修士論文

車載ネットワークにおける遅延時間の高時間分解能観測に基づく送信元識別手法

大平 修慈
奈良先端科学技術大学院大学
先端科学技術研究科
情報理工学プログラム

主指導教員: 藤川 和利 教授
情報基盤システム学 研究室（情報科学領域）

令和 2 年 3 月 13 日提出

本論文は奈良先端科学技術大学院大学先端科学技術研究科に
修士 (工学) 授与の要件として提出した修士論文である。

大平 修慈

審査委員：

藤川 和利 教授	（主指導教員）
林 優一 教授	（副指導教員）
新井 イスマイル 准教授	（副指導教員）

車載ネットワークにおける遅延時間の高時間分解能観測に基づく送信元識別手法*

大平 修慈

内容梗概

インターネットに接続する自動車が増加し，自動車内の車載ネットワークである Controller Area Network (CAN) へのサイバー攻撃が深刻な問題になっている．CAN は Electronic Control Unit (ECU) 間の通信に使用される車載ネットワークプロトコルであるが，CAN のデータフォーマットには送信元を識別する ID や認証する仕組みがないため，攻撃者から送信された不正なメッセージを区別することができない．したがって，CAN メッセージの送信元識別手法を確立することが必要となる．既存研究では，安価な計測デバイスを用いて，CAN トランシーバの遅延時間を観測し，送信元の識別を行うが，各 ECU の遅延時間の差が計測デバイスの時間分解能より低い場合，送信元を正しく分類できない．そこで，本研究では，ECU の識別精度を向上させるため，Time-Digital Converter (TDC) を用いた遅延時間の高分解能観測に基づく送信元識別手法を提案する．提案手法で用いる TDC は，原子核実験やハドロン実験で用いられ，FPGA で実装することでオシロスコープ等と比べると比較的低コストで実現できる．さらに，送信元識別において重要な特徴量を明らかにするために Relief-F と呼ばれるアルゴリズムを用いて特徴選択を行う．FPGA およびマイクロコンピュータにより計測デバイスを実装し，提案手法の ECU の分類に関する評価を行った．評価結果から，従来手法では研究室内の CAN のプロトタイプと実車でそれぞれ平均正解率は 81.43％と 76.75％であるのに対し，提案手法では 99.67％と 95.94％となり，提案手法の有効性を示すことができた．

*奈良先端科学技術大学院大学 先端科学技術研究科 修士論文, 令和 2 年 3 月 13 日.

キーワード

自動車セキュリティ, Controller Area Network, 送信元識別, 侵入検知システム, 機械学習

Physical-Layer Identification Based on High-Resolution Observation of Delay-Time in In-Vehicle Networks*

Shuji Ohira

Abstract

Currently, due to the increase in the number of automobiles that connect to the internet, cyber-attack on Controller Area Network (CAN) is becoming a severe problem. CAN is one of the in-vehicle network protocols for communicating among Electronic Control Units (ECUs) and it is a de-facto standard of in-vehicle networks. CAN bus is simple and has several vulnerabilities such as unable to distinguish spoofing messages due to no authentication and no sender identification. Hence, identifying the sender node of the CAN frame is a challenging task. In previous work, a delay-time based method to identify the sender node has been proposed. This method can identify ECUs with an inexpensive device to avoid requiring costly equipment. However, if different ECU ' s delay-time have similar variations, this approach may not correctly classify legitimate ECUs because the time resolution to measure the delay time will be coarse. Therefore, we should focus on enhancing the accuracy of sender identification. In this thesis, we propose a sender identification method based on high-resolution observation of delay-time using Time-Digital Converter. We implement the experimental devices using FPGA and microcomputer to evaluate the proposed method for the identification of legitimate ECUs. The conventional method identifies ECUs with

*Master's Thesis, Graduate School of Science and Technology, Nara Institute of Science and Technology, March 13, 2020.

a mean accuracy rate of 81.43% in the CAN bus prototype and 76.75% in a real-vehicle. In contrast, the proposed method achieves an accuracy rate of 99.67% in the CAN bus prototype and 95.94% in a real-vehicle.

Keywords:

Automotive Seucrity, Controller Area Network, Physical-Layer Identification, Intrusion Detection, Machine Learning

目 次

図 目 次

表 目 次

1. はじめに

自動車や路線バスといった様々な車両がインターネットと接続され，カーシェアリングやライドシェア等といった新たなサービスの概念が誕生している．これらの影響により，移動 (Mobility) をサービスとして捉える Mobility as a Service (MaaS) が注目されている．MaaS により，ユーザはスマートフォン等から移動に関する経路検索から予約・支払いまでを一度に行えるようになることや，移動の効率化により都市部での交通渋滞や環境問題，地方での交通弱者対策などの問題の解決を行うことが期待されている．

このような利便性のためにインターネットに接続する車両が増加する一方で，自動車内のネットワークである Controller Area Network (CAN) [1] へのサイバー攻撃が懸念されている [2] [3]．Nie らは車載システムのブラウザと CAN が持つそれぞれの脆弱性を悪用し，自動車の様々な機能が制御可能であることを実証した [3]．これらの攻撃は CAN の脆弱性に起因しており，CAN に対するセキュリティ対策が急務になっている．CAN は Electronic Control Unit (ECU) 間の通信に使用される車載ネットワークプロトコルであり，事実上の標準になっている．また，CAN のデータフォーマットには送信元を識別する ID や認証する仕組みがないため，攻撃者から送信された不正なメッセージを区別できない．

そこで，不正なメッセージを防止する目的で Message Authentication Code (MAC) を付加して CAN メッセージを認証することが考えられるが，CAN のデータフィールドは最大で 8 byte しかないため MAC による認証は容易に適用できるものではない．さらに，いくつかの認証手法 [4] [5] では事前共有鍵が必要となるが，その鍵交換の方法については検討されていない．

一方で，侵入検知システム (Intrusion Detection System: IDS) は，暗号化・認証手法とは異なり，その有効性や CAN への適用性において優位性がある．CAN における IDS として，信号の物理的特徴 (電圧 [6] [7] [8] [9]，クロックのずれ [10]，信号の遅延時間 [11] [12]) に基づく IDS がある．これらの手法の 1 つに，論理値を CAN の差動信号に変換する IC である CAN トランシーバにおける信号の立ち上がり・立ち下がりの遅延時間に着目した送信元識別手法 [11] がある．この手法では，安価な計測デバイスを用いて遅延時間を観測し，送信元識別可能であるこ

とが確認されている．ただし，各 ECU の遅延時間の差が計測デバイスの時間分解能より低い場合，ECU を正しく分類できない．そこで，遅延時間の高時間分解能観測により ECU の識別精度を向上させることが期待できる．また，温度変化に対し各特徴量 (遅延時間) が変化する可能性があるため，温度変化に対しロバストな手法を検討する必要がある．

本研究では，Time-Digital Converter (TDC) を用いた遅延時間の高分解能観測に基づく送信元識別手法を提案する．提案手法で用いる TDC は，原子核実験やハドロン実験で用いられ，FPGA で実装することでオシロスコープ等と比べると比較的低コストで実現できる．FPGA およびマイクロコンピュータにより計測デバイスを実装し，遅延時間に基づく従来手法と提案手法の ECU の分類に関する評価を行った．比較手法として時間分解能 20 ns で CAN バスプロトタイプと実車環境で識別精度を評価したところ，平均正解率はそれぞれ 81.43%と 76.75%であった．一方で，時間分解能 154 ps の TDC を用いた提案手法の評価結果では，研究室内の CAN バスプロトタイプで 99.67%，実車で 95.94%の平均正解率となった．

本研究による貢献を以下にまとめる．

1. TDC を用いた遅延時間の高分解能観測に基づく送信元識別手法を提案した．さらに，提案手法では Relief-F と呼ばれる特徴選択アルゴリズムを用いて分類精度の向上を図った．実験結果から，従来手法 [11] における CAN メッセージの分類の平均正解率が 81.43 % であるのに対し，提案手法では平均正解率 99.67 % となることがわかった．

2. 提案手法における特徴抽出の計算量は電圧ベースの送信元識別手法と同程度であり，かつ，提案手法の特徴抽出で用いられるデータ数は電圧ベースの送信元識別手法よりも少ない．したがって，電圧ベースの送信元識別手法よりも提案手法は軽量な処理で特徴抽出までを実行可能であることがわかった．

3. 遅延時間に基づく従来手法では検討されていなかった温度変化に対する特徴量の Concept Drift を確認し，温度変化にロバストな手法を検討した．検討したロバストな学習モデルでは 30 °C から 45 °C の全てのテストデータに

対し，CAN メッセージの分類の平均正解率は 99%以上となった．

本稿の構成は以下の通りである．第 2 章では，現在の車載ネットワークの事実上の標準である CAN について述べる．第 3 章では，自動車セキュリティに関する研究を包括的にまとめ，従来の送信元識別手法の問題点等を述べる．第 4 章では，TDC を用いた遅延時間の高分解能観測に基づく送信元識別手法を提案し，提案手法を構成する 3 つのフェーズについて説明する．第 5 章では，第 4 章で提案した手法の実装に関して述べる．第 6 章では，提案手法の送信元識別精度および侵入検知精度に関する評価を行う．第 7 章では，評価の結果から考察を行い，提案手法の妥当性および今後の展望について議論する．第 8 章では，本稿のまとめを行う．

2. Controller Area Network

本章では，現在の自動車に最も普及している車載ネットワークである CAN について述べる．また，いくつかの研究で指摘されている CAN の脆弱性についてまとめる．

2.1 CAN の概説

CAN は車載ネットワークの事実上の標準であり，一般に図 1 に示すようなバス型のネットワークトポロジである．CAN に接続する ECU は Micro Controller

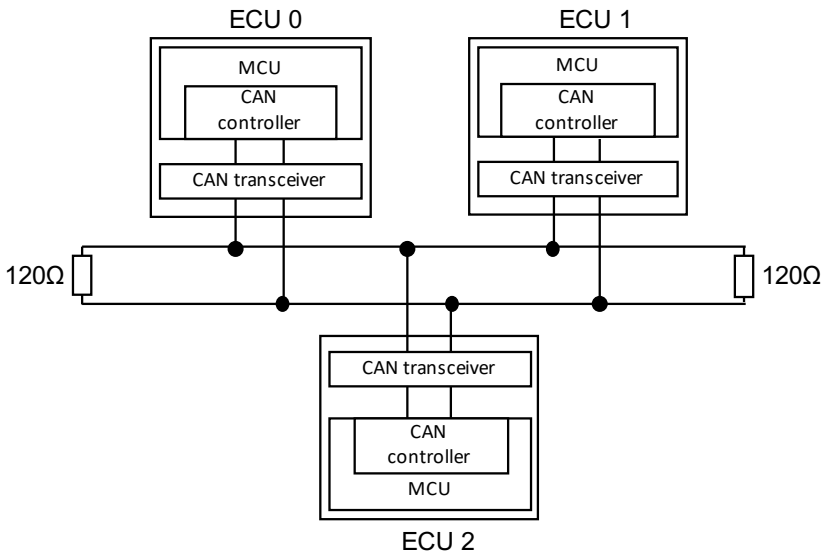
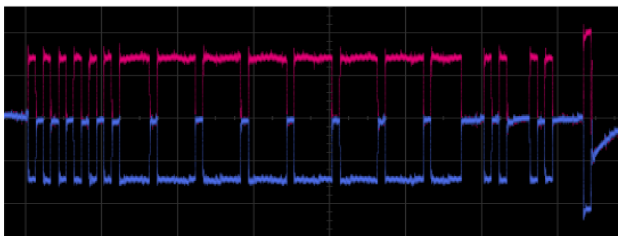


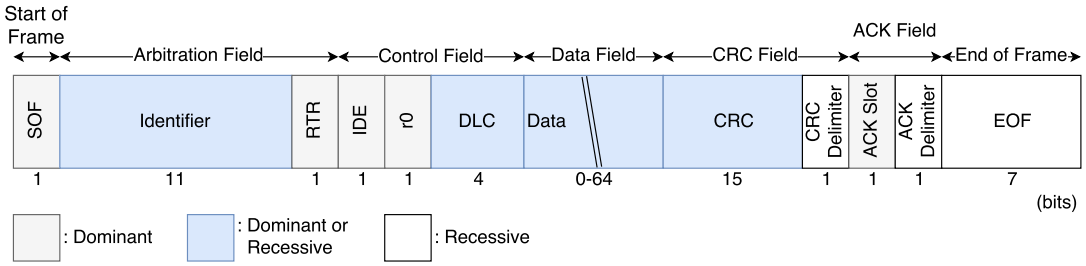
図 1: 典型的な CAN の構成

Unit (MCU)，CAN コントローラ，および，CAN トランシーバ から構成され，CAN コントローラ は CAN の仕様に従ってフレームを制御する．さらに，CAN トランシーバ は論理値 (0，1) を CAN の差動信号 (ドミナント，リセシブ) へ変換を行う．また，ISO 11898 によって高速 CAN 通信 (125 kbps ～ 1 Mbps) の仕様が規定されており，この仕様では最大 30 ノードを 40 m の最大バス長に接続できる．CAN は堅牢なノイズ耐性を実現するために，ツイストペアケーブルを用い

ている．ツイストペアケーブルはそれぞれ CAN-H，CAN-L と呼ばれ，ドミナントが送信されると，CAN-H には 3.5 V，CAN-L には 1.5 V の電圧がかかる．一方で，リセシブの場合 CAN-H と CAN-L は共に 2.5V となる．図 2 (a) に，CAN 信号の例を示す．前述の通り，CAN-H と CAN-L に電位差がある場合はドミナントであり，そうでない場合はリセシブを表している．複数のノードからドミナン



(a) CAN 信号の例



(b) CAN のデータフォーマット

図 2: CAN のデータフレーム

トとレセシブが同時に送信された場合，ドミナントが優先的に送信される．CAN ではこの特徴を利用して，複数のノードが同時にフレームを送信し，信号が衝突したとしても，優先度の高いフレームを送信を中断することなく送信できる．この仕組みはアービトレーションと呼ばれ，詳細は 2.1.2 項で述べる．通常，信号の反射を防ぐために CAN バスの両端が 120 Ω の抵抗で終端されている．したがって，CAN の合成抵抗の値は 60 Ω となる．

2.1.1 CAN におけるフレーム

CAN では 4 種類のフレーム (データフレーム，リモートフレーム，エラーフレーム，オーバーロードフレーム) が規定されている．1 つ目のデータフレーム

は，送信者から受信者へセンサデータ等を送信するためのフレームである．2 つ目のリモートフレームは，受信者がデータフレームの送信要求を行うために用いられる．3 つ目のエラーフレームは，送信した論理値と CAN の差動信号が異なるようなエラー等が発生した場合に送信される．4 つ目のオーバーロードフレームは，前のデータフレームと次のデータフレーム間に遅延を付加するのに用いられるが，CAN コントローラ や マイクロコンピュータ の処理能力が改善された現在ではほとんど使われなくなっている．

データフレームは図 2 (b) に示すように，複数のフィールドから構成される．以降では，各フィールドについて説明する．

スタートオブフレーム (Start Of Frame: SOF)

フレームの開始を表す 1 bit のドミナントで構成される．

アービトレーションフィールド

11 bit の識別子とフレームの種類を示す RTR が送信される区間で，フレームの優先順位を表す．識別子は，小さい値であるほど高い優先度のフレームとなる．本論文では，識別子を指して Arbitration ID と呼ぶ．RTR は Remote Transmission Request の略で，データフレームとリモートフレームを識別する．RTR がドミナントであるときデータフレームを表し， レセシブであるときリモートフレームを表す．アービトレーションフィールドとは調停フィールドとも呼ばれる．

コントロールフィールド

IDE，r0 と呼ばれる 2 つの予約ビットとデータ長 (Data Length Code: DLC) が送信される区間である．コントロールフィールドは制御フィールドとも呼ばれる．

データフィールド

データの内容が送信される区間である．この区間は可変長で，CAN では 0–8 byte のデータを送信できる．

CRC フィールド

フレームの伝送誤りをチェックする区間である．15 bit の Cyclic Redundancy

Check (CRC) と CRC の終了区切りを表す 1 bit のレセシブ (CRC デリミタ) からなる．

ACK フィールド

そのフレームを送信しているノード以外の受信ノードが，CRC フィールドまでを正常受信できた場合は，その合図として ACK スロットで 1 ビットのドミナント送信する．

エンドオブフレーム (End Of Frame: EOF)

フレームの終了を表す．7 bit のレセシブで表される．

2.1.2 アービトレーション

CAN では Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) 方式を採用しているため，フレームを送信しようとする ECU は，まず CAN のバスがアイドル状態であることを確認し，他の ECU がフレームを送信中の場合はフレームの送信が完了し，バスがアイドル状態になるまで待つ．2 つ以上の ECU が同じタイミングで送信を開始した場合，この送信要求の衝突はビット単位のアービトレーションによって解決される．このアービトレーションは Arbitration ID の値を用いて実行される．したがって，Arbitration ID を送信中，メッセージを送信している ECU は送信したビットとバスが表現するビットが同じかどうか比較する．もしリセシブを送信したにもかかわらず，バスがドミナントを表現している場合，リセシブを送信している ECU は送信権を失うため，メッセージの送信を取り止めなければならない．

また，もしデータフレームとリモートフレームが同じ Arbitration ID で，かつ，同じタイミングで送信を開始した場合，データフレームの RTR がドミナントであるため，データフレームが優先されるように設計されている．

2.2 CAN の脆弱性

Liu ら [13] によって CAN の脆弱性は 4 つに分類され，その脆弱性に対する攻撃手法は 5 つに分類された．図 3 に，Liu らの分類を示す．

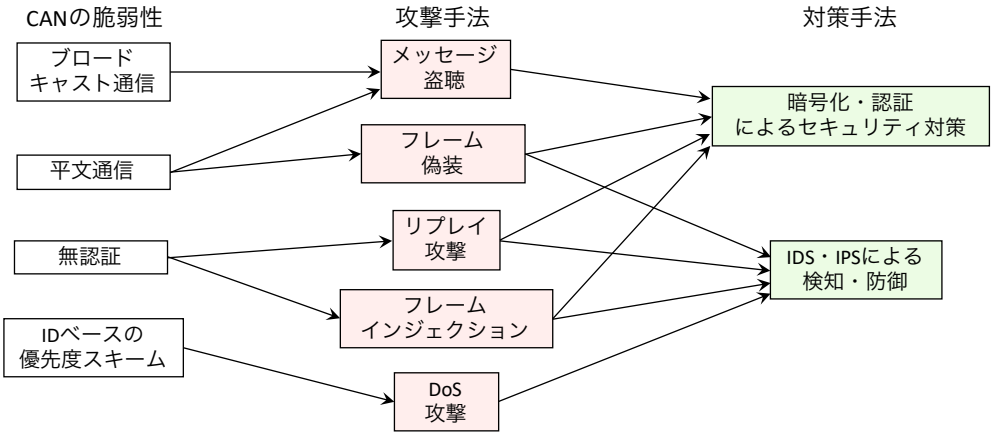


図 3: CAN における攻撃手法とその対策

まず，CAN の本質的な脆弱性としてブロードキャスト通信，平文通信，無認証，そして，Arbitration ID ベースの優先度スキームがある．これらに起因する攻撃手法を以降では述べる．1 つ目の攻撃手法として，ブロードキャスト通信と平文通信であることから，CAN に接続すれば誰でもメッセージ盗聴が可能であることが挙げられる．さらに，通信を盗聴し解析することで，車速に関するメッセージの偽装によるメータの操作といったフレーム偽装が可能となる．また，無認証であることからは，通信されているメッセージをそのまま再生する攻撃であるリプレイ攻撃や任意の CAN メッセージを送信するようなフレームインジェクションが可能であることが指摘されている．最後に，Arbitration ID ベースの優先度スキームからは攻撃者は通信されているメッセージの優先度以上のメッセージを大量に送信することでバスを圧迫させる DoS 攻撃に脆弱であると言われている．また，図 2 (b) に示すように，CAN のデータフォーマットには送信元情報を示すフィールドはないことから，受信者は CAN メッセージが送信された ECU を識別できないという問題も存在する．

以上の攻撃手法に対する対策手法として，暗号化・認証による対策がある．し

かし，CAN のデータフィールドは最大で 8 byte しかないため認証は容易に適用できるものではない．さらに，暗号化・認証に用いる鍵の管理についても自動車メーカ等で適切に鍵管理規則等を取り決めて実現する必要があるため，早急に実装することは難しい．一方で，IDS・IPS による検知・防御は，IDS・IPS を CAN バスに接続するのみで実行可能なことから，有効性と現在の自動車への実装の容易さにおいて優位性があるといえる．

2.3 自動車における Attack Surface

本節では，外部ネットワークからの攻撃の侵入経路について述べる．まず，インターネット等と接続された車載インフォテインメントシステム (In-vehicle Infotainment System: IVI) に着目する．図 4 に示すように，外部からの侵入は大きく分けて 3 つに分けられる [14]．また，Automotive Grade Linux (AGL) [15] といった Linux ベースの車載向けのプラットフォームを開発するプロジェクトが注目されていることから，ここでは Linux ベースの IVI を想定する．まず，外部 (遠距離) からの侵入経路として，携帯電話通信を取り扱う High Speed Synchronous Serial Interface (HSI) ドライバや，Wi-Fi 通信を取り扱うユーザ空間内の WPA サプリカントプロセスが挙げられる．これらの古いバージョンが IVI にインストールされている場合，攻撃者は IVI に侵入する可能性がある．さらに，IVI が自動的にある SSID のアクセスポイントに接続する設定になっている場合，攻撃者が悪意のある同一の SSID のアクセスポイントを立て，攻撃者が IVI の様々なポートへ侵入を試みる可能性もある．次に，外部 (近距離) からの攻撃者は，Bluetooth 通信を行うデーモンである Bluez の脆弱性を突いて侵入する可能性がある．また，外部ネットワーク以外の脅威として，攻撃者がソーシャルハッキング等によって自動車の運転者等にマルウェアをインストールするための USB デバイスを IVI へ接続させることも考えられる．この攻撃は，USB ソフトウェアスタックに欠陥がある場合や，あるいは，IVI のソフトウェア更新を USB 経由で行う場合に悪用される可能性がある．

また，CAN バスへの直接的な侵入経路として，ECU の診断ポートである On-

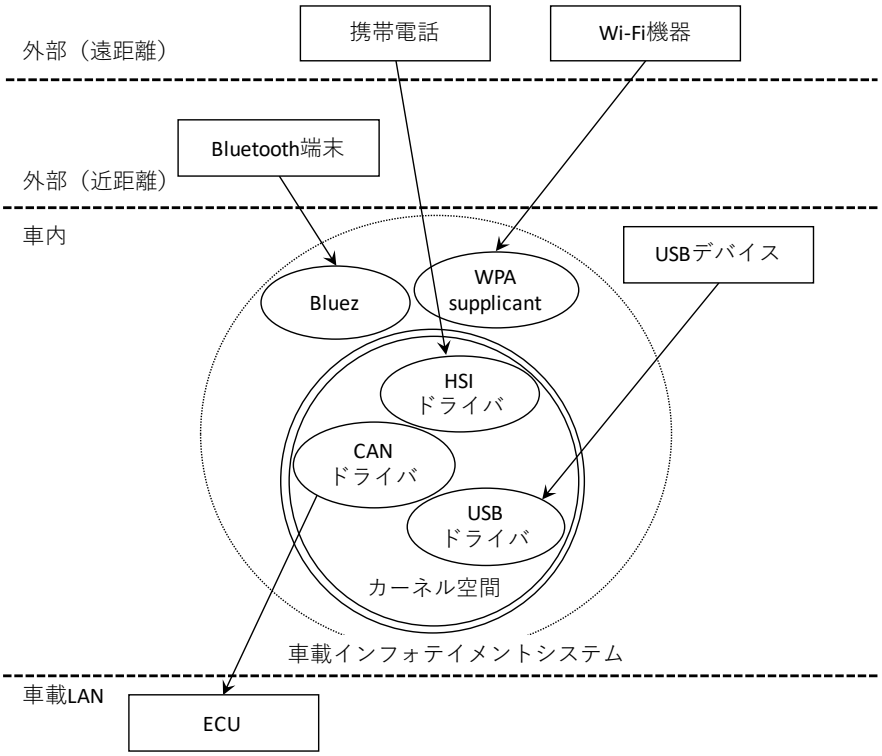


図 4: 車載ネットワークへの侵入経路 (Linux ベースの IVI)

Board Diagnostics-II (OBD-II) ポート[1] からの侵入 [16] や，車内に張り巡らされた CAN バスの配線の直接的な細工による侵入が可能である．本研究で行う評価においても OBD-II ポートから侵入を行い，攻撃検出の評価実験を行っている．

以上のことから，自動車の車載ネットワークは，直接または間接的に様々なインタフェースに接続されているため，その侵入経路は多様であることがわかる．そのため，自動車に対するセキュリティ対策には多段的・多層的な防御策が重要となる．

[1]自動車の故障診断ポート．一般的に，運転席の周辺に設置してある．

3. 関連研究

CAN のセキュリティ対策に関する研究について紹介する．ここでは，近年の自動車セキュリティに関する研究を包括的にまとめ，従来の送信元識別手法の問題点等を述べる．また，本研究で用いる高い分解能で任意の時間を観測可能な TDC と機械学習を用いたシステムにおける Concept Drift という概念について説明する．

3.1 暗号化・認証手法

本節では，CAN における暗号化・認証手法に対する手法について説明する．Herrewege らが提案した CANAuth [4] と呼ばれるプロトコルがある．このプロトコルは，CAN+ [17] という CAN のデータフィールドを拡張した下位互換性のあるプロトコルに Hash-based Message Authentication Code (HMAC) ベースの認証機能を追加している．また，CAN+は通常の CAN の 1 bit の中に 1 bit あたり 25 ns の overclocked bit と呼ばれるビットを最大 16 bit 埋め込み，データフィールドを拡張する．したがって，最も実車両に実装されているデータフィールドが 64 bit の CAN に対して適用するためには，全ての ECU に CAN+に対応したハードウェアが必要となり，CANAuth は現在普及している自動車への適用は現実的に難しい．

倉地らは，CaCAN [18] と呼ばれる CAN の集中監視システムを提案している．CaCAN は不正なデータフレームをエラーフレームで上書きすることで，不正なデータフレームを破壊する．この手法では，監視ノードが除去される，あるいは，監視ノードへ侵入されることでネットワーク全体が危険に晒される恐れがある．

AUTOSAR [19] という標準化団体は，車載ネットワークにおけるメッセージ認証に関する業界ガイドラインを公開している．AUTOSAR 準拠の CAN における認証手法として，LeiA [20] と vatiCAN [21] が提案されている．この 2 つの手法のうち，vatiCAN のみが実環境でのパフォーマンスの評価を行なっている．vatiCAN は，CAN への適用性と低いオーバーヘッドという点で優位性があるが，vatiCAN の頻繁なナンス更新スキームに対するリプレイ攻撃が可能であることが

示されている [22] .

さらに，Sancus [23] と呼ばれる軽量な Trusted Computing Base (TCB) を用いて LeiA と vatiCAN をより堅牢にするアプローチである VulCAN [22] が提案されている．VulCAN は，LeiA と vatiCAN とは異なり，ハードウェアによるメモリ保護を活用して return-oriented programming [24] といったコードを乱用する攻撃に対する強化を行う．しかしながら，VulCAN においても vatiCAN と同程度の帯域占有率の増加がある．

これらの認証手法は，基本的に認証情報の追加によるオーバーヘッドが生じ，バス占有率の増加を招くため，既存の CAN への適用は難しい．

3.2 Moving Target Defense

CAN メッセージのリバースエンジニアリング，および，フレーム偽装に対する防止手法として，ID-Hopping 機構 [25] が提案されている．この ID-Hopping 機構は，攻撃検出後，動的に全ての ECU が Arbitration ID を変更し，Targeted DoS 攻撃のターゲットとなる ECU を逃すことが可能となる．したがって， ID-Hopping 機構は，TCP/IP において IP アドレスをランダム化するような Moving Target Defense (MTD) と同様のアプローチといえる．しかし，CAN における MTD には，Arbitration ID の変更によって，変更のオーバーヘッドや各 Arbitration ID の優先度情報が失われるといったデメリットがある．

そこで，Arbitration ID の変更機能を CAN コントローラの回路にハードウェア実装する IDH-CAN [26] が考案された．この手法は，各 ECU 間でカウンタを同期しておくことで，全ての ECU がそのカウンタと対応する ID-Hopping Table を選択し，ハードウェア側で Arbitration ID を変更する．したがって，同期の失敗によって各 ECU のカウンタがずれてしまった場合，アプリケーション側でそれらのカウンタをリセットするといった例外処理を追加しなければならない．さらに，このリセットによる再同期にかかる時間が車載システムにおいて許容可能であるか検討する必要がある．

さらに，Arbitration ID 変更後も ID の優先度情報を保つことが可能な CAN-ID Shuffling Technique (CIST) [27] が提案されている．CIST は，アービトレーショ

ンフィールドが Arbitration ID の先頭ビットから順に値を比較することで決定されることを用いて，CAN の拡張 Arbitration ID (29 bit) の前半の優先度関係のみを保存し，後半をハッシュ関数を用いて乱雑に決定することで優先度情報を保つことができる．

ID-Hopping 機構，IDH-CAN，および，CIST は盗聴やリバースエンジニアリングを防止可能だが，実際の CAN へ適用する際には全ての ECU に対しハードウェアの変更が必要になり，実際の車載システムへの適用性に課題がある．

3.3 Intrusion Detection System

本節では，CAN における暗号化，認証手法，および，MTD の課題であった，CAN の帯域増加と全ての ECU に対するハードウェアの更新が無しで適用可能な手法について述べる．

3.3.1 メッセージの特徴に基づく IDS

メッセージ周期に基づく IDS [28] が提案されている．この手法は検出可能な攻撃に制限がある．例えば，この手法は攻撃者がターゲットとする ID に模倣した周期のメッセージを送信することで回避可能である．さらに，ID シーケンスに基づく IDS [29] が提案されている．この手法においても，IDS を回避可能な攻撃がある．攻撃者は正当なトラフィックと同様の ID シーケンスの悪意のあるトラフィックを送信することで回避を行う．

スライディングウィンドウにおけるエントロピーに基づく IDS [30] が提案されている．この手法は，攻撃者が一定の間隔毎に 1 つのスプーフィング攻撃を注入したとしても，スライディングウィンドウにおけるエントロピーの値が通常とほぼ変わらないため，IDS は攻撃を検出できない．攻撃を検出するためには，スライディングウィンドウの値を小さくする必要があるが，同時に IDS の偽陽性率の増加を招くことになる．

Deep learning を用いた IDS [31] [32] が提案されている．Deep learning 等の機械学習の推論は学習に比べ計算時間は大幅に少ないが，再学習が必要な場合には

多くの時間と高い計算資源が必要となる．したがって，ECU のソフトウェアアップデートによる CAN メッセージの追加が行われる場合，IDS は新たな学習モデルを再構築する可能性がある．

3.3.2 信号の物理的特徴に基づく IDS

メッセージの特徴に基づく IDS の欠点を改善するために，信号の物理的特徴に基づく IDS が検討されている．まず，CAN-H, CAN-L といった CAN におけるケーブルの特性インピーダンスの変動から不正なデバイスの接続を検出するアプローチとして時間領域反射 (Time Domain Reflectometry: TDR) を用いた手法 [33] が提案されている．この手法は CAN に対しパルスジェネレータからインパルス信号を印加し，返ってくる反射波形をオシロスコープで観測する．これより，TDR を用いた手法は不正なデバイスを検出する IDS が測定用の信号を CAN へ印加するアクティブな手法といえる．すなわち，自動車を運転中の場合，測定用の信号の印加により CAN 通信を妨害してしまう可能性がある．したがって，CAN のデータフレーム等を観測するだけで不正なデバイス・メッセージを検出可能なパッシブな手法を検討する必要がある．

パッシブな手法として，Murvay らが初めて電圧ベースの送信元識別手法 [9] を提案した．この後に，Viden [7] と Scission [6] が電圧ベースの送信元識別手法に拡張を行なった．

さらに，別のアプローチとして Cho らは周期的に送信されるメッセージのクロックのずれに基づいて送信元識別を行う手法 CIDS [10] を提案した．

しかしながら，Viden や CIDS は複数の CAN メッセージに基づいて正当な通信が行われているか判断するため，攻撃者が徐々に CAN メッセージを注入し閾値をシフトさせる Hill-climbing-style attack に脆弱であることが示されている [34]．この Hill-climbing-style attack に耐性を持つためには，SIMPLE [34] と呼ばれる手法と同様に，1 メッセージから得られる特徴のみを用いて送信元を識別可能でなければならない．

電圧ベースの 1 メッセージで送信元を識別する手法として，Choi らの手法 [8]，Scission [6]，および，SIMPLE [34] がある．これらと提案手法の比較は，7.2 節

で述べる．

CAN の伝送線路における遅延に基づく IDS [12] が提案されている．この手法では，遅延を観測するために，IDS は CAN バスの両端近くに 2 つのプローブポイントが必要となる．一般に，現代の自動車における CAN バスは複数のバスに分割されており，n 個のバスに分割されていると IDS のために $2n$ 個のプローブポイントが必要となる．そのため，この IDS は CAN への配線を増加させることになり，CAN を導入する目的の 1 つである配線の簡略化に反している．

各 ECU が送信する 1 bit の時間が異なることに着目した手法の Bit-Time-based CAN Bus Monitor (BTMonitor) [35] が提案されている．この手法は，同じ ECU から送信される 5〜50 個の CAN メッセージを 1 つのデータとして分類することで，平均正解率 99%を実現可能であることを示している．しかしながら，複数のメッセージに基づく手法であるため，Viden や CIDS と同様に，Hill-climbing-style attack に脆弱である可能性がある．また，BTMonitor は温度変化による特徴量の変動を複数の学習モデルを構築して適切な学習モデルを分類に用いるため，学習モデルの数だけ RAM 等の計算資源を浪費してしまうデメリットがある．

CAN トランシーバの信号の遅延時間に着目した送信元識別手法 [11] が提案されている．この手法では，安価な計測デバイスを用いて遅延時間を観測し，送信元識別可能である．また，この手法は CAN 信号の立ち上がり時のみ特徴となるデータをサンプリングするため，電圧ベースの送信元識別手法と比べて少ないサンプリング回数で CAN メッセージの送信元を識別可能である．さらに，1 メッセージで得られる特徴量のみを用いて送信元識別を行うため，Hill-climbing-style attack に対し堅牢である．ただし，各 ECU の遅延時間の差が計測デバイスの時間分解能より低い場合，ECU を正しく分類できないという問題がある．また，温度変化に対し各特徴量 (遅延時間) が変化する可能性がある．

3.4 CAN におけるセキュリティ対策手法のまとめ

本節では，これまでに述べた暗号化・認証手法，MTD，メッセージの特徴に基づく IDS，および，信号の物理的特徴に基づく IDS の比較を行う．Liu ら [13] の CAN に対する攻撃手法に基づく各対策手法の比較を表 1 に示す．また，メッ

セージの特徴に基づく IDS は Message-Based，信号の物理的特徴に基づく IDS は Physical-Layer Identification の頭文字の PLI として表記している．

表 1: CAN におけるセキュリティ対策手法の比較

	暗号化・認証	MTD	Message-Based	PLI
メッセージ盗聴	○	○	×	×
フレーム偽装	○	○	○	○
リプレイ攻撃	○	○	○	○
Frame Injection	○	○	○	○
DoS 攻撃	×	△	○	○
オーバーヘッド	有り	有り	無し	無し
ハードウェアの追加	全ての ECU	全ての ECU	無し	IDS のみ

暗号化・認証手法は，CAN に対する攻撃手法の DoS 攻撃以外を防止することができる．DoS 攻撃は CAN そのものの帯域を圧迫するため，暗号化・認証を施した CAN においても脅威となる．さらに，暗号化・認証手法は，認証を行うオーバーヘッドや，全ての ECU へ高速化を行うための暗号化処理を行う追加のハードウェアが必要になる．

MTD は，暗号化・認証手法においても防止可能であった 4 つの攻撃手法と，一部の DoS 攻撃を防止可能である [25]．しかしながら，MTD においても，Arbitration ID を変更するオーバーヘッドや，CAN コントローラ へ ID-Hopping 機構の追加が必要となる．

メッセージの特徴に基づく IDS は，データフィールドの暗号化や Arbitration ID の攪拌を行わないため，メッセージ盗聴を防ぐことはできない．しかし，それ以外の攻撃手法は検知可能である．メッセージ盗聴は直接的に自動車へ予期しない挙動をもたらすことはないことを踏まえると，メッセージの特徴に基づく IDS は暗号化・認証手法，および，MTD よりも対策手法として適しているといえる．また，メッセージの特徴に基づく IDS は基本的にソフトウェアレベルで実装されることから，暗号化・認証手法，および，MTD におけるオーバーヘッドやハードウェアの追加等は無い．しかしながら，各 IDS に特化した攻撃手法も明らかに

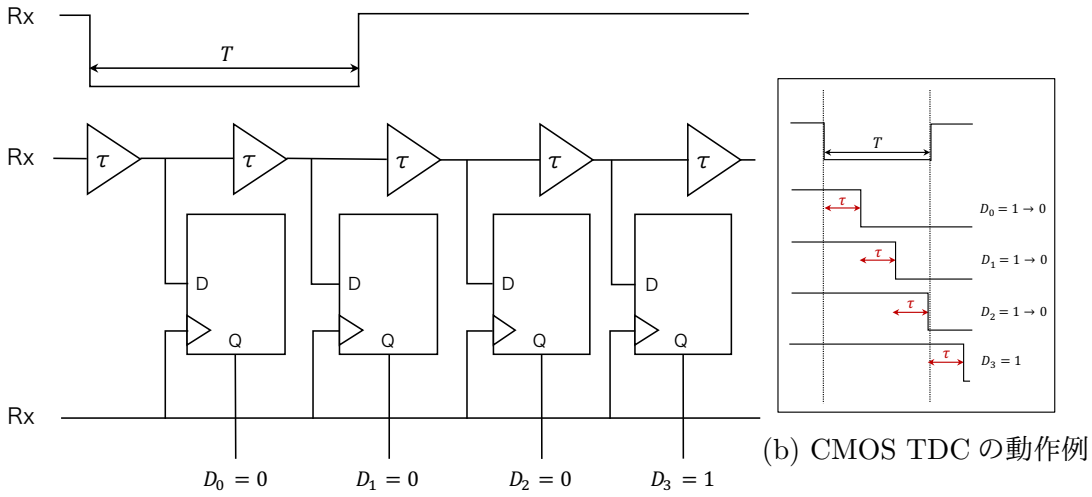
なっているため，それらを抜本的に解決する必要がある．

信号の物理的特徴に基づく IDS は，メッセージの特徴に基づく IDS と同様に，メッセージ盗聴を防ぐことはできないが，それ以外の攻撃手法は検知可能である．また，信号の物理的特徴に基づく IDS には，暗号化・認証手法，および，MTD のようなオーバーヘッドが無いが，IDS に物理的特徴を観測するためのハードウェアが必要となる．一方で，物理的特徴を偽装することは容易ではないため，信号の物理的特徴に基づく IDS は各 IDS に特化した攻撃手法 (例えば，エントロピーに基づく IDS に対するエントロピーを模倣した DoS 攻撃) でさえも検出することができる利点がある．そのため，本研究では，信号の物理的特徴に基づく IDS に着目する．さらに，本研究では，信号の物理的特徴に基づく IDS のうち Hill-climbing-style attack に対し堅牢で，かつ，データ取得に関するサンプリング回数が比較的少ない遅延時間に基づく送信元識別手法 [11] を拡張することを検討する．

3.5 Time-Digital Converter

Time-Digital Converter (TDC) は，Time-of-Flight (ToF) カメラにおける飛行時間計測，原子核実験，および，集積回路のテスト等で用いられる数十〜数百 ps 単位で時間計測を行うタイムディジタイザである．オシロスコープの時間分解能は数百 ps と高精度にパルス幅を計測可能であるが，その導入にコストがかかる．一方で，製品として TDC を含んだ IC が$23.80 程で販売されている [36] が，時間計測と同時に CAN の Arbitration ID の観測や観測期間を緻密に制御するためには TDC を直接制御する必要がある．原子核物理学の研究領域では，FPGA を用いて TDC を安価に実装する手法が研究されている [37] [38]．FPGA は，一般にオシロスコープよりは安価であり，FPGA に TDC を実装することで時間計測と同時に CAN の Arbitration ID の観測や観測期間の制御を行うことが可能となる．

まず，CMOS での TDC (以降，CMOS TDC) の実装 [39] について述べる．図 5 に，CMOS TDC の回路構成とその動作例を示す．CMOS TDC は，図 5 (a) のように，D 型フリップフロップと，1 つあたり τ ps の遅延セルから成る遅延線から構成される．ここで，CMOS TDC の動作を説明する．まず，T を被計測時間



(a) CMOS TDC の回路構成

(b) CMOS TDC の動作例

図 5: CMOS TDC の回路構成と動作例

とする．さらに，T 時間の間，論理値が 0 となる被測定信号 Rx (図 5 (a) の一番上) があるとする．この被測定信号 Rx を，CMOS TDC の 2 つの入力へ同時に入力する．入力後，図 5 (b) に示すように，入力信号は遅延セルによって τ だけ遅延し CMOS TDC 全体に伝搬していく．そして，被測定信号 Rx が立ち上がったタイミングで，それぞれの D 型フリップフロップの入力 D に入力されている信号が出力 Q に出力され，$D_0D_1D_2D_3$ が決定する．これより，CMOS TDC の出力 $D_0D_1D_2D_3 = (0, 0, 0, 1)$ が得られる．ここで，遅延 $\tau = 100\,\mathrm{ps}$ と仮定すると，出力 $D_0D_1D_2D_3 = (0, 0, 0, 1)$ より，3 つ目の遅延セルまで信号が伝搬されていることがわかる．したがって，$T = 3 \times 100\,\mathrm{ps} = 300\,\mathrm{ps}$ となる．一般的に，以上のような動作によって TDC は高い時間分解能を実現する．

続いて，FPGA における TDC の実装方式について述べる．FPGA における TDC の実装方式として，Tapped-Delay TDC [37] と呼ばれる TDC がある．Song ら [37] は，マルチビット加算器を用いて遅延線を構築した．図 6 に，Tapped-Delay TDC におけるマルチビット加算器を用いた遅延線の実装を示す．また，各加算器 (Adder) のブール方程式は次のようになる．

$$S = A \oplus B \oplus C_{\mathrm{i}} \tag{1}$$

$$C_{\mathrm{o}} = AB + (A + B)C_{\mathrm{i}} \tag{2}$$

式 (1), (2) の変数はそれぞれ，A と B は Adder の入力，C_{i} (carry-in bit) は一つ

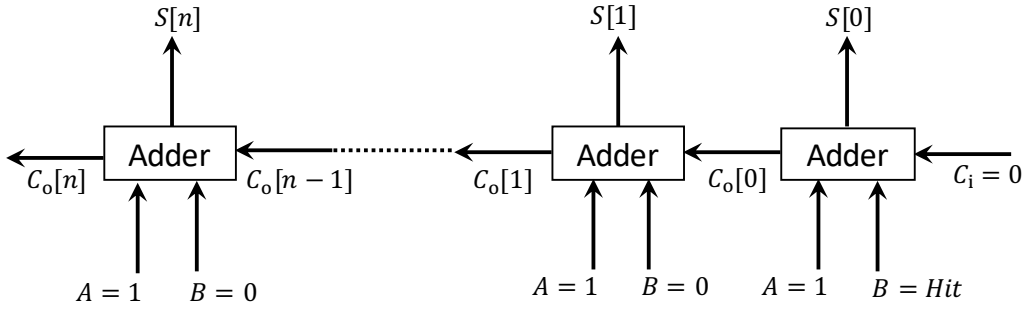


図 6: Tapped-Delay TDC における遅延線

前の Adder からの繰り上がりを表す入力，C_{o} (carry-out bit) は繰り上がりを表す出力，そして，S は加算結果の出力である．したがって，図 6 の遅延線は，ある信号 Hit が $Hit = 1$ になった時に，$C_{\mathrm{o}}[0] = 1$ が伝搬して各 Adder の出力 S が 0 となっていく．また，受信信号が送られてくる Rx を Hit に入力することで測定したい期間を観測できる．Tapped-Delay TDC も CMOS TDC と同様に，どの Adder まで信号が伝搬したかを各 Adder の S を取得し，時間計測を行う．

3.6 Concept Drift

機械学習を用いたシステムは，与えられた学習データの構造とその教師信号との関係を学習し，運用される．一般に，学習に用いられたデータの特性は，システムの使用期間にわたって恒久的に変化しないことが前提となる．ただし，現実的な環境において，外部の影響により基礎となるデータの分布が動的に変化する可能性があるため，この前提を満たすことはできない．このデータの特性または教師信号の変化は，Concept Drift [40] と呼ばれ，この Concept Drift によってシステムの分類精度が著しく低下する場合がある．Concept Drift は様々な分野で観測され，例えば，天気予報における天気や気温といったデータは季節によって周期的に変化する Concept Drift を持つ．

また，Concept Drift はその変化の仕方によっていくつかの種類に分類される．Concept Drift の種類を図 7 に示す．自動車の分野においては，材料の摩耗と温

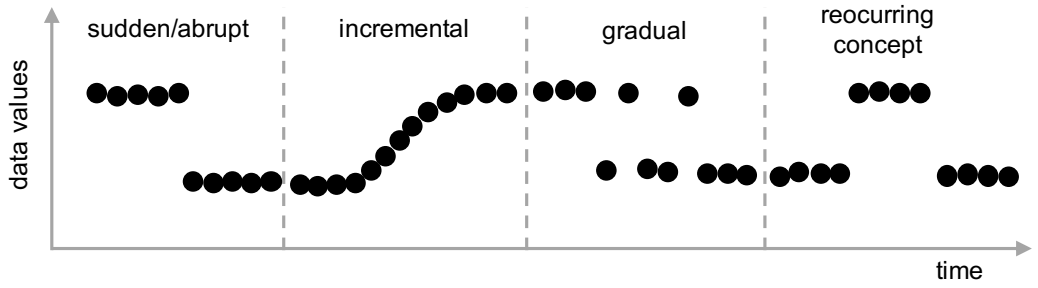


図 7: Concept Drift の種類

度の変化が電圧の増加的な Concept Drift をもたらし，車載ネットワークの変更やその電源電圧の変化が急激な電圧の Concept Drift を引き起こすと言われている [41]．したがって，高い分類精度を維持するには，Concept Drift に応じて学習モデルを適合させるか，Concept Drift した後のデータを使用して学習モデル再構築する必要がある．

また，CAN の信号の物理的特徴に基づく IDS においても，温度変化による特徴量の Concept Drift に対し自動的に学習モデルを追従させる手法 [41] や，複数の学習モデルを用意し温度によって適切な学習モデルを選択する手法 [35] が検討されている．

3.7 関連研究のまとめ

3 章の各節では，車載ネットワークにおける暗号化・認証手法，MTD，および，IDS についてまとめ，本研究に関連する TDC と Concept Drift について説明した．本節では，3 章を簡潔にまとめ，提案手法が解決する課題について述べる．

3.1，3.2 節では，これまで提案されてきた暗号化・認証手法，および，MTD は，基本的に暗号化処理等の追加によるオーバーヘッドが生じ，かつ，追加のハードウェアが必要であることについて述べた．一方で，3.3 節では，IDS は，暗号化・

認証手法，および，MTD とは異なり，従来の CAN に容易に適用可能であり，各 ECU に追加する処理が必要なく，オーバーヘッドも無いという利点があることを説明した．また，IDS の中でも信号の物理的特徴に基づく IDS はその特徴量を偽装することが困難であるため，メッセージの特徴に基づく IDS と比べ利点がある．さらに，信号の物理的特徴に基づく IDS のうち，遅延時間に基づく送信元識別手法は特徴量を取得するためのサンプリング回数が他の手法に比べ少ないという利点があるが，各 ECU の遅延時間の差が計測デバイスの時間分解能より低い場合，ECU を正しく分類できない．そのため，時間分解能を改善する必要がある．また，この手法で特徴量として用いられる遅延時間は温度の変化によって変化する可能性があり，特徴量の変化に対応する必要がある．

4 章では，3.5 節で述べた高時間分解能観測が可能な TDC を用いて遅延時間に基づく送信元識別の分類精度を改善し，3.6 節で述べた温度変化による Concept Drift に対しロバストな手法を提案する．

4. 遅延時間の高時間分解能観測に基づく送信元識別手法の提案

本章では，TDC を用いた遅延時間の高時間分解能観測に基づく CAN メッセージの送信元識別手法を提案する．以降の節では，提案手法の概要と要件，提案手法で用いる特徴量の定義，提案手法で用いる分類アルゴリズム，および，提案手法の実装に関して述べる．

4.1 提案手法の概説

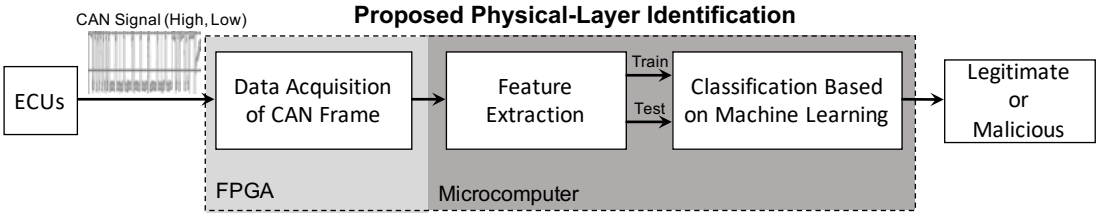


図 8: 提案手法の概要図

提案手法は，図 8 に示すように，データ取得，特徴抽出，分類の 3 つのフェーズから構成される．データ取得フェーズでは，TDC を用いて遅延時間がデジタルな値として取得される．次に，特徴抽出フェーズでは TDC から得られた遅延時間を平均，分散といった統計量に変換する．最後に，得られた統計量を用いて分類アルゴリズムにより CAN メッセージがどの ECU から送信されたか識別する．以降の節では，順番にそれぞれのフェーズについて述べる．

ここで，提案手法の要件を整理する．まず，提案手法の有効性を示すために，遅延時間を ECU の識別に用いたとしても電圧ベースの手法と同程度の識別精度を実現可能である必要がある．

さらに，電圧を用いた手法では連続な値をサンプリングしてデータ取得を行うため，シャノンの標本化定理より，CAN (500 kbps) の電圧の特徴的な振る舞いを観測するためには少なくとも 10 MHz でサンプリングを行う必要がある [6]．

このサンプリングレートはCANのボーレートに依存するため，Controller Area Network with Flexible Data rate (CAN FD) [42] のような1 Mbps以上の高速なボーレートの場合データ量が増加する．したがって，提案手法ではデータ取得フェーズにおけるデータ量を削減することを要件の1つとする．

次に，システム全体の計算量が従来手法の中で最も低い計算量である$\Theta(n)$以下となることを要件として定める．

そして，信号の物理的特徴は温度変化によるConcept Driftが起こる可能性があるため，特徴量のConcept Driftに対しロバストな手法である必要がある．

したがって，以下のように提案手法の要件を定める．

I. 電圧ベースと同程度の識別精度

II. データ取得フェーズにおけるデータ量の削減

III. システム全体が$\Theta(n)$以下の計算量

IV. 温度変化によるConcept Driftへのロバスト性

4.2 データ取得フェーズ

提案手法におけるデータ取得フェーズについて述べる．提案手法で取得するデータはCAN信号の立ち上がり・立ち下がり時間の遅延であり，以降ではその定義と観測方法について述べる．

4.2.1 遅延時間の定義

提案手法で用いる遅延時間は，従来のCANトランシーバ の信号の遅延時間に着目した送信元識別手法 [11] と同様であり，その問題点は3.3.2項で述べた．ここでは，遅延時間の原因と遅延時間の算出のための式について述べる．

遅延時間の観測を行なった例を図9に示す．図9 (a), (b) では，遅延時間の観測環境と観測された信号を示している．図9 (b) では，Node 1のTx信号がNode

2 の Rx 信号と比べわずかに遅延していることがわかる．提案手法ではこの遅延時間を ECU の分類に用いる．

また，この遅延時間は，CAN トランシーバ内のトランジスタのスイッチング時間に起因している．実際，CAN の信号レベルが変化する時の遅延時間は，トランジスタのスイッチング時間と出力の負荷容量 C_L が充放電される時間によって定まる．また，負荷容量 C_L は，トランジスタにおけるゲートの出力容量，入力容量，配線容量の 3 種類から決定される．

遅延時間の算出のための式について述べる．CAN における，CAN トランシーバのタイミング図を図 10 に示す．さらに，図 10 のように，t_1，t_2，t_3，および，t_4 を定義する．これより，以下の式が成り立つ．

$$t_3 - t_2 = t_4 - t_1 \tag{3}$$

IDS 側 (受信側) で観測できるのは t_4 のみであるため，近似を導入する．従来手法 [11] より，$t_1 \approx t_{\mathrm{bit}}$ と見なせるため，

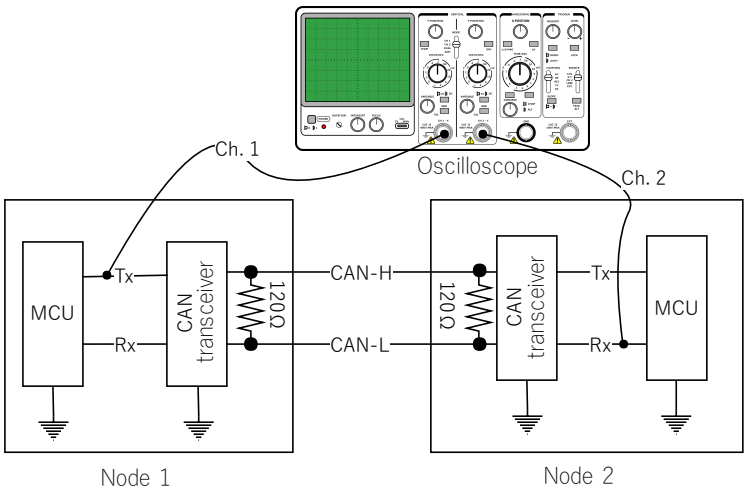
$$t_3 - t_2 \approx t_4 - t_{\mathrm{bit}} \tag{4}$$

となる．また，t_{bit} は CAN の 1bit の送信にかかる理想的な期間である．例えば，CAN が 500 kbps の場合，t_{bit} =2000 ns である．したがって，提案手法では $t_4 - t_{\mathrm{bit}}$ を観測することで CAN トランシーバの遅延時間を得る．

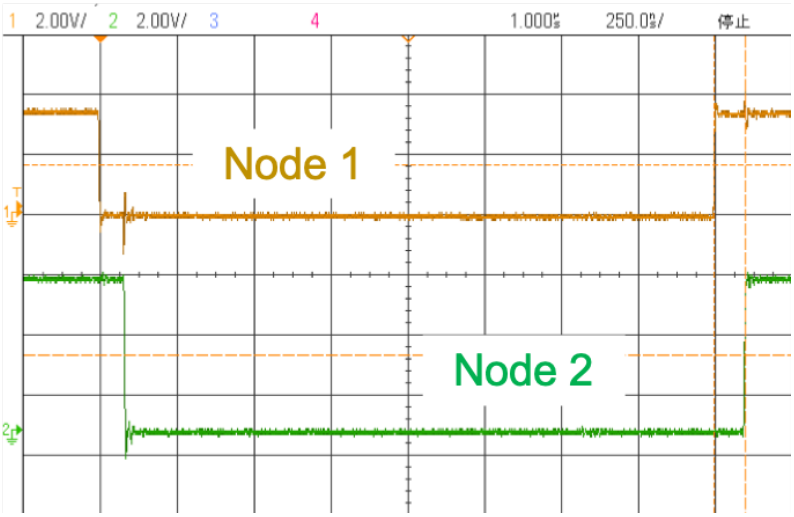
4.2.2 TDC による遅延時間の観測

提案手法で用いる TDC は Tapped Delay TDC [37] と呼ばれる TDC である．この TDC は，FPGA にある遅延素子を並べた回路に信号を入力し，並べた遅延素子のどこまで信号が伝達したかを D 型フリップフロップで保存し，その値を出力する．

CAN では，ある Arbitration ID は 1 つの ECU に割り当てられ，複数の ECU が同じ Arbitration ID を送信することはない．したがって，提案手法では Arbitration ID が正当な送信元から送信されているかを確認し攻撃を検出する．そのため，TDC で遅延時間を観測すると同時に，観測している遅延時間と Arbitration ID を紐付



(a) 観測環境



(b) Node 1 の Tx 信号と Node 2 の Rx 信号

図 9: 遅延時間の観測

ける必要がある．そこで，提案手法では CAN 信号の立ち上がり検出後，遅延時間，Arbitration ID，および，DLC を同時に FPGA 内のキューに加列する．

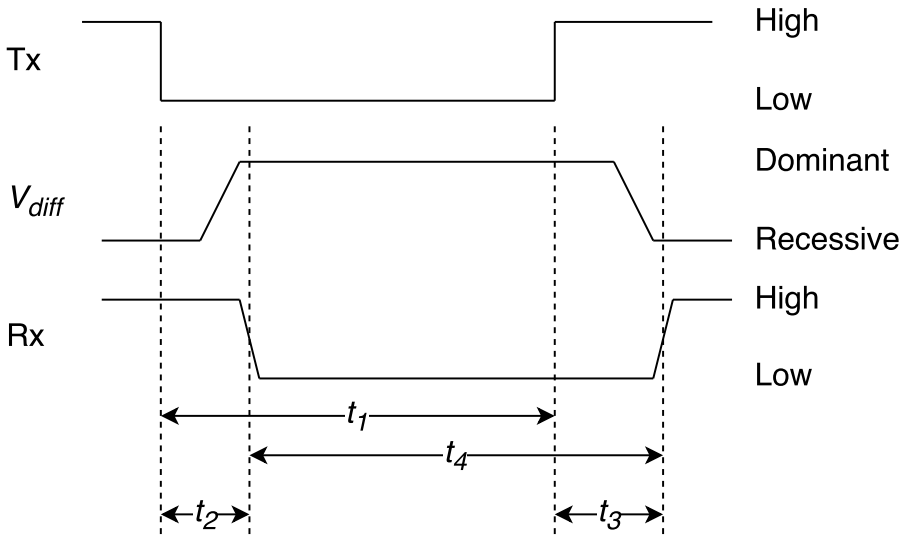


図 10: CAN トランシーバの信号のモデル化

4.3 特徴抽出フェーズ

遅延時間に基づく従来手法では特徴量として，平均と標準偏差を用いている．この 2 つの特徴量のみの場合，平均と標準偏差がおおよそ同じで左に裾が伸びる分布を持つ ECU と右に裾が伸びる分布を持つ ECU があった場合，送信元識別できないという問題がある．そこで，分布の歪度 (Skewness) を導入することで識別可能となる．このように，様々な特徴量を導入することでより効率的に ECU を分類できる可能性がある．したがって，従来手法よりも効率的に ECU を分類するために，提案手法では様々な統計量から適切な特徴量を選定する．

従来手法の 1 つである Scission [6] と同様に，提案手法では表 2 における統計量から効率的な特徴量を選定する．効率的な特徴量を決定するために，特徴量の重みを算出するアルゴリズムである Relief-F [43] を用いた特徴量のランク付けを行い，Relief-F の結果に基づいて特徴量を決定する．

表 2: 特徴選択を行う統計量のリスト，x は CAN メッセージの任意の時点で観測された遅延時間，N は 1 つの CAN メッセージで観測された遅延時間の数．

Feature	Description
Mean	$\mu = \frac{1}{N}\sum_{i=1}^{N} x(i)$
Standard Deviation	$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x(i)-\mu)^2}$
Variance	$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x(i)-\mu)^2$
Skewness	$skew = \frac{1}{N}\sum_{i=1}^{N}(\frac{x(i)-\mu}{\sigma})^3$
Kurtosis	$kurt = \frac{1}{N}\sum_{i=1}^{N}(\frac{x(i)-\mu}{\sigma})^4$
Root Mean Square	$rms = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x(i)^2}$
Max	$max = \max(x(i))_{i=1\ldots N}$
Min	$min = \min(x(i))_{i=1\ldots N}$
Energy	$en = \frac{1}{N}\sum_{i=1}^{N} x(i)^2$

4.4 分類フェーズ

CAN メッセージの送信元識別は，分類問題に帰着できる．したがって，提案手法では学習アルゴリズムを用いて解くことを検討する．提案手法では，いくつかの学習アルゴリズムの平均正解率を評価し，最も平均正解率が高いアルゴリズムを提案手法の分類フェーズで用いる．したがって，6.4 節では，さまざまな学習アルゴリズムの平均正解率を評価する．6.4 節の評価より，提案手法における分類フェーズにおける学習アルゴリズムとして Random Forest Classifier を選択する．また，Random Forest Classifier は，Multilayer Perceptron，および，Support Vector Machine といった学習アルゴリズムと比べ高速に学習モデルを構築可能であり，推論にかかる時間も Multilayer Perceptron よりも短く Support Vector Machine とおおよそ同様であることが示されている [44]．したがって，Random Forest Classifier は評価した学習アルゴリズムの中で最も高い平均正解率であり，かつ，比較的高速に学習・推論が可能であるため，提案手法で用いる学習アルゴリズムとして最も優れているといえる．以降では，特に明記しない限り，学習アルゴリズムは Random Forest Classifier とする．

ここで，平均正解率を比較する学習アルゴリズムについて述べる．提案手法では，関数値，距離，および，木によって構成される代表的な学習アルゴリズムを比較する．以下に，各学習アルゴリズムについて述べる．

Multilayer Perceptron

Multilayer Perceptron は，パーセプトロンと呼ばれる学習データの入力に対して望みの出力が得られるように構築されたネットワークモデルを多数の層にして学習するアルゴリズムである．提案手法で用いる Multilayer Perceptron は，入力層，中間層，および，出力層の 3 層から構成され，中間層のサイズ，活性化関数，および，最適化アルゴリズムは分類精度が高くなるように調節する．

K-Nearest Neighbor

識別したいデータに対し，最近傍にある K 個のデータの所属するクラスを調べ，それらが最も多く所属するクラスとして識別する方法を K 近傍法 (K-Nearest Neighbor) という．Multilayer Perceptron と同様に，分類精度が高くなるように最適な K を探索する．

Decision Tree

決定木 (Decision Tree) は，大小関係のような単純な識別規則を組み合わせて複雑な識別境界を得る方法である．提案手法の評価では，Scikit Machine Learning ライブラリのデフォルトの分類器を用いる．

Random Forest

Random Forest は，ブートストラップ法で抽出した学習データの説明変数をランダムに選択して，複数の決定木を構成し，それらの多数決によって識別結果を決める．決定木と同様に，評価で用いる Random Forest は，Scikit Machine Learning ライブラリのデフォルトの Random Forest Classifier を用いる．

Support Vector Machine (Gaussian Kernel)

Support Vector Machine は，クラス間のマージンを最大化することで，最

適な閾値のパラメータを学習し分類する学習アルゴリズムである．また，Support Vector Machine は，カーネルトリックと呼ばれる方法を用いて非線形な識別関数を構成できるように拡張可能であり，提案手法においてもカーネルトリックを用いた手法を用いる．さらに，提案手法ではカーネルは非線形特徴ベクトルが無限次元となり，高い分類精度が期待できる Gaussian Kernel を用いる．

4.5 遅延時間の Concept Drift

いくつかの電圧ベースの送信元識別手法 [7] [10] [34] [35] [41] では，温度変化による CAN メッセージの送信元情報となる特徴量の変化 (Concept Drift) が起こることが指摘されている．そこで，本研究で特徴量として用いる遅延時間も温度による Concept Drift が存在するかどうか実験による調査を行ったところ，温度変化に対し遅延時間が単調増加する ECU と遅延時間の変化が観測できない ECU が確認できた．温度変化による遅延時間の変化の詳細な実験結果については，6.6 節でまとめる．したがって，提案手法においても電圧ベースの送信元識別手法と同様に温度変化に堅牢でなければならない．

そこで，電圧ベースの送信元識別手法における温度変化に対するアプローチを以下に示す．

線形回帰による補正 [34]

CAN における差動信号の電圧と温度変化は相関があることに基づいて線形回帰により特徴量の補正を行う方式．

複数の学習モデルによる緩和 [35]

31 °C〜35 °C は同じ学習モデルを用いて，36 °C〜40 °C は 31 °C〜35 °C で用いた学習モデルとは異なる学習モデルを用いるといったように，複数の学習モデルを用意し各温度で最適な学習モデルを選択する方式．

特徴量のトラッキング [7] [10] [41]

間近に受信した数個の CAN メッセージを用いて学習モデルを更新して特徴量の変化をトラッキングし，学習モデルの劣化を回避する方式．

1 つ目の線形回帰による補正は，6.6 節で述べる通り遅延時間は温度変化に対し線形に増加しない場合も存在するため，提案手法には適用できない．2 つ目の複数の学習モデルによる緩和は，複数の学習モデルを用いるため 1 つの学習モデルを用いる場合よりもメモリ使用量が増加することが考えられる．3 つ目の特徴量のトラッキングのうち CIDS [10] と Viden [7] は，間近に受信した数個の CAN メッセージを学習に用いるため，Hill-climbing-style attack に脆弱であると指摘されている [34]．したがって，提案手法では遅延時間の統計量に温度を特徴量として加え，温度に対応した遅延時間を学習するモデルを構築する．

5. 提案手法の実装

本章では，提案手法を実装したプロトタイプ IDS と IDS を構成する要素の 1 つである TDC の実装について述べる．

5.1 プロトタイプ IDS の実装

本節では，提案手法の実装について述べる．CAN メッセージにおける遅延時間の観測区間を定義する．CAN は可変長のデータフィールドを持ち，さらに，ACK フィールドでは複数の ECU が ACK を送信する．そのため，データフィールドの長さが 0 byte である場合における ACK フィールドまでの区間 (34 bit) を観測区間として定義する．

提案手法の実装に関するブロック図を図 11 に示す．MCP2551 は CAN トランシーバであり，CAN の差動信号を論理値に変換，またその逆を行う．図 11 に示すように，FPGA には 5 つのモジュールが構成されている．

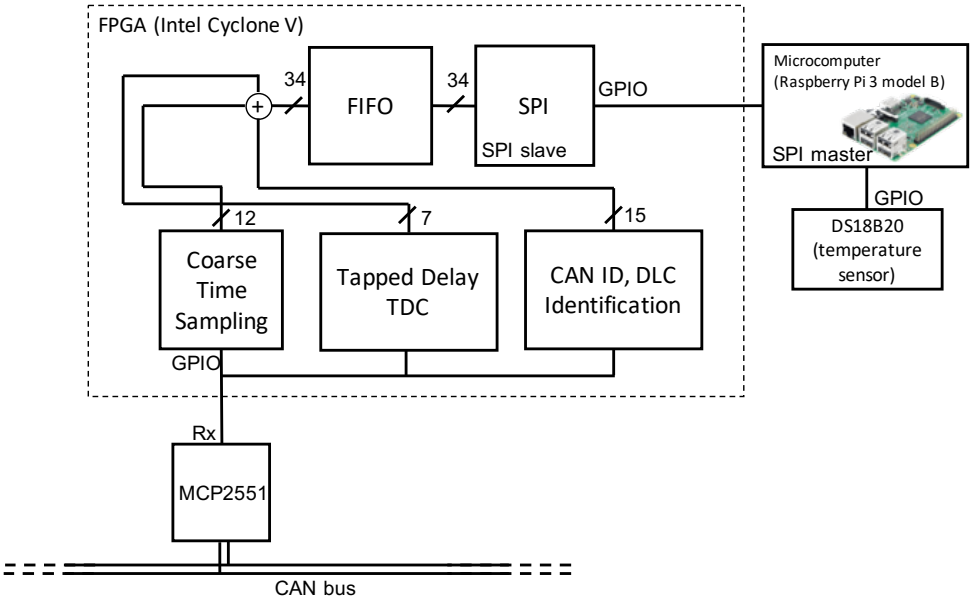


図 11: 提案手法の実装

1 つは，Tapped Delay TDC モジュールであり，154ps 単位で時間計測を行う．2

つ目は，Coarse Time Sampling モジュールと呼ばれ，FPGA 内のクロック (50MHz) の粒度で時間計測を行う．Tapped Delay TDC モジュールと Coarse Time Sampling モジュールは，前述の 34 bit の観測区間で常に，それぞれ 154ps と 20ns 単位でカウンタ値をインクリメントする．CAN の立ち上がり信号が入力されると，この 2 つのカウンタ値が FIFO モジュールに入力される．4 つ目は，CAN ID, DLC Identification モジュールであり，CAN メッセージの Arbitration ID と DLC をデジタル値に変換する．このモジュールも CAN の立ち上がり信号毎に，FIFO モジュールに Arbitration ID と DLC の値が入力される．最後は，SPI slave モジュールであり，SPI のマスタである Raspberry Pi から送信要求があると，FIFO モジュールにある Arbitraion ID や TDC のカウンタ値を Raspberry Pi 側に送信する．

さらに，遅延時間の測定アルゴリズムについて述べる．遅延時間の測定アルゴリズムをアルゴリズム 1 に示す．以下では，アルゴリズム 1 について説明する．

1〜6 行目では，各変数の定義を行う．また，1，2 行目の変数の定義では，CAN ID, DLC Identification モジュールと Tapped Delay TDC モジュールの出力をそれぞれの変数に格納する．モジュールの出力は FPGA 内のクロック CLK_{20ns} の状態によらず，常に最新の値が出力されるものとする．

7 行目以降は，CLK_{20ns} の立ち上がりエッジの度に実行される．

8 行目では，CAN の状態 (リセシブ，ドミナント) を変数 CAN_{register} に格納している．

9〜11 行目では，SOF を検出後，遅延時間の観測期間であることを表現する $chapture$ を 1 の状態にする．

12〜14 行目は，ACK の直前である 34 bit 目 で測定期間の終了を表すため $chapture$ を 0 の状態にし，20 ns 毎の遅延時間のカウンタ $coarse\_counter\_from\_SOF$ を初期化する．

16〜23 行目では，遅延時間のカウンタの増加，および，FIFO モジュールへのデータの送信を行う．

Algorithm 1 Sampling delay-time algorithm in the sampling circuit. $CLK_{20\text{ns}}$ is 50 MHz clock in FPGA.

Input: CAN_{Rx}

Output: $ID\_DLC, t_{\text{elapsed\_coarse}}, t_{\text{elapsed\_fine}}$

asssign $ID\_DLC[14:0] \Leftarrow$ CAN ID, DLC Identification Circuit
asssign $t_{\text{elapsed\_fine}}[6:0] \Leftarrow$ Tapped Delay TDC Circuit
$t_{\text{elapsed\_coarse}}[11:0] \Leftarrow 0$
$coarse\_counter\_from\_SOF[11:0] \Leftarrow 0$
$chapture \Leftarrow 0$
$CAN_{\text{register}}[1:0] \Leftarrow 0$
always positive edge $CLK_{20\text{ns}}$ **do**
    $CAN_{\text{register}} <= \{CAN_{\text{register}}[0], CAN_{Rx}\}$
    if SOF is detected **then**
        $chapture \Leftarrow 1$
    end if
    if $counter\_from\_SOF \geq 68\,000\,\text{ns}$ **then**
        $coarse\_counter\_from\_SOF \Leftarrow 0$
        $chapture \Leftarrow 0$
    end if
    if $chapture$ **then**
        $coarse\_counter\_from\_SOF \Leftarrow coarse\_counter\_from\_SOF + 20\,\text{ns}$
        if $CAN_{\text{register}}[1:0] == \text{b}'01$ **then**
            $t_{\text{elapsed\_coarse}} \Leftarrow counter\_from\_SOF$
            $t_{\text{elapsed\_fine}} \Leftarrow$ Tapped Delay TDC Circuit
            return $ID\_DLC, t_{\text{elapsed\_coarse}}, t_{\text{elapsed\_fine}}$
        end if
    end if
end always

アルゴリズム 1 の値を SPI 通信で受信した時の出力例を図 12 に示す．この出力例では，SPI 通信の受信プログラム開始から 104 個目の CAN メッセージで，Arbitration ID が 555，データフィールドが 8 byte の CAN メッセージの遅延時間を出力している．

```
[arbitration_ID]:0,[coarse_time]:66,[fine_time]:3C
[arbitration_ID]:0,[coarse_time]:12E,[fine_time]:2C
[arbitration_ID]:0,[coarse_time]:1F6,[fine_time]:2C
[arbitration_ID]:0,[coarse_time]:2BE,[fine_time]:26
[arbitration_ID]:0,[coarse_time]:386,[fine_time]:22
[arbitration_ID]:0,[coarse_time]:44E,[fine_time]:26
[arbitration_ID]:555,[coarse_time]:5DE,[fine_time]:34
[arbitration_ID]:555,[coarse_time]:7D2,[fine_time]:3E
[arbitration_ID]:555,[coarse_time]:89A,[fine_time]:32
[arbitration_ID]:555,[coarse_time]:962,[fine_time]:12
[arbitration_ID]:555,[coarse_time]:A2A,[fine_time]:32
[arbitration_ID]:555,[coarse_time]:AF2,[fine_time]:1E
[arbitration_ID]:555,[coarse_time]:BBA,[fine_time]:12
[arbitration_ID]:555,[coarse_time]:C82,[fine_time]:32
[packet_num]:104,[arbitration_ID]:555,[DLC]:8
```

図 12: IDS の出力例

また，提案手法を実装した IDS を図 13 に示す．FPGA として Intel Cyclone V (5CEBA4F23C7) を選択し，マイクロコンピュータとして Raspberry Pi 3 model B を使用した．

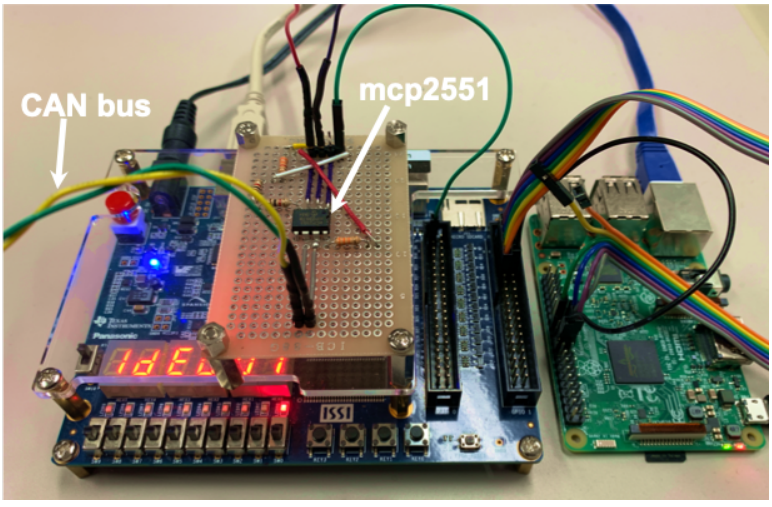


図 13: 提案手法を実装した IDS

5.2 TDC の実装

実装した TDC の時間分解能の性能について述べる．実装した TDC で 20 ns のパルスを計測したところ 92 個目の遅延素子まで信号が伝達した．同様に，40 ns のパルスを計測すると 183 個の遅延素子まで信号が伝達した．これより，1 つあたりの遅延素子の遅延時間の平均は $\frac{40-20}{183-92} = 219$ ps となる．続いて，真値 20 ns のパルスを 5 万回測定した値と真値との平均平方二乗偏差 (Root Mean Square Error: RMSE) を計算した．結果として，RMSE は 154.011 ps となり，実装した TDC は時間分解能は 154 ps であることがわかった．

TDC によって観測された遅延時間を図 14 に示す．2 つの ECU から 3 つの Arbitration ID をプロットしている．ECU a の Arbitaration ID は 50 ns の周辺にプロットされており，ECU b の Arbitration ID は 110 ns の周辺にプロットされている．したがって，2 つの ECU から送信される CAN メッセージの Arbitration ID に依らず，識別可能であるといえる．

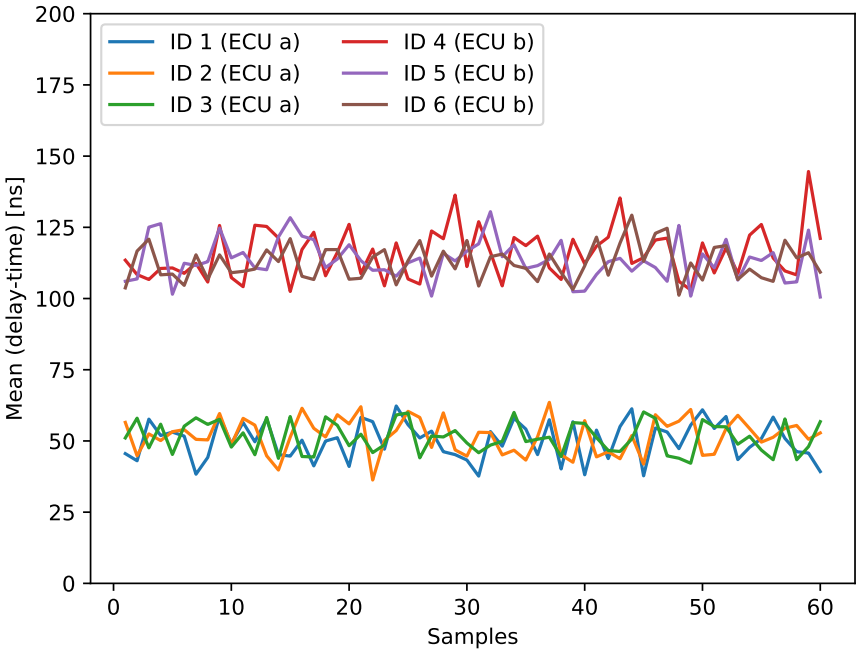


図 14: 実車 A の 2 つの ECU の遅延時間の比較

5.3 プロトタイプ IDS の受信性能

本節では，プロトタイプ IDS の受信性能について述べる．従来手法 [11] のプロトタイプの実装では，CAN メッセージを受信し終えてから計測結果を出力し終わるまでは新しい CAN メッセージを受信することができなかった．これはソフトウェア処理により計測を行っていることに起因している．一方で，提案手法のプロトタイプ IDS の計測を行う部分はハードウェアで構成されている．さらに，プロトタイプ IDS は CAN の信号の立ち上がり毎に計測データを FIFO モジュールへ挿入するため，FIFO モジュールのキューが溢れない限り CAN メッセージの計測結果を出力し続けることができる．受信性能の評価方法として，研究室内の CAN バスプロトタイプの 1 つの ECU から 1 万個の CAN メッセージを送信し，その送信間隔を段階的に減少させた時の CAN メッセージロス率を計測する実験を行う．また，CAN のボーレートは実際の車両で現在最も使用されている 500 kbps に設定した．

図 15 に，プロトタイプ IDS の CAN バス占有率を変化させた時の CAN メッセージロス率を示す．図 15 では，プロトタイプ IDS における FIFO モジュールのキューの長さが 512，2048，および，8192 の時のロス率を表している．これより，キューの長さが 512，2048 の場合では，バス占有率が増加するにつれてロス率が増加していることが確認できる．一方で，キューの長さが 8192 の場合では，ロス率が増加することはなかった．したがって，提案手法のプロトタイプ IDS の実装ではバス占有率が 100 % のような場合であっても CAN メッセージを見逃すことなく計測可能なことが確認できた．

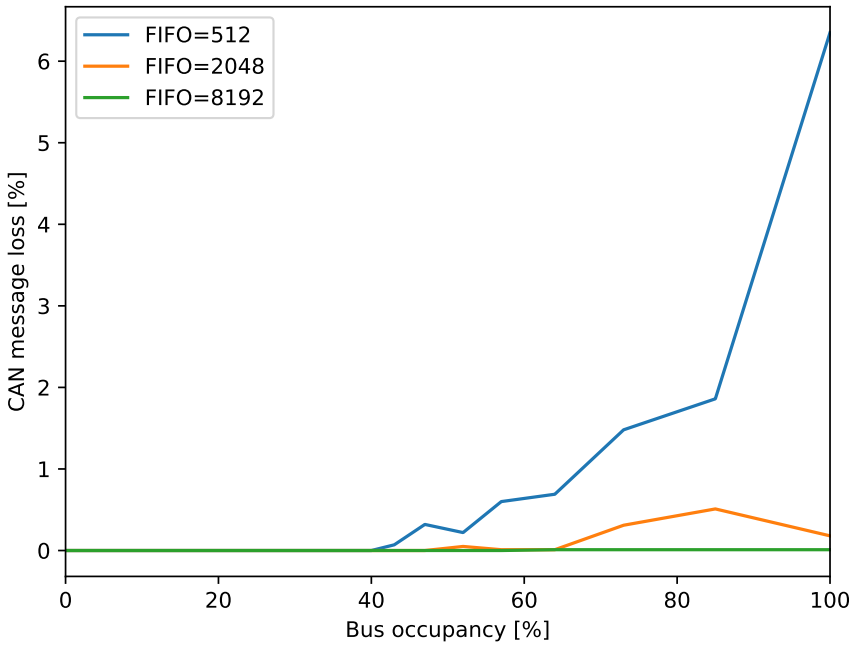


図 15: プロトタイプ IDS の CAN バス占有率を変化させた時の CAN メッセージロス率

6. 評価

本章では，まず評価環境と本研究で想定するアタッカーモデルについて述べる．次に，4.3 節で述べた統計量に関する Relief-F の結果から，提案手法で用いる特徴量を決定する．そして，遅延時間を用いた従来手法と提案手法の平均正解率，および，学習アルゴリズムを変更した時の平均正解率を評価する．さらに，想定するアタッカーモデルの検出率についても評価を行う．最後に，温度変化させた環境における提案手法の平均正解率の変動について評価を行う．

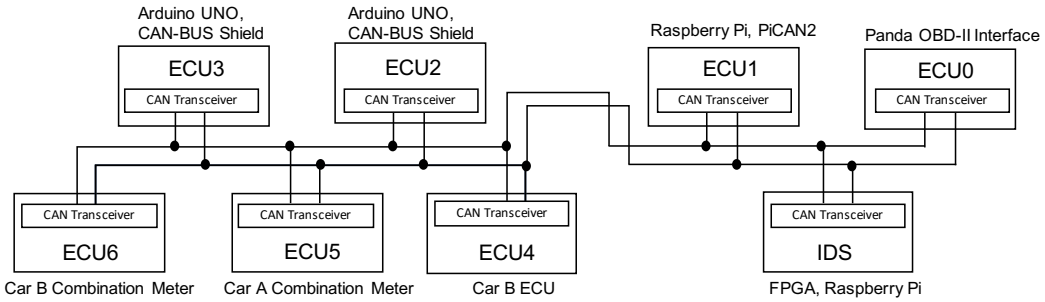
6.1 評価環境・アタッカーモデル

本節では，提案手法を評価するために 2 つの評価環境について述べる．図 16 (a) に研究室内の CAN バスプロトタイプを示す．ECU0 は Panda OBD-II インターフェース [2] である．ECU1 は Raspberry Pi 3 に CAN のインターフェースとなる PiCAN 2 board を乗せた模擬 ECU であり，ECU2，3 は Arduino UNO である．ECU4 はある実車 B の実際の ECU であり，ECU5，6 もそれぞれ実際の車両のコンビネーションメータである．ECU4，5，および，6 が送信する CAN メッセージは直接制御することはできないが，ECU4，5，および，6 は異なる Arbitration ID の CAN メッセージを周期的に送信しているため，提案手法ではこのメッセージを用いて分類を行う．

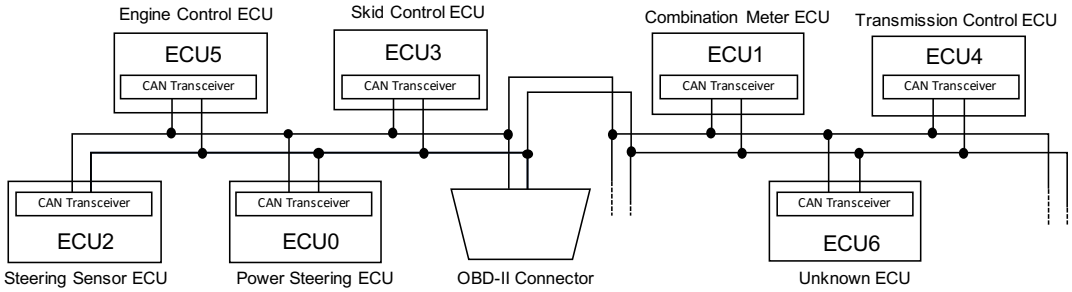
図 16 (b) に実車 A の CAN バスの一部を示す．CAN バスには 7 つの ECU が接続されており，それぞれがいくつかの CAN メッセージを周期的に送信している．提案手法の実車 A での評価は学習データのみ停車時と走行時の 2 つの走行パターンを行なった時のデータであり，テストデータは停車時のデータから構成される．

次に，評価で想定するアタッカーモデルを定義する．図 17 に想定するアタッカーモデルを示す．1 つ目のアタッカーモデルは Unmonitored ECU であり，Jeep Cherokee の攻撃事例 [2] と OBD-II ポート経由の攻撃事例 [45] に基づいている．Jeep Cherokee への攻撃は，通常は CAN メッセージの受診のみを行う正当な ECU が不正にソフトウェアアップデートされ，任意の CAN メッセージを送信可能に

[2] https://comma.ai/shop/products/panda-obd-ii-dongle



(a) CAN バスプロトタイプ



(b) 実車 A の CAN バス

図 16: 評価環境

した．OBD-II ポート経由の攻撃は，スマートフォンから車両情報を OBD-II ポート経由で取得できる社外品ドングルの接続を攻撃者が奪取することで，社外品ドングルから任意の CAN メッセージを送信可能にした．したがって，この 2 つの例は通常は IDS が監視していない ECU から CAN メッセージが送信される例となり，以降では Unmonitored ECU と定義する．

2 つ目のアタッカーモデルは Compromised ECU であり，外部のネットワークと通信可能なインターフェースを持つ ECU が悪用されることを想定している．このアタッカーモデルでは，Unmonitored ECU とは異なり，通常の場合でも IDS が監視している ECU からの攻撃となり，以降では Compromised ECU と定義する．

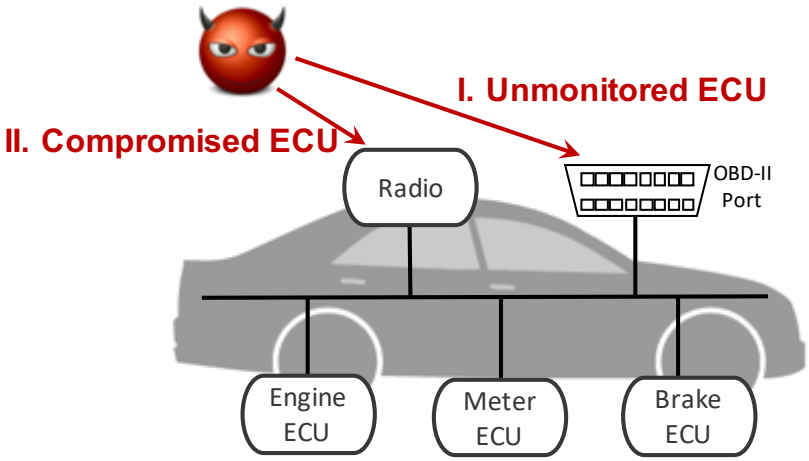


図 17: アタッカーモデル

6.2 特徴選択

4.3 節で述べたように，遅延時間を基本的な統計量に変換したデータに対し，Relief-F を適用し重要な特徴量を明らかにする．Relief-F の実行には機械学習ツールである Weka 3 Toolkit [46] を使用した．Relief-F の結果を表 3 に示す．学習モデルの複雑さの低減と特徴量の算出にかかる時間の削減のため，提案手法では Relief-F の重みが CAN バスプロトタイプと実車 A の両方で 0.01 以上である統計量のみを特徴量として用いる．その結果，Energy と Variance を除く 8 つの統計量が選定された．以降では，選定された統計量を特徴量として定義する．

6.3 遅延時間に基づく手法の比較評価

本節では，遅延時間に基づく従来手法 [11] と提案手法の比較評価を行う．CAN バスプロトタイプにおける比較を表 4 に示す．従来手法は，時間分解能 20 ns で特徴量として平均と標準偏差を用いる．一方で，提案手法では，時間分解能 154 ps で特徴量として Relief-F に基づいて選択された 8 つの統計量を用いる．また，学習アルゴリズムは従来手法で用いられた K-Nearest Neighbor を従来手法と提案手法で用いている．表 4 から，CAN バスプロトタイプにおける平均正解率はそれぞれ，従来手法は 81.43 %，提案手法は 99.57 % となった．したがって，時間

表 3: Relief-F による特徴のランク付け

ランク	CAN バス プロトタイプ	重み	実車 A	重み
1	Mean	0.11025	Stdev (fine time)	0.09311
2	Min	0.08773	Mean	0.05028
3	Root Mean Square	0.05644	Root Mean Square	0.04833
4	Max	0.04696	Min	0.04613
5	Kurtosis	0.03398	Kurtosis	0.04090
6	Stdev (fine time)	0.02949	Skewness	0.03694
7	Skewness	0.02307	Max	0.02468
8	Standard Deviation	0.01282	Standard Deviation	0.01746
9	Energy	0.00878	Energy	0.01639
10	Variance	0.00104	Variance	0.00723

分解能の改善と特徴量の選択によって平均正解率が改善されることが確認された．また，時間分解能 154 ps で特徴量として平均と標準偏差を用いた場合の平均正解率は 96.55% であり，時間分解能 20 ns で特徴量として Relief-F に基づいて選択された 8 つの統計量を用いた場合の平均正解率は 85.55% であった．したがって，CAN バスプロトタイプにおいては，時間分解能の改善が平均正解率に大きく寄与していることがわかった．

表 4: 遅延時間に基づく手法の比較 (CAN バスプロトタイプ)

	特徴量 (平均と標準偏差)	特徴量 (Relief-F による選択)
時間分解能 (20 ns)	81.43%	85.55%
時間分解能 (154 ps)	96.55%	99.57%

次に，実車 A における比較を表 5 に示す．実車 A における平均正解率はそれ

ぞれ，従来手法は 76.75％，提案手法は 94.10％ となった．したがって，実車 A においても時間分解能の改善と特徴量の選択によって平均正解率が改善されることが確認された．また，時間分解能 154 ps で特徴量として平均と標準偏差を用いた場合の平均正解率は 84.15% であり，時間分解能 20 ns で特徴量として Relief-F に基づいて選択された 8 つの統計量を用いた場合の平均正解率は 83.39% であった．したがって，実車 A においては，時間分解能の改善と特徴量の選択の両方が平均正解率に寄与していることがわかった．

表 5: 遅延時間に基づく手法の比較 (実車 A)

	特徴量 (平均と標準偏差)	特徴量 (Relief-F による選択)
時間分解能 (20 ns)	76.75%	83.39%
時間分解能 (154 ps)	84.15%	94.10%

6.4 送信元識別精度に関する評価

本章では，様々な学習アルゴリズムにおける平均正解率の比較と，各 ECU の平均正解率を示す．

6.4.1 CAN バスプロトタイプにおける送信元識別精度

まず，研究室内の CAN バスプロトタイプにおける提案手法の CAN メッセージの送信元識別能力を評価する．CAN バスプロトタイプのトポロジは図 16 (a) と同様である．

各 ECU から 9000 メッセージを送信し，その遅延時間を観測した．9000 メッセージから観測された遅延時間は，Relief-F の結果の重みが 0.01 以上であった 8 つの特徴量に変換され，その特徴量を 5 つの学習アルゴリズムで学習した．学習データとテストデータをそれぞれ 80%と 20%に分けて，学習モデルの評価を行う．提案手法の評価には $K = 5$ の時の層化 K 分割交差検証を用いた．

層化 K 分割交差検証の結果を，表 6 に示す．これより，Random Forest Classifier の平均正解率 99.67 % が最も高いことがわかった．また，Random Forest Classifier における層化 K 分割交差検証のある 1 回の時の混同行列を図 18 に示す．これより，最も高い正解率では 100.00%であり，最も低い正解率では 98.60%となった．

表 6: 各学習アルゴリズムにおける平均正解率 (CAN バスプロトタイプ)

学習アルゴリズム	平均正解率
Multilayer Perceptron	98.65%
K-Nearest Neighbor	99.57%
Decision Tree Classifier	99.48%
Random Forest Classifier	**99.67%**
Support Vector Machine (Gaussian Kernel)	98.98%

図 18: Random Forest Classifier による CAN バスプロトタイプにおける各 ECU の分類結果

6.4.2 実車 A における送信元識別精度

次に，実車 A における提案手法の CAN メッセージの送信元識別能力を評価する．CAN バスプロトタイプのトポロジは図 16 (b) と同様である．

実車環境では，それぞれの ECU から個別に CAN メッセージをキャプチャできないため，実車 A では合計で 20 万メッセージの遅延時間を OBD-II ポート経由で観測した．CAN バスプロトタイプと同様に遅延時間を 8 つの特徴量に変換し，学習した．また，CAN バスプロトタイプにおける評価と同様に，学習データとテストデータをそれぞれ 80%と 20%に分けて，$K = 5$ の時の層化 K 分割交差検証を行った．

層化 K 分割交差検証の結果を，表 7 に示す．これより，Random Forest Classifier の平均正解率 95.94 % が最も高いことがわかった．また，Random Forest Classifier における層化 K 分割交差検証のある 1 回の時の混同行列を図 19 に示す．これより，最も高い正解率では 100.00%であり，最も低い正解率では 91.67%となった．

表 7: 各学習アルゴリズムにおける平均正解率 (実車 A)

学習アルゴリズム	平均正解率
Multilayer Perceptron	95.56%
K-Nearest Neighbor	94.10%
Decision Tree Classifier	94.45%
Random Forest Classifier	**95.94%**
Support Vector Machine (Gaussian Kernel)	93.44%

6.5 攻撃者識別精度に対する評価

本節では，想定するアタッカーモデルに対する提案手法の攻撃検出精度について評価を行う．

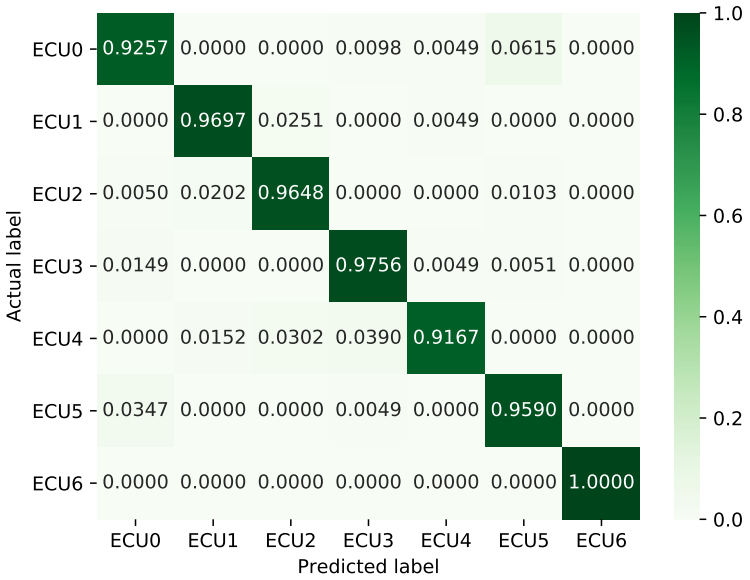


図 19: Random Forest Classifier による実車 A における各 ECU の分類結果

6.5.1 CAN バスプロトタイプにおける Unmonitoring ECU

本項では，提案手法の Unmonitoring ECU に対する攻撃者識別精度を評価する．CAN バスプロトタイプに ELM327 を取り付け，ECU3 が送信する Arbitration ID x を ELM327 からも送信を行った．この時の Arbitration ID x の送信元識別結果を表 8 に示す．

表 8: Unmonitored ECU と ECU3 が Arbitration ID x のメッセージを送信した際の分類結果

	Predicted: Attack	Predicted: Normal
Actual: Attack	100.00%	0.00%
Actual: Normal	0.68%	99.32%

表 8 の Attack ラベルは提案手法が Arbitration ID x のメッセージの送信元を ECU3 以外に分類した時のラベルであり，Normal ラベルは提案手法が Arbitration

ID x のメッセージの送信元を ECU3 に分類した時のラベルである．表 8 より，提案手法は ELM327 からの攻撃メッセージを 100.00 % で検出可能で，かつ，偽陰率 0.00 % であることが確認できた．

6.5.2 実車 A における Compromised ECU

本項では，提案手法の Compromised ECU に対する攻撃者識別精度を評価する．実車 A の OBD-II ポートに，Compromised ECU として CAN バスプロトタイプで用いた ECU2 (Arduino UNO) を設置し，車速に割り振られている Arbitration ID y を Compromised ECU からも送信した．この時の Arbitration ID y の送信元識別結果を表 9 に示す．

表 9: Compromised ECU と ECU3 が Arbitration ID y のメッセージを送信した際の分類結果

	Predicted: Attack	Predicted: Normal
Actual: Attack	100.00%	0.00%
Actual: Normal	5.70%	94.30%

表 9 における Attack ラベルと Normal ラベルも，表 8 と同様に，Attack ラベルは提案手法が Arbitration ID y のメッセージの送信元を ECU3 以外に分類した時のラベルであり，Normal ラベルは提案手法が Arbitration ID y のメッセージの送信元を ECU3 に分類した時のラベルを示す．表 9 より，提案手法は実車 A においても Compromised ECU からの攻撃メッセージを 100.00 % で検出可能で，かつ，偽陰率 0.00 % であることが確認できた．

6.6 Concept Drift における送信元識別精度の評価

研究室内の CAN バスプロトタイプの周辺温度を変化させるために，CAN バスプロトタイプに図 20 に示すような段ボールをかぶせ，段ボール内にヒートガンで熱風を送るようにした．図 20 の CH1〜4 は，各温度センサを表しており，段

ボール内には 4 つ設置されている．CH1〜3 の温度センサは実験中の温度を逐次確認するために使用し，CH4 の温度センサはプロトタイプ IDS に接続されており温度と遅延時間を同時に計測するために使用した．図 21 に，温度センサを追加したプロトタイプ IDS の出力例を示す． 図 21 に示すように，測定期間の終了時

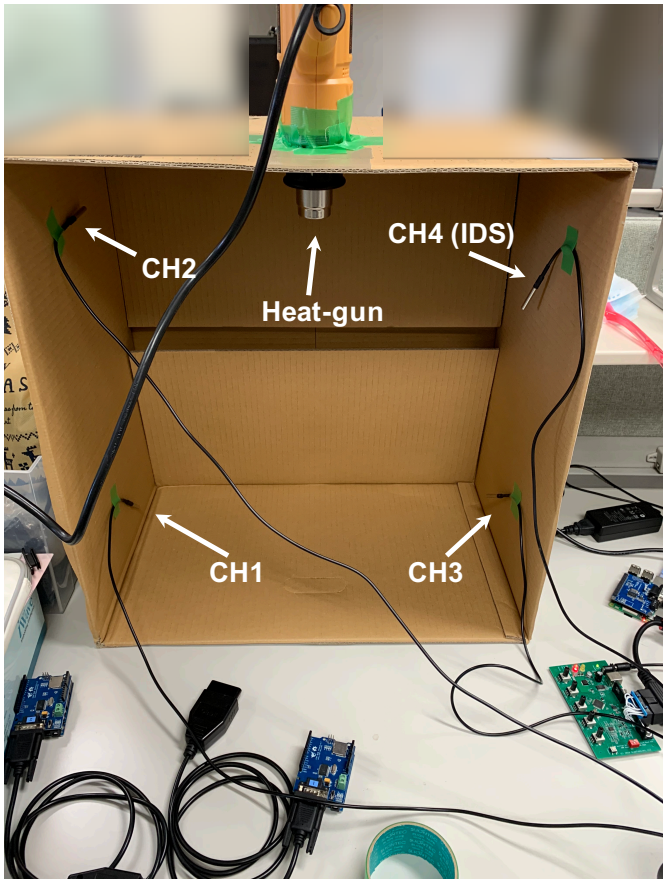


図 20: CAN バスプロトタイプの周辺温度を変化させるための実験環境

と同時に温度センサからデータを取得する．したがって，1 メッセージ毎に 1 度温度データを得る．

次に，CAN バスプロトタイプの ECU を用いて遅延時間の変化を調べた．表 10 に温度変化に対する遅延時間の線形回帰の結果を示す．また，表 10 は，30 °C〜45 °C のデータを用いて線形回帰を行った結果である．R^2 は独立変数 (温度) が従属変数 (遅延時間) のどの程度説明できるかを示す指標であり，Mean Squere

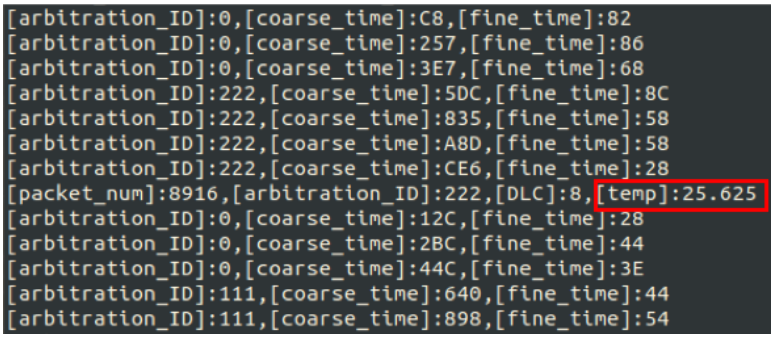


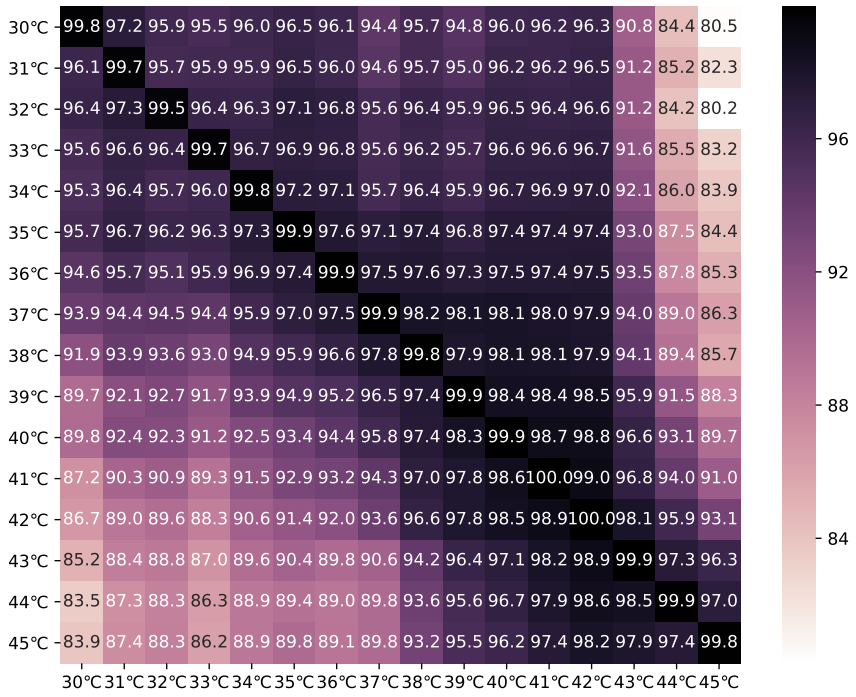
図 21: 温度センサを追加したプロトタイプ IDS の出力例

表 10: 温度変化に対する遅延時間の線形回帰の結果 (CAN バスプロトタイプ)

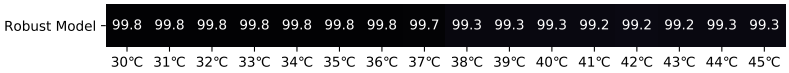
ECU (CAN トランシーバ)	R^2	MSE
ECU0 (TJA1040)	0.0006	0.9994
ECU1 (MCP2551)	0.8544	0.1456
ECU2 (MCP2551)	0.8242	0.1758
ECU3 (MCP2551)	0.6947	0.3053
ECU4 (TJA1040)	0.0948	0.9052
ECU5 (SE706)	0.0706	0.9294
ECU6 (TJA1042)	0.0102	0.9898

Error (MSE) は 1 から R^2 を引いた値となる．表 10 より，CAN トランシーバの種類によって決定係数 R^2 に差がみられることがわかる．CAN トランシーバが MCP2551 である ECU1，2，および，3 は MSE よりも R^2 が高い．一方で，CAN トランシーバが TJA1040，TJA1042，および，SE706 の場合，R^2 はほぼ 0 であった．したがって，温度変化によって Concept Drift が生じる ECU1，2，および，3 に対しては，温度変化に対応可能な学習モデルが必要となる．

以降では，4.5 節で述べた温度を特徴量として用いる学習モデルの評価を行う．まず，各温度における 8 つの遅延時間の特徴量のみを学習させたモデルを，温度の異なるデータに対して分類を行った時の平均正解率を図 22 (a) に示す．図 22 (a) の縦軸が学習データを示し，横軸が温度のテストデータを示す．つまり，縦軸 30 °C の学習データを用いて構築した学習モデルで，45 °C のテストデータを分



(a) 各温度における学習モデルの平均正解率の低下



(b) 温度変化に対してロバストな学習モデルの平均正解率

図 22: 温度変化における送信元識別精度

類した時の平均正解率は 80.5 % となる．図 22 (a) より，学習データの温度とテストデータの温度の差が大きくなるほど，平均正解率が減少することがわかった．

次に，温度を特徴量に加えた場合の学習モデルによる平均正解率を図 22 (b) に示す．図 22 (b) より，1 つの学習モデルのみで 30 °C から 45 °C のテストデータに対し，平均正解率 99 % 以上となった．したがって，温度を特徴量に加えることによって，温度変化による Concept Drift に対してロバストな学習モデルを構

築可能であることが確認できた.

7. 考察

本章では，6 章の評価結果に基づいて，従来手法と比較した時の提案手法の有効性を考察する．

7.1 遅延時間に基づく手法の比較

提案手法の送信元識別の平均正解率は CAN バスプロトタイプと実車 A の環境でそれぞれ，99.67%と 95.94%となった．従来手法 [11] と同程度の時間分解能 20 ns で観測したデータを分類した結果は，それぞれ 81.43%と 76.75%であった．これより，時間分解能を向上させることでより高い正解率で分類可能なことが確認できた．

さらに，CAN バスプロトタイプにおいて，時間分解能を変化させた場合の提案手法の平均正解率の評価を行った．時間分解能は TDC における複数の遅延セルを 1 つの遅延セルと見なすことで，時間分解能を変化させた．例えば，1 つの遅延セルで 0.154 ns だけ遅延する場合，4 つの遅延セルをまとめて 1 つの遅延セルと見なすことで，0.616 ns の時間分解能を得ることができる．時間分解能を変化させた場合の提案手法の平均正解率を表 11 に示す．また，平均正解率の算出には提案手法で選択した 8 つの統計量を使い，学習アルゴリズムは Random Forest Classifier を用いた．表 11 より，時間分解能が改善されることで，平均正解率が向上していることがわかる．したがって，時間分解能の改善に伴い，提案手法の平均正解率が向上することが確認できた．

表 11: 時間分解能を変化させた場合の提案手法の平均正解率

時間分解能 (ns)	20.000	7.700	3.850	1.540	0.616	0.154
平均正解率 (%)	87.20	93.02	97.47	98.31	98.46	99.67

7.2 従来手法との比較

表 12 に，信号の物理的特徴に基づく従来手法と提案手法の比較を示す．まず，各手法の平均正解率は，Choi らの手法では 96.48%，BTMonitor [35] では 1 つのメッセージのみを用いた場合 ($N = 1$) の 90.04%，および，それ以外の手法では 99%以上となっている．よって遅延時間を用いた提案手法も分類精度の高い電圧ベースの従来手法に匹敵する平均正解率であるといえる．

表 12: 送信元識別手法の比較, True Positive Rate (T.P.R.), Sampling Rate (S.R.), Best Number of Sampling per message (B.N.S.), Worst Number of Sampling per message (W.N.S.), Time Complexity (T.C.)

	Choi ら [8]	Scission [6]	SIMPLE [34]	BTMonitor [35]	提案手法
T.P.R.	96.48%	99.85%	100.00%	90.04% ($N = 1$)	99.67%
S.R.	2GS/s	20MS/s	500KS/s	50MS/s	-
B.N.S.	198×10^3	1980	47	10	5
W.N.S.	444×10^3	4440	111	28	14
T.C.	$\Omega(n \log n)$	$\Omega(n \log n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$
Cost	High	Low	Low	Low	Low

次に，サンプリングレートとコストについては Choi らの手法ではオシロスコープを用いて 2.5GS/s と非常に高いサンプリングレートを用いている．したがって，コストの側面においてこの手法は適していない．Scission，SIMPLE，および，BTMonitor はそれぞれ 20MS/s，500KS/s，50MS/s と比較的低いサンプリングレートを用いる．提案手法は電圧値のような連続値を元に送信元識別せず，立ち上がり時のみサンプルを行う．したがって，Scission，SIMPLE，BTMonitor，および，提案手法はコストの側面において優位性があるといえる．

続いて，各手法が 1 つの CAN メッセージ毎に行うサンプリング回数について議論する．Choi らの手法，Scission，および，SIMPLE の 1 メッセージあたりのサンプリング回数は，データフィールドの長さに依存する．したがって，データフィールドが最短 (0 byte)・最長 (8 byte) の場合を考える．最短の場合，データ

フィールドは 0 byte となるため，図 2 より，CAN メッセージの全体は 47 bit となる．CAN のビットレートが 500 kbps の場合，1 bit の送信時間は 2µs である．これより，各手法のサンプリングレートに $47 \times 2 \times 10^{-6}$ を掛けると，各手法の 1 メッセージあたりのサンプリング回数はそれぞれ 198×10^3，1980，47 となった．同様に，最長の場合，データフィールドは 8 byte となるため，CAN メッセージの全体は 111 bit となる．これを用いて，各手法の場合のサンプリング回数はそれぞれ 444×10^3，4440，111 となった．提案手法と BTMonitor の 1 メッセージ毎のサンプリング回数はデータフィールドの長さではなく，信号の 0 から 1 への遷移回数に依存する．そのため，提案手法と BTMonitor の最小と最大のサンプリング回数を，各ビットの遷移回数が少ない Arbitration ID 0x000 と遷移回数が多い Aribitration ID 0x555 で議論する．Arbitration ID 0x000 の場合，サンプリング回数 5 で，0x555 の場合，サンプリング回数 14 となった．また，BTMonitor は立ち上がりだけでなく立ち下がりの時にもデータ取得を行うため，BTMonitor のサンプリング回数は提案手法の 2 倍となる．これより，提案手法が最もデータ取得の段階で少ないサンプリング回数であることがわかった．言い換えると，提案手法はデータ取得の次の特徴抽出の段階において n が最も小さいため，他の手法と比べ軽量な処理で特徴抽出までを行える．

最後に，計算量について議論する．Choi らの手法と Scission では時間領域と周波数領域の特徴量を使用する．周波数領域の特徴量を算出するためにフーリエ変換を行うため，特徴抽出に $\Omega(n \log n)$ 要する．SIMPLE と BTMonitor は時間領域の特徴量で統計量として平均の算出を行うため，$\Theta(n)$ 要する．提案手法では，表 2 の統計量を用いるため，$\Theta(n)$ 要する．したがって，SIMPLE，BTMonitor，および，提案手法は他の手法に比べ軽量な処理であるといえる．

7.3 温度が変化する環境における手法の比較

6.6 節より，提案手法では各温度における各 ECU の遅延時間を学習することで 1 つの学習モデルのみで，30 °C〜45 °C のテストデータに対し，平均正解率 99% 以上となった．従来手法と比較すると，複数の学習モデルを用いる手法 [35] では学習モデルの数が増加するにつれメモリ使用量が増加するため，1 つの学習モ

デルを用いる提案手法の方が優位性があるといえる．さらに，学習モデルを逐次更新し特徴量をトラッキングする手法 [7] [10] [41] では Hill-climbing-style attack に脆弱である可能性があるが，提案手法は学習モデルの更新を行わないため，従来手法に比べ提案手法は Hill-climbing-style attack に堅牢であるといえる．一方で，CAN に供給される電圧がトポロジの変更等で変動する場合においても，CAN メッセージの送信元を特定するための特徴量が変動することが確認できている．そこで，今後の課題として，この Concept Drift に対応するために，少数の CAN メッセージのみ MAC 等で認証を行い，認証した CAN メッセージを用いて学習モデルを更新しする方法等を検討する．

7.4 今後の課題

提案手法の評価より，提案手法は最大で 99.67%で CAN メッセージの送信元を識別可能であることが示せた．しかしながら，提案手法を実際の車両に用いる場合，0.33%で CAN メッセージの送信元を誤ってしまうことは実用上問題となる．したがって，分類精度をさらに向上させることが必須となる．そこで，今後の課題として遅延時間と電圧の両方を用いて分類精度を向上させることを検討する．

CAN のトポロジの変更による電圧変動によって，遅延時間がわずかに変動することがわかっている．この変動により，送信元識別精度が低下する恐れがある．そのため，Scission における学習モデルの更新 [41] と同様に，事後確率が一定以上であるメッセージのみを用いて学習モデルを安全に更新する方法を検討する．

8. おわりに

自動車内部のネットワークである CAN へのサイバー攻撃が深刻な問題になっている．CAN のデータフォーマットには送信元を識別する ID がないため，攻撃者から送信された不正なメッセージを区別できない．したがって，CAN メッセージの送信元識別手法を確立することで，不正なメッセージの検知・無効化が可能となる．

CAN における IDS として，CAN トランシーバにおける信号の立ち上がり・立ち下がりの遅延時間に着目した送信元識別手法 [11] が提案されている．この手法では，各 ECU の遅延時間の差が計測デバイスの時間分解能より低い場合，ECU を正しく分類できない．そこで，遅延時間の高時間分解能観測により ECU の識別精度を向上させることが期待できる．

本研究では，TDC を用いた遅延時間の高分解能観測に基づく送信元識別手法を提案を行なった．FPGA およびマイクロコンピュータにより計測デバイスを実装し，提案手法の ECU の分類に関する評価を行った．評価結果から，提案手法では研究室内の CAN バスのプロトタイプで 99.67%，実車で 95.94%の平均正解率となった．

今後の課題として，遅延時間と電圧の両方を用いて分類精度を向上させる手法の検討と，CAN のトポロジの変更による電圧変動が起こった場合の学習モデルを更新する手法に関する検討を行う．

謝辞

主指導教員であり，適切な研究指導をしていただき，対外発表等の経験を積ませていただくなど様々な側面から研究のサポートをしてくださいました本学情報基盤システム学研究室の藤川和利教授に心から感謝致します．副指導教員であり，研究の方向性について的確な助言をくださいました本学情報セキュリティ工学研究室の林優一教授に心から感謝致します．副指導教員であり，研究や対外発表に関してご意見を頂き，多くのご助言を頂きました本学情報基盤システム学研究室の新井イスマイル准教授に心から感謝致します．学術論文誌や国際会議に論文投稿する際に，何度も何度も論文添削にご尽力をいただき，深謝しております．博士後期課程の 3 年間も何卒宜しくお願いします．研究についてのご助言や指導だけでなく，学内システムやネットワーク等の運用方法について等様々な側面において熱心にご指導くださいました本学情報基盤システム学研究室の垣内正年助教に心から感謝致します．研究方針や対外発表の練習において日頃から多くのご助言をくださいました本学情報基盤システム学研究室の油谷曉助教に心より感謝致します．また，急遽実車両で実験したいという私の勝手な要望も，快く受け入れて下さったことについても心から感謝しております．学術論文誌における研究方針を御教授して下さった広島市立大学情報科学研究科の井上博之准教授に心から感謝致します．自動車セキュリティ等に関する多角的な視点でのご助言や，社会人一年目で忙しい身であるにもかかわらず国際会議論文を添削して下さった北川智也氏に心から感謝致します．また，様々な面から研究活動を支援してくださいました本学総合情報基盤センターの辻元理恵女史，中野彩子女史に心から感謝致します．同一の研究分野の学生として研究の議論や国際会議論文の添削をして頂いた Araya Kibrom Desta 氏には心から感謝致します．研究面や生活面において，多くの支援をしていただいた本学情報基盤システム学研究室の皆様に心から感謝致します．

最後に私の意思を尊重し，研究活動に関する理解を示すとともに，経済面や生活面において多大な支援を頂きました家族に心から感謝致します．

参考文献

[1] Robert Bosch GmbH, "CAN Specification Version 2.0," http://esd.cs.ucr.edu/webres/can20.pdf, (Accessed on 07/08/2019).

[2] C. Miller and C. Valasek, "Remote Exploitation of An Unaltered Passenger Vehicle," *Black Hat USA*, vol. 2015, pp. 1–91, 2015.

[3] S. Nie, L. Liu, and Y. Du, "Free-Fall: Hacking Tesla from Wireless to CAN Bus," *Briefing, Black Hat USA*, pp. 1–16, 2017.

[4] A. Van Herrewege, D. Singelee, and I. Verbauwhede, "CANAuth-A Simple, Backward Compatible Broadcast Authentication Protocol for CAN Bus," in *ECRYPT Workshop on Lightweight Cryptography*, vol. 2011, 2011.

[5] B. Groza, S. Murvay, A. Van Herrewege, and I. Verbauwhede, "LiBrA-CAN: A Lightweight Broadcast Authentication Protocol for Controller Area Networks," in *International Conference on Cryptology and Network Security*. Springer, 2012, pp. 185–200.

[6] M. Kneib and C. Huth, "Scission: Signal Characteristic-Based Sender Identification and Intrusion Detection in Automotive Networks," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2018, pp. 787–800.

[7] K.-T. Cho and K. G. Shin, "Viden: Attacker Identification on In-Vehicle Networks," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 1109–1123.

[8] W. Choi, H. J. Jo, S. Woo, J. Y. Chun, J. Park, and D. H. Lee, "Identifying ECUs Using Inimitable Characteristics of Signals in Controller Area Networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 6, pp. 4757–4770, 2018.

[9] P.-S. Murvay and B. Groza, "Source Identification Using Signal Characteristics in Controller Area Networks," *IEEE Signal Processing Letters*, vol. 21, no. 4, pp. 395–399, 2014.

[10] K.-T. Cho and K. G. Shin, "Fingerprinting Electronic Control Units for Vehicle Intrusion Detection," in *25th USENIX Security Symposium (USENIX Security 16)*, 2016, pp. 911–927.

[11] 北川 智也, "車載ネットワークにおける信号の物理的特徴を利用した攻撃ノード識別手法の提案," http://library.naist.jp/mylimedio/dllimedio/show.cgi?bookid=100229936, 2018, NAIST 修士論文.

[12] Q. Wang, Y. Qian, Z. Lu, Y. Shoukry, and G. Qu, "A Delay Based Plug-in-Monitor for Intrusion Detection in Controller Area Network," in *2018 Asian Hardware Oriented Security and Trust Symposium (AsianHOST)*. IEEE, 2018, pp. 86–91.

[13] J. Liu, S. Zhang, W. Sun, and Y. Shi, "In-Vehicle Network Attacks and Countermeasures: Challenges and Future Directions," *IEEE Network*, vol. 31, no. 5, pp. 50–58, 2017.

[14] Craig Smith, "The Car Hacker's Handbook A Guide for the Penetration Tester," https://docs.alexomar.com/biblioteca/thecarhackershandbook.pdf, (Accessed: 2019-12-17).

[15] The Linux Foundation, "Automotive Grade Linux," https://www.automotivelinux.org/, (Accessed on 07/08/2019).

[16] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, T. Kohno *et al.*, "Comprehensive Experimental Analyses of Automotive Attack Surfaces," in *USENIX Security Symposium*, vol. 4. San Francisco, 2011, pp. 447–462.

[17] T. Ziermann, S. Wildermann, and J. Teich, "CAN+: A new backward-compatible Controller Area Network (CAN) protocol with up to 16x higher data rates," in *Proceedings of the Conference on Design, Automation and Test in Europe*. European Design and Automation Association, 2009, pp. 1088–1093.

[18] R. Kurachi, Y. Matsubara, H. Takada, N. Adachi, Y. Miyashita, and S. Horihata, "CaCAN-Centralized Authentication System in CAN (Controller Area Network)," in *14th Int. Conf. on Embedded Security in Cars (ESCAR 2014)*, 2014.

[19] AUTOSAR, "Classic Platform 4.4.0 - AUTOSAR," https://www.autosar.org/fileadmin/Releases_TEMP/Classic_Platform_4.4.0/Communication.zip, (Accessed on 12/29/2019).

[20] A.-I. Radu and F. D. Garcia, "LeiA: A Lightweight Authentication Protocol for CAN," in *European Symposium on Research in Computer Security*. Springer, 2016, pp. 283–300.

[21] S. Nürnberger and C. Rossow, "–vatiCAN–Vetted, Authenticated CAN Bus," in *International Conference on Cryptographic Hardware and Embedded Systems*. Springer, 2016, pp. 106–124.

[22] J. Van Bulck, J. T. Mühlberg, and F. Piessens, "VulCAN: Efficient Component Authentication and Software Isolation for Automotive Control Networks," in *Proceedings of the 33rd Annual Computer Security Applications Conference*. ACM, 2017, pp. 225–237.

[23] Sancus, "Sancus: Lightweight and Open-Source Trusted Computing for the IoT," https://distrinet.cs.kuleuven.be/software/sancus/index.php, (Accessed on 12/29/2019).

[24] S. Checkoway, L. Davi, A. Dmitrienko, A.-R. Sadeghi, H. Shacham, and M. Winandy, "Return-Oriented Programming without Returns," in *Pro-*

ceedings of the 17th ACM conference on Computer and Communications Security. ACM, 2010, pp. 559–572.

[25] A. Humayed and B. Luo, "Using ID-hopping to Defend against Targeted DoS on CAN," in *Proceedings of the 1st International Workshop on Safe Control of Connected and Autonomous Vehicles*. ACM, 2017, pp. 19–26.

[26] W. Wu, R. Kurachi, G. Zeng, Y. Matsubara, H. Takada, R. Li, and K. Li, "IDH-CAN: A Hardware-Based ID Hopping CAN Mechanism With Enhanced Security for Automotive Real-Time Applications," *IEEE Access*, vol. 6, pp. 54 607–54 623, 2018.

[27] S. Woo, D. Moon, T.-Y. Youn, Y. Lee, and Y. Kim, "CAN ID Shuffling Technique (CIST): Moving Target Defense Strategy for Protecting In-Vehicle CAN," *IEEE Access*, vol. 7, pp. 15 521–15 536, 2019.

[28] H. M. Song, H. R. Kim, and H. K. Kim, "Intrusion Detection System Based on the Analysis of Time Intervals of CAN Messages for In-Vehicle Network," in *2016 international conference on information networking (ICOIN)*. IEEE, 2016, pp. 63–68.

[29] M. Marchetti and D. Stabili, "Anomaly Detection of CAN Bus Messages through Analysis of ID Sequences," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 1577–1583.

[30] W. Wu, Y. Huang, R. Kurachi, G. Zeng, G. Xie, R. Li, and K. Li, "Sliding Window Optimized Information Entropy Analysis Method for Intrusion Detection on In-Vehicle Networks," *IEEE Access*, vol. 6, pp. 45 233–45 245, 2018.

[31] A. Taylor, S. Leblanc, and N. Japkowicz, "Anomaly Detection in Automobile Control Network Data with Long Short-Term Memory Networks," in *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2016, pp. 130–139.

[32] M.-J. Kang and J.-W. Kang, "Intrusion Detection System Using Deep Neural Network for In-Vehicle Network Security," *PloS one*, vol. 11, no. 6, p. e0155781, 2016.

[33] M. Rumez, J. Dürrwang, T. Brecht, T. Steinshorn, P. Neugebauer, R. Kriesten, and E. Sax, "CAN Radar: Sensing Physical Devices in CAN Networks based on Time Domain Reflectometry," *arXiv preprint arXiv:1910.02847*, 2019.

[34] M. Foruhandeh, Y. Man, R. Gerdes, M. Li, and T. Chantem, "SIMPLE: Single-Frame Based Physical Layer Identification for Intrusion Detection and Prevention on In-Vehicle Networks," in *Proceedings of the 35th Annual Computer Security Applications Conference*, 2019.

[35] J. Zhou, P. Joshi, H. Zeng, and R. Li, "BTMonitor: Bit-time-based Intrusion Detection and Attacker Identification in Controller Area Network," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 18, no. 6, p. 117, 2019.

[36] ams AG, "AS6500 Time-to-Digital Converter," https://ams.com/ja/as6500, (Accessed on 10/27/2019).

[37] J. Song, Q. An, and S. Liu, "A High-Resolution Time-to-Digital Converter Implemented in Field-Programmable-Gate-Arrays," *IEEE Transactions on Nuclear Science*, vol. 53, no. 1, pp. 236–241, 2006.

[38] J. Wu and Z. Shi, "The 10-ps Wave Union TDC: Improving FPGA TDC Resolution beyond Its Cell Delay," in *2008 IEEE Nuclear Science Symposium Conference Record*. IEEE, 2008, pp. 3440–3446.

[39] 小林春夫, "様々な時間デジタイザ回路アーキテクチャのタイミングテスト応用への比較検討," 第 *75* 回 *FTC* 研究会, 伊香保, 群馬 *(2016* 年 *7* 月*)*, 2016.

[40] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A Survey on Concept Drift Adaptation," *ACM computing surveys (CSUR)*, vol. 46, no. 4, p. 44, 2014.

[41] M. Kneib, O. Schell, and C. Huth, "On the Robustness of Signal Characteristic-Based Sender Identification," *arXiv preprint arXiv:1911.09881*, 2019.

[42] Robert Bosch GmbH, "CAN with Flexible Data-Rate Specification Version 1.0," https://can-newsletter.org/assets/files/ttmedia/raw/e5740b7b5781b8960f55efcc2b93edf8.pdf, 7 2019, (Accessed on 07/08/2019).

[43] I. Kononenko, "Estimating Attributes: Analysis and Extensions of RELIEF," in *European conference on machine learning*. Springer, 1994, pp. 171–182.

[44] M. Yazici, S. Basurra, and M. Gaber, "Edge Machine Learning: Enabling Smart Internet of Things Applications," *Big Data and Cognitive Computing*, vol. 2, no. 3, pp. 1–17, 2018.

[45] 北川智也, 垣内正年, 猪俣敦夫, 藤川和利, "車載の社外品ドングルに対する近接攻撃の検証," コンピュータセキュリティシンポジウム *2017* 論文集, vol. 2017, no. 2, 2017.

[46] T. C. Smith and E. Frank, "Introducing Machine Learning Concepts with WEKA," in *Statistical genomics*. Springer, 2016, pp. 353–378.

発表リスト

学術論文

Shuji Ohira, Araya Kibrom Desta, Ismail Arai, Hiroyuki Inoue, Kazutoshi Fujikawa, “Normal and Malicious Sliding Windows Similarity Analysis Method for Fast and Accurate IDS against DoS Attacks on In-Vehicle Networks.”, *IEEE Access*, Vol.8, pp.42422-42435, Feb. 2020.

大平修慈, 井上博之, 新井イスマイル, 藤川和利, “車載 LAN へ侵入するマルウェアの証拠保全を行うカーネル上のフォレンジック機構.”, 情報処理学会論文誌, Vol.60, No.3, pp.791-802, Mar. 2019.

国内会議

大平修慈, Araya Kibrom Desta, 新井イスマイル, 藤川和利, “TDC による遅延時間の高時間分解能観測に基づく CAN メッセージの送信元識別手法.”, 研究報告コンピュータセキュリティ（*CSEC*）, pp.1-8, Dec. 2019.

大平修慈, 新井イスマイル, 井上博之, 藤川和利, “車載インフォテインメントシステムにおけるホワイトリストと遅延付加による CAN バス上の DoS 攻撃緩和手法.”, コンピュータセキュリティシンポジウム *2018* 論文集, pp.1128-1133, Oct. 2018.