Computer Security Symposium 2019
21 - 24 October 2019

# ニューラルネットワークによる予測モデルを用いた制御システムネットワーク異常検知

三谷 昌平[1,a)]　吉永 直生[1]　山野 悟[2]

**概要**：制御システムの高度化が進展する一方で，不正制御による攻撃の脅威が拡大している．特に，正規の機器から正規のコマンドで行われる高度な不正制御をリアルタイムに検知することは困難である．このような攻撃に対し，制御システムを流れるコマンドとシステム状態との整合性を監視する対策が考えられるが，システム状態の中には運用で決まるものもあるため，それらのシステム状態を列挙することは容易ではない．本稿では，システム状態を明示的に定義することなく，事前に学習した予測モデルを用いてシステム状態に応じたシンプルなホワイトリストを逐次生成し，リアルタイムにパケット単位で異常を検知する手法を提案する．また，実システムのトラフィックデータを用いた評価結果を報告する．

**キーワード**：産業制御システム，ネットワーク異常検知，ホワイトリスト，ニューラルネットワーク，予測モデル

# Network Anomaly Detection in Industrial Control Systems based on Prediction Model with Deep Neural Network

Shohei Mitani[1,a)]　Naoki Yoshinaga[1]　Satoru Yamano[2]

***Abstract:*** Threats to Industrial Control Systems (ICS) are growing larger in terms of malicious control attacks, in contrast to the advancement of those systems. Especially, it's still difficult to detect the intelligent approach for malicious control, which is executed from an ordinary equipment with legitimate commands, especially in real-time. One approach to this threat is to check the consistency between commands and system state of ICS. However, some state are determined in operational phase. It is problematic to list these system state. We propose a network anomaly detection method. In our method, simple whitelists are sequentially generated by the pre-learnt prediction model corresponding to the continuous change of system state in ICS, then detect the malicious commands by checking each packets. It doesn't require us to define any system state explicitly. We report the result of an experiment using real system's traffic data.

***Keywords:*** ICS, Network Anomaly Detection, Whitelist, Neural Network, Prediction Model

## 1. はじめに

産業制御システム（Industrial Control System, ICS）は，電力・ガス・水道といった重要インフラや，製造プラント，ビル管理など幅広い分野で利用されている．

ICS の基本的な構成を，**図 1** に示す．ICS は，フィールド機器，制御コントローラ，中央監視機器，そしてそれらを接続する制御ネットワークから成るものを想定する．

ICS では，低コスト化・運用の効率化・高速化を目的として，汎用製品の導入やイーサネットを利用したネットワークプロトコルの標準化が進展している．また，インターネットへの接続が可能なインターフェースの導入も増加している．これらに伴い，従来はサイバー攻撃の対象になりにくいと考えられていた ICS においても，情報システ

[1] NEC セキュリティ研究所
Security Research Laboratories, NEC
[2] NEC 研究企画本部
Research Planning Division, NEC
[a)] s-mitani@az.jp.nec.com

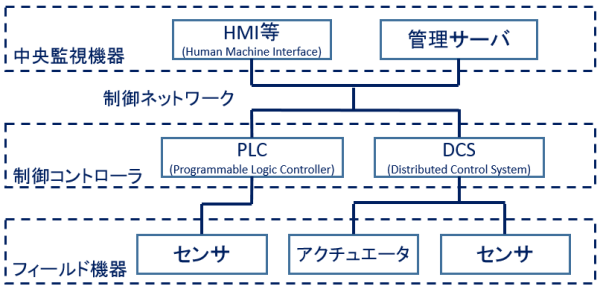


**図 1** ICS の基本的な構成
**Fig. 1** A typical structure of ICS.

ムと同様にサイバー攻撃の脅威が増大している [1].

ICS に対して攻撃が行われた場合，その被害は情報漏洩に留まらず，デバイスの設定値やプログラムの改竄による大規模な物理的損害が生じ得る [2]．ICS に対して大規模な損害を与えた攻撃として，2010 年にイランの核施設に侵入し，多数の遠心分離器に異常動作を引き起こして停止させた Stuxnet が知られる．Stuxnet は特定のシステムを標的として開発された高度なマルウェアによる標的型攻撃であり，複数の公開前脆弱性を利用した Zero-Day 攻撃であったことが明らかになっている．Stuxnet は監視システムに対してマルウェア感染の事実や遠心分離器の異常動作を隠蔽していた [3]．また，2016 年にはウクライナの送電網に対して Industroyer（または CrashOverride）と呼ばれる攻撃が行われ，周辺地域に停電を引き起こした．Industroyer のような攻撃が特に重大であるのは，バックドアを介して攻撃者が不正な制御コマンドを送信し，遮断器を直接開閉できたと考えられるためである [4].

ICS に対する高度な攻撃を検知することは，情報システムと同様に困難である．例えば中間者攻撃（Man-in-the-Middle，MitM）の場合，攻撃者は正規の機器になりすまし，正規のネットワークトラフィックを模倣した通信を行うと考えられる．さらに，HMI などの中央監視機器が一度マルウェアに感染するか，攻撃者による不正ログインに晒された場合，不正な制御コマンドが実際に正規の機器から送信されることになる [5].

サイバー攻撃の脅威が増大する一方で，ICS にはその運用上の特性から，情報システムと同様のセキュリティ技術をそのまま適用することは困難であるとされる．特に重要な要件は，可用性の維持である．ICS では，安全な状態を維持するために運転を継続する必要があり，異常時に動作を停止させることや通信の遮断や遅延により可用性を損なうことは好ましくないほか，即時の対応が求められる．そのため，セキュリティ技術にもリアルタイム性が要求される [1]．また，運用中の ICS を構成する機器の入れ替えやアップグレードは可用性を損なう恐れがあることから，古い機器が使われ続ける傾向にある．そのため，暗号化や認証といったセキュリティ対策を徹底することが難しく，既知の脆弱性が残ったソフトウェアが使われ続けることもある [6].

また，情報システムとは異なる ICS の特徴として，一般に ICS には複数のシステム状態が存在し，システム状態に応じて異なる制御が行われることが挙げられる．システム状態に整合しない制御が行われた場合，ICS が致命的な状態に陥る可能性がある [2]．また，ICS の制御ネットワークでは，制御や監視のために，プロセス値が常に流れていることも特徴である．攻撃者はプロセス値を改竄することで，不正制御を実行することや，監視員に誤った情報を伝えることができる．これらのことから，高度な不正制御の脅威を想定すると，ICS の特徴であるシステム状態も考慮した対処を行う必要があり，その手法が提案されている [7].

ICS をサイバー攻撃から防御するために，IDS（Intrusion Detection System）による攻撃検知を用いることができる．IDS は情報システムでも利用されており，一例として，適用箇所・検知モデル・監視対象という 3 つの観点から，次のように分類できる [8].

- **適用箇所 :Host-based と Network-based**

Host-based IDS は，PC や PLC などのホスト上で攻撃を検知する．Network-based IDS はネットワーク上で攻撃を検知する．

- **検知モデル :Signature-based と Behavior-based**

Signature-based 手法では，既知の脅威情報を入力して攻撃パターンとのマッチングを検知する．誤検知が少ないが，未知の攻撃は検知できない.

Behavior-based 手法では，システムの正常な振る舞いを定義し，そこからの逸脱を異常として検知する.

Behavior-based 手法は，さらに Specification-based 手法と Anomaly-based 手法に分けられる.

Specification-based 手法では，システムの設計情報を利用する．未知の攻撃を検知でき，誤検知も抑制できるが，設計情報から検知ルールを構築するコストが大きい．また，検知可能な攻撃の範囲は設計情報の内容に依存する.

Anomaly-based 手法では，正常時の振る舞いのパターンを入力し，そこからの乖離を検知する．未知の攻撃を検知できるが，誤検知が多くなる.

- **監視対象 :Packet-Based と Flow-based**

Packet-based 手法では，パケットのデータ部の内容まで監視する DPI（DeepPacket Inspection）が用いられる [9]．ヘッダ部は正常だがデータ部だけに異常が生じるような攻撃も検知できるので，例えば制御コマンド内の設定値を監視できる.

Flow-based 手法は，パケットヘッダ部から送受信関係やパケットフローの統計量の異常を監視する．監視パターンと計算量を抑制できるが，データ部の異常は検知できない．Flow-based 手法は，トラフィックの統計量を監視する Statistical-based の分析手法において良く用いられる.

Statistical-based 手法では，特徴量の設計が問題であり，またモデルの構築や検知にあたり，システムの状態がほぼ静的である必要がある．

我々は，可用性の観点から，ネットワーク上へ接続するだけで各ホストの動作に影響を与えることなく攻撃検知が可能な Network-based IDS へ適用可能な技術を検討した．

本稿では，未知の高度な不正制御をリアルタイムに検知するために，システム状態を考慮した Behavior-based, Packet-based(DPI) によるネットワーク異常検知手法を提案する．特に，未知の状態に応じて最適にホワイトリストを生成する手法について説明する．また，実システムのトラフィックデータを用いて，正規の機器から正規のコマンドで行われる不正制御を検知可能であることを示す．

## 2. 関連研究

### 2.1 システム状態を考慮した ICS のネットワーク異常検知

ICS において，システム状態を考慮して不正制御を検知する技術として，Hata ら [7] はシステムの運転状態に応じたモデルベースホワイトリストによる異常検知手法を提案している．モデルベースホワイトリストによる異常検知はネットワークスイッチで行われ，立ち上げ段階や制御段階といった運転状態に整合しない不正制御コマンドや異常プロセスを検知する．具体的には，システムを線形状態方程式でモデル化し，状態推定値を逐次的に計算するとともに，実際に観測された状態との誤差のリアプノフ安定性を評価し異常を検知する．

Kobayashi ら [10] も，システム状態とネットワークを統合したホワイトリストを用いる手法を提案している．これらはいずれも，Specification-based 手法に含まれる．

Hata らはまた，文献 [7] の手法を，設計情報の中に含まれない運用上のシステム状態に応じたホワイトリストへ拡張するために，システム状態を定常状態と過渡状態へ分離する手法を提案している [11]．この手法では，プロセス値の急峻な変化を検出し，状態を分離している．

しかしながら，安定状態と過渡状態の中で異なる制御のパターンを持つシステム状態がさらに複数存在する場合や，システム状態が明確な境界を持たず緩やかに遷移する場合には，それらの状態に応じたホワイトリストを用意することが困難であり，攻撃の見逃しが発生し得ると考えられる．

このような未知の状態パターンをデータから抽出するには，機械学習手法が適する．

### 2.2 機械学習による ICS の異常検知

本節では，教師ありネットワーク異常検知，Anomaly-based ネットワーク異常検知，および ICS における異常検知の関連研究を示す．

ICS に限らず，教師ありのネットワーク異常検知の一例として，VAE(Variational Auto-Encoder) を分類器として利用し，攻撃の種類または正常ラベルを出力する手法が提案されている [12]．また，半教師無し学習手法では，ラベル無しのトラフィックデータで学習した AE(Auto-encoder) を特徴抽出器として用いる手法がある [13]．

また，Packet-based 手法では，パケットデータ列から n-Gram や BoW 等の NLP(Natural Language Processing) 手法を用いて特徴ベクトルに変換することが多い [14]．これにより一般に過学習を抑制する．

Anomaly-based 手法では，例えば OCSVM(One-class Support Vector Machine) の利用が提案されている [15]．また, AE の階層化により誤検知を抑制する提案もある [16]．

上記のような手法は，パケットの特徴やトラフィック統計量の特徴を抽出して異常を検知することができるものの，ICS の特徴であるシステム状態等は表現されていない．

一方で，ICS 自体の異常を検知するために, LSTM(Long-Short Term Memory)[17][18] によって学習した予測モデルにより異常を検知する手法が提案されている [19]．また，Kiss ら [20] は，プロセス値の組をクラスタリングし，各クラスタからの逸脱によって異常を検知する手法を提案している．この手法では，上手くクラスタリングを行えば複数の運転状態を分離することができることを想定している．

### 2.3 課題

正規の端末から正規のコマンドで行われる高度な不正制御を検知するために，設計情報に含まれるとは限らないシステムの状態に応じて，制御トラフィックをリアルタイムに監視する手法の確立が課題である．

Kiss らの発想のように，運転状態をプロセス値のクラスタリングによって分離し，各クラスタに応じて Hata らの手法のように異なるホワイトリストを定義することによって，設計情報に含まれない状態にも応じたネットワーク異常検知が可能であると考えられる．しかし，このナイーブな方法には以下の問題が考えられる．

- ネットワークトラフィックとは無関係にシステム状態の特徴を抽出しても，その特徴がネットワークトラフィックの特徴を捉えているとは限らない．すなわち，ネットワークトラフィックが大きく異なればシステム状態も大きく異なるという前提条件が必要である．

そこで, 我々はプロセス値とトラフィックデータをニューラルネットワークによって学習することで，上記の問題を回避しつつ未知の状態に応じたネットワーク異常検知を行う手法を検討した．

## 3. 提案手法

提案手法の目的は，システム状態を考慮してネットワー

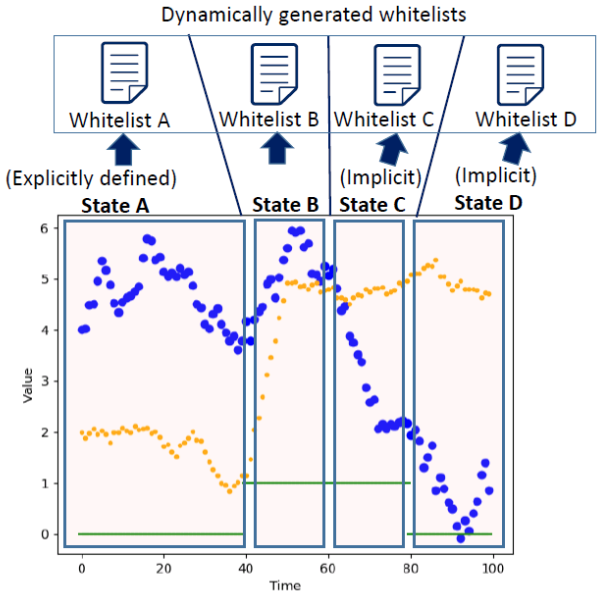


**図 2**　非明示的な状態に応じたホワイトリスト
**Fig. 2**　Updating whitelists based on implicit states.

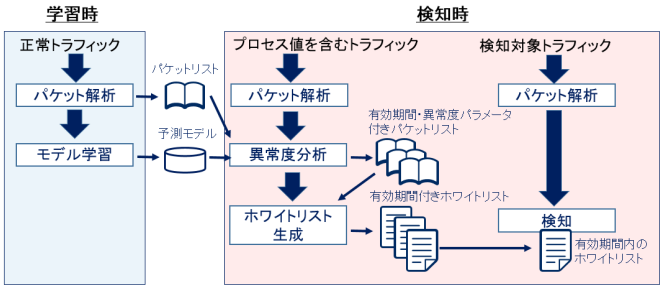


**図 3**　提案手法によるネットワーク異常検知の流れ
**Fig. 3**　Network anomaly detection with our method.

クパケットを監視し，不正制御による異常なタイミングでの制御コマンド発生や，異常な制御値を検知することである．提案手法では，あらかじめシステム状態を陽に定義するのではなく，プロセス値の集合から正常なネットワークトラフィックの発生パターンを直接推定する予測モデルを正常データから事前に構築する．予測モデルが最終的に生成するのは，送信元や送信先，データ部のフィールド値等を列挙したシンプルなホワイトリストであり，解釈や検知動作が容易である．

**図 2** に，トラフィックを高精度に監視するための状態に応じたホワイトリスト運用のイメージを示す．図中に 3 種類のプロセス値の変化を示している．当然ながら，明示的に定義された状態 A から状態 B への遷移に応じて，異なる内容のホワイトリスト A と B を利用するべきである．それだけでなく，状態 B から一見明確な境界を持たない未知の状態 C や運用上生じた未知の状態 D への遷移でトラフィックパターンが変化するならば，新たなホワイトリスト C,D を利用する．このようにして，必ずしも明示的に定義されているとは限らない状態に対応して，無数のパターンの最適なホワイトリストを生成し，高精度なトラフィック監視・攻撃検知を行う．

このようなネットワーク異常検知を実現するため，提案手法ではニューラルネットワークによる予測モデルを正常データから事前に学習しておく．そして，検知時には明示的でないシステム状態も対応したシンプルなホワイトリストを生成する．

本手法は，学習時の 2 段階の処理と，検知運用時（検知時）の 4 つの処理から成る．

- 学習時
  (1) パケット解析
  (2) モデル学習
- 検知時
  (1) パケット解析
  (2) 異常度分析
  (3) ホワイトリスト生成
  (4) 検知

本手法全体の流れを**図 3** に示す．左側に学習時の流れを，右側に検知時（検知運用時）の流れを示す．

学習時には，正常データを用いて予測モデルの学習を行う．

検知時には，予測モデルを用いて異常度分析を行い，複数の有効期間付きホワイトリストを生成する．状態を明示的に定義しない，連続的な状態に応じたホワイトリストである．最後に，検知機器が有効期間内のホワイトリストを参照して異常を検知する．

以下，検知時・学習時共通のパケット解析，学習時のモデル学習，検知時の異常度分析，ホワイトリスト生成，検知について順に説明する．

### 3.1　パケット解析

受信したパケットを解析し，パケット種別とプロセス値とを抽出する．パケット種別は，送信元や送信先，命令種別など，ヘッダからデータ部のフィールドまでを含む監視対象の値の組み合わせである．プロセス値は，連続変数を含むフィールドの値である．

### 3.2　モデル学習

ニューラルネットワークを用いて，過去のプロセス値の履歴から，未来の一定期間における各種別のパケットの発生確率とプロセス値，統計量等の確率分布を予測するモデルを学習する．出力として確率分布パラメータが得られるので，それから “未来の時刻に特定のプロセス値を持ったパケットが発生する” という事象の異常度を計算できる．

#### 3.2.1　LSTM

LSTM は再帰構造を持つニューラルネットワークであ

り，時刻ラベル $t$ で再帰のステップ数を表す．逐次的に与えられた入力ベクトル $\cdots, \mathbf{x}^{t-1}, \mathbf{x}^{t}, \mathbf{x}^{t+1}, \cdots$ に対して，逐次的に出力 $\cdots, \mathbf{y}^{t-1}, \mathbf{y}^{t}, \mathbf{y}^{t+1}, \cdots$ を計算することができる．

次のような時間ラベル $t$ を含む演算 $(\mathbf{h}^{t-1,(l)}, \mathbf{h}^{t,(l-1)}) \to \mathbf{h}^{t,(l)}$ を LSTM 層とする．

$$\begin{aligned}\mathbf{c}^{t,(l)} &= \mathbf{i}^{t,(l)} \circ f(W_{c,in}^{(l)}\mathbf{h}^{t,(l-1)} + W_{c,out}^{(l)}\mathbf{h}^{t-1,(l)} + \mathbf{b}^{(l)}_{c}) \\ &\quad + \mathbf{f}^{t,(l)} \circ \mathbf{c}^{t-1,(l)} \\ \mathbf{h}^{t,(l)} &= \mathbf{o}^{t,(l)} \circ f(\mathbf{c}^{t,(l)}) \end{aligned} \tag{1}$$

$\mathbf{c}^{t}$ は Memory Cell と呼ばれ，過去の履歴に関する情報を保持するベクトルである．非線形写像 $f(\cdot)$ としては tanh がしばしば用いられる．“$\circ$” はベクトル要素毎の積である．また，$\mathbf{i}^{t,(l)}, \mathbf{f}^{t,(l)}, \mathbf{o}^{t,(l)}$ はそれぞれ Input Gate，Forget Gate，Output Gate と呼ばれ，次のように計算する．

$$\begin{aligned}\mathbf{i}^{t,(l)} &= \sigma(W_{i,in}^{(l)}\mathbf{h}^{t,(l-1)} + W_{i,out}^{(l)}\mathbf{h}^{t-1,(l)} + \mathbf{b}_{i}^{(l)}) \\ \mathbf{f}^{t,(l)} &= \sigma(W_{f,in}^{(l)}\mathbf{h}^{t,(l-1)} + W_{f,out}^{(l)}\mathbf{h}^{t-1,(l)} + \mathbf{b}_{f}^{(l)}) \\ \mathbf{o}^{t,(l)} &= \sigma(W_{o,in}^{(l)}\mathbf{h}^{t,(l-1)} + W_{o,out}^{(l)}\mathbf{h}^{t-1,(l)} + \mathbf{b}_{o}^{(l)}) \end{aligned} \tag{2}$$

ただし，$\sigma(\cdot)$ はシグモイド関数である．

さらに，Peehole Connection を含むバリエーションでは，各 Gate にそれぞれ $\mathbf{c}^{t-1,(l)}, \mathbf{c}^{t-1,(l)}, \mathbf{c}^{t,(l)}$ からの寄与が追加される．

最適化の際は，後方の時刻から順に誤差逆伝播により各学習パラメータの勾配を計算できる．誤差逆伝播はミニバッチごとに規定の長さで打ち切り，別のミニバッチをサンプルする．

#### 3.2.2 予測分布モデル

ニューラルネットワークの出力は，単なる変数の予測値ではなく，文献 [21] に従い各パケットの発生確率およびプロセス値や統計量の確率分布を表すパラメータとした．さらに，出力する予測分布は未来の複数の時間間隔を含むものする．関数形として各変数が独立なガウス分布を仮定すると，

$$\begin{aligned}\mathbf{y}_i^t = (&r_i^{t+\tau_1}, \cdots, r_i^{t+\tau_M}, m_i^{t+\tau_1}, \cdots, m_i^{t+\tau_M}, \\ &\sigma_i^{t+\tau_1}, \cdots, \sigma_i^{t+\tau_M}) \end{aligned} \tag{3}$$

である．ここで，$r_i^t$ は時刻 $t$ における $i$ 番目のパケット種別の発生確率，$m_i^t$ は同様にプロセス値や特徴量の平均値，$\sigma_i^t$ は標準偏差である．$\tau_0, \cdots, \tau_M$ は相異なる時間間隔であり，$M$ は出力するホワイトリストの個数である．

損失関数は負対数尤度として以下のように書ける（添字は省略する）．

$$\begin{aligned}L &= -\log p(v, e|m, \sigma, r) \\ &= \log(\sigma + \epsilon) + \frac{(v-m)^2}{2(\sigma+\epsilon)^2} \\ &\quad - e\log r - (1-e)\log(1-r) \end{aligned} \tag{4}$$

**表 1** 異常度パラメータ付きパケットリストの例
**Table 1** Examples of packet list with parameter to calculate anomaly score.

Available Period :2019/11/1 10:00:00-10:10:00

| | IP.src | IP.dst | Object.ID | $r$ | $m$ | $\sigma$ |
|---|---|---|---|---|---|---|
| P1 | 192.168.1.1 | 192.168.1.2 | 5 | 0.9 | 30 | 2 |
| P2 | 192.168.1.1 | 192.168.1.2 | 6 | 0.8 | 40 | 20 |
| P3 | 192.168.1.3 | 192.168.1.2 | 15 | 0.7 | 20 | 1 |
| P4 | 192.168.1.3 | 192.168.1.2 | 16 | 0.02 | 50 | 10 |

Available Period :2019/11/1 10:10:00-10:20:00

| | IP.src | IP.dst | Object.ID | $r$ | $m$ | $\sigma$ |
|---|---|---|---|---|---|---|
| P1 | 192.168.1.1 | 192.168.1.2 | 5 | 0.8 | 30 | 5 |
| P2 | 192.168.1.1 | 192.168.1.2 | 6 | 0.01 | 45 | 10 |
| P3 | 192.168.1.3 | 192.168.1.2 | 15 | 0.04 | 25 | 5 |
| P4 | 192.168.1.3 | 192.168.1.2 | 16 | 0.3 | 50 | 8 |

ただし，$v$ は予測先時刻のプロセス値であり，$e$ は実際に予測先時刻でパケットが発生したかどうかを示す変数である．

### 3.3 異常度分析

LSTM の学習が完了すると，時刻 $t$ までの時系列データ $\mathcal{D}^t$ から予測分布を推定するモデル $g_\theta$ が得られる．$\theta$ は LSTM の学習パラメータである．

$$\begin{aligned}(&r_i^{t+\tau_1}, \cdots, r_i^{t+\tau_M}, m_i^{t+\tau_1}, \cdots, m_i^{t+\tau_M}, \\ &\sigma_i^{t+\tau_1}, \cdots, \sigma_i^{t+\tau_M}) \\ &= g_\theta(\mathcal{D}^t) \end{aligned} \tag{5}$$

これにより，時刻 $t$ において生成する $m$ 番目のパケットリストの $i$ 番目のパケットに異常度計算のためのパラメータ $(r_i^{t+\tau_m}, m_i^{t+\tau_m}, \sigma_i^{t+\tau_m})$ を付与する．異常度パラメータ付きのパケットリストの一例を表 1 に示す．項目は送信元 IP アドレス（IP.src），宛先 IP アドレス（IP.dst），制御プロトコルのデータ部に含まれる機能番号やオブジェクト番号（Object.ID），異常度パラメータ（$r, m, \sigma$）を表すが，あくまでも一例である．

$i$ 番目のパケットのプロセス値 $v_i$ に対する異常度は，Shannon 情報量異常度および Mahalanobis 距離をベースとして計算できる．重み $\lambda_i$ を用いて

$$\begin{aligned}&A_\theta(v_i|\mathcal{D}^t) \\ &= \max\left[\log r_i^{t+\tau_m}, \frac{\lambda_i(v_i - m_i^{t+\tau_m})^2}{2(\sigma_i^{t+\tau_m})^2}\right] \end{aligned} \tag{6}$$

と計算する．

$\lambda_i$ と $a_i$ は k-fold closs validation 等によって設定すべきパラメータである．特に考慮すべきナレッジが無い場合は，まず $\lambda_i$=0 として，正常データで異常判定が行われないか，または誤検知が許容可能なレベルに収まるように閾値 $a_i$ を決定する．次に $a_i$ を固定し， $\lambda_i$ を同様に決定する．

**表 2** ホワイトリストの生成
**Table 2** Whitelist Generation.

$r^{\cdots}, m^{\cdots}, \sigma^{\cdots}$ are outputs of the prediction model.
$a_i, \lambda_i$ are threshold parameters.
**for** m in 1 to Number of whitelists to be generated. **do**
  **for** i in 1 to Number of packets in packetlist. **do**
    **if** $\log r_i^{t+\tau_m} \geq a_i$ **then**
      $i^{th}$ packet is not included in $m^{th}$ whitelist.
    **else** $(\log r_i^{t+\tau_m} < a_i)$
      Put $i^{th}$ packet into $m^{th}$ whitelist.
      And set the range of value as follows.
      $Range \leftarrow m_i^{t+\tau_m} \pm 2\sigma_i^{t+\tau_m}\sqrt{\frac{a_i}{\lambda_i}}$
    **end if**
  **end do**
**end do**

**表 3** 有効期間付きホワイトリストの例
**Table 3** Examples of whitelist with available periods.

Available Period :2019/11/1 10:00:00-10:10:00

| | IP.src | IP.dst | Object.ID | Range |
|---|---|---|---|---|
| P1 | 192.168.1.1 | 192.168.1.2 | 5 | 24-36 |
| P2 | 192.168.1.1 | 192.168.1.2 | 6 | 0-100 |
| P3 | 192.168.1.3 | 192.168.1.2 | 15 | 17-23 |

Available Period :2019/11/1 10:10:00-10:20:00

| | IP.src | IP.dst | Object.ID | Range |
|---|---|---|---|---|
| P1 | 192.168.1.1 | 192.168.1.2 | 5 | 15-45 |
| P4 | 192.168.1.3 | 192.168.1.2 | 16 | 26-74 |

### 3.4 ホワイトリスト生成

予測モデルを用いた異常度分析により未来のトラフィックの異常度を計算し，閾値を下回っているパケット種別はその期間に発生しても良いと解釈し，そのパケット種別とプロセス値の許容範囲を，ホワイトリストに記載する．

この方法により，連続的で無数の状態を列挙することを回避し，状態に応じてホワイトリストを生成することで，状態依存性を表現する．

ホワイトリスト生成のための疑似コードを表 2 に示す．

また，表 1 に示した異常度付きパケットリストから最終的に生成される有効期間付きホワイトリストの一例を，表 3 に示す．“Range” はプロセス値の許容範囲である．

#### 3.4.1 検知

表 3 のようなホワイトリストを用いて，検査用機器にて有効時間内に受信したパケットを検査する．

## 4. 実験

### 4.1 ビル管理システムのトラフィックデータ

実験では，提案手法の ICS システムへの適用可能性，および正規の機器から正規のコマンドで行われるような高度な不正制御の検知可能性を検証することを目的とした．そのために，実験用のデータとして，実運用中のビル管理システムのトラフィックデータを収集した．

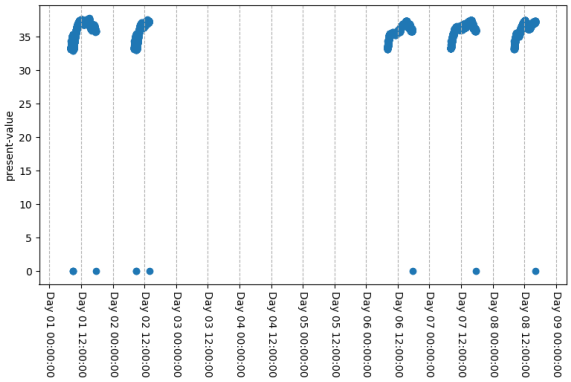


**図 4** 正常時における write パケットの温度制御値
**Fig. 4** Setting values of room temperature in write packet under the normal operation..

データの収集タイミングでは，ビルは正常運用中であった．ビルには管理サーバと監視 PC からなる中央監視機器が存在し，各階に設置された複数のコントローラを介して，ビル内のフィールド機器の状態を監視している．例えば空調機器や照明機器の監視・制御，人物センサや冷却設備・発電・受変電設備の監視等が行われているものである．

制御ネットワークの通信プロトコルとして BACnet/IP [22] が利用されていた．

**図 4** に，一例として write パケットに格納されるプロセス値（Presnt-value）を示す．このパケットは，温度制御のためのインバータの動作量を設定するパケットである．図 4 に示したのはパケットの 8 日分のプロットであり，この区間では平日の昼間にのみ制御コマンドが送信されていた．

### 4.2 模擬攻撃データ

例示した write パケットを攻撃者が送信し，異常な温度制御を試みることを想定し，正規の機器から正規のコマンドで行われるような高度な不正制御を模擬する攻撃データを作成した．模擬攻撃は 2 通りを準備した．

(1) パケット挿入による夜間の温度上昇：本来は気温が高く多数の機器が稼働している状態（昼）で発生する空調制御コマンドを，気温が低下途中で他の機器がほぼ稼働していない状態（夜）で送信する．

(2) パケット改ざんによる昼間の空調停止：本来は気温が低い状態（朝）や気温が高い状態（夕）で送信される，インバータ動作量 0 の制御コマンドを，気温が上昇しつつある状態（昼）で送信する．

攻撃 (1) は Day8 の write パケットの両側 2 箇所（Day 08 00:00:00 付近，Day 09 00:00:00 付近）に挿入し，攻撃 (2) は元々送信されていた write パケットの値のみを改ざんし

た（Day 08 12:00:00 付近）．いずれの攻撃も，正常時と全く同じ MAC アドレス，IP アドレスになりすまして行われることを想定した．パケットの種類やパケット内のプロセス値も，正常運用時であっても発生し得るパターンである．

### 4.3 実験方法

提案手法では，まず正常データを用いて予測モデルを学習した．学習には，Present-value の値が存在し，出現回数が多い 500 種類程度のパケットを用いた．

その学習済みの予測モデルを用いて，Day8 付近のデータに挿入した模擬攻撃を 2 つとも検知できるかどうかを検証した．

また，上記とは別途，Raspberry Pi [23] に検知アルゴリズムを実装し，ホワイトリストの生成や検知をリアルタイムに行えるかどうかを検証した．ホワイトリストの更新間隔は 1 分とした．

### 4.4 実験結果

二つの模擬攻撃の検知結果を**図 5** に示す．縦軸は異常度である．左上と右上の “Attack1” で示す攻撃 (1) と，中間の “Attack2” で示す攻撃 (2) の両方で異常度が増大し，二つの攻撃を検知することができた．特に，閾値として図 5 下図の赤い点線で示した Threshold=2.2 を取ると，誤検知無しで攻撃 (1)（30 パケット）と攻撃 (2)（30 パケット）を 100 %検知可能であった．この閾値はガウス分布を基準とすると $2.2\sigma$ に相当する．一方，より現実的な閾値 Threshold=3 （薄い桃色の点線で示す）を取ると，攻撃 2 で 2 パケット検知漏れが生じ，検知率は 93 %となった．また，2.2 より小さい閾値を取ると誤検知が生じる．

また，Raspberry Pi 上での検知動作では，シンプルなホワイトリストの検査であるため，ハッシュ探索を用いなくても 2.5ms ± 1.5ms 程度でパケット検査が完了していた．ホワイトリストの更新動作は 10s 程度であった．更新間隔が 60s であるため，十分に余裕がある．

## 5. 考察

提案手法における予測モデルは，検知性能向上のために平均値ではなく分布を予測する．一例として，温度制御のためのインバータ動作量の許容範囲の予測結果を**図 6** に示す．この動作量は前述の実験に用いたものとは異なり，write コマンドではない．現在値が変動している間，“Predicted range” として示すエラーバーの中心（予測平均）の変化は小さいが，その幅（標準偏差）は増大している．予測平均の変化が小さいのは，現在値 (present-value) が 50 付近へ不連続に変化するタイミングがまちまちであり，予測困難だからである．それでも，確率分布の予測によって，平均値の予測の不確実性が変動する状況に追従した監視ができていることがわかる．

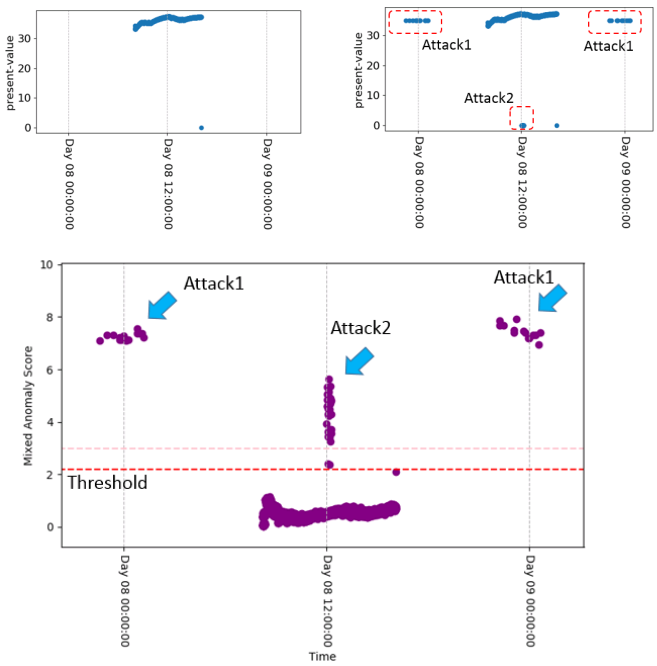


**図 5** 左上：正常時の Day8 の write パケット
右上：攻撃時の Day8 の write パケット（攻撃 (1)，攻撃 (2)）
下：攻撃時の Day8 における異常検知結果

**Fig. 5** Upper left: Write packet in normal state on Day 8. Upper right: Write packet under attack1, 2 on Day 8. Lower: Anomaly score for each packets on Day 8 under attack.

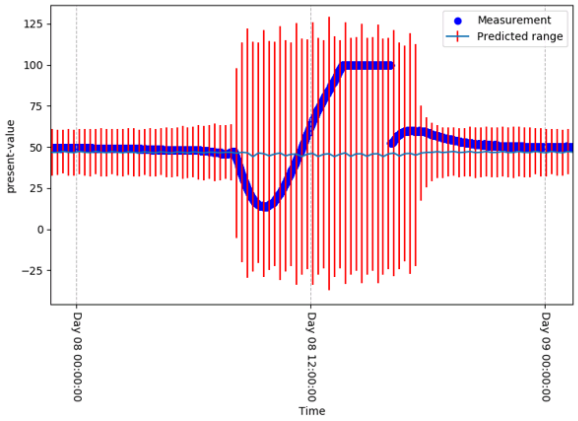


**図 6** インバータ動作量の値と予測許容範囲

**Fig. 6** Manipulating value of an inverter and the prediction of permissible range.

なお，本稿の提案手法は，離散的に定義される従来のホワイトリストをさらに一般化するものである．本手法により，従来手法ではカバーしきれない未知の攻撃を検知することができると考えられる．しかしながら，機械学習による Anomaly-based 手法であるため，本手法だけでは誤検知が生じやすい問題が残る．

今回の実験でも，正常なパケットであるにもかかわらず異常度が高いパケットが存在しており，検知期間が伸びれば誤検知の可能性は増す．例えば，今回異常度が高かった

正常パケットと異常度が最も低かった攻撃パケットとを判別するには，昼の状態と夕方の状態とを区別しなければならない．これらはそもそも類似した状態であるため，学習データの揺らぎ，学習結果の揺らぎ，そしてモデル自体の表現能力による誤差の影響を受けやすかったと考えられる．

モデル自体の最適化やデータの蓄積だけでなく，本手法と Specification-based 手法との組み合わせにより，検知率の向上と誤検知の抑制を達成することが期待できる．例えば，単純な方法として，Specification-based 手法において単一のシステム状態と考えられていたものの，実際には運用によって生じる複数の状態を含む場合，本手法でさらに詳細なホワイトリストを生成することが考えられる．また，本手法において十分なデータが無く，誤検知が増加するような状態において Specification-based 手法での検知に切り替えるといった方法が考えられる．

## 6. おわりに

本稿では，正規の機器から正規のコマンドで行われるような高度な不正制御を検知するために，システム状態を考慮したネットワーク異常検知手法を提案した．提案手法では，システム状態に応じて動的にホワイトリストを生成し，異常を検知する．特に，正常データを用いてニューラルネットワークによる予測モデルを学習することで，プロセス値の集合から正常なパケットの発生パターンを直接推定する．それにより，システム状態を明示的に定義・抽出することなく，逐次的に最適なホワイトリストを生成できる．最終的に生成するのは，送信元や送信先，データ部のフィールド値等を列挙したシンプルなホワイトリストであり，解釈や検知動作が容易である．

また，実運用中のビル管理システムのトラフィックデータを用いて，正規の機器から正規のコマンドで行われる 2 種類の模擬攻撃をパケット単位でリアルタイムに検知できることを確認した．

今後，学習モデルの拡張や Specification-based 手法との組み合わせ等による誤検知の抑制・検知精度の向上，また，より複雑な運用を行うシステムでの評価を実施していく．

### 参考文献


[1] Stouffer, K., Lightman, S., Pillitteri, V. :Guide to Industrial Control Systems (ICS) Security, *NIST Special Publication 800-82* (2011).

[2] Angle, M.G., Madnick, S., Kirtley, J.L., et al. :Identifying and Anticipating Cyber Attacks that could cause Physical Damage to Industrial Control Systems, *IEEE Power and Energy Technology Systems Journal* (2019).

[3] Falliere, N., Murchu, L.O., Chien, E. :W32.Stuxnet Dossier, Symantec, Version 1.4 (2011).

[4] Cherepanov, A. :WIN32/INDUSTROYER A New Threat for Industrial Control Systems, ESET (2017).

[5] Elmrabet, Z., Elghazi, H., Kaabouch, N., et al. :Cyber-Security in Smart Grid: Survey and Challenges, *Computers & Electrical Engineering*, 67, 469-482 (2018).

[6] Kshetri, N., Voas, J., :Hacking Power Grids: A Current Problem, *Computer*, 50(12), 91-95 (2017).

[7] Hata, K., Sawada, K., Nakai, T., et al. :Model Based Anomaly Detection Focused on The Operating States, *SICE*, 11-14 (2018).

[8] Zarpelao, B.B., Miani, R.S., Kawakani, C.T., et al. :A survey of intrusion detection in Internet of Things, *Journal of Network and Computer Applications*, 84, 25-37 (2017)

[9] Kim, B.K., Kang, D.H., Na, J.C., et al. :Abnormal Traffic Filtering Mechanism for Protecting ICS Networks, *International Conference on Advanced Communication Technology*, IEEE, 436-440 (2016).

[10] Kobayashi, N., Shimizu, K., Nakai, T., et al. :Intrusion Detection Method Using Enhanced Whitelist Based on Cooperation of System Development, System Deployment, and Device Development Domains in CPS, *International Conference on Network-Based Information Systems*, Springer, Cham, 430-444 (2018).

[11] 秦 康祐, 望月 明典, 澤田 賢治 他 :ホワイトリスト運用を目的とした制御システムの状態分離アルゴリズムの検討, *SCIS*, 2E1-4 (2018).

[12] Lopez-Martin, Manuel, et al. :Conditional Variational Autoencoder for Prediction and Feature Recovery Applied to Intrusion Detection in IoT, *Sensors*, 17(9), 1967 (2017)

[13] Niyaz, Q., Sun, W., Javaid, A.Y., et al. :A deep learning approach for network intrusion detection system, *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (BIONETICS)*, ICST (2016)

[14] Vinayakumar, R. et al. :ScaleNet: Scalable and Hybrid Framework for Cyber Threat Situational Awareness Based on DNS, URL, and Email Data Analysis, *Journal of Cyber Security and Mobility*, (8)2, 189-240 (2019)

[15] Shang, W., et al. :Industrial Communication Intrusion Detection Algorithm Based on Improved One-Class SVM, *World Congress on Industrial Control Systems Security (WCICSS)*, IEEE (2015)

[16] Mirsky, Y., Doitshman, T., Elovici, Y., et al. :Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection, arXiv: 1802.09089v2 [cs.CR] (2018).

[17] Hochreiter, S., Schmidhuber, J. :Long Short-term Memory, *Neural Computation*, 9(8), 1735-1780 (1997)

[18] Sutskever, I., Vinyals, O., Le, Q.V. :Sequence to Sequence Learning with Neural Networks, *NIPS* (2014)

[19] Goh, J., Adepu, S., Tan, M., et al. :Anomaly Detection in Cyber Physical Systems using Recurrent Neural Networks, *IEEE HASE* (2017).

[20] Kiss, I., Genge, B., Haller, P., et al. :Data Clustering-based Anomaly Detection in Industrial Control Systems, *IEEE 10th International Conference on Intelligent Computer Communication and Processing (ICCP)* (2014)

[21] Nix, D.A., Weigend, A.S. :Estimating the mean and variance of the target probability distribution, *International Conference on Neural Networks*, IEEE (1994).

[22] Bushby, S.T. :BACnet$^{TM}$ : A Standard Communication Infrastructure for Intelligent Buildings, *Automation in Construction*, 6, 529-540 (1997).

[23] Sforzin, A., Conti, M., Gomez, F., et al. :RPiDS: Raspberry Pi IDS A Fruitful Intrusion Detection System for IoT, *IEEE UIC /ATC /ScalCom /CBDCom /IoP /SmartWorld*, 440-448 (2016).