Isolation Forestを用いた IoT向け異常検知手法に関する考察

菅田 大輔[1,a)]　石井 将大[1,2,†1]　松浦 知史[2]

概要：本論文では、Isolation Forest を用いた IoT 向け異常検知手法の改善を行なった。この研究の背景に IoT（Internet of Things）の普及がある。2030 年には約 300 億台ものデバイスが利用されると予測されており、すべての IoT 機器のセキュリティ確保が重要な課題となっている。特に 2016 年の Mirai 型マルウェアによる DDoS 攻撃はその必要性を浮き彫りにした。本研究は専門的なセキュリティ対策が難しい、家庭内ネットワークなどの小規模環境に着目し、軽量で高速な異常検知システムの提案を目指した。Isolation Forest は軽量で高速に動作する異常検知手法として注目されているが、これを用いた従来の方法には二つの問題が存在する。閾値を手動で設定する必要がある点と、閾値による異常判定のアルゴリズムの精度に限界がある点である。これらの課題を解決するため、異常判定をする際にロジスティック回帰を応用した判定アルゴリズムを提案する。提案手法を二つのデータセットで実験した結果、精度がそれぞれ 84.4%から 91.1%、84.6%から 94.5%に改善したことを確認した。

キーワード：Isolation Forest， IoT， IDS， 異常検知

A Study on Anomaly Detection Method for IoT using Isolation Forest

Daisuke Sugata[1,a)]　Masahiro Ishii[1,2,†1]　Satoshi Matsuura[2]

Abstract: This paper presents an improved anomaly detection method for IoT environments using Isolation Forest. The motivation for this study lies in the widespread adoption of the Internet of Things (IoT). By 2030, it is estimated that around 30 billion devices will be in use, making the security of all IoT devices a critical issue. The necessity of this was highlighted by the 2016 DDoS attack caused by Mirai malware. This study focuses on small-scale environments, such as home networks, where implementing specialized security measures is challenging, and aims to propose a lightweight and fast anomaly detection system. Although Isolation Forest is recognized for being a lightweight and fast anomaly detection method, traditional approaches using it have two main issues: the need for manual threshold setting and limitations in the accuracy of anomaly detection algorithms based on these thresholds. To address these challenges, we propose an anomaly detection algorithm that incorporates logistic regression for decision-making. Experiments on two datasets showed that the accuracy of the proposed method improved from 84.4% to 91.1% and from 84.6% to 94.5%, respectively. abstract

Keywords: Isolation Forest, IoT, IDS, Anomaly Detection

1. はじめに

[1] 東京工業大学 情報理工学院 数理・計算科学系
Department of Mathematical and Computing Sciences, School of Computing, Tokyo Institute of Technology
[2] 株式会社 YY セキュリティ研究所
Security Laboratories, YY Corporation
[†1] 現在，国立研究開発法人 ZZ 研究所
Presently with National Institute of ZZ
[a)] sugata.d.aa@m.titech.ac.jp

2. 研究方法

2.1 Isolation Forest の説明

Isolation Forest（以下，iForest）は，外れ値検出のためのアルゴリズムである。iForest は異常データが少数であり、離れているという前提に基づいている。ランダムにデータを分割していくと、異常データは相対的に早く分離される。iForest は以下のステップで実行される。

1. データの分割

ランダムに選んだ特徴量から、ランダムに選んだ値をもとにデータを分割する。これを一定回数繰り返し、複数のツリーを作成する。

2. 異常スコアの算出

データがツリーの枝に到達するまでの平均パスをもとに、異常スコアを算出する。具体的な計算式は以下の通りである。

ここで、$E(h(x))$ はデータ点 x の平均パス長、$c(n)$ はデータセットのサイズ n に依存する定数である。

3. 異常判定

先ほど計算した異常スコアをもとに異常検知を行う。通常、トレーニングデータの異常スコアの上位 10%を閾値として設定し、それを超えたデータを異常と判定する。

2.2 IDS の概要

本研究では、小規模な IoT 環境に適した Intrusion Detection System（IDS）の設計について検討する。具体的には、Isolation Forest を用いた異常検知手法が提案され、その有効性を評価する。

2.3 全体の設計

IDS の設計は以下の 3 つのセクションに分けられる：

(1) **データの前処理**：
- 入力データを適切な形式に変換する。
- 不必要な特徴量の削除やデータの標準化、ラベルエンコーディングを行う。

(2) **特徴量選択**：
- 判定に重要な特徴量を選択し、過学習を防ぎ、検知精度を向上させる。
- Random Forest を用いて特徴量の重要度を算出し、重要な特徴量を選択する。

(3) **攻撃の判定**：
- iforest を用いて通信が攻撃通信であるかを判定する。
- 特徴量の選択後、Isolation Forest で異常検知を行う。

2.4 実装

実装は以下のステップで行う：

(1) **データの前処理**：

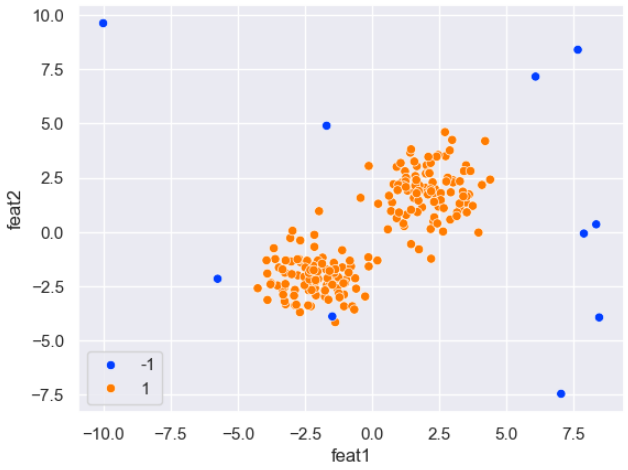

図 1 デモデータに関する図の説明（和文）
Fig. 1 Description of the dim_vs_accu figure (English).

- はじめに、不必要な特徴量の削除を行う。（データの標準化を行う必要なし？）Iforest に入力できるのは数値データだけなので、カテゴリカルデータを数値データに変換する。one-hot エンコーディングを用いてラベルのエンコーディングを行う。

(2) **特徴量選択**：
- Random Forest を用いて特徴量の重要度を算出する。その後、重要度をもとに上位 1 割の特徴量を使用する。

(3) **攻撃の判定**：
- 攻撃通信、正常通信のそれぞれでトレーニングされたサブシステムが、Isolation Forest によって異常検知を行う。
- それぞれのサブシステムの結果を 2 通りで組み合わせ、最終的な判定を行う。

3. iForest の問題点の整理

Isolation Forest を異常検知手法として使用する際にどのような問題があるのかを明らかにするための事前実験を行った．はじめにデモデータを使用して，iForest の挙動を確認した．その考察をもとに特徴量選択手法の提案を行なった

使用したデモデータは、(2.5, 2.5) と (-2.5, -2.5) を中心とした正常データ群と、10 から-10 の範囲に一様に分布した異常データ群からなる．二次元の場合のデモデータを図 1 に示す．

はじめに、デモデータと特徴量数の関係を調査した．図 2 に示すように、特徴量数が増えるにつれて、異常検知の精度が単調に向上することがわかった．また、精度ののびは増加に反比例して緩やかになっていることもわかる。ゆえに、iForest は目標とする精度に対して十分な特徴量数が存在すると言える。

続いて、デモデータにノイズ特徴量を含めたときの精度

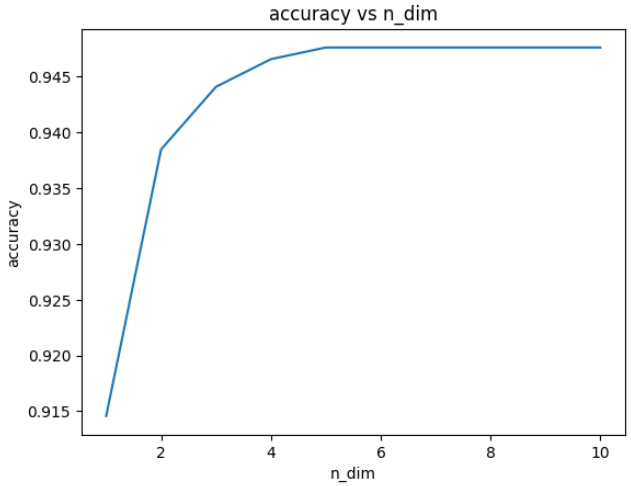

図 2　dim_vs_accu に関する図の説明（和文）
Fig. 2　Description of the dim_vs_accu figure (English).

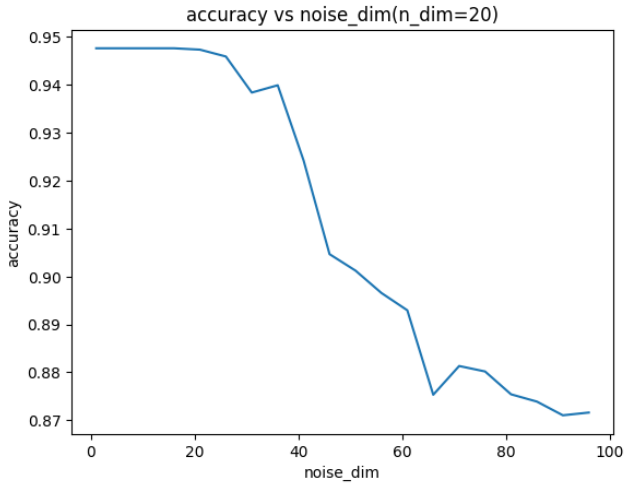

図 3　noise_accu に関する図の説明（和文）
Fig. 3　Description of the noise_accu figure (English).

の悪化について調査した。パケット通信を監視して得られたデータセットの全ての特徴が、異常検知に有効なわけではない。そして、iForest は特徴量同士の重みづけを行わないため、判定に有効でない特徴量が混ざると精度が低下すると考えられる。図 3 に示すように、ノイズとなる特徴量が混ざると精度が低下することがわかった。また、今回の実験の場合だと、ノイズ特徴量が判定に有効な特徴量数の 2 倍以上になると、精度が急激に低下することがわかった。

前の実験から、データセットからノイズとなる特徴量を取り除くことが重要であると考えられる。ところで、iForest はツリーベースの異常検知手法である。そこで、同じくツリーベースの Randam Forest から特徴の重要度を算出すれば、ノイズとなる特徴量を取り除けるのではないかと考えた。図 4 は、Random Forest で算出した特徴量の重要度を表している。このグラフは、ノイズ特徴量を判別できていることがわかる。そして、実際にノイズ特徴量を取り除いた場合の精度を調査したところ、精度は???%から 0.945%まで向上した。この結果から、Feature Importance による特徴量選択手法は精度の向上に有効ではないかと考えた。

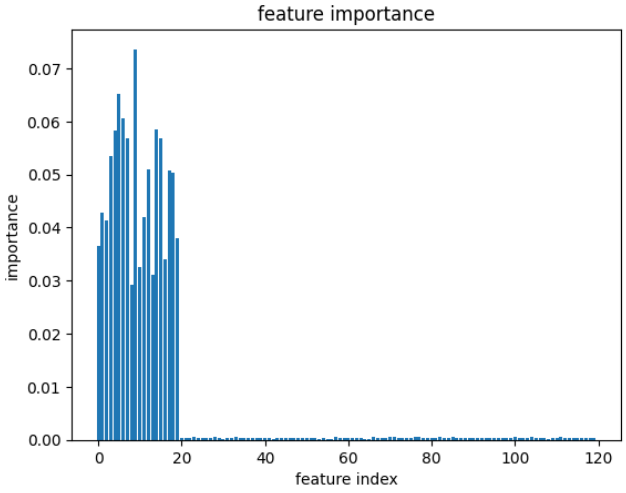

図 4　select_noise に関する図の説明（和文）
Fig. 4　Description of the select_noise figure (English).

表 1　組み合わせアルゴリズム（和文）
Table 1　combination algorithm (English).

Normal subsystem	Attack subsystem	combination result
Normal	Anomaly	Normal
Anomaly	Normal	Anomaly
Normal	Normal	Anomaly
Anomaly	Anomaly	unknown

3.1　より効果的な判定の組み合わせ方について

Iforest を用いた異常検知手法では、攻撃データと正常データのそれぞれでトレーニングされたサブシステムが、Isolation Forest によって異常検知を行う。最終的な判定は、これらの結果を組み合わせて行う。AbuAlgaham らの研究では、表 1 のようにそれぞれの判定器の結果を組み合わせて最終的な判定を行う。

しかし、表 1 のようなマッピングによる判定手法は以下の 2 つの問題を抱えている。

- それぞれの判定器の異常スコアの閾値を手動で設定する必要があること
- 異常判定の境界線が直線であるため、異常スコアの分布によっては誤判定が多くなること

実際に異常スコアの分布を調査したところ、図 5 と図 6 のように、異常スコアの分布が斜めになっていることがわかる。このような場合、直線的な分割では正確な判定ができない。

そこで、本研究では、異常スコアの分布によって適切な判定ができるように、異常検知の境界線を、ロジスティック回帰を用いて決定する手法を提案する。この手法では、

それぞれのサブシステムの異常スコアを入力とし、ロジスティック回帰によって異常判定を行う。この手法を用いることで、1.iForest の閾値を設定する必要がない 2. 閾値による直線的な分割よりも正確な判定が行える という利点がある。

4. 結果と考察

4.1 比較するアルゴリズム

本研究では、以下の 2 つの比較を行った。特徴量エンジニアリングの効果の比較を行うため、特徴量選択を行わなかった場合と、Random Forest を用いた特徴量選択手法を用いた場合の結果を比較する。また、判定の組み合わせアルゴリズムの比較には、2 つのサブシステムの結果をマッピングする手法と、Rogistic Regression を用いて判定を行う手法の結果を比較した。

(1) **特徴量エンジニアリングの比較**：
- 特徴量選択なし
- Random Forest を用いた特徴量選択手法

(2) **判定の組み合わせアルゴリズムの比較**：
- マッピングして判定
- Rogistic Regression を用いて判定

4.1.1 実験環境

実験は Mac Book Pro 2017 2.3GHz Intel Core i5, 8GB RAM で行った。また、実験に用いたプログラムは Python3.10.4 で実装した。Isolation Forest や Random Forest の実装には、scikit-learn のライブラリを用いた。

4.1.2 データセット

本実験では、以下の 2 つのデータセットを用いた。

(1) **NSL-KDD**：KDDCUP99 の問題点を解決するために提案されたデータセットであり、データの冗長性や攻撃データの割合を調整したもの。

(2) **UNSW-NB15**：既存のデータセットの問題点を解決し、現代のネットワークトラフィックと低フットプリント攻撃を包括的に反映するために作成されたデータセット。

4.1.3 評価指標

本研究では、以下の 4 つの評価指標を用いてモデルの性能を評価した。

(1) **Accuracy**：正確度を示し、予測が実際のクラスと一致する割合を示す。

(2) **Precision**：予測が正常と判定されたデータのうち、実際に正常であるデータの割合を示す。

(3) **Recall**：実際に正常であるデータのうち、正常と判定されたデータの割合を示す。

(4) **F1-score**：Precision と Recall の調和平均を示す。

4.2 結果

はじめに、2 つのデータセットに対してトレーニングデータの異常スコアの分布を調査した。横軸は正常データで訓練された iForest が算出した異常スコアを示し、縦軸は攻撃データに対して iForest が計算した異常スコアを表している。それぞれのゼロ点は異常スコアの閾値を示し、負の方向に大きくなるほど異常スコアが高く、正の方向に大きくなるほど異常スコアが低いことを意味する。各データ点の色はラベルを示しており、赤が攻撃データ（1）、青が正常データ（0）を表している。図 5 はデータセット UNSW における異常スコアの分布を示しており、図 6 はデータセット NSL における異常スコアの分布を示している。

図 5 UNSW1（和文）
Fig. 5 Description of the collectness_UNSW figure (English).

表 2 UNSW でのモデルの性能評価結果（和文）
Table 2 Performance evaluation results of models (English).

model	accuracy	precision	recall
特徴量選択なし	0.7506	0.7949	0.7506
特徴量選択なし、ロジスティック回帰で判定	0.8685	0.8666	0.8685
特徴量選択あり	0.8440	0.9321	0.8440
特徴量選択あり、ロジスティック回帰で判定	0.9112	0.9146	0.9112

UNSW データセットでは、データが対角線方向に分布しており、2 つのサブシステムによる判定が相反するデータが多いことがわかる。また、右下に正常データが多く存在し、この領域の判定は適切に行われている。攻撃データと正常データの境界線が対角方向に位置しているため、従来の直角的な分割に比べて、ロジスティック回帰を用いて斜めに分割することで精度が向上する可能性がある。

図 6 に示されているデータセット NSL の異常スコア分布では、右下と左下にそれぞれ攻撃データと正常データが集中しており、これも判定が適切に行われていることを示している。同様に、2 つのデータの境界線が対角線方向に位置しているため、ロジスティック回帰を用いることで精度が向上する可能性がある。

それぞれの実験結果は表 2, 表 3 の通りである。表 2 を見ると、特徴量の選択をすることで、精度が 0.7506 から 0.8440 に向上したことがわかる。また、ロジスティック回帰を用いて判定を行うことで、精度が 0.8440 から 0.9112

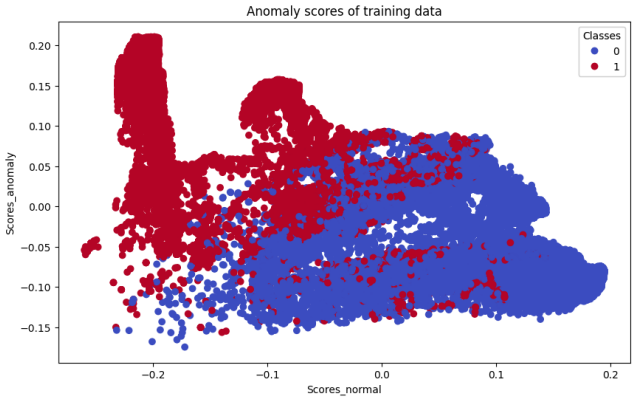

図 6　NSL1（和文）

Fig. 6　Description of the collectness_UNSW figure (English).

表 3　NSL でのモデルの性能評価結果（和文）

Table 3　Performance evaluation results of models (English).

model	accuracy	precision	recall	f1
特徴量選択なし	0.7873	0.8749	0.7873	0.8171
特徴量選択なし、ロジスティック回帰で判定	-	-	-	-
特徴量選択あり	0.8466	0.9174	0.8466	0.8775
ロジスティック回帰で判定	0.9452	0.9453	0.9452	0.9452

に向上したことがわかる。同様に、表 3 から、データセット NSL でも、特徴量の選択をすることで精度が 0.7873 から 0.8466 に向上し、ロジスティック回帰を用いて判定を行うことで精度が 0.8466 から 0.9452 に向上したことが確認できた。

4.3　考察

(うまくいった理由) 今回の実験から、ロジスティック回帰を応用した異常判定手法により、精度が向上することがわかった。これは、今回使用した 2 つのデータセットにおいて、異常スコアの分布の境界線が斜めでうまく分割できたから。

適用範囲境界線が曲面であったり、データセットが混ざってしまっている場合、うまくいかない

トレードオフモデル構築のコストの比較・ロジスティック回帰のコスト vsiforest の探索（）の計算量（時間）を比較する

ロジスティック回帰におけるランダムサンプルしての精度の比較（調べてもいい）サンプリングの適用範囲（データ量と精度のトレードオフ）

5.　おわりに

おわりにを書く.

謝辞　謝辞を書く.

参考文献