Vol.2021-DPS-186 No.64
Vol.2021-CSEC-92 No.64
2021/3/16

Auto-Encoderによる異常検知手法のための実数を使用した秘密計算方法

松本 麻里[1,a)]　古田 雅則[1]

概要：秘密データを含むビッグデータの有効活用とプライバシー保護を両立させる方法として，データを秘匿化したまま計算を行う秘密計算の需要が高まっている．近年では機械学習を含むデータ分析に秘密計算を用いる研究が多く行われている．本稿では，Auto-Encoder による異常検知への活用を目的として，平文と同等の検知率を目指した高精度な秘密計算実現に取り組む．秘密分散による秘匿化データを用い，学習および推論では実数を含む秘密計算を実施し，NSL-KDD データセットを用いた評価において，小規模でのノード構成においては平文と同等の約 77%の検知率が得られた．

キーワード：秘密計算，機械学習，異常検知，Auto-Encoder

Secret computation method using real numbers for anomaly detection method by Auto-Encoder

Abstract: As a method of achieving both effective utilization of big data including secret data and privacy protection, there is an increasing demand for secret calculation that performs calculations while keeping data confidential. Recently, much research has been done using secret calculations for data analysis, including machine learning. In this paper, we will work on the realization of highly accurate secret computation aiming at the detection rate equivalent to plaintext, with the aim of utilizing it for anomaly detection by the auto-encoder. Using concealed data by secret sharing, secret computation including real numbers is performed in learning and inference, and in evaluation using NSL-KDD data set, about 77% detection equivalent to plain text in small-scale node configuration The rate was obtained.

Keywords: secret computation, machine learning, anomaly detection, Auto-Encoder

1. はじめに

近年、パソコンやスマートフォンといった従来型 ICT（Information and Communication Technology）端末のみでなく，自動車，家電，ロボットといったあらゆるモノがインターネットに接続する IoT 化が進み，新たなサービスの提供や装置の自動化による利便性向上が期待される [1]-[3]．工場，ビル等の社会インフラの運用 · 保守分野においては，基幹システムの運用 · 保守高度化に加え，作業者の安全性向上や作業効率向上に向けた自動化が加速している．システムの構成要素である様々なエッジ機器が混在する基幹システムの安定運用の鍵は，各機器の状態に関する情報のクラウドへの集約と集約した情報を用いた機器の遠隔監視である．社会インフラの運用 · 保守の分野では，自動化加速要求の増加に伴い，今後もますますエッジ機器の IoT 化拡大が進むと考えられる．しかし，それに伴いマルウェア等の攻撃機会が増加し，攻撃手法も日々新たな未知のマルウェアが発生している．従来のパターンマッチングによる攻撃検出方法のみでは外部からの攻撃を防ぐことが難しくなっているのが現状であり，近年ネットワーク異常検知に機械学習を用いる方法が多く検討されている [5]-[7]．

例えば，工場内等の環境下でこのようなネットワーク異常検知を実施する際，多数のエッジ機器からの膨大なデータを工場内のエッジサーバに集約し，サーバの計算資源を用いてネットワークの異常検知を実施することが考えられ

[1] （株）東芝
1, Komukai Toshiba-cho, Saiwai-ku, Kawasaki-shi 212–8582, Japan
[a)] mari.matsumoto@toshiba.co.jp

る．一般に，エッジサーバの計算能力上限は制限されることから，膨大な計算量が必要となる異常検知の処理時間短縮化が課題となる．高い計算能力を保有するクラウドサービスを利用することは，処理時間短縮化を実現する有効な課題解決手段の一つである．しかしながら，現状のクラウドにおける異常検知は，エッジ機器間の通信データやセンサデータ等の工場で取得した秘匿性の高いデータを平文で取り扱う必要があり，重要データが第三者に流出するリスクが発生する．

秘密計算は，情報を暗号化したままの状態で演算可能な技術であり，秘密計算を用いることで，工場の重要データを安全に守った状態でデータ分析などに利用することができる．このような背景からこれまでに様々な統計手法や機械学習手法を秘密計算で行われてきた [11]-[13].

本稿では異常検知に多く用いられる Auto-Encoder において，秘密分散で秘匿化したネットワークデータを用いて学習および推論を実施しその推論結果から異常検知を行う．本稿では，データの秘匿化の有無にかかわらず，異常検知率の保持を可能とすることを目的とし，秘密分散法および復号時に行うアルゴリズムについて提案し，評価を行う．

1.1 本稿の成果

本稿では，Auto-Encoder による異常検知を秘匿化データを使用して実施することを目的とし，その秘匿化のための秘密分散方法および復号方法を提案する．Auto-Encoder は入出力を 0〜1 の実数とし，秘密計算では扱いが難しい指数や除算という実数演算を含む．本稿では秘密分散の範囲に制限を加え，復号化した推論結果が体 Q 近傍であった場合に，復号化した推論結果を体 Q とこの推論結果の値の差分に変換を行う秘密分散および復号方法を提案する．そして，ネットワーク異常検知に多く用いられる NSL-KDD データセットにより上記手法を用いて評価を実施し，入出力 10 程度の小規模のノード構成の Auto-Encoder では学習および推論データに秘密分散処理の有無による異常検知率に殆ど差が生じないことを示した．

2. 準備

2.1 表記方法

a を b で定義することを a:=b とし，ベクトルを $\vec{a}$:=(a_1, $\cdots$,a_n) とする．また，行列を A:=($\vec{a_1}$,$\vec{a_2}$,$\cdots$,$\vec{a_m}$) と書く．行列やベクトルの内積は (・) で表し，要素ごとの積は（×）で表し，演算子がないときはスカラー倍を表す．

2.2 Shamir の（k,n）しきい値秘密分散

秘密計算には，大きく分けて準同型暗号 [4] と秘密分散 [8]-[10] の 2 つの方式がある．秘密分散法は，「シェア」というパーツに分解して複数のパーティ間で共有する方法であり，一般的に準同型暗号よりも計算コストを抑えることができ，データ解析などには実用的であると言われている．本稿では，n 個のシェアを生成し，k 個以上のシェアからは秘密が復元できるが，k 個未満のシェアからでは秘密情報が全く漏れない Shamir の (k, n) しきい値秘密分散をベースとした秘密分散法を提案する．以下にベースとなる Shamir の (k, n) しきい値秘密分散について述べる．ここではデータ保有者が自身の持つデータを外部のパーティ（P）に安全に預けたい状況を考える．

分散：データ保有者がデータ a を n 箇所のパーティ $\{P_1,\cdots,P_n\}$ に安全に分散したいとする．このときデータ保有者は a∈Q となるような体 Q を選択し，以下の手順で分散を行う．

(1) k-1 個 ($1 < k \leq n$) のランダムな Q の元 $r_1,\cdots,r_{k-1}$ を選択し，a を切片とする k-1 次多項式 $M(x) = \sum_{i=1}^{k-1} r_i x^i + a$ を構成する．

(2) n 個の Q の元 $x_1,\cdots,x_n$ を選択し，$M(x_1),\cdots,M(x_n)$ を求める．(ただし、任意の $t \neq t'$ について，$x_t \neq x_{t'}$)

(3) $M(x_t)$ を P_t に送信する．このとき，$M(x_t)$ は P_t における a のシェアであり，$[a]_t := M(x_t)$ と表記する．

復元：データ保有者は P_t に格納している $[a]_t$ から以下を行うことで元のデータ a を復元することができる．

(1) $\{P_1,\cdots,P_n\}$ からパーティ k 個を選択し，$\{P_{t_1},\cdots,P_{t_k}\}$ とする．（ただし，任意の $j \neq j'$ について，$t_j \neq t_{j'}$）

(2) 選択した P_{t_j} から $[a]_{t_j}$ を受け取る．

(3) $a = \sum_{j=1}^{k} \lambda_{t_j} [a]_{t_j}$ を計算し，復元する．このとき λ_{t_j} は，Laglange 補間法における Lagrange 係数である．

2.3 秘密計算

ここでは秘密分散により生成されたシェアによる演算について述べる。平文とシェアを区別するために，平文 a の暗号文を $[\![a]\!]$ と表す。

2.4 正規化方法

秘密分散により生成された整数のシェアを Auto-Encoder で学習および推論を実施する前に max-min normalization によって正規化を行う．整数 x を式 (1) を用いて正規化する．

$$x_{nm} = \frac{(x - x_{min})}{(x_{max} - x_{min})} \tag{1}$$

ここで，x_{nm} は x を正規化した値であり，x_{max} および x_{min} はそれぞれベクトル $\vec{x} := (x_1, \cdots, x_n)$ における最大値と最小値である．行列に含まれるベクトルごとにこの正規化を実施する．

2.5 Auto-Encoder

Auto-Encoder は，教師無し学習の一つであり，そのため学習時の入力データは訓練データのみで教師データは使用しない学習手法である．その特徴から異常検知への活用も多く検討されている [14]．Auto-Encoder は，学習データと近い値の再構成を行うモデルを生成する．Auto-Encoder は式（2）で表されるエンコーダと，式（3）で表されるデコーダの２つの要素から構築される．

$$h = \sigma(W_{ij} \cdot a + b_{ij}) \tag{2}$$

$$z = \sigma(W_{ji} \cdot h + b_{ji}) \tag{3}$$

$$\|a - z\| \tag{4}$$

ここで，σ は Activation 関数，W_{ij} および W_{ji} は重み，b_{ij} および b_{ji} はバイアスを表している．式 (2) で表されるエンコーダは入力ベクトル a を隠れ層にマッピングする．このとき，入力ベクトル a は，アフィン変換により隠れ層ベクトル h に変換される．式（3）で表されるデコーダは，隠れ層ベクトル h をエンコーダと同様にアフィン変換を用いることで入力ベクトル a と同じベクトル空間に変換する．また，式（4）で示した元の入力ベクトル a と再構成されたベクトル z の差は再構成誤差と呼ばれ，Auto-Encoder はこの再構成誤差が最小になるように学習が行われる．Auto-Encoder を異常検知に用いる場合，Auto-Encoder に正常データを学習させ，正常データしか再構成できないモデルを構築する。このモデルに異常データが入力された場合，再構成された出力データは入力データとの間に大きな誤差が生じる．そのため，入力データと出力データの差を見ることで異常判定が可能となる．

2.6 異常検知手法

異常検知には推論時に Auto-Encoder へ入力する推論データと，推論データを学習モデルを介して得られた推論結果を用いて再構成誤差を算出し，その再構成誤差を使用する．すなわち，再構成誤差は，入力する推論データベクトル $\vec{y} := (y_1, \cdots, y_n)$ に対し，

$$e = \sum_{i=1}^{n} (y_i - y_{pred_i}) \tag{5}$$

となる．ここで，y_i は規格化した推論データであり，y_{pred_i} は学習モデルを介して得られた推論結果を規格化したデータである．式（5）に示す再構成誤差があるしきい値以上となった場合を異常と判定する．

3. 提案方法

本稿では，データ保有者がもつ平文を秘密分散することで得られたシェアデータを複数サーバに送信し，各サーバで Auto-Encoder によって学習および推論を実施した後，その推論結果を基に異常検知を行う方法を提案する．

3.1 提案フロー

これより，本手法の異常検知の手順について述べる．本手法の流れを図１に示す．本手法は大きく分けて学習と推論の２つの処理に分けることができ，学習・推論で使用するデータはともに本稿で提案する秘密分散方法によって秘匿化したものを使用する．なお，その秘密分散方法および復号方法については次節で詳細を述べる．学習時の処理では，データ保有者が学習に使用するデータとして，パケットデータやセンサデータ等を用意し，それらのデータを秘密分散によって秘匿化を行う．シェアデータを生成し，生成されたシェアデータは 2.4 節で述べた方法を用いて規格化を行う．この規格化の処理は，Auto-Encoder の入力データの範囲が 0〜1 間の実数である必要があるため実施するものである．学習データの入力データがXで表される行列である場合，シェアは $[\![X]\!]$ で表すことができ，このシェアを規格化したものを $[\![X_{nm}]\!]$ とすると，シェアデータはこの規格化の後のデータ $[\![X_{nm}]\!]$ を各サーバへ送信する．各サーバで受信した規格化されたシェアデータを用い，Auto-Encoder によって学習モデルを生成する．例えば，サーバ P_t における Auto-Encoder による秘密計算は，入力データである学習データが $[\![X_{nm}]\!]_t$ と表されるとき，式 (2) および式 (3) に示される Auto-Encoder のエンコーダおよびデコーダは以下のように表すことができる．

$$[\![h]\!]_t := [\![\sigma(W_{ij} X_{nm} + b_{ij})]\!]_t \tag{6}$$

$$[\![z]\!]_t := [\![\sigma(W_{ji} h + b_{ji})]\!]_t \tag{7}$$

また，式（8）に示す再構成誤差についてもシェアの入力ベクトルと，再構成されたベクトル z の差が最小になるよう各サーバで学習を行う．

$$\| [\![X_{nm}]\!]_t - [\![z]\!]_t \| \tag{8}$$

ここで使用する Auto-Encoder は，入力層，出力層，隠れ層１層の計３層で構成され，層の数，各層のノード数は，学習を行う全てのサーバで同じものとする．このような条件下で実施された学習によって得られた学習モデルは，各サーバで格納される．

続いて，推論時の処理では，学習時の処理と同様に推論データを秘密分散によりシェアデータを生成し，そのシェアデータの規格化を行う．そして，規格化したシェアデータを各サーバに送信する．各サーバにて，格納された学習モデルによって推論処理を実施する．ここで得られた推論結果（すなわち，各サーバにおける Auto-Encoder の学習

モデルを介した出力データ）を各サーバで格納する。このようにして各サーバで得られた推論結果のうち，任意のサーバから復号に必要な k 個の推論結果をデータ保有者に送信する．そして，データ保有者は，受信した k 個の推論結果を復号前に整数に変換する必要がある．ここでの整数への変換は，規格化を行ったときに使用した式 (1) を利用する．式 (1) の x_{max}，x_{min} は規格化したときに使用した値と同一のものを使用し，x_{nm} を受信した推論結果として，x を求める．x に小数点以下の数が含まれる場合には四捨五入により整数化する．上記のように，得られた整数化された各サーバの推論結果から，推論結果を復号する．このとき，推論結果の復号値が体Q近傍の値となった場合には，後述の修正処理を行い，平文の推論結果を得る．異常検知については，2.6 節に記載したように，平文の推論データと復号した推論結果の差分による再構成誤差を指標として行う。このような手順にそって異常検知を行うことで，サーバ側には平文データを公開せず，秘密計算により Auto-Encoder による学習および推論を実施することができる．

3.2 提案する秘密分散条件の定義

前節の提案フローで述べたように，データの秘匿化には秘密分散をを用いるが，2.2 節で述べた従来の Shamir の (k,n) しきい値秘密分散法では生成したシェアで学習および推論を実施した結果から異常検知を行った場合平文と同程度の検知率を得るのは難しい．その理由の一つは，シェアを用いた秘密計算に実数を含むことにある．本提案では，秘密計算部分である Auto-Encoder の演算は平文の演算と同様のアルゴリズムを使用し，秘密分散および復号方法の変更によって Auto-Encoder を秘密計算で実施するための方法を以下に提案する．

分散：データ保有者がデータ a を n 箇所のパーティ $\{P_1,\cdots, P_n\}$ に安全に分散したいとする．このときデータ保有者は a∈Q となるような体 Q を選択し，以下の手順で分散を行う．

(1) k-1 個 $(1 < k \leq n)$ のランダムな Q の元 $r_1,\cdots,r_{k-1}$ を選択し，a を切片とする k-1 次多項式 $M(x) = \sum_{i=1}^{k-1} r_i x^i + a$ を構成する．

(2) n 個の Q の元 $x_1,\cdots,x_n$ を選択し，$M(x_1),\cdots,M(x_n)$ を求める．(ただし、任意の $t \neq t'$ について，$x_t \neq x_{t'}$) なお，$x_1, \cdots, x_n$ が $x_1 < \cdots < x_n$ の関係にあり，a の最大値が A_{max} であるとき，A_{max} のシェアの中の最大値であるシェア $M_{max}(x_n)$ は，

$$M_{max}(x_n) = \sum_{i=1}^{k-1} r_i x_n^i + A_{max} \qquad (9)$$

と表せる．本提案においては，この $M_{max}(x_n)$ が体 Q の最大値未満（すなわち，$M_{max}(x_n) < Q$）となるように係数 $r_1, \cdots, r_{k-1}$ および $x_1, \cdots, x_n$ が決定されるものとする．

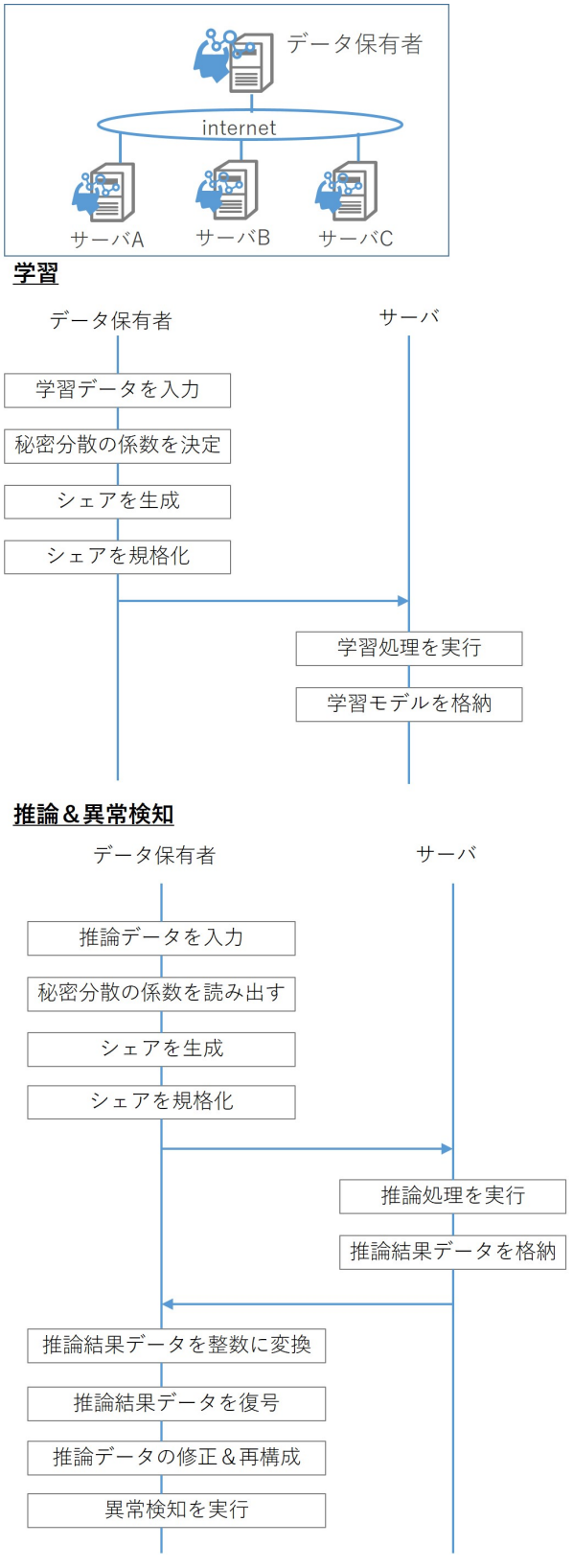


図 1 本稿の異常検知の手順
Fig. 1 Anomaly detection procedure in this paper

(3) $M(x_t)$ を Pt に送信する．このとき，$M(x_t)$ は P_t における a のシェアであり，$[a]_t := M(x_t)$ と表記する．

復元：データ保有者は P_t に格納している $[a]_t$ から以下を行うことで元のデータ a を復元することができる．

(1) $\{P_1,\cdots,P_n\}$ からパーティ k 個を選択し，$\{P_{t_1},\cdots,P_{t_k}\}$ とする．（ただし，任意の $j \neq j'$ について，$t_j \neq t_{j'}$）

(2) 選択した P_{t_j} から $[a]_{t_j}$ を受け取る．

(3) a' $= \sum_{j=1}^{k} \lambda_{t_j}[a]_{t_j}$ を計算し，復元する．このとき λ_{t_j} は，Laglange 補間法における Lagrange 係数である．復号が正しく行われている場合は，a'=a である．

(4) 復号値 a'$\simeq$ Q の場合，a' $\rightarrow$ Q - a' に変換する．例えば，この変換を行う否かの判定値を q_{th} したとき，この q_{th} は Q 未満かつ Q と同桁数の整数を設定する．（すなわち，$q_{th} <$ Q であり，例えば Q=2^{19}-1 のとき，q_{th}=10^5 とする．）

$$a' \rightarrow \begin{cases} a' & \text{if } a' < q_{th}, \\ Q - a' & \text{if } a' \geq q_{th}. \end{cases} \tag{10}$$

上記の定義に従って，得られた復号された推論結果を用い，異常検知を実施する．異常検知の指標である再構成誤差は，規格化し平文の推論データと，規格化した復号工程（4）を経た復号化された推論結果の差分となる．規格化には 2.4 節に示した式（1）の max-min-normalization を行い，x_{max} および x_{min} を使用する．そのため，分散工程（2）で定義した平文のデータの最大値が A_{max} が体 Q と比較して $A_{max} <<$Q であった場合，復号した推論結果 a' の値が体 Q 近傍であると，本来の復号した推論結果が検知結果に正しく反映されない恐れがある．そのため，復号における工程 (4) で，復号後の推論結果のデータ分布範囲を平文の推論データの分布範囲と同程度になるように設定することで検知率向上へつながる．

4. 実験

本手法の有効性を確認するために，NSL-KDD データセットを用いて実験を行った．

4.1 データセット

本稿では侵入検知用ベンチマークデータセットである NSL-KDD を用いて検証を行った [15]．NSL-KDD はカナダの University of New Brunswick において開発されたデータセットで，攻撃パターン 40 クラスにラベル付けされている．本稿では評価のために正常・異常の 2 クラスに大別した．また、データセットには Ethernet の通信に関する 41 種類の特徴量が含まれる．それらは，基本的な特徴量，コンテンツの特徴量とトラフィックの特徴といった 3 つのカテゴリーに分けられる．基本的な特徴には，各 TCP/IP コネクションのパケットヘッダから抽出された情報が含まれる．コンテンツの特徴には，各 TCP/IP コネクションのパケットペイロードを評価した情報が含まれる．

表 1 NSL-KDD データセットのデータ件数
Table 1 Number of data in NSL-KDD dataset.

	学習データ	評価データ
Normal	67343 件	9711 件
Abnormal	—	12833 件

4.1.1 NSL-KDD データセットの予備分類

NSL-KDD データセットには 41 種類の特徴量が含まれているが，Auto-Encoder による学習の前に予め検知に感度の高い特徴量を決定木を用いて抽出する．決定木は目的変数に属する確率を複数の説明変数の組み合わせで算出する方法である．特徴量ごとに決定木 [16] による分類結果から Accuracy を求め，特に Accuracy が 90 %以上となる特徴量 9 種類を抽出した．この 9 種類の特徴量については表 2 に示す．

表 2 NSL-KDD データセットの予備分類により選択した特徴量
Table 2 Features selected by preliminary classification of NSL-KDD dataset.

特徴量
protocol type
service
coun
serror-rate
srv-serror-rate
rerror-rate
srv-rerror-rate
same-srv-rate
dst-host-rerror-rate

4.2 実験結果

4.2.1 評価方法

評価方法は混同行列から求めた Accuracy，Precision，Recall，F-measure を用いた．Accuracy は全データの中で異常とした割合，Precision は異常とした中で実際に異常データであった割合，Recall は異常データを正しく異常とした割合を示す．また、F-measure は Precision と Recall の調和平均を用いて算出することができる．Accuracy，Precision，Recall，F-measure は式（11）〜（14）で求めることができる．

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{11}$$

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

$$Recall = \frac{TP}{TP + FN} \tag{13}$$

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{14}$$

ここで TP(True Positive) は異常データを正しく異常とした数，FN(False Negative) は異常データを誤って正常とした数，FP(False Positive) は正常データを誤って異常とした数、TN(True Negative) は正常データを正しく正常とした数を表す。

4.2.2 従来の秘密分散法を使用した場合の検知率との比較

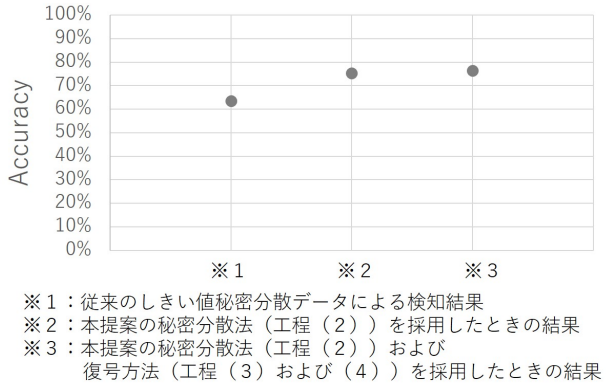
表 2 に示した 9 種類の特徴量を用い，図 1 に示した手順に従って学習および推論を行い，異常検知を行う．ここでは体 $Q = 2^{19} - 1$, 秘密分散先 n は n=8 とする．図 2 は，従来の秘密分散の工程で秘匿化し，学習，推論および異常検知を個なった場合，本稿の秘密分散法を用い，復号は従来の工程で行った場合，秘密分散および復号ともに本稿提案のものを採用した場合の Accuracy を比較する．これより、従来の秘密分散形式による Accuracy が約 64%であったのに対し，本稿の秘密分散方法を採用することで Accuracy は約 75%まで向上し，さらに復号処理についても本稿提案の手法を採用することで Accuracy は約 77 %に向上する．このときの Precision，Recall および F-measure についても比較する．また、表 2 の 9 種類の特徴量を使用し，秘密分散および復号の処理を行わず、Auto-Encoder による学習および推論結果から異常検知を行ったときの結果と，上記の本稿で提案した秘密分散および復号方法を採用したときの結果を比較したものが表 3 である．これより，Accuracy，F-measure ともに数%秘密分散を行ったときの結果のほうがよくなっていることがわかる．このように，従来の秘密分散法では平文による検知結果と 10%以上の低い検知率しか得られなかったが，本稿で提案する秘密分散方法を採用することで，演算部分は平文と同様の処理方法であっても秘密分散の有無によらない検知結果が得られた．

表 3 秘密分散の有無による検知結果の比較
Table 3 Comparison of anomaly detection with and without secret sharing.

	Accuracy	Precision	Recall	F-measure
秘密分散なし	73.2%	94.2%	56.4%	70.6%
秘密分散あり	76.9%	92.3%	64.8%	76.1%

4.2.3 本提案における秘密分散を行ったときの秘密分散値と検知率の依存性

図 3 は，検知率（Accuracy, Precision, Recall, F-measure）の秘密分散における多項式 $M(x) = \sum_{i=1}^{k-1} r_i x^i + a$ の係数 r_i および多項式の次数 k-1 の依存性について示す．ここでは体 $Q = 2^{19} - 1$, 秘密分散先 n=8 とする．図 3 では，横軸



※1：従来のしきい値秘密分散データによる検知結果
※2：本提案の秘密分散法（工程（2））を採用したときの結果
※3：本提案の秘密分散法（工程（2））および
復号方法（工程（3）および（4））を採用したときの結果

図 2 本稿の秘密分散法および復号方法の検知率に与える効果
Fig. 2 Effect on detection rate of secret sharing method and decryption method of this paper

を復号値のしきい値である k が k=2 のときの多項式の係数 r_1 としたときの検知率を示す．これより，k=2 の秘密分散において，検知結果のいずれのパラメータも係数 r_1 に依存しないことを確認できた．これは、秘密分散後は max-min normalization によって 0〜1 の範囲に規格化を行うことから，係数 r_1 が異なるデータ群を使用していても，それ以降の演算は 0〜1 の実数による同様のノード構成の演算を実施するため，係数 r_1 依存はないと考えられる．

図 4 は横軸を多項式の次数に係る k としたときの検知率を示す．k=2,3,4 のときの検知率を示したものであり，図より，k の大きさに関わらず検知率を示すいずれのパラメータもほぼ一定となる．k 依存性についても k=2 の時の係数 r_1 依存性と同様の理由から次数に関わらず検知率にほぼ変化がなかったものと考えられる．

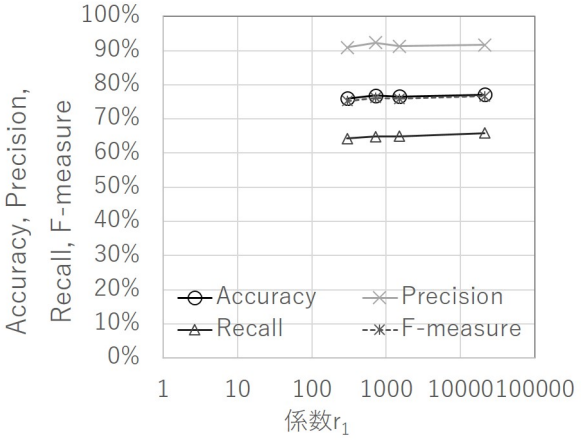


図 3 検知率の秘密分散時の多項式の係数依存性
Fig. 3 Polynomial coefficient dependence of detection rate

5. おわりに

Auto-Encoder によるネットワーク異常検知をモチーフとした，学習および推論データの秘匿化および復号方法に

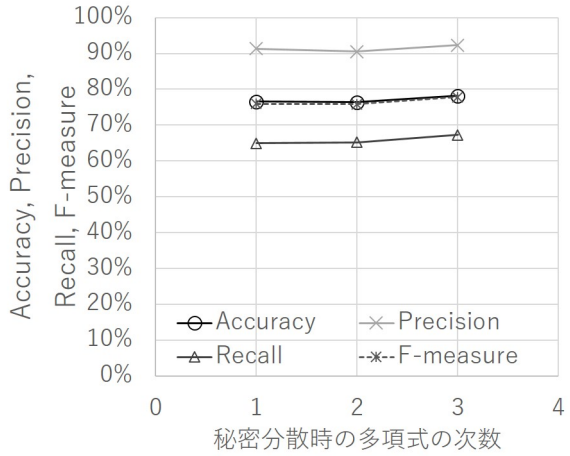


図 4 検知率の秘密分散時の多項式の次数依存性
Fig. 4 Polynomial degrees dependence of detection rate

ついての提案をした．秘匿化手法は，(k,n) しきい値秘密分散方法をベースとし，シェア生成時の多項式の係数範囲に制限を設けることで従来の (k,n) しきい値秘密分散方法を用いて秘匿化したデータを使用した場合と比較して，異常検知結果が向上した．一方で，Auto-Encoder の入出力ノード数の増加により検知率の劣化を確認しており，今後の課題として，ノード数の拡大やネットワークの大規模化対応に向けた理論改善の検討を行う．

参考文献

[1] 総務省：情報通信白書平成３０年版 (online)，入手先 ⟨http://www.soumu.go.jp/johotsusintokei/whitepaper/h30.html⟩ (2021.02.08).
[2] IPA：情報システムの本格利活用時代の到来 (online)，入手先 ⟨https://www.ipa.go.jp/files/000061384.pdf⟩ (2021.02.08).
[3] Zhi-Kai Zhang, Michael Cheng Yi Cho, Chia-Wei Wang, Chia-Wei Hsu, Chong-Kuan Chen, Shiuhpyng Shieh,：*IoT Security*, Ongoing Challenges and Research Opportunities, 2014 IEEE 7th International Conference on Service-Oriented Computing and Applications (2014).
[4] C. Gentry: *Fully homomorphic encryption using ideal lattices*, In STOC, vol. 9, pp. 169-178(2009).
[5] 吉田圭吾, 濱田芳博, 足立直樹, 相羽慎一, 石川史也, 上口翔悟, 上田浩史, 宮下之宏, 畑洋一: *DoIP* と *SOME/IP* におけるセキュリティ分析と侵入検知の検討, Computer Security Symposium 2019(CSS2019) 講演論文集, pp.48-54(2019).
[6] 西村賢太, 山本萌花, 掛井将平, 瀧本栄二, 毛利公一, 齋藤彰一: *IoT* ゲートウェイで動作するコンテナの異常検知手法の提案, Computer Security Symposium 2019(CSS2019) 講演論文集, pp.136-143(2019).
[7] 長友誠, 油田健太郎, 岡崎直宣, 林美娘: ブロックチェーンを用いた分散機械学習におけるパラメータ異常検知システムの提案, Computer Security Symposium 2019(CSS2019) 講演論文集, pp.1343-1348(2019).
[8] Adi Shamir: *How to share a secret*, Communications of ACM, vol. 22, pp. 120-126(1978).
[9] Y. Desmedt, Society and Group Oriented Cryptography: *a New Concept*, Proc. of CRYPTO'87, LNCS 293, pp.120-127(1987).
[10] G. Blakley: *Safeguarding cryptographic keys*, Proc of AFIPS. 48, pp.313-317(1979).
[11] 田中哲士，山田真徳，菊池亮: 秘密分散ベース秘密計算を用いたニューラルネットワークのコスト評価, CSEC-73 No.21, Vol.2016, pp.1-8(2016).
[12] 三品気吹，濱田浩気，五十嵐大，菊池亮: 秘密実数演算を用いた高速かつ高精度なロジスティック回帰とデータ標準化, Computer Security Symposium 2020(CSS2020) 講演論文集, pp.1142-1149(2020).
[13] 五十嵐大: 効率的シフト量秘匿シフトプロトコルの構成による，速度と精度を両立する秘密計算上の浮動小数点数の実現, Computer Security Symposium 2020(CSS2020) 講演論文集, pp.1134-1141(2020).
[14] Jinwon An, Sungzoon Cho: *Variational Autoencoder based Anomaly Detection using Reconstruction Probability*, SNU Data Mining Center Special Lecture on IE (2015).
[15] University of New Brunswick：NSL-KDD dataset(online)，入手先 ⟨http://www.unb.ca/cic/datasets/nsl.html⟩ (2021.02.08).
[16] L.Breiman: *Random Forests*, Machine Learning, 45,1, pp.5-32 (2001).