PRICAR: Privacy Framework for Vehicular Data Sharing with Third Parties

Mert D. Pesé[*], Jay W. Schauer[†], Murali Mohan[†], Cassandra Joseph[†], Kang G. Shin[†], and John Moore[‡]

[*]Clemson University, mpese@clemson.edu
[†]University of Michigan, {jschauer, cassmjo, kgshin}@umich.edu
[‡]Ford Motor Company, jmoor422@ford.com

***Abstract*—In-vehicle infotainment (IVI) platforms are getting increasingly connected. Besides Original Equipment Manufacturer (OEM) apps and services, the next generation of IVI platforms are expected to allow integration of third-party apps, such as usage-based insurance (UBI). Under this anticipated business model, vehicular sensor and event data can be collected and shared with selected third-party apps. However, third-parties can be malicious and have easier access to the driver's data. Several research projects and commercial products also show the possibility of leaking sensitive private information such as vehicle location via seemingly benign vehicular sensors, which can, in turn, harm the driver's privacy. Furthermore, increasing privacy regulations worldwide, such as GDPR, make privacy a major issue for the automotive industry. To overcome these problems, we present PRICAR, a framework for privacy-preserving vehicular data collection and sharing with third-parties. It enforces three data privacy goals — *minimization*, *anonymization* and *sanitization* — while focusing on the last one. We describe and evaluate how to sanitize user data before sharing it with a third-party by adapting two well-known techniques from anomaly detection, *Change-Point Detection* (CPD) and *Entropy-Based Detection* (EBD).**

Keywords*—*Automotive Privacy, Data Sanitization

I. Introduction

Vehicles are becoming increasingly connected. According to [36], automotive applications are expected to constitute 53% of Internet of Things (IoT) data transmitted over 5G by 2023. In future, in-vehicle infotainment (IVI) platforms will likely allow third-party services/apps to collect data from the vehicle using a built-in data connection. Together with the sheer amount of data generated in vehicles (25 GB/h [19]), data sharing capabilities with third-parties could turn into a lucrative after-market business for OEMs. One prominent upcoming IVI platform is Android Automotive OS (AAOS) [13] which is an Android build with some additional vehicle-specific modules for accessing vehicular sensors through Android apps. Numerous OEMs have signed up to use AAOS in their vehicles [28]. Each OEM will be free to choose which third-party apps to allow in their OEM-specific Play Store. Nevertheless, AAOS is highly preferable for third-parties as they will be able to provide an app to multiple OEMs at once instead of customizing it individually for each proprietary platform. However, this potential data monetization may introduce security and privacy risks to the driver. According to Frost & Sullivan [51], data security and privacy are among the most critical drivers or inhibitors in next-generation mobility services.

Privacy attacks on IVIs are an emerging and serious concern. The European Union (EU) has established a privacy standard called *General Data Protection Regulation* (GDPR) in May 2018 [5] that gives more consent opportunities over an individual's data. Although GDPR is only binding for EU residents and entities, OEMs are global companies selling cars worldwide. Hence, GDPR adherence is of great importance to North American and Asian OEMs. Even in the US, there are state-specific privacy laws, such as the California Consumer Privacy Act (CCPA) [1] and its more stringent 2023 update, the California Privacy Rights Act (CPRA) [17].

Although automotive privacy is an area of growing concern and interest in recent years due to increasing connectivity, it has not yet attracted the immediate attention from automotive OEMs or academic researchers. There has been some initial work on classifying the privacy impact of certain vehicular sensors [50]. Issues with the permission model on Android Automotive have also been briefly discussed [47], [48]. This is in stark contrast to a myriad of privacy attacks designed for vehicles [50]. Nevertheless, an end-to-end privacy framework for the connected vehicle ecosystem which satisfies the privacy goals laid out by regulation is missing. To the best of our knowledge, PRICAR is the first automotive privacy framework that attempts to mitigate privacy concerns of sharing drivers' data with automotive third-party entities in connected vehicles.

Adding privacy protection to the vehicle ecosystem comes with some unique challenges as it involves multiple stakeholders, such as the driver/owner of the vehicle (”user”), the car manufacturer (”OEM”), and finally the third-party service provider (”third-party”). In order to design a system architecture for a privacy-preserving framework, we first have to define *privacy goals*. We do this based on existing privacy regulation, specifically GDPR, and compliment them by additional privacy principles. The core principles of GDPR are laid out in Sec. II-B before the definition of our three key privacy goals of data *minimization*, *anonymization* and *sanitization* in Sec. III. In addition to meeting the aforementioned privacy goals in the vehicle, the underlying system design must consider automotive-specific constraints, such as limited in-vehicle computational resources and overall added cost to the OEM (e.g., cellular bandwidth). Although a detailed evaluation of the aforementioned cost-specific metrics are outside of this paper's scope, we still consider them during the draft of our framework.

An overview of the entities involved in PRICAR is depicted in Fig. 1. Third-party apps are submitted to the OEM's app store. The user's vehicle will transmit the generated driving data based on the selection of the offered third-party app to

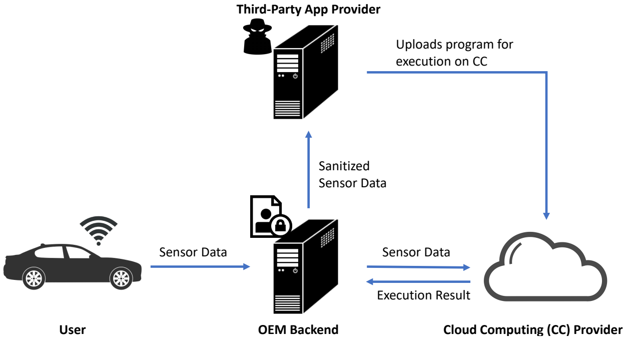


Fig. 1: Entities in the connected vehicle ecosystem

the OEM's backend. The third-party app provider expects their apps to be used locally on their end without sharing their source code and thus protect their Intellectual Property (IP). Since the OEM is responsible for ”protecting” the privacy of user data, it cannot directly share the data with the third-party since it will relinquish control of the data at this point and cannot guarantee GDPR adherence. As a result, we introduce a neutral entity (”cloud computing provider”) unaffiliated with either OEM or third-party. The concept of having a neutral entity is not entirely new. The Solid Project [10] lets people store their data securely in decentralized data stores called *pods*. Users can choose to share the data in their pods with any third-party. In the automotive domain, the European Automobile Manufacturers' Association introduced *neutral servers* which can facilitate data access to third-parties by offering multi-OEM data access on one server [21]. However, the neutral server or pod does not prevent a third-party from replicating data they have legitimate access to once permission is granted. This also means that even if a user revokes access to their data at some point, the third-party can still hold a copy of the data if previously accessed. This is a major concern for OEMs due to possible liability because of privacy regulations.

As a result, the neutral entity needs to execute the third-party's code in a sandboxed environment and share the result of the execution with the OEM. This has the advantage that the third-party does not have to share its IP with the OEM. Furthermore, the OEM will rather share the execution result instead of vehicular sensor data with the third-party and can run sanitization algorithms on the result first. The rationale behind data sanitization — one of our three privacy goals — are *inference attacks*, i.e., when a malicious or benign-but-curious third-party decides to mine more context out of data that has not been agreed on in the Terms of Service (ToS) with the OEM at app approval time. Some examples [31], [46] are geolocation inference by merely using speed or steering angle values. If execution results can be sanitized at the OEM first before relaying them to the third-party, potential indefinite storage of raw sensor data at the third-party, as well as ”sneaking” in geo-coordinates into a list of speed/steering angle values can be avoided by our system architecture. As a result, the extension of neutral entities by this component is one of the major contributions of this paper.

Part of this paper focuses on the sanitization module, called *Privacy Check* (PC), by adapting two techniques known from anomaly detection, namely *change-point detection* (CPD) and *entropy-based detection* (EBD). We elaborate on them in Secs. V-B and V-C, respectively, before evaluating their performance for a real-world use case in Sec. VI. Experimental results show that a combination of both approaches is well-suited for the detection of privacy attacks launched by third-parties under the given threat model. In summary, this paper makes the following contributions:

1) Definition of privacy goals based on existing regulations;
2) Open-source framework [45] for privacy-preserving vehicular data collection and sharing called `PRICAR` that fully complies with defined privacy goals;
3) Extension of neutral entity concept to enforce purpose and storage limitation of driver data;
4) Adaptation of two existing anomaly detection techniques for data sanitization module;
5) Evaluation of data sanitization module on real vehicle apps and data.

II. Background

A. Data Collection from Vehicles

First, we briefly introduce how vehicular data is collected. Vehicular sensor data is collected from a set of Electronic Control Units (ECUs) residing inside the vehicle. ECUs are usually interconnected with each other by an on-board communication bus, or in-vehicle network (IVN), with the Controller Area Network (CAN) being the most popular in current vehicles. CAN data can be collected from ECUs through the OBD-II port which is a physical interface below the steering wheel in all US cars since 1996. In recent years, wireless connectivity in vehicles has gained popularity. According to [6], 250M vehicles are connected to the Internet of Things (IoTs) in 2020. Soon, third-party services will obtain driver data using a built-in data connection through novel connected platforms, such as BMW CarData [15] or Android Automotive [13].

Most vehicular data collection platforms operate in similar ways. For instance, in the case of BMW CarData, the user/driver of a vehicle wants to install a third-party app from the OEM's app store. In this example, since the app has to pass a review process of BMW, the chances of getting the app published on BMW's app store is high if the app is not obviously over-privileged (e.g., using GPS permission for a fuel consumption tracking app) or requesting permission for sensors which the app description does not correlate with. For Android Automotive, third-party apps are ”subject to additional review” [3] which can include manual vetting from the OEM itself. An example workflow for data collection and distribution to third-parties is shown in Fig. 2.

Alice finds an interesting third-party app from the malicious third-party entity, Mallory, which she wants to install. Alice is then given an overview of sensors the app requests for proper functioning, akin to app permissions shown at installation time in a mobile phone app store. If agreed, BMW transmits a copy of the collected telematics data to Mallory's server over their business-to-business (B2B) interface, but does not have any influence on what happens afterwards with the data. In the case of Android Automotive, the data would be transmitted to the third-party directly, without passing any OEM backend first. However, the permission model and vetting process are identical to BMW CarData. Although the

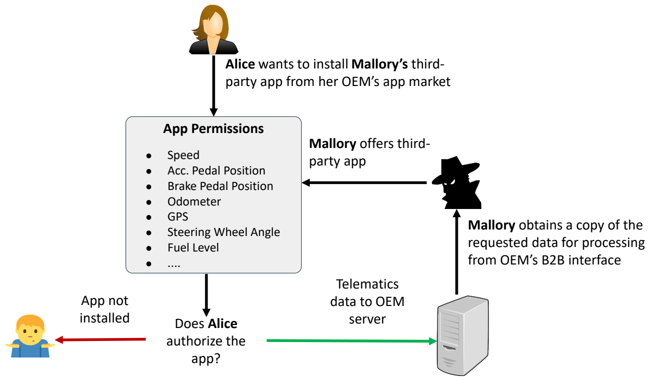

Alice wants to install Mallory's third-party app from her OEM's app market
App Permissions
Speed
Acc. Pedal Position
Brake Pedal Position
Odometer
GPS
Steering Wheel Angle
Fuel Level
....
Mallory offers third-party app
Mallory obtains a copy of the requested data for processing from OEM's B2B interface
Telematics data to OEM server
App not installed
Does Alice authorize the app?

Fig. 2: Threat model (adapted from BMW CarData [15])

idea behind Android Automotive is to have a shared Play Store across car brands, Pesé *et al.* have discovered that OEMs choose to feature exclusive apps in their respective Play Stores [48]. In the following, we assume that the OEM is trustworthy. OEMs are committed to only accepting trustworthy service providers as business partners for CarData, such as insurance companies. If Mallory has a valid use-case for their app, the OEM might approve Mallory as a trustworthy entity. Since the user and OEM do not have any influence on what happens to the data after giving permission to run that app, Mallory can use all the collected data as they like.

B. Privacy Regulation

As mentioned in Sec. I, there are several new privacy regulations that have been passed in recent years. The General Data Protection Regulation (GDPR) is the most comprehensive of them and will affect all global carmakers conducting business in the European Union.

GDPR distinguishes between *data subjects*, *data controllers* and *data processors*. GDPR ensures adequate protection of the privacy rights of data subjects, i.e., drivers in our context. Data controllers dictate how and why data is going to be used by the organization and thus have the most responsibility when it comes to protecting the privacy and rights of the data subject. Data processors process any data on behalf of the data controller. Since OEMs control the data shared with third-party app providers, they fall under the category of data controllers and are subject to increased compliance obligations [12]. These are summarized as seven core principles in the following [44]:

1) **Lawfulness, Fairness and Transparency:** Relates to the legality of data collection and transparency of collected data.
2) **Purpose Limitation:** Use collected data only for the specific purposes for which it was collected.
3) **Data Minimization:** Only request data that is required for a purpose.
4) **Accuracy:** Relates to upkeeping the accuracy and completeness of such data and what a consumer's rights are for correcting inaccuracies.
5) **Storage Limitation:** Constrain the amount of time that personal data can be stored for.
6) **Integrity and Confidentiality:** Relates to security of data transmission and storage, e.g., by encrypting and pseudonmyizing data.
7) **Accountability:** Have appropriate measures and records in place to take responsibility for what you do with personal data and how you comply with the other principles.

The California Consumer Privacy Act (CCPA) is the first large privacy law enacted in the US, although it applies only to residents of California. The most important difference from GDPR is prior consent versus opting out [2]. GDPR requires users to give clear and affirmative consent prior to having data collected and processed. CCPA requires businesses to make it possible for consumers to opt out of having data disclosed or sold to third-parties. Among others, the definition of ”businesses” was quite vague in the original definition of CCPA. To overcome these ambiguities, CCPA was amended to form the California Privacy Rights Act (CPRA) [18].

Furthermore, according to voluntary guidelines passed by the North American Alliance of Automobile Manufacturers [14], there are three categories called "covered information" which the OEM must ask for explicit permission:

- **Driving Behavior:** Information about how a person drives a vehicle. Examples stated in that document are vehicle speed, seat belt use, and information about braking habits.
- **Geolocation:** Information about the precise geographic location of a vehicle.
- **Biometrics:** Information about a driver's physical or biological characteristics that can be used to identify the person.

C. Privacy in Automotive Industry

To adhere to GDPR standards, OEMs have been either sending privacy notices or updating their privacy policies to inform their consumers of how their data is being used. Privacy policies from Honda [20], Ford [22], Volvo [23], and BMW [8] outline *who* is responsible for the data collected, *how* they use it, *what* GDPR law solidifies their use, and retention of the data. Regarding who is responsible for the data, OEMs list themselves and either "affiliates" (Honda and Ford), "partners" (BMW) or "joint controllers" (Volvo). Volvo's joint controller, Polestar, is only allowed access to analyze the vehicle data and not consumer data, but BMW's partners can collect personal data. In addition to this, third-parties that the customer permits within BMW vehicles are not under the influence of BMW [8]. Each OEM states that the main purpose of collecting consumer data is for the advancement and betterment of the car, as well as services they provide. However, how long they retain the data varies across OEMs and the type of data. For instance, Honda [20] promises to retain data for up to five years. Volvo [23] varies their retention based on whether it pertains to the battery of their hybrid and electric vehicles. What we currently see from privacy policies of the automotive industry is the lack of standardization of GDPR implementation. Each OEM has its own implementation which can, in turn, be deficient to the consumer.

Although the idea behind `PRICAR` does not necessarily have to be restricted to the automotive domain, the following three reasons explain why it has been specifically crafted as an end-to-end privacy framework for connected vehicles:

1) Vehicular data is very rich compared to the data collected by mobile phones and smart home devices. Smartphones also collect numerical data such as from IMU sensors that can be leveraged for similar applications such as driver behavior scoring. Despite that, (i) kinematic data from vehicles is higher quality and more precise compared to noisy IMU data, and (ii) vehicles contain other information such as seat belt status, battery voltage, etc. (see Table I) that cannot be captured by phones. As a result, third-party app developers have more access to data and can create more versatile applications, but also craft more sophisticated inference attacks.
2) Compared to other domains such as mobile app stores, vehicles are a more closed ecosystem with significantly fewer apps that are subject to manual vetting by the OEM upon acceptance into their respective app store. Mobile app stores have automated malware checkers when a third-party app is uploaded. Errors or shortcomings during the manual vetting process might make the OEM liable. Furthermore, no precedent has been set in the automotive privacy domain yet which can paradoxically lead to elevated caution in OEMs.
3) The design of neutral entities as neutral servers is well understood in the automotive context [21] and offers a good opportunity to extend for full privacy compliance. In other domains (e.g., mobile), this concept is unfortunately relatively scarce, although promising solutions such as the Solid Project [10] (see Sec. IV) exist.

III. Privacy Goals

Despite realizing some GDPR principles (see Sec. II-B) such as data minimization in the presented threat model, several core principles are missing in the current design of vehicular data collection and sharing platforms. Even the permission model (which is covered by data minimization) is vulnerable to privacy attacks from Sec. IV-A. In what follows, we will define three privacy goals to comply with GDPR principles and discuss how a novel privacy-preserving framework can meet them. The focus of this paper lies on data sanitization (see Sec. III-C) due to a wide range of existing work for the former two of data minimization (see Sec. III-A) and anonymization (see Sec. III-B). Prior work on these two privacy goals are summarized in the respective subsections whereas related work on data sanitization is presented in the next section (see Sec. IV).

A. Data Minimization

Although current telematic systems (e.g., BMW CarData, Android Automotive) enforce a permission model as depicted in the previous subsection, their design is still weak and allows third-parties to launch privacy attacks. The main reasons behind a vulnerable permission model are (i) its coarse-graininess and (ii) its lack of understanding for the average driver/customer. Pesé *et al.* [47] showed that the current permission model of Android Automotive OS (AAOS) is not fine-grained enough and offers opportunities for privacy attacks. Android defines four protection levels (normal, dangerous, signature, privileged) and each permission is assigned a unique protection level. AAOS enforces third-party apps to only request normal and dangerous permissions, whereas the latter are reserved for OEM-native apps. A normal permission can be regarded as *zero-permission*, i.e., no explicit user consent is necessary when the user installs a third-party app with that permission. This stands in contrast to dangerous permissions that require explicit user consent at installation time. AAOS 10 defines 92 permissions in its `android.car.permission` package, of which 8 are labeled normal or dangerous. Multiple vehicle properties, such as RPM and gear can be summarized in one permission which leads to the permission model to be coarse-grained. This design choice can be exploited by inferring sensors protected by dangerous permissions using other sensors protected by normal permissions. One example is speed, which is protected by a dangerous permission, but can be easily calculated by gear position and engine speed (RPM) [11] which are zero-permission. As a result, any third-party app can infer the drivers' speed without explicitly requesting it from the driver. Location inference attacks as shown in Sec. IV-A further display the utmost danger of having access to vehicle speed due to the possibility of user tracking.

A proper way to conduct data minimization would be to expand on the existing permission model by more permissions and assign a unique permission to each vehicle property/sensor. Pesé *et al.* [50] quantified the privacy risk of 20 vehicular sensors based on existing privacy attacks. Although the general separation between third-party- and OEM-specific permissions is favorable, OEMs need to be careful with permissions that can be accessed by third-parties. The privacy risk is a good metric to define a threshold and assign all permissions above that threshold dangerous protection level whereas the rest can be assigned normal protection. Finally, OEMs cannot expect the average driver to fully understand the functioning of requested permissions/sensors and its possible impact/implications. Pesé *et al.* [46] showed in a survey with 100 participants that users are relatively comfortable to share certain sensitive sensors with both OEMs and third-parties, which can, in turn, be leveraged to launch privacy attacks. Hence, it is important for OEMs to be very transparent in explaining background information about sensors. Third-parties must also explain why each specific permission is required for their app to avoid over-privileged apps (which can theoretically be caught at the initial submission stage to the OEM). Each customer will also have the right to revoke any permissions at any time (even if it comes at the expense of the app not functioning any more). All in all, the privacy goal of data minimization covers GDPR principles (1), (2), (3) and (4).

B. Data Anonymization

The privacy goal of data anonymization will remove any personally identifiable information (PII) from the data upon sharing it with the third-party app provider. This can be done by altering data in such a way that the data subject can no longer be identified *directly* or *indirectly*. One can distinguish between *direct identifiers* and *quasi-identifiers*. Direct Identifiers can identify an individual just by themselves while Quasi-Identifiers can only identify an individual in combination with others. For instance, the Vehicle Identification Number (VIN) is a unique number assigned to each vehicle that can help identify the driver by database queries [9]. Even accumulating location data can help identify the driver by analyzing home/work pairs [38]. Due to recent advances in location inference attacks [31], [46], speed and steering wheel angle can be considered a direct identifier. Table I gives an overview of how 20 frequently collected vehicular sensors can be categorized into direct identifiers and quasi-identifiers.

TABLE I: Direct and quasi-identifiers for 20 most frequently collected vehicular sensors

Vehicle Sensor	**Direct Identifier**	**Quasi-Identifier**
Location	✓	✓
VIN	✓	
Outside temperature		
Odometer		
Current speed	(✓)	✓
Average speed		
Maximum speed		(✓)
Fuel consumption		(✓)
Throttle position		(✓)
RPM		(✓)
Steering wheel angle	/✓)	✓
Airbag status		
Seat belt status		(✓)
Battery level		
Tire pressure		
Hard braking		✓
Make/model/year		✓
Fuel level		
Check engine light on		(✓)
Oil level		

There are six types of data anonymization techniques: pseudonymization, generalization, data masking, swapping, perturbation and synthetic data generation [25]. In the automotive context, we recommend pseudonymizing the VIN, i.e., replacing it with a randomly generated unique identifier that is attached to all data that the third-party is requesting. Furthermore, generalizing data, e.g., by truncating digits or assigning values to pre-defined buckets with a minimum and maximum value can help in reducing the accuracy of privacy attacks which usually rely on high-quality data. Another possible technique to reduce data utility is downsampling. It decreases data quality without altering or replacing values from the original dataset. Furthermore, low-pass filtering can also be used for signal anonymization by eliminating high-frequency components that describe idiosyncrasies of the signal and end up with the more general parts. Finally, differential privacy is getting increasingly popular in recent year since it can offer a more robust and mathematical approach to anonymize data. In the automotive context, Gazdag *et al.* [35] apply the techniques of low-pass filtering, aggregation and differential privacy to empirically demonstrate the effect on driver fingerprinting and location inference attacks (see Sec. IV-A). Similar de-anonymization techniques are applied to the same attacks by Li *et al.* [42]. All in all, the privacy goal of data anonymization covers GDPR principles (2) and (6).

C. Data Sanitization

Despite minimization (through the permission model) and anonymization of data, it is still possible for third-parties to mine additional context to conduct a privacy attack. Sharing minimized and anonymized data with a third-party directly violates two essential GDPR principles, namely (2) Purpose Limitation and (5) Storage Limitation. Since we relinquish control over the data at the time we transmit them to the third-party, there is no feasible way for the OEM to control the proper use of the data on third-party servers as agreed upon in the Terms of Service (ToS) if the third-party is rogue. Besides possibly mining additional content that has not been agreed upon in the ToS (violates (2)), the third-party could also store the data as long as they want (violates (5)). Fig. 3 shows an example of this which we will also use in our experimental evaluation. We assume a rogue third-party insurance company that might not be malicious per se, but curious. In the ToS with the OEM, they agree on using three permissions (acceleration, braking, steering angle) just for the purpose of calculating a driving score for each user to assess their insurance premium upon. As a result, they can only use this data for the specified purpose. Nevertheless, using these three permissions, it is also possible to conduct driver fingerprinting, i.e., distinguish multiple drivers in the same vehicle. The insurance company could use this information to see if the main driver in the policy avoids adding other people in their household to the policy. Given the ToS and privacy regulation, this is illegal, but the data controller (OEM) cannot technically stop the rogue third-party from doing that. On the other hand, the third-party wants to make sure not to share their source code directly and openly with the OEM due to Intellectual Property (IP) protection.

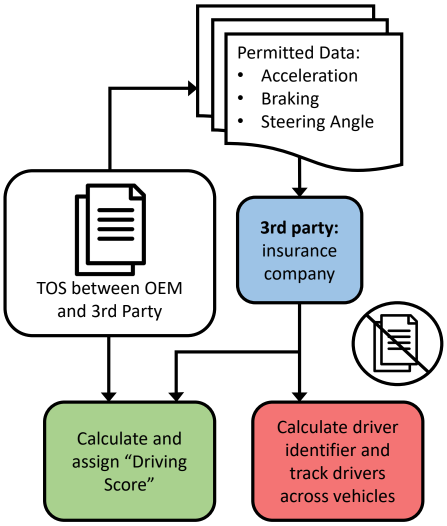

Permitted Data:
• Acceleration
• Braking
• Steering Angle
TOS between OEM and 3rd Party
3rd party: insurance company
Calculate and assign "Driving Score"
Calculate driver identifier and track drivers across vehicles

Fig. 3: Example justifying the need for data sanitization

One way to solve this problem is to introduce a *neutral entity* independent from both OEM and third-party. The neutral entity would get sensor data from the OEM as input, execute the third-party app code and then share the result with the OEM first. The latter could then decide if the result looks *legitimate*, i.e., satisfies the purpose of the third-party app, or *illegitimate*, i.e., has been snuck into a list of legitimate results. Based on the labeling as legitimate or illegitimate, the OEM can then either relay the result to the third-party or remove it and flag the third-party after repeated incidents. All in all, the privacy goal of data sanitization covers GDPR principles (2) and (5).

IV. Related Work

A. Inference Attacks

To understand the threats against drivers' privacy better, it is essential to survey and analyze the landscape of existing inference attacks on seemingly benign vehicular data that demonstrate how more context can be mined out of them. Inference attacks clearly violate the GDPR principle of (2) Purpose Limitation. Pesé *et al.* [50] categorized and summarized academic literature on this topic:

1) Driver Behavior Analysis: Vehicular data is so rich that individual driving behavior can be analyzed and a driving score assigned to each driver [30]. Various automotive usage-based insurance (UBI) companies use this score to adjust the premium rate. In order to protect drivers' location privacy, a range of other sensors, such as speed, acceleration and turns can also be used to calculate a driving score.

2) Driver Fingerprinting: The goal behind this attack is to distinguish different drivers using the same car by analyzing vehicular sensor data during trips. It has been shown by Enev *et al.* [32] that using 15 sensors, it was possible to identify 15 different drivers with 100% accuracy. Kar *et al.* [40] showed how drivers can be distinguished before even starting their trip. The main privacy issue behind fingerprinting drivers is to conclude that other drivers than the main (authorized) driver have used that vehicle. An attacker could also monetize this information. Automotive UBI companies, especially, are interested in this information since this might violate their terms and/or lead to a change in the insurance premium.

3) Location Inference: It has been shown that traveled routes of the user can be inferred by merely using the speed trace of a trip [31], [34], [54]. Pesé *et al.* [46] showed that geolocation can even be inferred by the less sensitive steering wheel angle sensor alone. Furthermore, the Vehicle Identification Number (VIN) can be leveraged to obtain knowledge about the rough location.

B. Data Sanitization in Vehicular Context

As mentioned in the introduction, the European Automobile Manufacturers' Association has already a *neutral server* concept [21]. Neutral servers can be established to provide easy access to vehicle data for third-party service providers who are interested, eliminating the necessity of entering into an agreement with a vehicle manufacturer. These servers maintain complete neutrality, implying that they are managed and funded by an impartial entity rather than the manufacturers themselves. Naturally, these impartial server operators must incorporate advanced security and data safeguarding techniques. Neutral servers are based on the *extended vehicle* concept [24] that mandate third-party access through an off-board facility, i.e., without a direct connection to a (moving) vehicle. As displayed in Sec. V, our system design follows these ideas.

Several companies doing business in the area of automotive data follow the concept of neutral servers. Examples include HERE Maps [7], Otonomo [26], as well as IBM for the aforementioned BMW CarData platform [41]. However, the current state of neutral servers have one significant shortcoming: They cannot restrict purpose nor storage limitation which are key GDPR principles. Once data access permissions are granted to a rogue third party, they can choose to indefinitely store the data even if access is revoked after some time. There is also no way to control if a rogue third party calculates more context out of the data than intended.

C. Data Sanitization in Non-Vehicular Context

Solid [10], [33] is an initiative for web decentralization spearheaded by the innovator behind the World Wide Web. This endeavor seeks to fundamentally transform the functioning of current web applications, leading to authentic data ownership and heightened privacy. Its primary objective is to construct a framework for linked-data applications that operate in a completely decentralized manner, firmly within the grasp of users rather than external entities. The aim of Solid is to empower users with comprehensive authority over their personal data, encompassing control over access and storage location. Any data can be stored by users in a *pod*. The main drawbacks of Solid are comparable to the ones of neutral servers: Solid cannot prevent third parties from replicating data they had legitimate access to at any given point in time.

Specific applications of Solid-like concepts exist in healthcare where electronic health records (EHRs) are real-time patient records [16] that can be made available to any medical professional upon request. EHRs contain a wide range of personally identifiable information (PII) and are thus subject to similar considerations as described in this paper. A rogue doctor does not need to adhere to record deletion once the patient-doctor relationship has ended. However, the main difference between the automotive and medical contexts are the type of data: Whereas vehicular data types are mostly numerical, health data is usually categorical. As a result, PRICAR cannot be directly applied to the medical context.

Another way to preserve privacy, but still being able to perform calculations on it is a technique called homomorphic encryption [43]. It enables a specific mathematical operation to be conducted on the original data, resulting in an outcome comparable to a different mathematical operation carried out on the encrypted data. This capability facilitates the execution of computations directly on data that is encrypted. Fully homomorphic encryption, which supports both addition and multiplication, offers greater versatility, as it permits the evaluation of any computation in a homomorphic manner. However, these fully homomorphic encryption methods are still prohibitively expensive for practical use. On the other hand, partially homomorphic encryption techniques, which support only one type of operation (either addition or multiplication) are comparatively more efficient. These schemes become particularly useful when basic statistical calculations (e.g., SUM, AVERAGE) need to be performed on uploaded data, enhancing the security of user data stored in the cloud.

V. System Design

A. Overview

After introducing the three privacy goals of *minimization*, *anonymization* and *sanitization* in the previous section, we present a privacy-preserving framework, called PRICAR, for vehicular data collection and sharing with third-parties. Fig. 4 depicts the architecture, with four involved entities: Vehicle, OEM Backend, Cloud Computing (CC) Provider, and the Third-Party Service Provider. The CC Provider plays the role of the *neutral entity* to satisfy the privacy goal of data sanitization. Each entity with its respective modules and the data flow between the entities is detailed next.

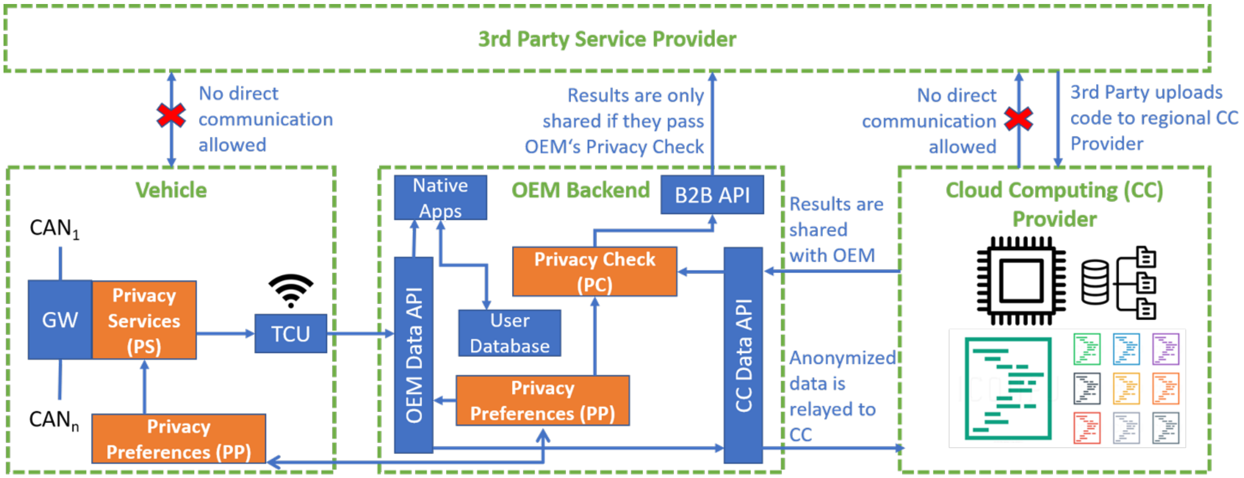
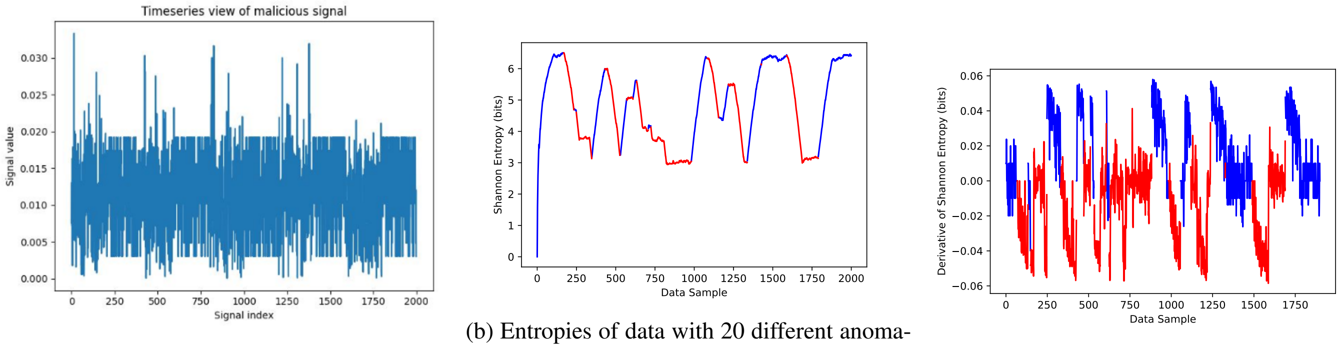

3rd Party Service Provider
No direct communication allowed
Results are only shared if they pass OEM's Privacy Check
No direct communication allowed
3rd Party uploads code to regional CC Provider
Vehicle
CAN$_1$
GW
Privacy Services (PS)
TCU
CAN$_n$
Privacy Preferences (PP)
Native Apps
OEM Backend
B2B API
Privacy Check (PC)
OEM Data API
User Database
Privacy Preferences (PP)
CC Data API
Results are shared with OEM
Anonymized data is relayed to CC
Cloud Computing (CC) Provider

Fig. 4: System architecture overview

First, a third-party app/service provider develops an app and submits it to the OEM-managed app store. This can be achieved by a web app provided by the OEM. During this process, they have to specify the name, description, required permissions of this app, as well as the third-party endpoint (i.e., network address and port) [49]. As we discussed in Sec. III, it is essential for the third-party to specify why each permission is required to comply with GDPR principles (1) and (2), and to avoid the app being over-privileged. The OEM reserves the right to reject any third-party app. This could be because of lack of value or the app being blatantly over-privileged [4]. Since OEM-run app stores will only feature a manageable number of apps, this initial submission process can thwart obviously malicious third-party apps. Nevertheless, we anticipate that in the future, with a rising number of apps, this might not be possible any more and numerous rogue third-party apps might go unnoticed. Due to the introduction of a neutral entity, the third-parties will also need to upload their source code together with information about the build environment upon approval by the OEM. The submitted code will not go or be stored by the OEM backend, but will be used to spin a container on the neutral entity, the CC provider. This could be a Docker instance running on Amazon AWS or Google Cloud. For I/O with the OEM backend, we provide a networking script that is automatically included in each container and a standardized developer API. There are three significant advantages to this: (i) Each app is sandboxed and clearly separated from each other; (ii) The OEM does not have to provide any computational resources, but can rent cloud computing services from well-known, trusted companies. The rent can be charged to the third-party as part of their recurring service fee for instance; (iii) Since OEMs are global companies, the cloud computing provider can be chosen according to geographic region and availability. Cars sold in China can rely on Tencent Cloud, whereas US cars can use Amazon AWS.

Once the OEM-managed app store contains third-party apps, the driver can select and install an app. During installation, permissions that require explicit user consent (e.g., *dangerous* permissions in Android) are prompted to the user. All permissions of a certain app are stored in a module called *Privacy Preferences* (PP) on the vehicle. Then, data is generated in vehicles by Electronic Control Units (ECUs). Note that we do not make any assumptions on the used in-vehicle network architecture, i.e., PRICAR is transparent on the vehicle architecture and Fig. 4 merely shows an example. The requested data by an app is bundled and passed on to the Telematic Control Unit (TCU). Before that, a module called *Privacy Services* (PS) applies data anonymization techniques as discussed in Sec. III. Note that this can also be performed at the OEM backend, but we recommend this to be done locally inside the vehicle. The reason behind this is that techniques such as downsampling will save bandwidth and other generalization or pseudonymization techniques are very light-weight and with very low computational overhead to in-vehicle ECUs. Also, note that the vehicle is only allowed to communicate with the OEM backend. No direct communication between vehicle and third-party will be allowed at any point. After the TCU transmits the data via the vehicle's built-in cellular connection, the OEM backend will receive it via its *OEM Data API*. There are two categories of apps: OEM-native apps and third-party apps. In the former case, the OEM can execute them locally on its backend. In the latter case, the data is passed on to the *CC Data API* and forwarded to its respective container on the CC provider. The third-party's code parses the received input, executes it and returns output values, called *results*. Currently, we only support numeric output values. The container on the CC Provider cannot talk to its third-party endpoint by only whitelisting network connections to the OEM backend.

The results are then sent back to the OEM backend and pass through a *Privacy Check* (PC) module — the core module responsible for data sanitization. Sections V-B and V-C will present two techniques used to realize the PC module. If the results pass the PC module, they are further forwarded to the third-party's endpoint via the *Business-to-Business* (B2B) interface. In case of a negative outcome in the PC module, there are certain options: (i) If any result is flagged, the OEM can suspend the sharing of results with the third-party immediately. The experimental evaluation in Sec. VI shows that our algorithms have false positives, so third-party apps might get flagged incorrectly. This would reduce utility, but prevent privacy violations at any price; (ii) The OEM will count the number of violations. If the number of violations over a given



(a) Malicious Result Vector with $w = 200$

(b) Entropies of data with 20 different anomalies

(c) Gradient of (b)

Fig. 5: Exploratory Data Analysis

time period exceeds a heuristically determined threshold, the OEM will then suspend communication and get in touch with the third-party to resolve or rectify the situation. Although this greatly benefits utility, numerous privacy violations might happen before the OEM takes an action.

B. *Change-Point Detection (CPD)*

The privacy goal of sanitization tries to prevent third-parties from mining additional context — the *Privacy Check* (PC) module inspects results passing through to determine if they are a privacy violation. Let us revisit the example from Fig. 3. Assume that the third-party insurance company computes a new driving score a_i for a driver every minute. Imagine that the insurance company wants to fingerprint the driver and tell itself that another driver is present. Since the CC provider cannot talk to the third-party endpoint, the only way to inform itself is to share the driver ID information b_i as a result. After three minutes, the third-party will now output b_3 instead of a_3 and then continue transmitting driving scores again:

$$\vec{v} = (a_0, a_1, a_2, b_3, a_4, a_5, \ldots) \qquad (1)$$

All output values ("results") are collated chronologically in a vector $\vec{v}$ in the PC module. Hence, the results are buffered on the OEM backend and not immediately released to the third-party endpoint. This is acceptable since third-party apps will usually not have any real-time, safety-critical purpose. The challenge now is to detect the outlier, i.e., b_3 in this vector. If the range of b_i was larger than a_i, this would be easy to detect since the OEM knows the range of expected results *a priori*. Nevertheless, a smart attacker could scale down the values to fit in the range of a_i. Hence, we assume that both legitimate a_i and illegitimate scores b_i have been normalized to a range between 0 and 1.

In order to detect b_3, a technique called *change-point detection* (CPD) can be used. CPD operates on time-series and tries to identify times when the probability distribution of this time-series has changed. There is a myriad of CPD algorithms [52] and some leverage statistical features such as mean, variance, correlation or spectral density. For instance, using the CPD algorithm ED-PELT [39], the PC module looks at the statistical distribution of results and detects anomalies. More specifically, as results for a given third-party app are sent from the CC provider to the PC module for approval, they are collated chronologically in a vector $\vec{v}$. Then ED-PELT is run on batches of results taken from vector $\vec{v}$ of size n. The size parameter also determines how many results have to be buffered before the CPD algorithm can be run on it. If ED-PELT detects a change-point in the batch, then that indicates that there is a statistical anomaly that could be the result of the third-party attempting to mine additional context. The batch is held until further review can be conducted, then it is either released to the third-party or denied. If no change-points are detected, the batch is immediately released to the third-party.

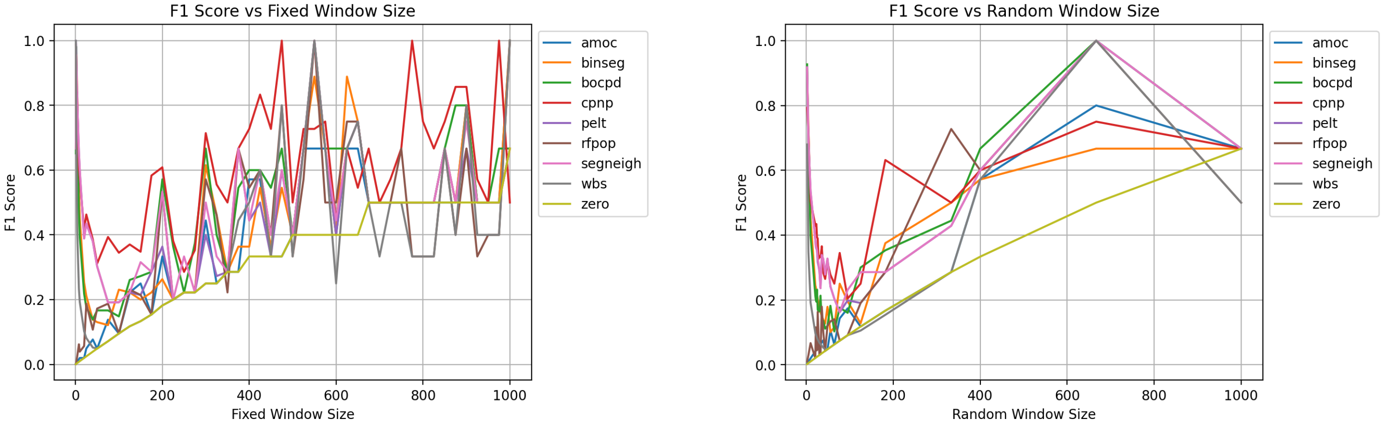
Fig. 5a displays a result vector which has illegitimate data points injected by a malicious third-party. The attacker uses a fixed window size $w = 200$, i.e., they use 200 legitimate data points first and then inject 200 illegitimate data points, etc. In our evaluation, we will analyze the effect of w. Furthermore, an attacker could also switch between legitimate and illegitimate scores *randomly*, i.e., without a fixed window size. Instead of specifying a fixed window size, we introduce the random window size w as the statistical mean of random change-point intervals. The OEM needs to have some knowledge about what legitimate results look like to distinguish legitimate from illegitimate results. For this purpose, the OEM needs some ground truth which can be requested after approval of the app.

C. *Entropy-Based Detection (EBD)*

In addition to traditional CPD algorithms mentioned earlier, we also wanted to explore CPD based on the entropy of the data rather than the data itself. We will call this *entropy-based detection* (EBD). Other researchers have found success with using EBD for anomaly detection using a sliding window entropy calculation [53]. To find the entropy of the application's data, the Shannon entropy is calculated over a sliding window taken over the data. As new data comes in, the oldest data points are dropped from this sliding window. As the data is not necessarily sent in an integer format, the data is first rounded to two decimal places. To calculate the Shannon entropy H of the data in a sliding window X, we use the following equation:

$$H(X) = -\sum p(X) \log p(X) \qquad (2)$$

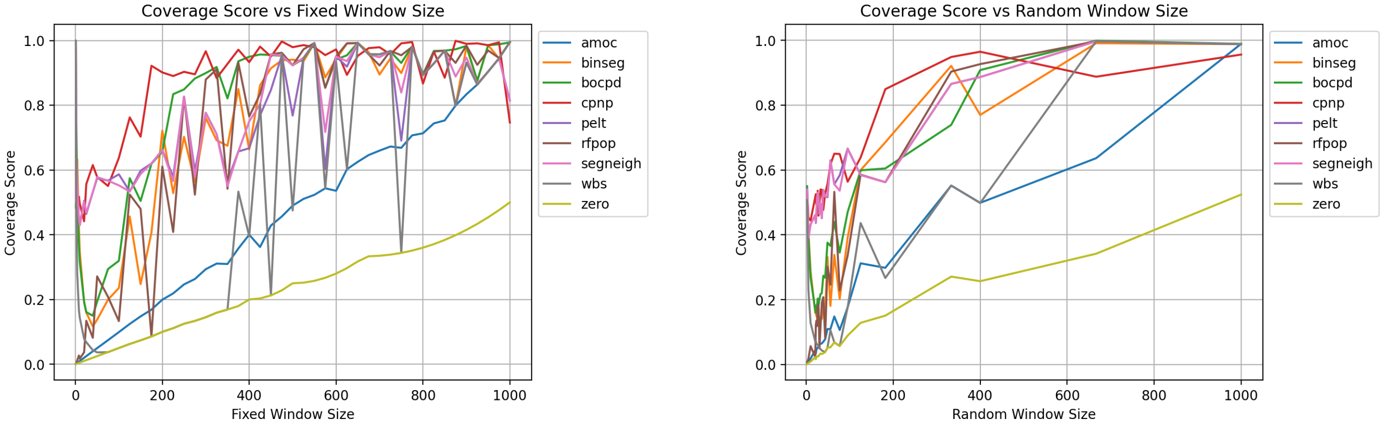
For each datum x_i contained in the (rounded) input data, we compute $P(x_i)$ = # of occurrences of x_i / N with N being



(a) Fixed window sizes F1-Score

(b) Random window sizes F1-Score

Fig. 6: CPD F1 score for varying window sizes



(a) Fixed window sizes coverage score

(b) Random window sizes coverage score

Fig. 7: CPD coverage score for varying window sizes

the total number of data points in the sliding window. Once the entropy is calculated, there are a few different methods of change-point detection that we can use to classify the original points as either legitimate or illegitimate. Wu *et al.* [53] use a straightforward method of marking all data in a certain range of entropies as *valid* and all data outside that range as *anomalies*, using simulated annealing to tune that range. While a similar approach can be attempted for our data, the results would not be optimal — mainly due to the fact that this method is well-suited for anomalies like DoS attacks and that the anomaly is not intermittent like ours. Fig. 5b depicts the entropy of one of our data samples, with blue parts of the graph corresponding to normal data and red parts of the graph corresponding to malicious data. By inspection, we can see that it is difficult to come up with a vertical range of entropies inside which legitimate data points lie. Nevertheless, illegitimate segments of the data tend to correspond to areas with decreasing entropies compared to legitimate data. Fig. 5c shows a plot of the gradient of the Shannon Entropy of the same data as in Fig. 5b. From our observations, for regions with a negative derivative, we will mark the data as illegitimate and otherwise as legitimate. As Sec. VI shows, the F1-score is relatively high for various experiments. This approach runs into issues though when there are long stretches of illegitimate data — it would flag the initial entropy change but would not consider the subsequent low-entropy run of data as anomalous. This might not be a huge issue if the system design would stop the application passing on data once the data was flagged as anomalous.

VI. Experimental Evaluation

A. Dataset

To evaluate the CPD algorithms in achieving our privacy goal of data sanitization, as well as determine which of them performs best, we created a realistic dataset of result vectors that would appear in the Privacy Check module for a third-party insurance app. We have two types of results — driver scores [27] and driver fingerprints [37] — calculated from a one-hour driving data using a 2016 Ford Explorer. `PRICAR` is built on top of the open-source data collection, translation and sharing framework DETROIT [49]. We forked DETROIT and added the Privacy Check module, as well as the Cloud Computing Provider entity. The driving score calculator takes GPS coordinates, RPM and speed as input and outputs a driving score from 0 to 100. The fingerprinting script uses 40 different sensors, such as fuel consumption and engine coolant temperature and outputs a driver ID from 0 to 9. PRICAR's source code and evaluation data are available on Github [45].

B. Change-Point Detection (CPD)

We benchmark CPD algorithms using the Turing Change Point Detection Benchmark (TCPDBench) [52], a benchmark evaluation of 28 CPD algorithms in total. In this setup, we define the driving score as the only approved calculation using the driver data, a legitimate result. Then, the driver fingerprints are unapproved, illegitimate results. We then evaluate if the PC module can detect these illegitimate results. The performance of various CPD algorithms is measured with two metrics: F1-score and coverage score. The authors of TCPDBench [52] define the coverage score as follows. For two sets $\mathcal{A}, \mathcal{A}' \subseteq [1, T]$ the Jaccard index, also known as Intersection over Union (IoU), is defined as $J(\mathcal{A}, \mathcal{A}')$. Following [29], the covering metric of partition $\mathcal{G}$ by partition $\mathcal{G}'$ is defined as

$$C(\mathcal{G}', \mathcal{G}) = \frac{1}{T} \sum_{\mathcal{A} \in \mathcal{G}} |\mathcal{A}| \cdot \max_{\mathcal{A}' \in \mathcal{G}'} J(\mathcal{A}, \mathcal{A}'). \tag{3}$$

We ran 9 CPD algorithms through the benchmark qwhile varying fixed and random window sizes w. Fig. 6 shows the F1-score as defined above for window sizes up to 1000. Coverage scores are depicted in Fig. 7. Generally, for both F1- and coverage-scores, the metrics improve with larger (fixed and random) window sizes. Our testing shows CPNP and PELT performed the best on average, with F1-scores of 0.558 and 0.477 and coverage scores of 0.767 and 0.698 for fixed and random window sizes, respectively.

Furthermore, we were interested in the execution time of CPD algorithms. Since the PC module is on the OEM backend and not on the vehicle, we can assume that computational resources are not the bottleneck. Nevertheless, some CPD algorithms are, computationally, very intensive. We benchmarked the execution of 9 fast CPD algorithms and summarized them in Table II. Our benchmark uses Python 3 and R and was run on Ubuntu 18.04 LTS with 128GB of ECC DDR4 RAM and two Intel Xeon E5-2683V4 CPUs. We averaged the latencies across three runs on the random20 series (random window size, 20 changepoints). Our best-performing algorithms CPNP and PELT have a very small execution time (around one second), just like most of the other CPD algorithms. Only BOCPDMS, ECP and KCPA seem to take longer than a minute. Our evaluation also showed that the window size does not have any effect on the execution time of CPD algorithms.

C. Entropy-Based Detection (EBD)

Fig. 8a depicts the F1-score of the heuristic from Sec. V-C against both random and fixed window sizes. For most window sizes, the F1-score is above 75%. The F1-score can be improved even further. If the entropy drops (has negative gradient) for multiple consecutive time steps before classifying a given data point as anomalous, the specificity (true negative rate) can be improved by 4–10% (depending on window size) while having a negligible effect on the recall (true positive rate). As shown in the previous section, PELT and CPNP algorithms had the best performance in detecting change-points between legitimate and illegitimate data. We wanted to see if calculating the sliding window entropies of the data as a pre-processing step would further improve the performance of these CPD algorithms. Fig. 8b shows the F1-score of PELT and CPNP on random window sizes by themselves (in blue), as well as combined with EBD (in red). For window sizes larger than 50, the combination is shown to outperform CPD. Compared to EBD alone (see Fig. 8a), we can also see a slight improvement by using the combined approach. For instance, using CPNP on entropy data yields an F1-score of over 0.9, while EBD alone stands around 0.8 for the same window size. As a result, we recommend the combined approach for best data sanitization performance.

TABLE II: Execution times of CPD algorithms

Algorithm	**Execution Time**
AMOC	1.028s
BINSEG	1.022s
BOCPD	2.640s
CPNP	1.135s
PELT	1.042s
RFPOP	0.836s
SEGNEIGH	2.810s
WBS	0.936s
ZERO	0.205s

VII. Limitations

The current system design of PRICAR makes certain assumptions which can be regarded as limitations.

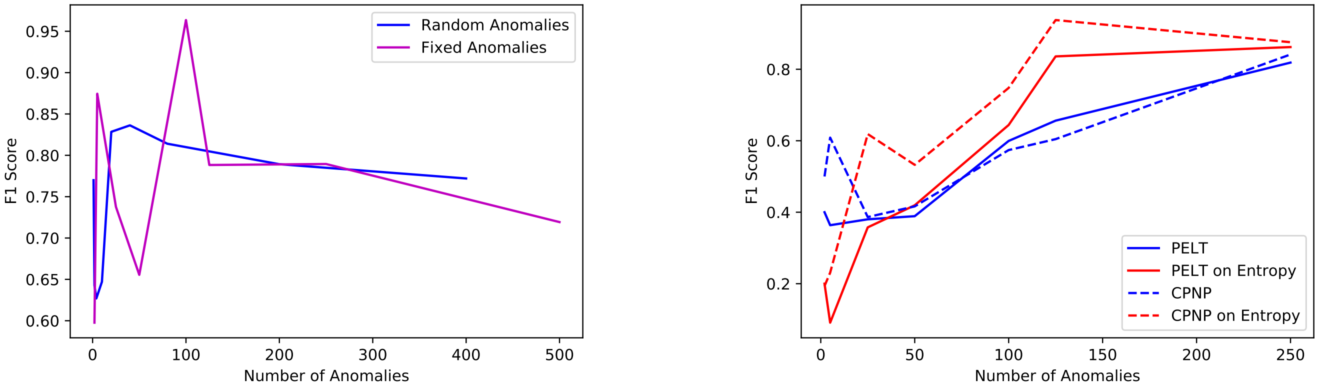
Numeric output values. Currently, the execution result computed at the cloud computing provider needs to be a numeric value since the algorithms in the PC module operate on time-series data. Categorical data are not covered yet.

Plaintext result. Third-parties might want to obfuscate their results before outputting them to the PC module which would affect the proper functioning of CPD algorithms. In our current design, we expect the results in plaintext.

Ground truth poisoning. Since the OEM requires ground-truth data from the third-party during the vetting process, it is possible that even this data might be malicious. We suggest a more rigorous vetting process (since it is already manual) to confirm that the ground truth "makes sense".

VIII. Conclusion and Future Work

In this paper, we have presented the first privacy-preserving vehicular data collection and sharing platform called PRICAR. We summarized existing privacy attacks and depicted the existing threat model for vehicular data sharing. Based on GDPR principles, we defined the three privacy goals of data minimization, anonymization and sanitization. The focus of the latter part of this paper was data sanitization. We demonstrated how driver privacy can be preserved by change-point and entropy-based detection, and evaluated their performance.



(a) EBD fixed and random window sizes F1-Score

(b) Comparison of CPD (blue) against EBD+CPD (red)

Fig. 8: EBD performance metrics

Acknowledgement

The work reported in this paper was supported in part by a Ford–UM Alliance Grant, the NSF under Grant 2245223 and the ARO under Grant W911NF-21-1-0057. Assistance from undergraduate researcher Erich Shan is also gratefully acknowledged.

References

[1] "Basics of the california consumer privacy act of 2018," https://www.privacypolicies.com/blog/california-consumer-privacy-act/.

[2] "Ccpa vs gdpr: Compliance with cookiebot cmp." [Online]. Available: https://www.cookiebot.com/en/ccpa-vs-gdpr-compliance-with-cookiebot-cmp/

[3] "Distribute android apps for cars." [Online]. Available: https://developer.android.com/training/cars/distribute

[4] "Distribute android apps for cars." [Online]. Available: https://developer.android.com/training/cars/distribute#opt_in

[5] "The eu general data protection regulation (gdpr) is the most important change in data privacy regulation in 20 years." https://eugdpr.org/.

[6] "Gartner says by 2020, a quarter billion connected vehicles will enable new in-vehicle services and automated driving capabilities," https://www.gartner.com/newsroom/id/2970017.

[7] "Here developer neutral server and consent management services." [Online]. Available: https://developer.here.com/documentation/marketplace-consumer/user_guide/topics/neutral_server.html

[8] "Informations sur la politique de confidentialité." [Online]. Available: https://www.bmw.fr/fr/footer/metanavigation/data-privacy.html

[9] "Research any vehicle in seconds," https://www.vehiclehistory.com/.

[10] "Solid project." [Online]. Available: https://solidproject.org/

[11] "Transmissions," http://craig.backfire.ca/pages/autos/transmissions.

[12] "What does the gdpr have to do with car oems?" https://upstream.auto/blog/dgpr/.

[13] "What is android automotive? android open source project," https://source.android.com/devices/automotive/start/what_automotive.

[14] "Consumer privacy protection principles privacy principles for vehicle technologies and services," 2014.

[15] "Bmw group launches bmw cardata: new and innovative services for customers, safely and transparently," May 2017. [Online]. Available: https://www.press.bmwgroup.com/global/article/detail/T0271366EN/bmw-group-launches-bmw-cardata:-new-and-innovative-services-for-customers-safely-and-transparently?language=en

[16] "What is an electronic health record (ehr)? — healthit.gov," Sep 2019. [Online]. Available: https://www.healthit.gov/faq/what-electronic-health-record-ehr

[17] "California consumer privacy act (ccpa)," Nov 2020. [Online]. Available: https://www.cookiebot.com/en/ccpa-compliance/

[18] "California privacy rights act (cpra): Ccpa vs cpra," Nov 2020. [Online]. Available: https://www.cookiebot.com/en/cpra/

[19] "The connected car 'data explosion': the challenges and opportunities," Sep 2020, https://www.information-age.com/tconnected-car-data-explosion-123473363/.

[20] "Honda global: Privacy policy (privacy notice)," Dec 2020. [Online]. Available: https://global.honda/voice-control-system/legal/GB/privacy.html

[21] "Acea neutral server concept," Nov 2021. [Online]. Available: https://www.cardatafacts.eu/vehicle-data-available-service-providers/

[22] "Ford® privacy policy," Jun 2021. [Online]. Available: https://www.ford.com/help/privacy/

[23] "Volvo car privacy notice," Apr 2021. [Online]. Available: https://www.volvocars.com/mt/legal/privacy/privacy-car

[24] "What is the extended vehicle concept?" Nov 2021. [Online]. Available: https://www.cardatafacts.eu/extended-vehicle-concept/

[25] "Data anonymization: Use cases and 6 common techniques," Nov 2022. [Online]. Available: https://satoricyber.com/data-masking/data-anonymization-use-cases-and-6-common-techniques/

[26] Admin, "The otonomo neutral server for automotive data," Jan 2023. [Online]. Available: https://otonomo.io/neutral-server/

[27] AkashKabra11, "Akashkabra11/driver-behavior-scoring," https://github.com/AkashKabra11/Driver-Behavior-Scoring.

[28] R. Amadeo, "Android Automotive OS review: Under the hood with Google's car OS," May 2021, https://arstechnica.com/cars/2021/05/android-automotive-os-review-under-the-hood-with-googles-car-os/.

[29] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 5, pp. 898–916, 2010.

[30] S. Arumugam and R. Bhargavi, "A survey on driving behavior analysis in usage based insurance using big data," *Journal of Big Data*, vol. 6, no. 1, pp. 1–21, 2019.

[31] R. Dewri, P. Annadata, W. Eltarjaman, and R. Thurimella, "Inferring trip destinations from driving habits data," in *Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society*. ACM, 2013, pp. 267–272.

[32] M. Enev, A. Takakuwa, K. Koscher, and T. Kohno, "Automobile driver fingerprinting," *Proceedings on Privacy Enhancing Technologies*, vol. 2016, no. 1, pp. 34–50, 2016.

[33] C. Esposito, R. Horne, L. Robaldo, B. Buelens, and E. Goesaert, “Assessing the solid protocol in relation to security and privacy obligations,” *Information*, vol. 14, no. 7, p. 411, 2023.

[34] X. Gao, B. Firner, S. Sugrim, V. Kaiser-Pendergrast, Y. Yang, and J. Lindqvist, “Elastic pathing: Your speed is enough to track you,” in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2014, pp. 975–986.

[35] A. Gazdag, S. Lestyán, M. Remeli, G. Ács, T. Holczer, and G. Biczók, “Privacy pitfalls of releasing in-vehicle network data,” *Vehicular Communications*, vol. 39, p. 100565, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214209622001127

[36] N. Gibbs, “Automakers seek new revenue streams from wireless services, upgrades,” May 2020, https://europe.autonews.com/automakers/automakers-seek-new-revenue-streams-wireless-services-upgrades.

[37] A. Girma, X. Yan, and A. Homaifar, “Driver identification based on vehicle telematics data using lstm-recurrent neural network,” in *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2019, pp. 894–902.

[38] P. Golle and K. Partridge, “On the anonymity of home/work location pairs,” in *International Conference on Pervasive Computing*. Springer, 2009, pp. 390–397.

[39] K. Haynes, P. Fearnhead, and I. A. Eckley, “A computationally efficient nonparametric approach for changepoint detection,” *Statistics and Computing*, vol. 27, no. 5, pp. 1293–1305, 2017.

[40] G. Kar, S. Jain, M. Gruteser, J. Chen, F. Bai, and R. Govindan, “Predriveid: pre-trip driver identification from in-vehicle data,” in *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*. ACM, 2017, p. 2.

[41] A. Khizhniak, “Bmw delivers iot services to 1m car owners by using ibm cloud foundry,” May 2019. [Online]. Available: https://www.altoros.com/blog/bmw-delivers-iot-services-to-1m-car-owners-by-using-ibm-bluemix/

[42] H. Li, D. Ma, B. Medjahed, Y. S. Kim, and P. Mitra, “Analyzing and preventing data privacy leakage in connected vehicle services,” *SAE International Journal of Advances and Current Practices in Mobility*, vol. 1, no. 2019-01-0478, pp. 1035–1045, 2019.

[43] ——, “Data privacy in the emerging connected mobility services: architecture, use cases, privacy risks, and countermeasures,” *SAE International Journal of Transportation Cybersecurity and Privacy*, vol. 2, no. 11-02-01-0004, pp. 49–61, 2019.

[44] G. Madzudzo and M. Cheah, “Data protection and connected vehicles: Privacy policy analysis from a consumer perspective,” November 2020.

[45] mdp93, “Mdp93/pricar_secdev.” [Online]. Available: https://github.com/mdp93/PRICAR_SecDev

[46] M. D. Pesé, X. Pu, and K. G. Shin, “Spy: Car steering reveals your trip route!” *Proceedings on Privacy Enhancing Technologies*, vol. 2, pp. 155–174, 2020.

[47] M. Pesé, K. Shin, J. Bruner, and A. Chu, “Security analysis of android automotive,” *SAE Int. J. Adv. & Curr. Prac. in Mobility*, vol. 2, pp. 2337–2346, April 2020, https://doi.org/10.4271/2020-01-1295.

[48] M. D. Pesé, “A first look at android automotive privacy,” SAE Technical Paper, Tech. Rep., 2023.

[49] M. D. Pesé, D. Chen, C. A. Campos, A. Ying, T. Stacer, and K. G. Shin, “Detroit: Data collection, translation and sharing for rapid vehicular app development,” in *2022 19th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 2022, pp. 397–406.

[50] M. D. Pesé and K. G. Shin, “Survey of automotive privacy regulations and privacy-related attacks,” in *SAE Technical Paper*. SAE International, April 2019, https://doi.org/10.4271/2019-01-0479.

[51] Pymnts, “Who controls data in web-connected vehicles?” Jun 2018, https://www.pymnts.com/innovation/2018/data-sharing-smart-cars-privacy/.

[52] G. J. van den Burg and C. K. Williams, “An evaluation of change point detection algorithms,” *arXiv preprint arXiv:2003.06222*, 2020.

[53] 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.

[54] L. Zhou, Q. Chen, Z. Luo, H. Zhu, and C. Chen, “Speed-based location tracking in usage-based automotive insurance,” in *Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on*. IEEE, 2017, pp. 2252–2257.