o，GPT-3.5-turbo，o3-mini についても DarkWrt を用いた評価を行い，LLM モデルの違いにより検知性能に顕著な差異が見られることを確認した．

キーワード：不正機能，IoT 機器，データセット，大規模言語モデル

Verification of the Effectiveness of LLMs in Detecting Potentially Unwanted Functions in IoT Devices

Misato Ito[1,a)]　Daichi Uezono[2,b)]　Ryu Kuki[2,c)]　Iwaki Miyamoto[2,d)]　Takayuki Sasaki[3,e)]
Katsunari Yoshioka[4,f)]

Abstract: In recent years, security threats arising from the embedding of potentially unwanted functions in IoT devices have become increasingly severe. While conventional signature-based detection methods face difficulties in identifying potentially unwanted functions with diverse implementation patterns, detection approaches utilizing large language models (LLMs) for source code analysis have gained attention. However, their effectiveness in IoT firmware environments remains insufficiently validated. This study investigates whether LLMs demonstrate practical effectiveness in detecting potentially unwanted functions within IoT firmware.In the evaluation experiments, source code from DarkWrt, a dataset embedding potentially unwanted functions in OpenWrt, and the ROSARUM backdoor detection benchmark were input to Claude Opus 4 for potentially unwanted function determination.The results demonstrated that Claude Opus 4 achieved a detection rate of 99% (9.9/10 files) with a false positive rate of 0% (0/84 files) on DarkWrt, and a detection rate of 85.7% (18/21 files) with a false positive rate of 0.0086% (7/81,753 files) on ROSARUM. These results demonstrate that the proposed method possesses high detection accuracy. Furthermore, evaluation in three real router environments showed that Claude Opus 4 identified an average of 0.47% of the total analyzed files as suspected of containing potentially unwanted functions, and verification confirmed that known potentially unwanted functions were included among these detections. Additionally, manual analysis of files detected by Claude Opus 4 revealed one previously unknown potentially unwanted function. The findings from these real-world device evaluations suggest the effectiveness of the proposed method in practical environments.Moreover, evaluation using DarkWrt was conducted for GPT-4o, GPT-3.5-turbo, and o3-mini, confirming that notable performance differences exist among different LLM models.

Keywords: Potentially unwanted function, IoT devices, Dataset, Large Language Model

1. はじめに

近年，IoT 機器は社会インフラの重要な構成要素となっているが，その複雑な製造プロセスに起因するセキュリティ脅威が深刻化している．中でも，悪意のある関係者によって故意に埋め込まれる不正機能は重大な脅威となっている [1], [2], [3], [4]．論文 [4] では，このような不正機能を「意図性と不正性を満たした脆弱性」と定義しており，意図性とは悪意を持って故意に埋め込まれていること，不正性とは社会的に許容できないことを意味している．本研究では論文 [4] で定義されている不正機能の定義を採用し，この定義に基づく不正機能を大規模言語モデル (LLM) による検知対象とする．

既存研究により，IoT 機器における不正機能検知には 2 つの主要な課題が明らかにされている．第一に，既存のセキュリティ監査技術やバックドア検知技術は特定の脆弱性に特化しており，多様な実装の IoT 機器や巧妙化したバックドアに対する汎用的適用が困難である [1], [2], [3]．第二に，IoT ファームウェアは多様なプログラミング言語・形式が混在する複合的環境を形成しており，従来の単一解析手法では包括的な不正機能検知が困難である．近年，LLM がコード解析分野で注目されているが，既存の LLM ベース検知手法は単一言語・単一形式のコードを前提としており，IoT ファームウェアの複合的環境への適用可能性は未検証である [5], [6]．特に，IoT ファームウェアに埋め込まれた不正機能の LLM による検知の有効性については十分に論じられていない．

本研究では，LLM による IoT 機器の不正機能検知の有効性を検証する．段階的な検証アプローチの初期段階として，主な解析対象は人間可読ファイル群 (ソースコード，設定ファイル，スクリプトなど) としたが，市販ルーターに対する検知では一部のバイナリファイルも解析対象とした．

実験では，まず論文 [5] を参考にしたプロンプト構築を行い，IoT 機器の不正機能データセットである DarkWrt[4] を用いて複数 LLM モデルの比較評価を実施した．次に，LLM による不正機能検知の汎化性能を検証するため，バックドア検知ベンチマーク ROSARUM[7] を用いて検証を行った．ROSARUM はオープンソースプロジェクトベースのバックドアに加え，実際のルーターファームウェアから発見されたバックドアも含んでおり，DarkWrt と比較してより多彩な起源を持つデータセットである．さらに，実環境での適用可能性を検証するため，不正機能が報告されているものも含む，3 種類のルータのファームウェアに対して提案手法を適用した．

実験結果として，比較評価した 4 つの LLM モデル (GPT-4o，GPT-3.5-turbo，o3-mini，Claude Opus 4) の中で，Claude Opus 4 が不正機能検知において最も高い性能を示した．なお，Claude Opus 4 は Claude Code を通じて利用されており，ファイルアクセス方法や操作環境が他のモデルと異なることに留意が必要である．DarkWrt では 10 回実験を行い，平均性能を評価した．DarkWrt では検知率 99%(9.9/10 ファイル)，誤検知率 0%(0/84 ファイル)，ROSARUM では，検知率 85.7%(18/21 ファイル)，誤検知率 0.0086%(7/81753 ファイル) を達成した．3 種類のルータ製品のファームウェアの解析においては，検査対象の 5965 ファイルから 20 個を不正機能の可能性があるファイルとして検知し，そのうち 1 個は実際に既知の不正機能，もう 1 個は未知の不正機能であることを確認した．それ以外の 18 個については今後詳細分析を実施予定である．発見した不正機能は情報処理推進機構 (IPA) に報告する予定である．

本論文の貢献は以下の通りである．

- DarkWrt および ROSARUM ベンチマーク，ならびに市販 IoT 機器を用いた検知実験により，Claude Opus 4 が実用環境において高い適用可能性を有することを実証した．
- GPT-4o，GPT-3.5-turbo，o3-mini，Claude Opus 4 の比較評価により，モデル選択が検知性能に大きく影響することを明らかにした．

2. 関連研究

既存研究にて，IoT 機器における不正機能の事例調査，データセットの構築および検知手法の調査や提案が行われている．

IoT 機器の脆弱性分類とセキュリティ監査・バックドア検知技術に関する包括的な分析が実施されている [1], [2], [3]．これらの研究では，既存のセキュリティ監査技術における主要な課題として，特定の脆弱性への特化による汎用性の欠如，高いプラットフォーム依存性，偽陽性・偽陰性率の高さ，および単一検知手法の限界が指摘されている．

不正機能検知手法の評価基盤として，複数のデータセットが構築されている．具体的には，IoT 機器に埋め込まれ

[1] 横浜国立大学理工学部
College of Engineering Science, Yokohama National University
[2] 横浜国立大学大学院環境情報学府
Graduate School of Environment and Information Sciences, Yokohama National University
[3] 横浜国立大学先端科学高等研究院
Institute of Advanced Sciences, Yokohama National University
[4] 横浜国立大学大学院環境情報研究院/先端科学高等研究院
Faculty of Environment and Information Sciences, Yokohama National University / Institute of Advanced Sciences, Yokohama National University
[a)] ito-misato-wh@ynu.jp
[b)] uezono-daichi-ct@ynu.jp
[c)] kuki-ryu-dr@ynu.jp
[d)] miyamoto-iwaki-yk@ynu.jp
[e)] sasaki-takayuki-yv@ynu.ac.jp
[f)] yoshioka@ynu.ac.jp

る不正機能の事例調査に基づいて，OpenWrt に不正機能を埋め込んだデータセットである DarkWrt が構築されている [4]．また，実際に発見されたバックドア 7 個と人工的に作成したバックドア 10 個から構成される ROSARUM データセットも構築されている [7]．

不正機能の検知手法として，ROSA と呼ばれるファジングベースのコードレベルのバックドア検知手法が提案され，その有効性が ROSARUM を用いた評価で実証されている [7]．ROSA をさらに改良するため，正常な IoT 機器と検査対象 IoT 機器に同一のテストデータを用いたバイナリファジングを行い，システムコールの発行順序を比較する改良手法も提案されている [8]．

LLM を活用した悪性コードおよび悪性パッケージの検知実験が実施されている [5], [6]．しかしながら，これらの手法は検知対象が Python 単一言語で記述されたコードやパッケージレポジトリに限定されており，IoT ファームウェアのような多様なプログラミング言語，設定ファイル，スクリプト，バイナリファイルが混在する複合的な環境への適用可能性は限定的である．

3. 課題/リサーチクエスチョン

既存研究より，IoT 機器における不正機能検知には 2 つの主要な課題が明らかになっている．第一に，既存のセキュリティ監査技術とバックドア検知技術は特定の脆弱性に特化しており，多様な IoT 機器や巧妙化するバックドアに対する汎用的適用が困難である．第二に，IoT ファームウェアは C 言語，アセンブリ，シェルスクリプト，設定ファイルなど多様な言語・形式が混在する複合的環境を構成しており，単一の解析手法による全体的な検知が困難である．既存の LLM ベース検知手法は単一言語・単一形式のコードを対象としているため，複合的な IoT ファームウェア環境における不正機能検知の有効性は十分に検証されていない．

これらの課題に対して，本研究では「IoT 機器中の不正機能検知において LLM は有効性を示すのか」という問いに答えることを目的とする．具体的には，IoT 機器を模したデータセットと実際のファームウェアを対象として，多様な IoT 環境における検知性能と複合的環境での検知可能性を実験的に明らかにする．この実験により，既存手法の限界を補完する新たな検知アプローチの実現可能性を示し，IoT セキュリティ分野における LLM 活用の指針を提示する．

4. 検証手法

IoT 機器のファームウェアに埋め込まれた不正機能を検知する LLM の有効性を多角的に検証するため，以下の段階的な 4 つの実験を設計・実施した．

4.1 DarkWrt における LLM モデル比較

DarkWrt データセット（不正機能ファイル 10 個，正常機能ファイル 84 個）を用いて，LLM モデル選択が検知性能に与える影響を評価した．評価対象モデルは，汎用性が高い GPT-4o，コスト効率性に優れた GPT-3.5-turbo，推論特化した o3-mini，コード解析に特化した Claude Opus 4 の 4 種類とした．各モデルに対して 4.8 章で述べたプロンプトを適用し，検知率と誤検知率で性能比較した．各モデルにつき 10 回の検知実験を実施した．

4.2 DarkWrt における Claude Opus 4 による検知

前実験において最も優れた検知性能を示した Claude Opus 4 を対象に，プロンプト設計が検知精度に与える影響を分析した．不正機能の詳細な定義を明示的に含む第一のプロンプトと不正機能の詳細な定義を含まない第二のプロンプトにより検知実験を行った．この実験の目的は，不正機能の具体的な定義を提示することが LLM の判断プロセスに与える影響を検証および分析することである．各プロンプト条件において同一の DarkWrt データセットを用いて検知実験を実施し，検知率，誤検知率の分析を通じて，Claude Opus 4 の推論プロセスの特性を明らかにした．各条件につき 10 回の検知実験を実施した．

4.3 公開データセット ROSARUM における検知

LLM による不正機能検知の汎化性能をより多彩なデータセットで検証するため，公開ベンチマークデータセット ROSARUM を用いた評価を実施した．4.1 節で述べた実験で最高性能を示した Claude Opus 4 を主要評価対象とし，ROSARUM の全 17 個のバックドアに対する検知実験を実施した．比較対象として，GPT-4o については計算資源の制約を考慮し，ファイル構成と複雑度に基づいて選定した代表的な 5 つのバックドアでの評価を併用した．各バックドアにつき 1 回の検知実験を実施した．

4.4 市販ルーターファームウェアにおける実証実験

LLM ベース不正機能検知手法の実環境適用可能性を評価するため，市販 IoT 機器の実際のファームウェアを対象とした検証を実施した．実験対象として，過去に意図性と不正性の観点から不正機能の可能性を有する脆弱性 (隠し Telnet) が報告されている機種を含む，市販無線ルーター 3 機種のファームウェアを選択した．解析手法として，各機器のファームウェアイメージは各製造者の公式サイトから入手し，Firmware Mod Kit および Unblob を用いてファームウェアイメージを展開した．抽出したファイルに対し，Claude Opus 4 による検知実験を実施した．検知結果の妥当性評価は，既に報告されている脆弱性情報と照合することで行った．各機器につき 1 回の検知実験を実施した．

4.5 実験環境

4.5.1 LLM モデル

本研究では，異なる性能特性を持つ 4 種類の LLM モデル (GPT-4o，GPT-3.5-turbo，o3-mini，Claude Opus 4) を選択し，比較評価を実施した．OpenAI 社のモデル (GPT-4o，GPT-3.5-turbo，o3-mini) は OpenAI API 経由でアクセスし，API リクエストの送信とレスポンス処理には openai ライブラリ（バージョン 1.75.0）を使用し，temperature=1.0 と設定した．Claude Opus 4 は検知対象となるソースコードが配置されたディレクトリに移動後，Claude Code ツールを通じてコマンドライン環境から実行した．各モデルの特徴は以下の通りである．

- **GPT-4o**: OpenAI 社が開発したマルチモーダル対応の大規模言語モデルであり，テキスト処理と推論において優れた性能を発揮する [9].
- **GPT-3.5-turbo**: 同社の軽量版モデルで，処理速度とコスト効率に優れる [10].
- **o3-mini**: 同社の推論特化型モデルであり，STEM 分野における論理的推論に最適化されている [11].
- **Claude Opus 4**: Anthropic 社が開発した大規模言語モデルで，複雑なコーディングタスクと高度な推論に最適化されている [12].

これらのモデル選択により，汎用性 (GPT-4o)，経済性 (GPT-3.5-turbo)，推論特化 (o3-mini)，コード特化 (Claude Opus 4) という異なる設計思想を代表し，不正機能検知タスクにおける各手法の有効性を多角的に評価することを可能とした．

4.5.2 データセット

DarkWrt: DarkWrt は，オープンソースのルーター用 Linux ディストリビューションである OpenWrt に不正機能を埋め込み構築したデータセットである [4]．本実験では，94 個のファイル (不正機能 10 ファイル，正常機能 84 ファイル) を検知対象とした．検知対象となる 8 種類の不正機能（10 ファイル）の詳細を表 1 に示す．

ROSARUM: ROSARUM は，Dimitri らによって開発されたバックドア検知手法評価用の公開ベンチマークデータセットである [7]．本データセットは，実際に発見されたバックドア 7 個（Authentic）と研究用に人工的に作成されたバックドア 10 個（Synthetic）の計 17 個で構成されている．Authentic バックドアには，実際のルーターメーカーが開発したファームウェアに含まれていたバックドア 4 個と，オープンソースプロジェクトへのサプライチェーン攻撃で注入されたバックドア 3 個が含まれている．Synthetic バックドアは，9 個はソフトウェア脆弱性検出評価用ファジングベンチマークである MAGMA の標準プログラムをベースとして構築され，1 個は論文の説明用に作成されたものである．

4.6 評価基準

評価指標として検知率と誤検知率，抽出率を採用した．正解ラベルが既知の実験 (DarkWrt，ROSARUM) では検知率と誤検知率を，正解ラベルが未知の実機ファームウェアでは抽出率を用いて評価を行った．検知率=不正機能であると検知したファイル数/不正機能ファイル総数，誤検知率=不正機能であると検知したファイル数/正常機能ファイル総数，抽出率＝検知ファイル数/総ファイル数と定義した．

検知成功の定義として, DarkWrt では不正機能が実装されている 10 個のファイルの検知を成功事例とし, ROSARUM ではバックドアパッチが適用されたファイルの検知を成功事例として定義した．市販ルーターファームウェアを用いた実験では，既知の脆弱性情報との照合により検知結果の妥当性を評価した．

実験では，OpenWrt プロジェクトのソースコード群および OSS ライブラリに対するメーカーパッチファイル群を「正常機能ファイル」と仮定し，評価を実施した．

4.7 出力形式

検知実験では，JSON 形式で以下の 6 項目を含む構造化された出力を指示した．(1) 不正機能が埋め込まれているファイルパス，(2) 不正機能に該当する行，(3) 不正機能に該当するコード部分，(4) 不正機能のタイプ，(5) 影響範囲，攻撃シナリオを含む不正機能の説明，(6) 不正機能であると判断した理由である．

4.8 使用したプロンプト

本研究では，IoT ファームウェアに埋め込まれた不正機能の検知において 2 種類のプロンプトを構築し，適用した．本節では，各プロンプトの構成および特徴について詳述する．

4.8.1 OpenAI 製 LLM モデルに用いたプロンプト設計

先行研究 [5] を参考にプロンプトを構築した．構築したプロンプトの特徴は以下の通りである．

- 言語を英語とし，LLM の理解精度向上を図った
- 権限昇格，情報漏洩，設定変更，キルスイッチ，隠しアカウント，自動実行機能，Telnet 起動，痕跡消去を検知対象とした
- 難読化された文字列を用いた不正行為に特に注意を払う指示，正当なシャットダウン，権限昇格操作を不正機能から明示的に除外する指示をした

このプロンプトにより，誤検知率の削減と検知精度の向上を同時に達成することを目指した．

以下が使用したプロンプトである．

表 1: DarkWrt の不正機能一覧

番号	不正機能名	説明
1	隠しアカウント (passwd ファイル)	隠しアカウントのユーザ名や権限情報を設定
2	隠しアカウント (shadow ファイル)	隠しアカウントのパスワードを設定
3	アカウント権限昇格機能	パスワードなしでの SUDO 実行やアカウントを root グループに追加
4	パケット盗聴機能	通信をパケットキャプチャし，特定のファイルに格納
5	情報転送機能 (情報収集・送信)	情報を収集し，指定したサーバへ送信
6	情報転送機能 (cron)	情報転送機能を定期的に自動実行
7	Telnet 起動・痕跡消去機能	特定のパケットを受信することで Telnet の起動や不正機能ファイルを消去
8	自動実行機能	Telnet 起動機能と痕跡消去機能をバックグラウンドで自動実行
9	ファイアーウォール設定変更機能	指定ポートのファイアウォール設定を監視・変更
10	キルスイッチ	強制シャットダウンスクリプトを作成・有効化し，即座に再起動

Open AI プロンプト

You are an excellent code analyzer. Please accurately analyze whether there are any malicious functions embedded in the code, especially in the context of obfuscated strings being used for illegal actions.
When analyzing, please pay special attention to the following points:

- Consider the context and intent of the code
- Distinguish between ordinary vulnerabilities and malicious functions
- Minimize false detections

Task
Analyze the given code and determine if it contains any malicious functions.
Constraints

- Output the judgment result in the specified JSON format
- Select the type of risk from the following:Privilege Escalation, Information Leakage, Configuration Modification, Kill Switch, Hidden Accounts, Automatic Execution, Telnet/Backdoor, Trace Erasure
- Explain the judgment reason
- Extract the referenced code portion accurately

Definition of malicious function
{Describe the definition of malicious functions.}

Exclude legitimate shutdown and privilege escalation operations from malicious functions.

Target malicious functions
{Provide 8 types Target malicious functions}
Output format
{Designated output format}
code
{Target code for detection}

4.8.2 Claude Opus 4 に用いたプロンプト

Claude Opus 4 は高度なコーディング能力や推論能力を有する大規模言語モデルである．Claude Opus 4 の持つ高度なコード理解能力を考慮し，OpenAI 製 LLM モデルに適用したプロンプトとは異なり，詳細な不正機能の種類や，具体的な判断基準の提示を意図的に省略した．代わりに，不正機能の定義，全ファイルを対象とした包括的検知の指示，出力形式の 3 つのみをプロンプトとして与えた．

この設計方針は，Claude Opus 4 が過度な制約なしに自律的なコード解析を実行できるという仮定に基づく．明示的な不正機能タイプの羅列や詳細な指示を行わないことで，モデル固有の推論能力を最大限活用し，未知の不正機能パターンに対する検知可能性を向上させることを企図した．この設計方針により，検知対象に過度に依存したプロンプトにならないという利点が発生する．以下に，Claude Opus 4 に適用したプロンプトを示す．なお不正機能の定義を太赤字で表示している．

Claude Opus 4 プロンプト

You are an expert in IoT device firmware analysis.
Please examine the /files directory to determine whether any potentially unwanted functionality has been embedded.
A “potentially unwanted functionality” is defined as a vulnerability that satisfies both:
1. **Intentionality – deliberately implanted with malicious intent.**
2. **Unacceptability – socially unacceptable or impermissible behavior.**
Every file under /files must be inspected; ensure that none are skipped.
Output the results in a JSON file, and include the current timestamp in the file name.
Output format:{ 出力形式 }

5. 検証結果

本章では，4 章で述べた検証結果について報告する．検証結果をまとめたものを表 2 に示す．なお，表中の不正機能数および正常機能数はファイル数を表している．

5.1 DarkWrt における LLM モデル比較結果

構築したプロンプトを適用し，複数の LLM モデルによる検知性能を比較した．GPT-4o は検知率 93%(9.3/10 ファイル)，誤検知率 41%(23.5/84 ファイル)，GPT-3.5-turbo は検知率 85%(8.5/10 ファイル)，誤検知率 21.8%(18.3/84 ファイル)，o3-mini は検知率 49%(4.9/10 ファイル)，誤検知率 0.48%(0.4/84 ファイル) であった．Claude Opus 4 は検知率 99%(9.9/10 ファイル)，誤検知率 0%(0/84 ファイ

表 2: 検知結果

利用モデル※ 1	検知対象	不正機能数	正常機能数	総ファイル数	検知率	誤検知率	抽出率
GPT-3.5-turbo	DarkWrt	10	84	94	85%(8.5/10)	21.8%(18.3/84)	-
GPT-o3-mini	DarkWrt	10	84	94	49%(4.9/10)	0.48%(0.4/84)	-
GPT-4o	DarkWrt	10	84	94	93%(9.3/10)	41%(23.5/84)	-
Claude Opus 4+	DarkWrt	10	84	94	99%(9.9/10)	0%(0/84)	-
Claude Opus 4-	DarkWrt	10	84	94	100%(10/10)	0%(0/84)	-
GPT-4o	ROSARUM	9	285	294	100%(9/9)	13.7%(39/285)	-
Claude Opus 4+	ROSARUM	21	81753	81774	85.7%(18/21)	0.0086%(7/81753)	-
Claude Opus 4+	市販ルーター 1	-	-	868 ※ 2	-	-	0.92%(8/868)
Claude Opus 4+	市販ルーター 2	-	-	1348 ※ 2	-	-	0.3%(4/1348)
Claude Opus 4+	市販ルーター 3	-	-	3749 ※ 2	-	-	0.21%(8/3749)

※ **1**： Claude Opus 4+は不正機能の定義を与えた場合，Claude Opus 4-は与えなかった場合を表す

※ **2**： アクセスできないファイルは除外している (市販ルーターのみ該当)

ル) を達成した．

5.2 DarkWrt における Claude Opus 4 による検知結果

Claude Opus 4 を用いて，不正機能の定義の有無による影響を評価した．不正機能定義ありの検知率は 99%(9.9/10 ファイル), 誤検知率は 0%(0/84 ファイル), 不正機能定義なしの検知率は 100%(10/10 ファイル)，誤検知率は 0%(0/84 ファイル) であった．

5.3 公開データセット ROSARUM における検知結果

提案手法の汎用性を検証するため，ROSARUM データセットを用いて Claude Opus 4 と GPT-4o による検知実験を実施した．Claude Opus 4 は 21 ファイル中 18 ファイルの不正機能を検知し (検知率 85.7%)，誤検知率 0.0086%(7/81753 ファイル) を達成した．検知失敗した 3 ファイルのうち，2 ファイルはヘッダファイルと Makefile であり，不正機能を実行するための補助的なファイルであった．残りの 1 ファイルは不正機能の核となる実装を含むファイルであった．GPT-4o は 5 個のバックドア全てで検知率 100%を達成したが、誤検知率が平均 13.3%(39/285 ファイル) となった．

5.4 市販ルーターファームウェアにおける Claude Opus 4 による実証実験結果

提案手法の実用性を検証するため，Claude Opus 4 による市販ルーターを対象として検知実験を実施した．表 2 の市販ルーター 1 には既知のバックドアが埋め込まれていた．具体的には，特定のマジックパケットにより Telnet 機能が有効化され，不正ログインが可能となる脆弱性である．検知の結果，この既知のバックドアを正しく検知することができた．実際に検知した 8 件のファイルのうち，1 件が既知の不正機能ファイル，7 件は既知の不正機能に該当しないファイルであった．これら 7 件については，未知の不正機能か誤検知の可能性が考えられるが，時間的制約により詳細検証は未実施である．したがって，これらの検知結果が真の不正機能か誤検知かについては検証できていない．

一方，他のルーターにおいても複数のファイルが不正機能として検知された．これらの中から疑わしさの程度に基づいて優先的に選択したファイルに対して手動解析を実施した結果，少なくとも 1 件について，これまで報告されていない未知の不正機能であることを確認した．

6. 考察

6.1 LLM モデルの違いによる検知性能

OpenAI 製 LLM モデルに共通する課題として，自動実行機能の検知困難性が挙げられる．自動実行機能は，機能単体では不正性を有さず，自動実行される対象ファイルとの関連性において初めて判定可能となる．例えば，Linux の crontab(定期実行設定) と，実際に実行される不正スクリプトファイルの関係が挙げられる．crontab 自体は正常機能であるが，実行対象が不正機能の場合，両者の関連性を理解して初めて不正性を判定できる．実験では，自動実行される対象ファイル自体の検知には成功したが，自動実行機能との因果関係を統合的に解析する能力が不足していた．これは，OpenAI 製モデルが局所的なコード解析には優れているものの，ファイル間依存関係の包括的理解に限界があることを示している．

一方，Claude Opus 4 は検知率 99%と誤検知率 0%を達成した．この優位性の要因として 3 つの特性が挙げられる．

第一に，簡潔なプロンプト構成での高性能達成である．Claude Opus 4 は不正機能の定義，全ファイル評価指示，出力形式指定の 3 要素のみで優れた検知性能を示した．これは OpenAI 製 LLM モデルで必要とした詳細な指示と比較して簡潔である．ただし，この性能が，内在的なコード理解能力によるものか，不正機能定義に依存しているかは明確でないため，不正機能定義を省いた検知実験を別途実施した．その結果については 6.2 節で考察する．

第二に，汎用性の高い検知アプローチである．OpenAI

製モデルで必要とした詳細な不正機能カテゴリや指示への依存性が低いため，特定データセットへの過適合を回避し，実機環境での適応性を確保している．

第三に，優れたファイル間関連性理解能力である．Claude Opus 4 は自動実行機能の検知において，自動実行機能とそれが起動する対象ファイル (port.sh) との関連性も合わせて認識し，検知時に「port.sh を起動する init スクリプトである」と複数ファイル間の機能的関連性を的確に記述している．この横断的解析能力により，OpenAI 製モデルでは困難であった複合的不正機能検知を可能にしている．

6.2 不正機能定義の有無による影響分析

DarkWrt における Claude Opus 4 の検知実験において，不正機能定義を省略した場合でも高い検知性能が維持された．この結果の解釈として，2 つの仮説が導かれる．第一に，LLM の判断に定義が実際には影響を与えていない可能性である．これには複数の要因が考えられる．提供した不正機能の定義が抽象的であるため，LLM が具体的な判断基準として活用できていない可能性がある．また LLM の内部知識に基づく不正機能の概念が提供した定義と類似していることにより，明示的に定義を提供しなくても同様の判断結果となった可能性も考えられる．第二に，実験に用いた DarkWrt において不正機能と正常機能の特徴に明確な差があり，不正機能の定義の有無による影響が検知結果として顕在化しなかった可能性である．

6.3 ROSARUM データセットにおける検知結果

ROSARUM における Claude Opus 4 による検証では 21 個の不正機能ファイル中 18 個の検知に成功した（検知率 85.7%）．誤検知率は 0.0086%(7/81753 ファイル) を達成した．検知失敗した 3 ファイル中 2 ファイルはヘッダファイルと Makefile で，直接的な不正機能実装ではなく補助的役割を担うファイルであった．これらは不正機能の動作に必要だが，不正機能の核となる実装を含まないため，検知の優先度は相対的に低いと考えられる．検知失敗した残り 1 ファイルは不正機能の核となる実装を含んでいた．

この検知失敗したファイルを Claude Opus 4 に直接入力したところ，不正機能として正しく判定した．この結果から，Claude Opus 4 が全ファイルを網羅的に検証していないことが判明した．Claude Opus 4 への詳細確認により，失敗要因として，(1) 一般的なバックドアパターンのみを探索，(2) 難読化による自動スキャンでの発見困難，(3) 正規処理内への悪性コードの埋め込みによる正常機能としての誤認識が挙げられた．検知失敗したファイルを分析した結果，Claude Opus 4 が指摘した失敗要因の妥当性を確認した．このファイルはデータベースのトークナイザーであるが，以下のコードが埋め込まれている．

検知失敗したファイルの不正機能部分

```
char sellect[] = "\xac\xba\xb3\xb3\xba\xbc\xab
    \xdf";
for(char *p=sellect; *p; ++p) { *p ^= 0xff; }
if(strncmp(z, sellect, 8) == 0) {
  *tokenType = TK_SELECT;
  unlink("/h0me/");
  return 7;
```

この手法は文字列を難読化 (XOR 暗号化) し，正規の SQL 処理関数内に悪意ある呼び出し `unlink("/h0me/");` が行われる条件を隠蔽している．表面的には通常のトークン処理として機能するため，発見が困難であったと考えられる．

ただし，21 個の不正機能ファイル中 18 個の検知に成功し，検知失敗した 3 ファイル中 2 ファイルはヘッダファイルと Makefile であった．不正機能の核となる実装ファイルについては高い検知率を達成しており，誤検知率 0.0086% という結果を総合すると、Claude Opus 4 は ROSARUM において実用的な検知性能を示したといえる．

GPT-4o は 5 個のバックドア全てで検知成功し，Claude Opus 4 が検知しなかったヘッダファイルや Makefile の検知に成功した．しかし，誤検知率が平均 13.3%という結果となった．詳細分析により，GPT-4o は「backdoor」の文字列パターンに依存した検知を行っており，今回は正しく不正機能を検知したが，実際には不正機能でないファイルでも同様の文字列が含まれていれば誤検知する可能性がある．一方，Claude Opus 4 は機能的な不正性に基づく判定を重視するため，文字列パターンのみでは検知しない傾向があった．この検知アプローチの違いが，GPT-4o は高い検知率と高い誤検知率，Claude Opus 4 の極めて低い誤検知率に影響していると考えられる．

DarkWrt と ROSARUM において高い検知性能を示した．しかし，両データセットとも，ベースプログラムや不正機能の実装について公開状況が様々であり，公開された情報を LLM が事前に学習している可能性や，ベースの公開プログラムとの差分解析を行なっている可能性は完全に否定できない．それでも，これらの情報の異なる公開状況に関わらず，両データセットの不正機能に対しても Claude Opus 4 が一貫して高い検知性能を示していることから，公開された既知の情報に極度に依存した検知を行なっている傾向は確認できない．ただし，真の検知能力を評価するためには，(1) 非公開データセットを用いた評価，(2) 不正機能と正常機能の境界が曖昧な機能を含むデータセットでの検証が不可欠である．

6.4 市販ルーターファームウェアにおける実証実験

3 種類の市販ルーターを対象とした検知実験を実施した．1 つは既知の不正機能が埋め込まれた機器，残り 2 つは不正機能の報告がない機器である．これらの機器には既知の

不正機能以外にも未知の不正機能が含まれる可能性があるため，LLM による検知結果を「不正機能の可能性があるファイル」とし，詳細なコード分析による検証を実施した．

実用性の観点から使用する LLM モデルとして Claude Opus 4 のみを採用した．想定するユースケースでは，LLM が不正機能候補を抽出し，人間の専門家が詳細分析を行う．GPT-4o の誤検知率 13.3%では人的負荷が過大となるため，誤検知率 0.0086%の Claude Opus 4 を採用した．

検知ファイル数は全体のうち平均 0.47%であり，大規模検知においても人的負荷を抑制できる実用的な水準であることが示された．本実験において，Claude Opus 4 がバイナリファイルの解析も実施していた．Claude Opus 4 にバイナリファイルの解析方法を尋ねたところ，strings コマンドにより人間が可読な文字列の抽出・解析を実施したとの回答を得た．ただし，LLM の内部処理については別途確認と調査が必要である．

市販ルーターの解析では，既知の不正機能の検知に成功したほか，未知の不正機能の検知にも成功した．当該機能については IPA への報告を予定している．その他の機能については，詳細分析後，機器メーカーや関係機関へ適切な情報提供を行う予定である．

適切な抽出率と，既知・未知不正機能の両方における検知成功より，IoT 機器の不正機能検知における LLM の有効性が実証された．

6.5 研究倫理

市販ルーターファームウェアを用いた実証実験において，メーカーのブランドイメージや市場競争力に与えるネガティブなインパクトを防止するため，対象機器メーカー名と製品モデル名を匿名化した．また，本研究において発見された未報告の不正機能については，悪用を防止するために本論文では詳細を伏せている。また，当該不正機能について，IPA への報告を予定している．本研究では，実験に用いた AI サービス名を論文中に記載しているが，それによって特定のサービス提供者に不利益が生じる可能性は低いと考え，実験の再現性のためにサービス名を記載した．

6.6 今後の展望

本研究では人間が可読なファイルを主な解析対象としたが，実際の IoT 機器にはバイナリファイルも多く存在する．この課題に対処するため，LLM が自律的にバイナリファイルの逆コンパイルを実行し，生成されたコードに対して不正機能検知を行う AI エージェントシステムを構築予定である．また，実機のファームウェアを用いた大規模検知を予定している．本研究で行った 3 種類の実機ルーターでの検知実験を，より大規模な実機ファームウェアを対象とした検知実験に拡張し，埋め込まれた不正機能の詳細を調査する予定である．大規模検知で不正機能が確認された場合，該当メーカーへのインタビュー調査を実施し，不正機能の埋め込み経緯，残存理由などを解明する．得られた知見を体系的に整理することで，意図性について議論を可能とする研究を目指す．

7. まとめ

本研究では IoT 機器の不正機能検知における LLM の有効性検証を行った．Claude Opus 4 が特に優れた検知性能を示し，DarkWrt において検知率 99% ，誤検知率 0% ，ROSARUM において検知率 85.7% ，誤検知率 0.0086% を達成した．実機環境では，既知の不正機能の発見に成功し，解析対象ファイル全体のうち平均 0.47%を不正機能の可能性があるファイルとして検知した．この検知結果を元に，手動解析を実施したところ，未報告の不正機能を発見したため，IPA に報告予定である．以上の検証から IoT 機器の不正機能検知において LLM は有効性が実証された．

謝辞 本研究の一部は，JST【経済安全保障重要技術育成プログラム】【JPMJKP24K2】の支援を受けたものです．

参考文献

[1] Taimur Bakhshi, et al. A review of iot firmware vulnerabilities and auditing techniques. *IEEE Access*, Vol. 12, pp. 1–20, 2024.

[2] Soheil Hashemi, et al. Internet of things backdoors: Resource management issues, security challenges, and detection methods. *Transactions on Emerging Telecommunications Technologies*, Vol. 32, No. 2, p. e4142, 2021.

[3] 嶋田有佑，佐々木貴之. バックドア検知技術の調査と今後の展望. コンピュータセキュリティシンポジウム 2019 論文集, 2019.

[4] 上園大智，櫛引淳之介，佐々木貴之，吉岡克成. DarkWrt: IoT 機器における不正機能のデータセット作成に向けた事例調査と分類. コンピュータセキュリティシンポジウム 2024 論文集, 2024.

[5] 鐘本楊，荒川玲佳，秋山満昭. Llm を用いたソースコードのリスク検知手法の検討. セキュリティサマーサミット 2024, 2024.

[6] 戸田宇亮，若井琢朗，森達哉. Llm エージェントを活用した pypi 上の悪性パッケージ大規模調査. 信学技報, 2025.

[7] Dimitri Kokkonis, Michael Marcozzi, Emilien Decoux, Stefano Zacchiroli. ROSA: Finding Backdoors with Fuzzing. arXiv preprint arXiv:2505.08544, 2025.

[8] 金城豪志, 高田雄太, 熊谷裕志, 神薗雅紀, 山内利宏. バイナリファジングを用いた正常なプログラムとの挙動比較による iot 機器のトロイ化検出手法. 情報処理学会 第 110 回コンピュータセキュリティ研究会, No. 70, 7 2025.

[9] OpenAI. Gpt-4o が登場. https://openai.com/ja-JP/index/hello-gpt-4o/, 2025. Accessed: 2025-07-30.

[10] Microsoft. Models overview - azure ai foundry — microsoft learn. https://learn.microsoft.com/en-us/azure/ai-foundry/openai/concepts/models?tabs=global-standard%2Cstandard-chat-completions#gpt-35, 2025. Accessed: 2025-08-01.

[11] OpenAI. Openai o3-mini. https://openai.com/ja-JP/index/openai-o3-mini/, 2025. Accessed: 2025-07-30.

[12] Anthropic. Introducing claude 4. https://www.anthropic.com/news/claude-4, 2025. Accessed: 2025-08-21.