# CAN 通信における汎用的な攻撃検出を目的とした時系列データ解析

福田 國統[†1]　礒山 芳一[†1]　濱田 芳博[†1]　畑 洋一[†1]

**概要**：車載ネットワークの高度化にともない，通信手法である Controller Area Network（CAN）におけるセキュリティ対策の重要性は高まり続けている．CAN の保護手法である異常検出は，関連データの利用により推定精度が向上するが，各データの意味，および，それらデータ間の関係性が既知の場合に限定される．そこで，本研究では，すべての CAN データを一律にバイト単位に分離，関連性の高いデータセットを抽出し，そのデータセットを用いた汎用的な異常検出を提案する．実車の CAN ログへ適用し，データ内容が不明な場合にも適用可能であり，なりすまし攻撃に対して有効なことを示す．

**キーワード**：Controller Area Network（CAN），異常検出，機械学習

## 1. はじめに

近年，車両の安全性，利便性を向上させるために，一台あたり 70~100 台の Electron Control Units ( ECU ) が車両制御ネットワークを介して制御情報をやり取りし[1]，また，様々なサービスとネットワークを介して車両外と情報を共有している．しかし，発達したこの複雑なシステムは，利便性とともにセキュリティの脆弱性を生み出している．その脆弱性に関して，複数の具体的な攻撃手法が示されてきた[2,3,4]．これらの攻撃は，車外ネットワークから車内の通信ネットワークへなりすましフレームを注入して，車載ネットワーク内の ECU を不正に制御し，安全な運転に反した問題を引き起こす危険性がある．このためのセキュリティ対策としては侵入検知システムが知られており，特に未知のサイバー攻撃を検知可能なアノマリ型が多く研究されてきている．これらの検知システムでは，なりすましフレームを周期的な正常フレーム間に挿入することで発生する，送信周期やコンテキストデータの急激な変化を検出する．しかし，近年示されたより巧妙な手法として，攻撃により周期的に送信される正常フレームを停止させ，その後に，攻撃用のなりすましフレームを送信する手法が提示された[5]．このような攻撃の検知は従来の車載侵入検知方式では難しい．そのため，新たな検知手法の検討が必要である．

その手法の一つとして，攻撃の判断に監視対象のデータのみならず関連したデータを用いる手法が提案されている[6]．しかし，従来の手法では，コンテキスト内のデータの構造や，そのデータ間の関係性の情報が必要であった．

本論文は，このようなデータ構造，および関連性が不明な場合においても，なりすましフレームを用いた攻撃を検知可能な手法を提案する．具体的には，CAN 通信ログから相関が高いデータを抽出し，データセットを作成する．そして，そのデータセットを用いて CAN 通信に対して異常検知を行う．

†1 住友電気工業株式会社 〒554-0024 大阪市此花区島屋 1-1-3
SUMITOMO ELECTRIC INDUSTRIES, LTD. 1-1-3, Shimaya, Konohana-ku, Osaka, 554-0024, Japan

### 1.1 本論文の構成

本論文を以下の通り構成した．第 2 節では，CAN 通信とそのセキュリティ脅威について紹介し，第 3 節では，現在の車載ネットワークのセキュリティ技術を概説し，第 4 節では従来の車載侵入検知システムの課題を示す．そして，第 5 節で今回の提案手法を説明する．第 6 節では，本提案手法を実際の車両データを用いて評価し，本論文のまとめと今後の展開を第 7 節にて説明する．

## 2. CAN 通信とそのセキュリティ脅威

### 2.1 CAN のトポロジ

車載ネットワークである CAN には，CAN プロトコルと CAN-FD プロトコルの二種類の形式が存在する．CAN プロトコルは ISO 11898-1(2003)で標準化され，CAN-FD プロトコルは，その改良版として ISO11898-1(2015)にて標準化された[7,8]. バス型やスター型のトポロジに複数の ECU を接続し，その中で通信調停により送信権を得た ECU が，CAN では最大 8 バイト，CAN-FD では最大 64 バイトのコンテキストをブロードキャスト送信する．これにより，制御システム用途での低遅延なメッセージ送信を可能としている．

### 2.2 送信パターン

車載ネットワークにて伝達する CAN フレームは，送信周期において 2 種類に大別される．周期的に送受信されるフレームは，速度，エンジン回転数，アクセル開度といった車両において重要な制御情報の伝送を担う．もう一つは，非周期に送信されるフレームであり，ドアの開錠・施錠といった突発的なイベントの送信に使用される．

### 2.3 CAN の構造的なセキュリティ脅威

Koscher らは CAN が抱える 3 つの構造的な脆弱性を指摘した[9]．(1) ネットワーク上の制御情報を容易に解析可能 (2) なりすましフレームの挿入が容易 (3) Denial of Service（DOS）攻撃に弱い．なりすましフレームの挿入や DOS 攻撃は，CAN バスに接続された攻撃用の ECU を介して実行される．

## 3. 車載ネットワークのセキュリティ対策

### 3.1 従来のセキュリティ対策

車載ネットワークのセキュリティ対策は，2 つに分類できる．(1) セキュア通信：ネットワークプロトコルにおけるセキュリティ対策の実施．(2) 侵入検知システム：ネットワークプロトコル上で動作し，アプリケーションやネットワークにおける不審な動作の検出を行う．

### 3.2 侵入検知システム

#### 3.2.1 技術分類

侵入検知システムは，その設置形式の違いからネットワーク型とホスト型の 2 つに分類される[6]．ホスト型のセキュリティ対策機器は 1 つのノードに設置されるため，システムや接続されたネットワークセグメントの監視を行うことが出来る．一方，ネットワーク型はすべてのネットワークセグメントが監視可能な位置に設置される．

ここで，現在の車両内部では，70 を超える ECU を複数のネットワークセグメントに接続した構造を取っている．そのため，ネットワーク型侵入検知システムを適用することで, 多数の ECU を少ない侵入検知システムにより監視することが出来る．

#### 3.2.2 侵入検知システムの分類

侵入検知システムは，シグネチャー型とアノマリ型の 2 種類に大別される[11]．これらは，監視対象の不審な動作から，システムへの攻撃信号の侵入を検知する．シグネチャー型は，監視対象と既知の誤った使用例が一致した場合に検知する．一方，アノマリ型は，監視対象の異常な動作を，正常な動作からの逸脱として検知する．このため，アノマリ型は, 監視対象の正常な動作を学習する必要がある．

一般的に，シグネチャー型よりもアノマリ型は誤検知率が高い．しかし，車両の運用期間が長く，運用中に未知のサイバー攻撃の発生が予測されるため，アノマリ型が効果的な手法と考えられる．

#### 3.2.3 従来の車両侵入検知システム

以上のことから，車載ネットワークの侵入検知システムにおいて，監視の効率からネットワークベースが，システムの潜在的な脆弱性に備えるためにアノマリ型が多く研究されている．

Larson らは，仕様ベースの侵入検知を CANopen2.1 に適用した[12]．ここでは，CANopen2.1 プロトコルで定義されるルールから逸脱した場合に検知が行われる．

また，周期性のフレームに対して通信周期を監視する手法が提案されている．Otsuka らは，CAN-ID 別に最大受信遅延を学習することで，プロトコルによる送信周期のゆらぎに対応している[13]．また，揺らぎを確率密度関数として学習し、監視を行う手法も提案されている[14]．

さらに，エントロピーベースの検知手法を Müter らが提案している[15]．ここでは，フレームの通信周期や，通信フレームのデータフィールドに含まれるコンテキストデータの時間変動からエントロピーを算出し，監視を行った．

Markovitz らは，CAN フレームのデータフィールドに含まれるコンテキストデータの変動範囲を監視する方式を提案した[16]．この変動範囲は，実際に車両を走行させた際の実効的なコンテキストデータの可変範囲を使用している．

Wasicek らは，情報の物理的意味に基づいた検知手法を提案した[17]．車両物理モデルから，車両内や車両外から収集した情報を用いて監視対象のコンテキストデータを算出し，監視を行った．

## 4. 従来の検知システムの課題

### 4.1 なりすましフレームの挿入と課題

ここで，なりすましフレーム挿入における 2 つの攻撃モデルを定義する．一つ目のモデルは，Shared Bus モデルである．送信 ECU によって正常なフレームが送信される中，攻撃者は攻撃 ECU から，なりすましフレームを挿入する．このため受信 ECU は, なりすましが挿入された CAN-ID について，正しいフレームとなりすましフレーム両方を受信する．二つ目のモデルは，Occupied Bus モデルである．攻撃者は正常な CAN フレームの送信を停止した後, 攻撃 ECU からなりすましフレームの挿入を行う．このため，受信 ECU は，なりすましが挿入された CAN-ID について，なりすましフレームのみを受信する．

### 4.2 侵入検知方式の課題

攻撃者は Occupied Bus モデルにより，通信周期の変動を起こさずに，なりすましフレームを挿入できるため，通信周期を監視する 4 種類の従来方式では，攻撃の検知が困難である[12,13,14,15]．また，攻撃者は Occupied Bus モデルにより，コンテキストデータの時系列での大きな変動を発生させることなく偽造データを挿入することが可能であり，コンテキストデータの時系列変化を監視する従来方式でも，この攻撃の検知は容易ではない[15]．

コンテキストデータの値を監視する Markovitz や Wasicek らの従来方式を用いる場合，Occupied Bus モデルによる攻撃であっても検知が可能である．しかし，これら 2 種類の方式にはそれぞれ別の課題がある．攻撃者がコンテキストデータの可変範囲内で攻撃を行う場合，検知が難しい[16]．また，コンテキストデータの値を物理モデルにより監視する方式では，物理モデルの準備のためのコストが高いことが課題となる[17]．

一方，関連データを利用した手法が提案されている[6]．この手法では，監視対象のコンテキストデータと，監視対象と関連するコンテキストデータに対して機械学習を適用し，車両状態の予測を行うモデルを生成し，モデルの予測をもとに正常と判断されたデータのみを選択し続ける手法である．より具体的には，車速を監視対象とした場合に，アクセル，ブレーキのデータを使用して次のステップの車

速を推定する．そして，推定から外れたデータを攻撃として検知する．この手法を検証した結果，監視対象のデータのみを使用して推定する場合よりも，関連するデータを使用した場合の方が，推定精度が向上し，Shared Bus 型の攻撃への対処が可能となっている．

上記のように，関連するデータを使用することは，推定精度の向上，および，攻撃に対する頑強性を高める効果がある．また，Shared Bus 型の攻撃に対する報告例だが，Occupied Bus 型においても，同様の対処が期待される．しかし，その利用には，以下の 2 つの制限が存在する．（1）分割：CAN に含まれるコンテキストの構造，つまり，8 バイトのデータの区切り，および，その意味が既知であること．（2）データの関連性：コンテキストデータ間の関連性が既知であること．（1）に関しては，CAN に含まれるコンテキストのデータは，メーカ，車種，モデル毎にその仕様が異なる．また，（2）に関しても，ECU の増加に伴い，それらの関係性，依存性が複雑化すると予測され，その関係性を把握し，事前に異常検知システムに入力することは，負担となることが予測される．

## 5. 提案手法

### 5.1 アイディア

前述したように，関連データを使用した検出手法は有効と考えられるが，そのためには，CAN コンテキストの仕様を把握する必要があった．本章では，これらの問題を克服するためのアイディアと，提案手法の構成の説明を行う．

- コンテキストの分割：車両それぞれのコンテキスト構造毎に侵入検知システムを対応させることは，システムの適用範囲に制限を与える．そのため，既に CAN メッセージからコンテキストの構造を抽出する手法が提案されている[16]．今回は，より一般的な手法を志向して，全てのコンテキストをその内容によらず，1 バイトに分割する．1 バイト単位で分割することにより，1 ビットから数バイトに及ぶデータ幅のある複数のコンテキストをまとめて処理することが可能となる．
- データの関連性：物理的に関連するデータを使用するには，スピード，アクセル，ブレーキ等の様に，データ間の明確な関係性が既知の場合に制限される．そこで，キャプチャした正常な CAN のコンテキストから，関連するコンテキストデータを自動的に抽出する．ここでは，コンテキストデータの時系列的変化の特徴から，関連データを抽出する．

以上のアイディアを用いて，提案手法では事前にコンテキストデータの構造や，データ間の関連性の情報を必要としない，CAN 通信全体への適用性を持った汎用的な攻撃検知手法を提案する．

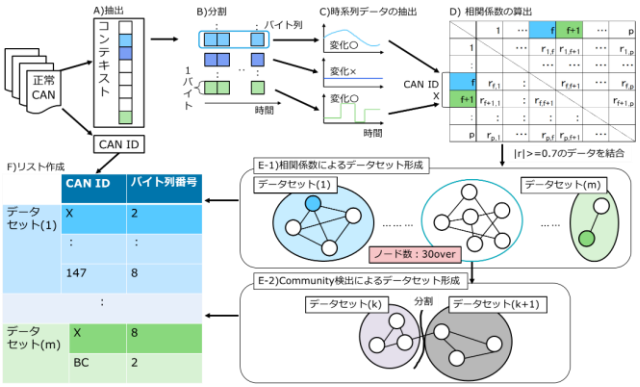


図 1　データセット作成手法

### 5.2 概要

提案する侵入検知システムは，下記の 3 ステップにて実行される．

**(1) データセット作成**

正常な CAN フレームのログをバイト単位で分割する．分割されたバイト列から関連するデータを抽出し，データセットを作成する．このとき，データセットに含まれるデータの CAN-ID とバイト列番号のリストを作成し，後のモデル選択に使用する．

**(2) 学習**

作成されたデータセットを用いてモデルの学習を行う．データセット 1 組に対して 1 つの推定モデルを生成する．学習した推定モデルは，検知に使用される．

**(3) 検知**

推定モデルを用いて，検査対象の CAN フレームの異常検知を行う．検査対象の CAN フレームと，推定モデルの予測値を比較し，異常検知を行う．検知結果に従って，モデルから正常と思われる CAN フレームを生成する．

以下に，より詳細な各ステップの処理手順を説明する．

### 5.3 データセット作成

この節では，データセット作成で行われる車両から取得した CAN フレームのログから，コンテキストの分割と類似データの発見，データセットの作成について説明を行う（図 1）．

**A）分離：**CAN フレームから CAN-ID とコンテキストを分離する．

**B）分割：**ログのコンテキストすべてを 1 バイト単位で分割する．そして，同一の CAN-ID とバイト列番号を持つバイトを時系列に並べ，バイト列を作成する．図 1 の B）では，CAN-ID が X のコンテキストをバイト単位（水色，青，緑色）で分割し，各バイトデータを時刻順に整列させた．1 バイトの時系列データが，バイト列を形成する．

**C）時系列データの抽出**：作成したバイト列の内，バイト列の値が時間変化するものを抽出する．図 1 の C)で示された，バイトの時系列変化のグラフの内，時系列変化している青色と緑色が時間変化するバイト列として抽出される．

抽出した時間変化するバイト列を用いて，以下の D)相関係数の算出を行う．

**D）相関係数の算出**：バイト列からデータセットを作成する．そのために，バイト列間の類似性の尺度が必要となる．今回は，時系列変化するデータ間の類似性を定量的に評価する指標として，相互相関係数を用いる．相互相関係数は，式（1）で表せる．

$$r_{xy}=\frac{\sum_{i=1}^{n}(x_i-\mu_x)(y_i-\mu_y)}{\sqrt{\sum_{i=1}^{n}(x_i-\mu_x)^2}\sqrt{\sum_{i=1}^{n}(y_i-\mu_y)^2}} \quad (1)$$

ここで，$r_{xy}$ はバイト列 $x$，$y$ の相互相関係数，$x_i$，$y_i$ はそれぞれのバイト列の $i$ 番目の要素，$\mu_x$，$\mu_y$ は $x$，$y$ の平均であり，$n$ はバイト列のサンプリング数である．

一般的に二つの CAN フレームの発生頻度は異なり，そのままでは相互相関係数を得られない．そこで，サンプリング数の補正を行う．このとき，サンプリング数の多いバイト列に少ない方のバイト列を合わせることで，時系列変化の特徴が減少することを防ぐ．

加えて，バイト列の値が 0 から 1 に収まるようにデータを正規化する．その後，ログから得られた全ての時系列変化のあるバイト列間で相互相関係数を算出する．計算された相互相関係数の一覧を模式的に図 1 の D)に示す．得られた相関係数から，以下の 2 段階の手順でデータセットを作成する．

**E-1) 相関係数によりデータセットの作成**

バイト列間の関連性を，バイト列で構成されるネットワークとして表現する．まず，各バイト列をネットワークの構成要素（ノード）（図 1：E-1）の丸記号）とする．それらノードの中で，相関係数の絶対値が 0.7 以上のノードのみを接続する．接続されたバイト列の集合をグループとする．これにより，全ての時間変化するバイト列の集合から，関連性の高いバイト列のグループが形成される．一つのグループを構成するノード数が 30 以下の場合，そのグループをデータセットとする．データセットは，後の学習や検知においてモデルへの入力単位となる．一方，グループ内のノード数が 30 以上の場合は，下記の E-2）に移行する．

**E-2) Community 検出によるデータセットの作成**

30 以上のバイト列で構成されたグループは，さらに分割する．分割の手法として，ネットワークの密集性（モジュラリティ）に注目したコミュニティ検出を用いる[18]．コミュニティ検出により，ノード数が 30 以下となるまで再帰的にグループを分割する．30 以下となった段階で，分割されたグループをデータセットとする．

データセットのノード数を制限する目的は，データセットを構成するバイト列の増加により，後の学習において多大な時間を消費することや，関連性が低いデータがデータセットに含まれるのを防止するためである．

**F)リスト作成**：データセットを作成する際には，データセットの構成リストとして，データセットに含まれる CAN-ID とバイト列番号のリストも合わせて作成する．

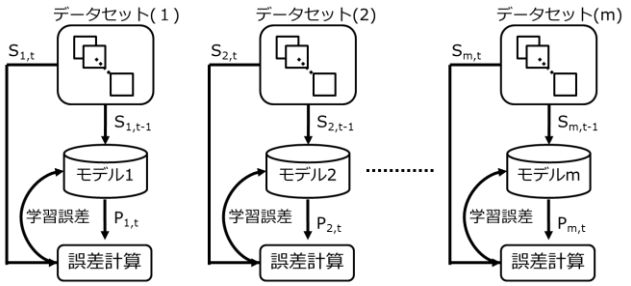


図 2　提案手法の学習モデル

**5.4 学習**

この節では，作成したデータセットを用いて，推定値を予測するモデルを生成する手法を説明する．

作成する推定モデルは，過去のデータセットの変動から，データセットの予測を行う．ここで予測にあたり，以下の理由からデータの時系列的変化に着目した．

Markovitz らは，コンテキストデータを，Multi-Value，Counter，Constant，Sensor に分類した[16]．ここで，Constant は，定数域のため推定対象から除外する．センサーの計測値である Sensor は，物理量であるため，その変化が連続的である．また，Counter は変化が単調であり，時間に対する規則性を持つ．車両の状態を表す Multi-Value も，走行状態において，その変化にはある程度の規則性が期待される．つまり，CAN の時間変化するコンテキストは，時系列的な特性を多く含むと考えられる．

この様な連続性に特徴を持つデータを処理するのに適した手法として，機械学習の一種である Recurrent Neural Network (RNN)がある．一般的に機械学習で使用されるニューラルネットワークは，各入力データを独立して扱うため，それまでの入力履歴などを反映することが出来ない．一方，RNN は，自身の出力を再入力することで連続的なデータとして扱うことが可能である．しかし，RNN の性質上，10 ステップ以前の状態を反映できないことが知られている．これを CAN に置き換えると，10 ミリ秒毎に送信する CAN フレームは，100 ミリ秒程度しか反映されないこととなる．これは，車両における運動の時間スケールとしては十分とは言えない．そこで，より長期的な依存関係を学習することが可能な Long Short Term Memory (LSTM)ネットワーク[19]を本提案手法では使用した．

図 2 に示すように，現在の 1 ステップ前のデータセット（$S_{i,t-1}$）から現在のデータセット（$P_{i,t}$）を予測させ，その推定値と実際に取得された現在のデータセット（$S_{i,t}$）との学習誤差が最小となるようにモデル $i$ を最適化する．これにより，1 ステップ前のデータセットから，予測値を出力するモデルを生成する．図 2 に示すように，この推定モデルは，データセットごとに作成される．つまり，m 個のデ

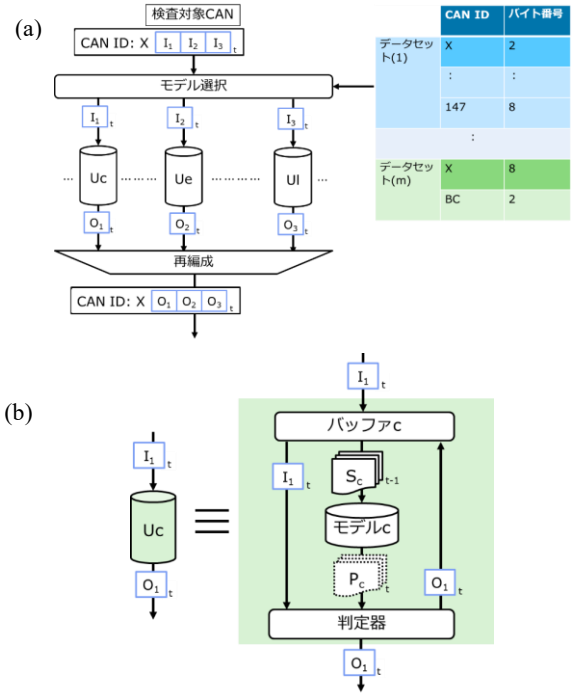


図 3　検知方法の構造り

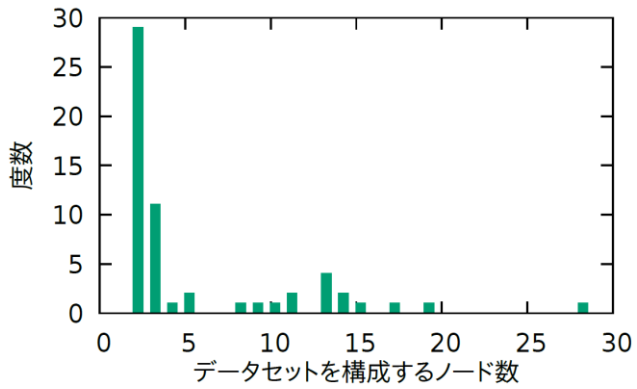


図 4　作成されたデータセットのサイズ分布

ータセットが作成された場合，m 個のモデルを学習し，生成する．

### 5.5 検知

この節では，生成した推定モデルを用いて車両内の CAN 通信を監視し，異常を検知する手法を説明する．

CAN バスから受信した CAN フレームはモデル選択に入力される．モデル選択の内部で，入力された CAN フレームは，CAN-ID とコンテキストに分離される．また，コンテキストは，さらに 1 バイト単位に分割され，データセット作成で得たリストを用いて，対応した推定モデルを持つユニット（U）に送信される．図 3 (a) では，3 バイトのデータ（$I_1$，$I_2$，$I_3$）を持つ CAN-ID を例に示している．それぞれのデータに対応した 3 つのユニット（Uc，Ue，Ul）に送信される．そのため，1 つの CAN-ID に対して最大 8 個のユニットが稼働する．各ユニットは，ユニット出力（$O_1$，$O_2$，$O_3$）を出力し，CAN フレームに再編成される．

ここで，ユニットの内部処理について図 3 (b)で説明する．ユニットは，1 ステップ前のデータセットを保持するバッファ（上段），1 ステップ前のデータセットから現在のデータセットを予測する推定モデル（中段），推定されたデータセットと新しい CAN のバイト列を比較し，ユニット出力を決定する判定器（下段）で構成されている．モデル選択からバイトデータ($I_1$)を入力されたバッファは，保持していた現在から 1 ステップ前のデータセット（$S_{c,t-1}$）を推定モデル（モデル c）に入力する．推定モデルは，予測した現在のデータセット（$P_{c,t}$）を出力する．推定された $P_{c,t}$ は，判定器に入力される．入力されたバイトデータと，推定したデータセット内のバイトデータの差分から，攻撃を閾値判断する．閾値よりも小さければ，正常と判断し，入力されたバイトデータ（$I_1$）をそのまま出力する．一方，閾値よりも差分が大きければ攻撃と判断し，推定したバイトデータを出力する．つまり，判定器は入力バイトデータ，推定バイトデータ，どちらを $O_1$ として出力するか決定する．また，バッファが保持しているデータセットは判定器の出力により更新する．そのため，攻撃を受け，入力値が異常と判断された場合，バッファには，推定値が保存される．そのため，攻撃を連続的に受信した際には，推定値から推定を重ねる手法である．

## 6. 提案手法の評価

本章では，提案手法の有効性を示すために実際の車両から取得した CAN フレームへ適用した結果を示す．まず，データセットの作成，そして学習結果を示す．また，生成した推定モデルを用いた Occupied Bus 型のなりすまし攻撃への応答と，全 CAN ログへの適用結果から提案手法の汎用性を評価した．

### 6.1 データセットの作成

本稿では，テストコースを約 550 秒間，平均時速 80 km にて走行した車両の全 CAN フレームを用いた．CAN の形式は，最大 8 バイトのコンテキストを含む CAN プロトコルである．CAN フレームには，約 200 個の CAN-ID が含まれ，それらすべてのコンテキストを 1 バイト毎に分割した結果，約 450 バイト列に時間変化が確認された．このバイト列を用いて学習に使用するデータセットを作成した．

その結果，61 個のデータセットが作成された．データセットに含まれるバイト数の分布を図 4 に示す．このデータセットの作成の段階で，データ間の相互相関係数が 0.7 未満なため，データセットを構成できないバイト列が，約 30%程度存在した．以下では，作成された 61 個のデータセットを用いた学習について説明する．

### 6.2 推定モデルの学習と推定精度の評価

本節では，推定モデルの学習条件と，作成したデータセットを用いた学習結果を示す．

学習において，作成したデータセットを学習用データと

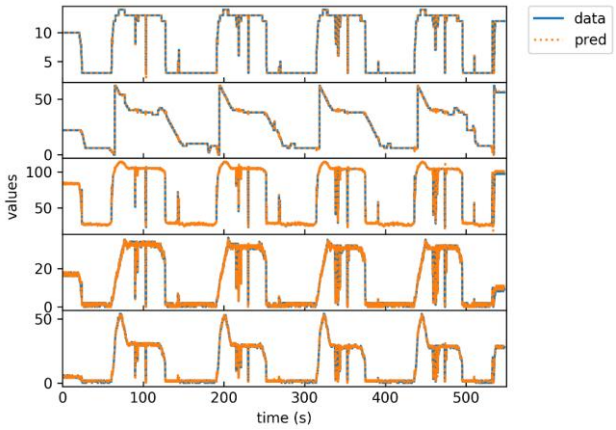


図 5　推定値と観測されたデータセットの比較

検証用データに分割した．学習データである 385 秒間の走行データを用いて推定モデルの最適化を行った．推定モデルの構造は，入力層，LSTM ユニットにより構成された隠れ層，出力層となっている．入力層と，出力層は，データセットに含まれるバイト列の数と同じ個数の全結合ユニット，LSTM は，50 ユニットで構成されている．推定値と，正解のデータセットの誤差は，平均二乗誤差関数（MSE）により評価し，モデルの学習を行った． 学習により生成した推定モデルの推定精度を確認した．推定モデルが予測したデータセットの推定値と，実際のデータセットの出力の一例を図 5 に示す．データセット は，このように，時間変化の類似したバイト列により構成されている．表示しているデータセットにおいて，推定値（オレンジ色：点線）がデータセットの出力（青色：実線）をほぼ再現していることが確認できる．したがって，推定モデルは，実際のデータセットを予測可能なことが示された．

### 6.3　なりすまし攻撃に対するデータセット使用の効果

続いて，学習，および検知時に使用するデータの違いによる，推定モデルの挙動を比較するために，以下の条件で推定モデルを構築した（図 6）．

推定モデル A：推定対象のデータのみで学習

推定モデル B：本提案手法のデータセットを用いて学習

図 6 の上段に示す推定対象のデータのみで学習した推定モデル A と，下段のデータセットを用いて学習した推定モデル B を作成した．図 6 において，推定対象のデータは，推定モデル A，B ともに水色で示されている．

これらのモデルへの攻撃として，推定対象の CAN フレームが停止させられ，停止された CAN-ID の不正な信号が入力される Occupied Bus 型のなりすまし攻撃を用いた．なりすまし信号として，一定時間の間，定数を入力した．定数は，攻撃の直前の入力値を使用した．図 6 において攻撃は，検知に用いる t=1，2 の区間のデータ（赤色）として示している．また，推定モデル B の場合は，検知においてデータセットを入力するが，攻撃の区間における推定対象の

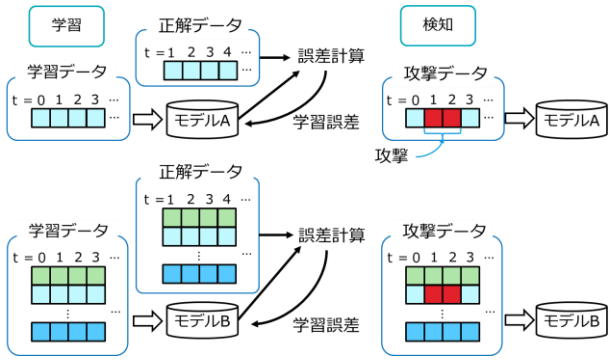


図 6　Occupied Bus 型のなりすまし攻撃に対するデータセットの効果の検証手法

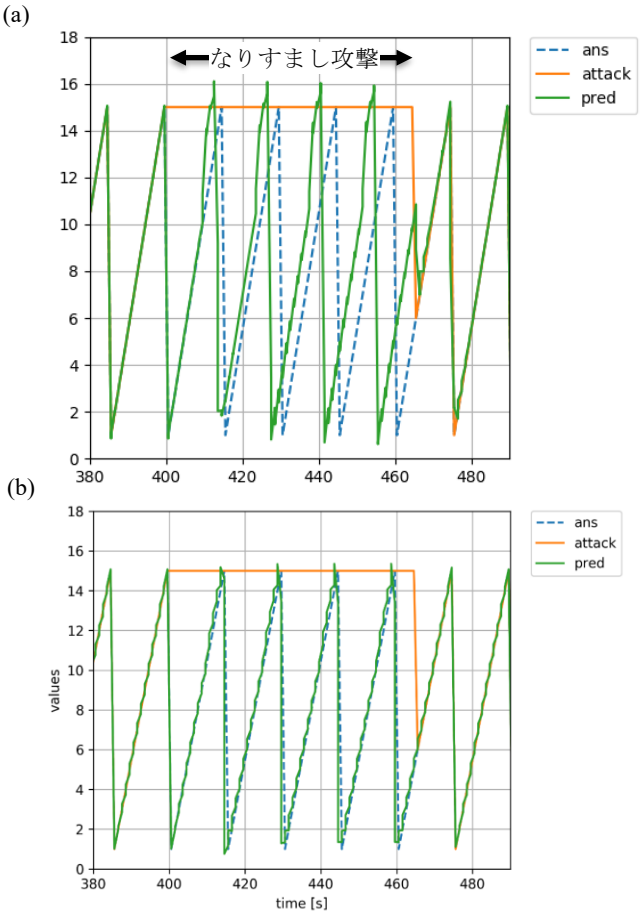


図 7 Occupied Bus 型なりすまし攻撃に対する挙動．
(a)推定データのみを使用，
(b)データセットを使用した場合

以外のデータセットのバイト列（図 6 下段の緑色と青色のバイト列）は，正常な値を取り続けることとした．

推定モデル A と推定モデル B に対して，上記のなりすまし攻撃を行い，そのときの推定モデルの出力を検証した．図 7 にそれぞれの推定モデルの結果を示す．図 7 の青色の点線（ans.）は正常な CAN データ，オレンジ色の実線（attack）がなりすまし攻撃，緑の実線（pred.）が各モデルの推定値を表している．図 7 (a)の推定対象のデータのみを学習，検査データとした推定モデル A では，なりすまし攻撃を受け

た 400 秒から 465 秒の範囲において，推定値が正解の波形を再現出来ていない．一方，攻撃の値にも追従していないため，なりすまし攻撃を検知し，入力された値ではなく，推定値を出力していることが確認できる．攻撃を始めた直後の 400 秒から 410 秒の間は，正解の値を推定できているが，自身の 1 ステップ前の推定値を次のステップの推定に使用し続けることで，推定の累計誤差が蓄積した結果，本来の時間変化から逸脱したと考えられる．

一方, 図 7 (b)のデータセットを用いて学習と推定を行ったモデル B では，正常な CAN データが受信不可能な攻撃範囲においても，推定値が正解値をよく再現している．本推定手法も攻撃中は，自身の予測値をもとに推定を重ねている．しかし，データセットに含まれる他の正常なバイト列が，モデル B における推定誤差の蓄積を抑えた結果，正常な推定が維持されたと考えられる．

以上のことから，本提案手法においてデータセットを利用することで，Occupied Bus 型なりすまし攻撃を検出可能であり，また，攻撃を受け，正常な CAN コンテキストが受信できない場合においても，正常値に近い推定が可能であることを確認した．

### 6.4 提案手法の汎用性

提案手法の汎用性を調査するために, CAN フレーム全体へ適用した場合の推定精度を検証した．推定モデルの予測と実際の入力間の誤差から, 推定モデルの精度を評価する．データセット作成の際に述べた様に，車両から取得した CAN データから，約 70%のバイト列のデータを取得し，61 個のデータセットに再編成した．それぞれのデータセットに対応する推定モデルを生成し，バイトデータの変動範囲に対する平均二乗誤差の割合を推定誤差として用いた．バイト列に対する推定モデルの推定誤差を図 8 に示す．最も頻度が大きい推定誤差は，1~2%であり，全体的にも低い推定誤差の範囲にデータが集約されていることが見て取れ，提案手法の有効性を示している．推定誤差が 10%以下のバイト列が，全体の約 88%だった．

一方, 12%にあたる 41 個のバイト列が 10%以上の推定誤差を示した．これらの推定誤差が大きいバイト列は，データの時間的規則性が低く，意味のある CAN コンテキストの断片，あるいは，複数のデータが 1 バイトに集合した結果と考えられる．

## 7. まとめと今後の課題

本稿では，コンテキストを一律に分割し，分割データの類似性からデータセットを自動作成する，汎用的な CAN 通信に対する異常検出手法を提案した．提案手法を実車両における CAN のログに対して適用し，時間変化するバイト列の 7 割からデータセットを抽出した．推定対象のデータのみの場合より，データセットを用いることで Occupied Bus 型のなりすまし攻撃を受信中の予測精度が向上した．

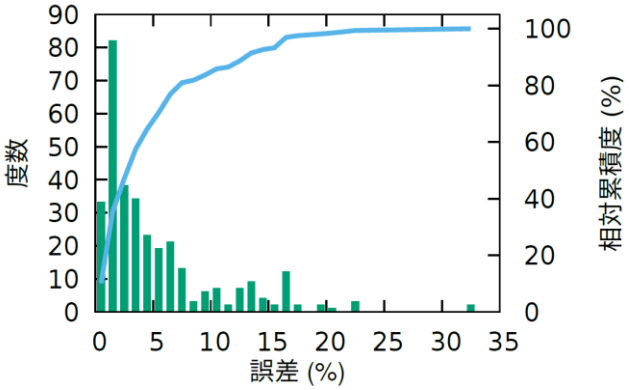

図 8 推定誤差の度数分布（緑棒：左軸）と
相対累積度数分布（青線：右軸）

そして，取得した CAN ログに対して網羅的に提案手法を適用した結果, 全体の約 88%において 10%以下の推定誤差が得られ，提案手法の汎用性が確認された．

さらに, 本手法は，コンテキストの意味にかかわらず， 1 バイト毎に分割して処理を行うため，その適用範囲は今回適用した CAN 通信に限定されず，他の時系列変化を持つ通信手法に応用可能と考えられる．

今後は，バイト列間の類似性の尺度や，類似性を元にデータセットを作成する手順を最適化する．また，様々な環境における走行データを収集する．異なる環境におけるデータを蓄積することによって，新たなデータ間の関連性が発見され，データセットを構成するデータの増加と，推定モデルの精度向上が予測される．そして，最適化された推定モデルに対して，さまざまな攻撃パターンを用いた性能評価を行う．

一方，データセットに所属しないデータが存在した．データには，本質的にデータセットを構成することが困難なものもが存在すると思われる．それは，ドアロック，シートベルト，ワイパー等の突発的な入力により一意に値が決定するデータである. これらのイベント送信型のデータは，提案手法での攻撃検知が困難と思われ，その対応は今後の検討課題となる．

## 参考文献

[1] Charette, R.N., “This Car Runs on Code.” , http://spectrum.ieee.org/transportation/systems/this-car-runs-on-code, accessed July. 2017. car-runs-on-code, accessed July. (2017).
[2] Checkoway, S., McCoy, D., Kantor, B., Anderson, D. et al., “Comprehensive Experimental Analyses of Automotive Attack Surfaces,” Proceedings of the 20th USENIX Conference on Security, (2011).
[3] Miller, C., and Valasek, C., “Adventures in Automotive Networks and Control Units,” presented at DEF CON 21, August (2013).
[4] Miller, C., and Valasek, C., “Remote Exploitation of an Unaltered Passenger Vehicle,” presented at DEF CON 23, August (2015).
[5] Miller, C., and Valasek, C., “Advanced Can Injection Techniques for Vehicle Networks,” presented at DEF CON 24, August (2016).
[6] 亀岡良太，吉田康太，西村勇人，汐崎充，久保田貴也，白畑

正芳，藤野毅：車載ネットワークにおける深層学習を用いた時系列解析による攻撃検知手法，SCIS2018 (2018).
[7] International Organization for Standardization, “Road vehicles-Controller area network (CAN) - Part 1: Data link layer and physical signaling,” ISO11898-1, Rev. (2003).
[8] International Organization for Standardization, “Road vehicles-Controller area network (CAN) - Part 1: Data link layer and physical signaling,” ISO11898-1, Rev. (2015).
[9] Koscher, K., Czeskis, A., Roesner, F., Patel, S. et al., “Experimental Security Analysis of a Modern Automobile,” 2010 IEEE Symposium on Security and Privacy, (2010).
[10] Scarfone, K.A., and Mell, P.M., “Guide to Intrusion Detection and Prevention Systems (IDPS),” Special Publication (NIST SP) - 800-94, February, (2007).
[11] Chandola, V., Banerjee, A., and Kumar, V., “Anomaly Detection: A Survey,” ACM Computing Surveys, 41(3), (2009).
[12] Larson, U.E., Nilsson, D.K., and Jonsson, E., “An Approach to Specification-Based Attack Detection for In-vehicle Networks,” 2008 IEEE Intelligent Vehicle Symposium, (2008).
[13] Otsuka, S., Ishigooka, T., Oishi, Y., and Sasazawa, K., “CAN Security: Cost-Effective Intrusion Detection for Real-Time Control Systems,” SAE Technical Paper 2014-01-0340, (2014).
[14] Hamada, Y., Inoue, M., and Horihata, S. et al., ”Intrusion detection by Density Estimation of Reception Cycle Periods for In-Vehicle Networks: A Proposal,” presented at the 14th escar Europe Conference, November 16-17, (2016).
[15] Müter, M., and Asaj, N., “Entropy-Based Anomaly Detection for In-Vehicle Networks,” 2011 IEEE Intelligent Vehicle Symposium (IV), (2011),
[16] Markovitz, M. and Wool, A., “Field Classification, Modeling and Anomaly Detection in Unknown CAN Bus Networks,” presented at the 13th escar Europe Conference, November 11–12, (2015).
[17] Wasicek, A., Pesé, M., Weimerskirch, A., and Burakova, Y. et al., “Context-aware Intrusion Detection in Automotive Control System,” presented at the 5th escar USA Conference, USA, June 21–22, (2017).
[18] Newman, M. E. J. and Girvan, M.: Finding and Evaluating Community Structure in Networks, Physical Review E, Vol. 69,026113, 1–15, (2004)
[19] Hochreiter, S., and Schmidhuber, J., “Neural Computation” Vol 9, No8, 1735-1780, (2015).