ExHPD: Exploiting Human, Physical and Driving Behaviors to Detect Vehicle Cyber Attacks

Qian Chen*, Paul Romanowich†, Jorge Castillo*, Krishna Chandra Roy*, Gustavo Chavez‡, and Shouhuai Xu§
*Department of Electrical and Computer Engineering, The University of Texas at San Antonio, San Antonio, TX, 78249 USA †Department of Psychology, Gonzaga University, Spokane, WA, 99258 USA
‡Automotive & Transportation, Southwest Research Institute, San Antonio, TX, 78238 USA
§Department of Computer Science, University of Colorado Colorado Springs, Colorado Springs, CO 80918 USA.
Work done when affiliated with The University of Texas at San Antonio, San Antonio, TX, 78249 USA

***Abstract*—As increasingly more vehicles are connected to the Internet, cyber attacks against vehicles are becoming a real threat with devastating consequences. This highlights the importance of detecting vehicle cyber attacks before fatal accidents occur. One natural method for tackling this problem is to adapt existing approaches for detecting attacks in enterprise networks, but which has achieved limited success. In this paper, we propose a new approach to treat vehicles as cyber-physical-human systems, leading to a novel framework called Exploiting Human, Physical and Driving Behaviors to detect vehicle cyber attacks (ExHPD). The framework has 4 detectors: a human detector, a physical behavior-based detector, a driving behavior-based detector, and an integrated physical and driving behavior-based detector. As proof of concept, we recruited 50 drivers to conduct IRB-approved simulation-based driving tests. Experimental results show that ExHPD is effective to detect vehicle cyber attacks and avoid deadly crashes by offering drivers adequate time to safely pull over their compromised vehicle. The impact of driver's impulsiveness (one aspect of human factors) on the detectors' effectiveness and limitations of the present study are discussed. Future research directions towards an ultimately usable solution are outlined.**

***Index Terms*—Cyber-Physical-Human System and Vehicle Cybersecurity.**

I. Introduction

The vehicle industry is increasingly adopting the Internet of Things (IoT) techniques to provide important functions such as smart mobility [1]. Some researchers also believe that the use of IoT techniques can improve transportation safety by reducing 400k-600k crashes each year [2]. Achieving this would be a significant contribution because 1.25 million people are killed by vehicle crashes around the world each year [3], including 37k fatalities in the United States that incur $242 billion economic loss in productivity, property damage and medical costs, and $594 billion in the loss of life, the pain and decreased quality of life [4], [5]. Adopting the IoT technology is a double-edged sword as it renders vehicles vulnerable to vehicle cyber attacks, such as the following. (i) A cyber attacker can remotely control a compromised vehicle and make profits by extorting the driver (e.g., turning off their engines [6]), similar to ransomware attacks against computers. Although a compromised vehicle may be repaired by re-installing its software system, a driver may prefer to pay a relatively small ransom to "unlock" the vehicle rather than paying a larger amount of fees or waiting for a long time for software re-installation. (ii) A cyber attacker can also obstruct traffic by commanding compromised vehicles under their control to benefit the attacker. For example, by manipulating the GPS services of the compromised vehicles to route those vehicles to specific (e.g., heavy traffic) roads while possibly making profits by instructing subscribing drivers to avoid such specific roads [7]. (iii) A cyber terrorist can abuse compromised vehicles as weapons to attack pedestrians, deliver explosive devices, or breach physical security barriers [8]. Although some cyber attacks mentioned above have yet occurred, the number of vehicle cyber attacks is increasing rapidly: 60 attacks among the documented 170 incidents occurred in 2018, and 51 attacks in the first quarter of 2019 (i.e., 300% more than what occurred in the first quarter of 2018) [9].

Vehicles are important for society and vehicle cyber attacks would become inevitable. It important to detect those cyber attacks before devastating consequences occur. This research problem has started receiving attention. Several studies have focused on leveraging *physical* behaviors of Electronic Control Units (ECUs) and In-Vehicle Network (IVN) to detect vehicle cyber attacks.

Our contributions. We make three contributions. First, different from most of previous cyber security research which considers a vehicle as a cyber-physical system, we propose viewing vehicles as *cyber-physical-human* (CPH) systems, where *cyber* means the IoT-enabled data communication and processing components, *physical* means the mechanical and electrical components, and *human* means the vehicle driver. Due to legal and ethical concerns, this CPH view will remain valid even after the level 5 self-driving is available. Human driver is the ultimate detector who would stop driving a vehicle when diagnosing any problems. However, a driver cannot and is not capable of detecting small changes of the vehicle incurred by cyber attacks, which calls for automated detectors to detect vehicle cyber attacks timely, ideally before crashes occur. The involvement of a driver makes it imperative to understand the impact of human factors on vehicle cyber security that distinguish vehicle cyber security from other application settings.

Manuscript received October, 2020. Corresponding author: Q. Chen (email: guenevereqian.chen@utsa.edu)

Second, we propose **Ex**ploiting **H**uman, **P**hysical and driver's **D**riving behaviors to detect vehicle cyber attacks (ExHPD for short). The ExHPD framework has 4 detectors: the driver as a Human Detector (HD); the vehicle Physical Behavior-based Detector (PBD); the driver's Driving Behavior-based Detector (DBD); and an integrated Physical and Driving Behavior-based Detector (PBDB). We hypothesize that human factors impact the detectors' effectiveness in detecting vehicle cyber attacks. As human factors are very broad topics, in this work we focus on driver's *impulsiveness* or more specifically its quantifiable manifestation known as *delay of gratification* or *Delay Discounting* (DD). In psychology, DD rates correspond to how quickly a reward loses value as a function of time; a high DD rate implies quick reward devaluation. Inspired by the fact that DD rates are related to risk-taking car driving [10], and impulsive drivers produce more driving errors [11], we propose investigating the relationship between drivers' impulsiveness to the driver whose vehicle is crashed by cyber attacks.

Third, as a proof of concept, we design a simulated a hardware-in-the-loop driving simulation testbed that is extended from *OpenDS* [12]. The testbed design and driving simulation test source code will be open sourced, which have independent value. 50 college-age drivers are recruited to conduct IRB-approved simulated driving tests, during which we wage simulated cyber attacks against the testbed (vehicle) by modifying its input signals. We find that low-DD drivers are more effective HD than high-DD drivers, but HD does not detect crashes until it is too late. We show that PDB, DBD, and PDBD using both supervised and unsupervised machine learning approaches are effective, including detecting 19 attacks that cause crashes and offering (in most cases) at least 4 seconds surviving time for drivers to safely stop the compromised vehicles to avoid crashes. The driver's impulsiveness has no significant impact on PBD's effectiveness. DBD and PDBD are more effective for less impulsive drivers in detecting vehicle cyber attacks.

Paper outline. Section II introduces related prior work. Section III describes the ExHPD framework. Section IV presents a case study with experimental results. Section V discusses the limitation of this study and future research directions. Section VI concludes the paper.

II. Related Work

We divide related prior studies into three categories: those which investigate how to detect and protect from vehicle cyber attacks, those which investigate how to use human's driving behavior to identify different drivers (including information of driving testbed, simulated/real-world driving tests and machine learning detection models), and those which investigate human's impulsiveness via DD rates.

A. Vehicle Cyber Attack Detection and Protection

The problem of detecting vehicle cyber attacks has been investigated by Cho and Shin [13], who propose leveraging clock skews to fingerprint a vehicle's transmitter Electronic Control Units (ECU) and model vehicle clock behaviors. The compromised or alien ECUs can also be detected by leveraging time and frequency domains of ECU signals [14]. Kneib and Huth [15] develop a signal characteristics-based sender identification and detection mechanism to detect attacks from infiltrated ECUs or additional devices. Choi et al. [16] propose detecting attacks by leveraging voltages of in-vehicle network (IVN). Nowdehi et al. [17] learn the normal behavior of IVN dynamics from historic data to detect deviations in IVN traffics. These studies leverage *physical* behaviors of ECUs and IVNs to detect attacks. By contrast, we propose leveraging vehicle physical behaviors and driver's driving behaviors to detect attacks, while investigating the impact of driver's impulsiveness on our detectors' effectiveness.

Kerrache et al. [18] consider the human factor and their impacts to assure trustiness among inter-vehicle communication using Online Social Networks (OSNs). Towards realizing intervehicle trust, the researchers combine the calculation of intervehicle trusts with and without considering Human Honesty Factor (HHF) using Advogato1 classification. Using the map of Laghouat city in Algeria (via 4km2), where the experimental region has 4 randomly deployed RSUs and testing on different vehicles and their drivers, this approach achieves 95% detection rate and also reduces the detection error ratio by about 3%. The researchers later improve their previous work, and propose TACASHI in the following paper [19] to prevent malicious drivers from provoking unwanted situations such as stolen vehicles using location-related honesty (LRH). By using the NS-2.35 simulator and a benchmark dataset, the experimental results also provide a 95% confidence for misbehavior detection as the previous work. Moreover, even under the worst-case scenario, TACASHI takes only 5 seconds for honesty estimation, which can timely prevent terrorist attacks or stolen vehicles.

Mekki et al. [20] apply evolutionary game theory (EGT) to dynamically select appropriate access technology for cooperating or accessing the conventional cloud through the 4G-LET link. Vehicles can change access strategies according to different conditions until reaching equilibrium. Using the NS 3 simulator, researchers evaluate the EGT algorithm by a case study of downloading services. The experimental results indicate that EGT is better suited for highways, vehicles traveling to the same destination, or simply vehicles with low mobility. However, it suffers from scalability issues in comparison with their previous work [21], where the Q-learning algorithm was used to provide the same service. Yahiatene et al. [22] propose a blockchain-based architecture to improve the security of the software-defined vehicular network (SDVN) while protecting user privacy in a fully distributed network by ensuring data anonymity. The transactions in the blockchain consist of shared content between the vehicular social network (VSN) entities. This study also introduces a miner selection algorithm distributed miners connected dominating set (DM-CDS) based on a trust model and network parameters. The trust model identifies misbehavior in VSN leveraging connectivity, fitness, and satisfaction measures of the nodes.

The efficient vehicle cybersecurity solutions development and deployment require high quality of experience (QoE) and

quality of service (QoS) networks. Abar et al. [23] introduce a fog computing-based high throughput information-centric networking (ICN), which contains cloud-based computing, storage, and networking facilities. Specifically, the new architecture is efficient to enhance QoE sensitive applications such as multimedia content delivery for future communication networks. Jabri et al. [24] propose a new decentralized vehicular fog architecture and use fuzzy logic-based gateway selection module to solve the challenges of a large amount of vehicular cloud access traffic. The multi-objective optimized gateway selection module reduces communication cost in terms of bandwidth consumption and cellular link usage. Fabian et al. develop an original architecture and a programmable objective function to improve QoS in the IoV [25]. This new architecture significantly improves QoS such as packet delivery ratio, packet loss, and energy consumption.

B. Driving Tests and Driver Identity Detection

The problem of identify driver identity has been investigated by Hallac et al. [26]. The research team recruits 64 drivers to operate 10 Audi vehicles for 2,098 hours covering 110,023 kilometers on real roads in Ingolstadt, Germany. The driving behavior dataset is composed of sensor signal values of single-turns and straight ways to build driver identification models using the random forest, SVM and multinomial logistic regression methods. The experimental results show that turns are better than straight ways for detecting variations across drivers.

Nagoya University and Toyota Central R&D Labs model drivers' behavior via spectral analysis. They recruit 12 participants who complete a driving test on a driving simulator that shows a two-lane expressway following a lead vehicle [27]. Twelve drivers conduct the five-minute long test four times. Three of the four driving test behavior datasets are used for training a Gaussian Mixture Model, which is tested by the other dataset. The research shows that (i) velocity, following distance, gas or brake pedal angles and their their dynamics provide a 89.6% driver identification rate, and (ii) a model taught with the gas/brake pedal feature outperforms a model taught from the other two features.

Later improvements in papers [28] and [29] show that using features like brake and gas pedal angles can achieve a higher drive detection accuracy. Van Ly et al. [30] investigate the binary classification of drivers, which is useful when identifying two family members who share a car. Their experiment uses a 2008 Volkswagen Passat Variant 3.6L modified to include sensors and vision systems at different times of the day to increase driving variation in congested and non-congested traffic. They show that using inertial sensors to capture driving behavior and using machine learning to classify driving events can reduce driving danger. Jafarnejad et al. uses 19 dynamic signals currently available in production cars to detect 75% of the drivers in less than 65 seconds in a 5 drivers scenario [31].

Meseguer et al. [32] develop a mobile platform called DrivingStyle to classify driving styles and vehicle fuel consumption of different drivers. This platform can promote eco-style driving to reduce fuel consumption fuel economy and enhance driving safety. The architecture of DrivingStyle is comprised of an Android app that collects data from an OBD-II Bluetooth device; a data center to store all collected data; a neural network that is trained by the most representative routes for in-situ analysis, and another neural network for off-site analysis. 534 drivers' driving data was used to evaluate the efficiency of DrivingStyle. The experimental results illustrate that when a drive adopts an efficient driving style, at least 15% fuel consumption can be reduced.

Our driving test and driving simulation environment are different from all these prior efforts as we designed driving simulation tests focused on collecting human psychology data, vehicle physical behaviors, and human driver's driving behaviors when the vehicle (i.e., simulator) was operated normally as well as when it was under various cyber attacks. We also used both supervised and unsupervised machine learning and deep learning methods to predict and detect these attacks efficiently.

C. Prior Work on DD Research

While we are the first to investigate the impact of driver's DD rate on the effectiveness of vehicle cyber attack detectors, DD is known as a trans-disease process [33], [34], whereby findings from one disorder (e.g., cigarette use) can inform other seemingly unrelated disorders (e.g., pathological gambling). We hypothesize that individuals who are more impulsive during a DD task is also more impulsive during a driving task.

In psychology, DD describes how quickly a commodity loses value as a function of time, and DD is one aspect of impulsiveness. Previous research shows that DD exhibits both state and trait-like properties, depending on the individual's context [35]. For example, there is a robust literature showing DD state-like properties for individuals who quit smoking [36], whereby smokers are more impulsive (i.e., devalue hypothetical monetary outcomes more quickly) than former smokers and nonsmokers. That is, DD seems to be malleable depending on different individual states. In addition, evidence shows that positive correlations between DD tasks completed at different times and use as a prospective measure, suggesting more trait-like properties. DD has been shown to be useful as a prospective measure for smoking cessation success across many different studies [37].

There is also a large body of research that suggests DD is a trans-disease process, whereby findings from one disorder (e.g., cigarette use) can inform other seemingly unrelated disorders (e.g., pathological gambling). If DD were domain-specific, then the previous literature would have little relevance for driving behavior. However, this trans-disease process predicts that individuals that are more impulsive during a DD task will also be more impulsive during a simulated driving task. In addition, the variation in driver errors during the driving simulation suggest a novel application to driver authentication. Previous DD research has also shown that people discount real and hypothetical rewards at similar rates and hypothetical rewards are a valid proxy [33], [34]. Therefore, a DD task using hypothetical rewards is a cost-efficient behavioral method to differentiate between drivers who are more or less likely to make driving errors.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3069951, IEEE Internet of Things Journal

In the present study we proposed to leverage drivers' DD rates as a feature to design more accurate detectors to protect the vehicles from cyber attacks. We also investigated which drivers are more vulnerable to vehicle cyber attacks using the DD rate as a measurement.

III. The ExHPD Framework

A. Model, Human Factor, and Design Objective

System model. Figure 1 shows the vehicle system model with (i) a cyber sub-system, such as an in-vehicle network and IoT-enabled vehicle-to-everything (V2X) communications (e.g., for downloading music or other Internet-based entertainment services); and (ii) a physical sub-system, which consists of vehicle safety-critical functionalities such as braking and steering. We consider the setting that each vehicle is operated by a driver, rather than self-driving vehicles.

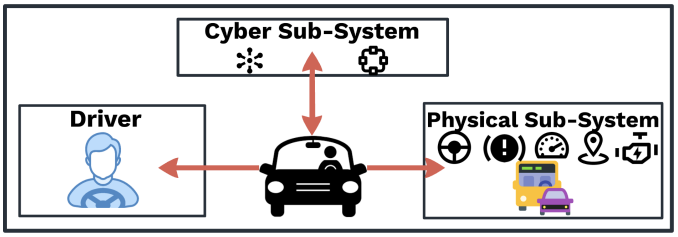


Figure 1: The CPH system model of vehicles.

Threat model. We consider cyber attacks against the vehicle's cyber sub-system to control safety-critical functions such as the steering wheel, brake, and accelerator. Only one type of vehicle cyber attack each time. Physical attacks against the physical sub-system and attacks against the driver are excluded.

Vehicle cyber attacks vs. vehicle failures. Vehicle cyber attacks are different from common vehicle failures. Vehicle mechanical failures such as worn brake lines, engine, and transmission problems or tires blowout normally have some signals for drivers to notice (e.g., noise and dashboard warning lights). These signals last a while before the vehicle breaking down or causing a traffic accident/crash. Vehicle cyber attackers aim at undermining the safety of the people in and around the car, usually immediately disable/malfunction vehicle safety-critical systems. The vehicle system therefore is not allowed to provide in-time warnings or noticeable signals to drivers before crashes occurring. The levels of vehicle cyber attacks that compromise vehicle safety-critical functions can be defined as severe or high that depend on how quickly a crash occurs since such vehicle cyber attack is launched. The vehicle cyber attacks are severe if they lead to car crashes immediately after being performed.

Easy/hard noticed vehicle cyber attacks. During the time period when the vehicle cyber attacks start to perform until the attacks stops, severe/high level attacks can cause vehicle crashes if drivers uses the vehicle's malfunctioning components to control the vehicle. These attacks are easy noticeable by human drivers. However, the same attacks can be "temporally elevated", which will not result in crashes if drivers do not use compromised safety-critical components. In this case, the vehicle still performs normally, and these attacks are hard to notice or not noticeable by human drivers.

Human factor. We need to consider the driver's human factors and their impact on this CPH vehicle system. Human factors are a broad area and in this paper, we focus on one specific human factor, namely *impulsiveness*. Impulsiveness is a human personality trait and a multidimensional psychological construct that describes when individuals cannot inhibit a response, do not plan before an action is taken, and/or have difficulty delaying gratification [38]. In Psychology, one aspect of impulsiveness is quantified by the Delay Discounting (DD) rate, which describes how quickly a commodity loses value as a function of time. Figure 2 highlights our insight into the relationship between impulsiveness and driving behavior, which has not been investigated until now.

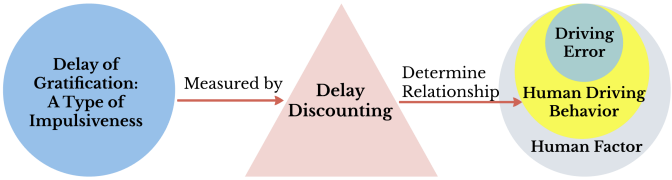


Figure 2: The connection between impulsiveness and driving behavior, which we propose to investigate and is broader than the driving errors investigated in the literature.

Design objective. We propose treating vehicles as cyber-physical-human systems because a human driver is a natural detector for determining whether or not a vehicle is safe to drive. When a driver notices that something goes wrong with the vehicle (e.g., changes in physical behaviors like brake malfunctioning), the driver would stop driving it before an accident occurs. The *design objective* is to create a framework for detecting vehicle cyber attacks *as early as possible* (or before crashes occur) and alerting the driver to stop the vehicle immediately to save human lives from potential accidents. Moreover, the detectors should be practical, meaning that they make realistic assumptions and are applicable to a wide range of scenarios (e.g., accommodating driver differences).

B. Framework

As highlighted in Figure 3, the framework has 4 detectors: (i) *manual* Human Detector (HD), which is the driver that may be able to detect attack-caused vehicle malfunctions. (ii) *automatic* Physical Behavior-based Detector (PBD), which aims to detect vehicle cyber attacks by leveraging attack-caused changes in a vehicle's physical behaviors; (iii) *automatic* driver's Driving Behavior-based Detector (DBD), which aims to detect vehicle cyber attacks by leveraging attack-caused driving behavior changes; (iv) *automatic* integrated Physical and Driving Behavior-based Detector (PDBD).

Human Detector (HD). The driver can detect a vehicle's physical behavior changes that are caused by cyber attacks, such as the vehicle going around in a circle; the vehicle steering is uncontrollable; the vehicle's engine emits strange sounds

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3069951, IEEE Internet of Things Journal

and/or vibrations; the vehicle's battery drains abruptly; the vehicle steering, brake, and/or accelerator is less sensitive than normal; the vehicle navigation system gives wrong directions. Humans can recognize these matters, although they are hard to formalize. This compels us to investigate the impact of human factors on the effectiveness of HD. While it is intuitive that a careful driver would recognize many cyber attacks, this has to be (in)validated by experiments.

Physical Behavior-based Detector (PBD). Cyber attackers may not incur substantial physical behavior changes until causing crashes, meaning that some attacks are not noticed by a driver until it is too late to avoid an accident. This reiterates the importance of designing automatic detectors to detect vehicle cyber attacks that cannot be recognized by HD. We propose detecting cyber attacks by leveraging the vehicle's physical behavior changes that may not be noticed by human drivers, such as: steering, brake and/or accelerator sensitivity, which may only make a vehicle slightly move to the right/left or cause slightly slower or faster speed than usual but may not be noticed by the driver. We define the following features to measure vehicle physical behavior changes.

- F_1 *Vehicle longitudinal position*, namely the distance between the starting point and the current vehicle location on the trajectory map. This feature is measured by the GPS tracking system.
- F_2 *Vehicle lateral position*, namely the distance between the center of a vehicle and the center of the driving lane.
- F_3 *Vehicle speed*, can be measured by the GPS tracker.

Driving Behavior-based Detector (DBD). Vehicle cyber attacks may cause a driver's driving behavior to change, such as the following. (i) The driver steers the wheel more/less frequently and/or with a higher/lower degree than usual, for example when the steering wheel drifts/pulls slightly or when the steering wheel is harder/easier to turn. (ii) The driver presses the brake/accelerator more frequently than usual, for example when the brake and accelerator are less sensitive. (iii) The driver takes a substantially longer time to drive over a certain distance, for example when driving from home to school under similar traffic situations. (iv) The driver makes more driving errors than usual, for example by running red lights and/or crossing lanes. For measuring these driving behavior changes, we define the following features.

- F_4 *Steering angle*, namely the angle between the front of the vehicle and the steering wheel direction.
- F_5 *Brake pedal position*, namely the pressure transferred from the brake pedal to the brake pads to stop the vehicle.
- F_6 *Accelerator pedal position*, namely Deflection Angle of the electronic throttle control to vehicle speed.
- F_7 *Reaction time*, namely the time interval between a driver is instructed to start a driving task (e.g., lane changing) and when the driver finishes the driving task.
- F_8 *DD Rate*, which measures a driver's impulsiveness and is representative of the driver's human factor.

Integrated Physical and Driving Behavior-based Detectors (PDBD). The preceding three detectors can be collectively used together to achieve higher effectiveness. It is also reasonable to treat alerts from the three detectors seriously, meaning that drivers should stop driving a vehicle immediately when they receive an alert from these detectors. With measurement data of the eight features, PBD, DBD, and PDBD can be trained using machine learning techniques. Note that F_7 is affected when drivers encounter new situations corresponding to vehicle cyber attacks [39]. F_8 is relevant as more impulsive drivers produce more driving errors [40].

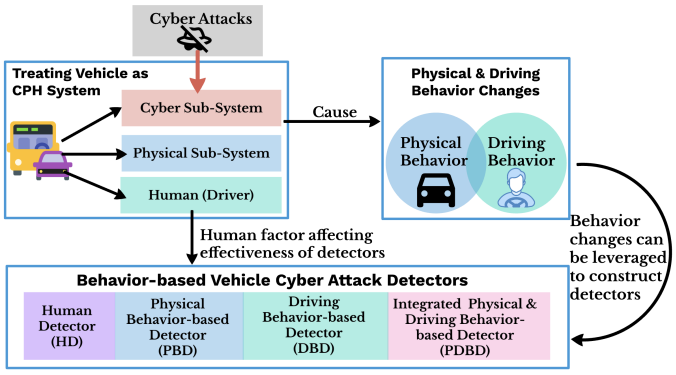


Figure 3: The ExHPD framework with 4 detectors: HD, PBD, DBD and PDBD

C. Detector Effectiveness Metrics

Detector effectiveness is evaluated using standard metrics [41], [42]: (i) *accuracy*, the percentage of correct predictions among all predictions; (ii) *true-positive rate* (TPR), the percentage of correct predictions for attacks; (iii) *false-positive rate* (FPT), the percentage of normal behaviors that are predicted as attacks; (iv) *false-negative rate* (FNR), the percentage of attacks that are predicted as normal behaviors; (v) *precision*, the percentage of true-positive among positive predictions; and (vi) *area under the curve* (AUC), the percentage of the area below the *receiver operating characteristic* (ROC) curve in the ROC plot. We also define (vii) *detection delay*, the time between when an attack is waged and when the attack is detected; (viii) *# crashes avoided*, the number of crashes that are avoided, assuming a driver can immediately and safely stop driving a vehicle upon receiving the detector's alert; and (ix) *surviving time*, the time a detector offers to a driver to safely stop the vehicle before it crashes.

IV. Case Study

We recruit drivers to conduct simulation-based driving tests to show the usefulness of the ExHPD framework, which are approved by the local Institutional Review Board (IRB).

A. Simulation-based Driving Testbed

Our driving testbed is extended from OpenDS [12], which is an open source driving simulator software built upon the Java video game engine called jMonkeyEngine (jME). OpenDS contains multiple pre-defined driving tasks, such as the *reaction test* that requires drivers to reduce vehicle speed or change lanes when signals are presented on the screen. We extend the pre-defiend reaction test to incorporate our driving

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3069951, IEEE Internet of Things Journal

tasks. Figure 4 highlights the testbed. As shown on the left, a computer monitor runs the extended OpenDS software and a Logitech G29 driving force racing wheel and pedal [43] for a driver to control the vehicle. On the right are two pictures indicating a driver unsuccessfully vs. successfully finishing a driving task.

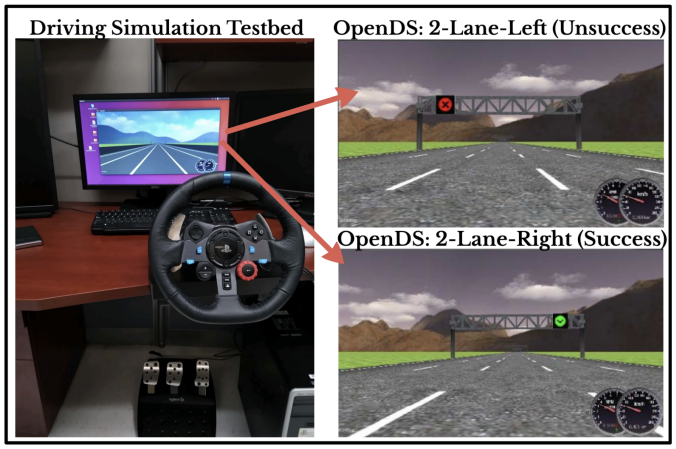


Figure 4: Experimental testbed with a computer running the extended OpenDS software.

Figure 5 is a package UML diagram that highlights our extensions to OpenDS. The extensions include 2 added classes to the `trigger` package and 13 modified classes in packages `main`, `niftyGUI`, `reactionCenter`, `jasperReport` and `drivingTask`, where a dashed arrow indicates a dependency relationship between a pair of packages (i.e., A pointing to B means package A imports or accesses package B). The modifications are: 3 classes of package `niftyGui` are extended to display instructions to drivers and shutdown the GUI; 1 class of package `reactionCenter` is extended to set up a brake timer, hold speed, and add a sound effect to notify drivers when they succeed or fail a driving task; 3 classes of package `trigger` and 3 classes of package `drivingTask` are extended as well as another 3 classes are added into package `drivingTask` to reset the driving simulation test, open the driving simulation test GUI, and shutdown the driving test; 1 class of package `main` and 1 class of package `jasperReport` are extended to reflect and support modifications in the other classes. Also, the default simulator launching interface is extended to support anonymous participation, such that a driver is only asked for an index number.

To collect feature measurements F_1-F_8, we build a *five-lane straight road* driving test model and remove the hundreds of small segments provided by OpenDS. This is reasonable as OpenDS' default reaction test has a poor graphics performance (lower than 30 frames per second or FPS), even if we run it on a high-end computer with a modern GPU. This poor performance is inherent to OpenDS' design of driving track loading. In the pre-defined driving simulation test, the 1.1-kilometer straight driving track is composed of a large number of small road segments whose lengths are the same as the length of the simulated vehicle; when hundreds of driving test models are loaded from the `scene` XML file into the memory, human driver's reactions to situations are significantly delayed. As a side-product, two bugs of software are found and fixed by changing the shape of the road model to match roadside barriers. This change can prevent the vehicle from driving away, and reaction time collection failures caused by vehicles move around gantries.

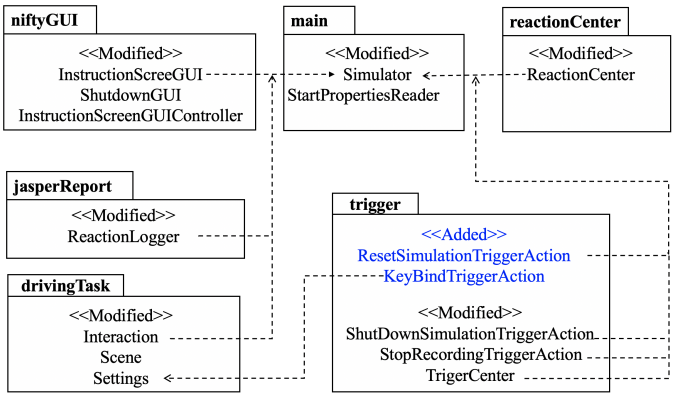


Figure 5: Illustrating our extension to OpenDS, including 2 *added* classes and 13 *modified* classes.

B. Driving Tests, Attacks and Data Collection

Driving tests. Three driving tests dubbed Test 1, 2, and 3 are designed as mentioned above. Test 1 is designed for practice (i.e., making drivers familiar with the testbed) with no cyber attacks. Test 2 is normal driving with no cyber attacks. Test 3 has cyber attacks waged against the simulated vehicle, but the drivers have not been told about these attacks. Data collected in Test 2 is used as the normal physical and driving behaviors, and the data collected in Test 3 can reflect the physical and driving behaviors under cyber attacks. In any test, no other vehicles are on the track when a driver takes a driving test. A driver drives on a 22-gantry and 1.1 kilometers (km) straight road comprised of 10 lane-change and 10 speed-change tasks, where the distance between two gantries is set to be 50 meters and is divided into 200 units (i.e., 0.25 meters per unit). A driver is given one driving task every gantry (50 meters).

Table I lists 20 driving tasks. A one-lane-change task asks a driver to change one lane to the left or right; a two-lane-change task asks a driver to change two left/right lanes at a time. After finishing any lane-changing tasks, a vehicle should be driven back to the middle lane, and the driver should operate the vehicle in the middle lane until the next change-lane task is given. A speed-change task asks a driver to decrease the vehicle speed to 20 km/h from 60 km/h in 10 seconds. The speed is accelerated back to 60km/h when speed-change tasks are finished. A driver typically finishes a driving test within 5 minutes when no cyber attacks against the vehicle. At the end of Test 3, the driver is asked to complete a post-test survey.

Simulated attacks. 3 vehicle cyber attacks are simulated in Test 3 such that each is waged twice. Attacks 1 and 4 are waged against the steering system leading to the vehicle to turn the *opposite* direction. The attacks are simulated by

the newly added OpenDS class `KeyBindTriggerAction`, which reverses the OpenDS's input signals received from the vehicle's steering system (i.e., the vehicle's Logitech G29 driving force racing wheel). Attacks 2 and 5 are waged against the vehicle's brake. Attacks 3 and 6 are waged against the vehicle's accelerator. These two kinds of attacks are simulated by modifying OpenDS' class `KeyBindTriggerAction` to respectively set the signals received from the vehicle's brake and accelerator to 0, meaning that the brake/accelerator stops functioning immediately. As shown in Table I, each attack is waged 50 units (or 12.5 meters) before the corresponding task starts. The attack finishes 50 units before the task ends, and the duration of each attack is 200 units.

These three attacks are all severe/high levels which can cause vehicle crashes when the driver using the malfunctioning steering system, break or accelerator to control the vehicle. However, these attacks will not lead to crashes if drivers do not use malfunctioning components when they are compromised. For example, when the vehicle's brake is compromised, if drivers do not press the brake during that period of time (200 units), the vehicle still performs "normally" at that time, and the vehicle cyber attacks would not be noticed by drivers. Therefore, to create a driving test containing both human noticeable and not noticeable vehicle cyber attacks, we launch these the three types of cyber attacks when drivers perform different driving tasks. Attack 1 and Attack 4 are easy noticed attacks, but the other four attacks are hard noticed attacks.

Data collection and pre-processing. In Test 3, we observe crashes caused by cyber attacks. A driver's test is terminated when the vehicle crashes or when the driver cannot operate the vehicle back to the five-lane roadway. We collect data by measuring the 8 features F_1-F_8 mentioned in the framework during the aforementioned Tests 2 and 3. Features F_1-F_6 are collected by OpenDS' software sensors 20 times per second (i.e., sampling time—one measurement per 0.05 seconds). F_7 is counted from the point in time at which a driving task sign appears on the screen (near each gantry) to the point in time at which the driver finishes (or fails) the task in question; F_7 is in the range of 0 to 10 seconds (precision: millisecond), where the value of 10 seconds means the driver fails to finish a driving task. Since there are 20 driving tasks during each test, there are 20 reaction times for each driver, assuming the driver finishes the driving test. For each driver, F_8 is derived from a driver's impulsiveness (elaborated below). In Test 2 (absence of attacks), a driver is represented on average by 5,300 feature measurements F_1-F_6, 20 measurements for F_7 and 1 measurement for F_8; there is no significant variance in the number of measurements for the 50 drivers because they all finish the driving test within a similar period of time. In Test 3 (presence of attacks), some vehicles crash, and the drivers have to terminate the driving test, which causes a large variance in the number of measurements between the 50 drivers. On average, each driver has 7,000 measurements for F_1-F_6, 20 measurements for F_7 and 1 measurement for F_8.

For machine learning purposes, we pre-process the data by using measurement-centered data representation. Consider driving test c, where $c \in \{2, 3\}$. For the d-th ($1 \leqslant d \leqslant 50$) driver, the ℓ-th ($1 \leqslant \ell \leqslant 20$) driving task is represented by a sequence of $m_{c,d,\ell}$ measurements of features F_1-F_8, where the value of $m_{c,d,\ell}$ depends on the measurement frequency (which further depends on the driver's driving speed) and there being accidents or not. The data corresponding to the d-th driver is represented as $N_{c,d} = \{(n_{c,d,\ell,i,1}, \ldots, n_{c,d,\ell,i,8})\}_{1\leqslant\ell\leqslant 20, 1\leqslant i\leqslant m_{c,d,\ell}}$ where $n_{c,d,\ell,i,j}$ is the value of the j-th feature (F_j where $1 \leqslant j \leqslant 8$) obtained at the i-th measurement during the ℓ-th driving task of the d-th driver in driving test c. Since feature F_7 is measured once per driving task, the same value is used for every measurement during the task, namely $n_{c,d,\ell,1,7} = \ldots = n_{c,d,\ell,m_{d,\ell},7}$. Since each driver only has one DD rate (i.e., there is only one measurement for feature F_8), the same value is used for each measurement, meaning that $n_{c,d,\ell,i,8}$ is the same for $1 \leqslant \ell \leqslant 20$ and $1 \leqslant i \leqslant m_{c,d,\ell}$. Given $N_{c,d}$ for $1 \leqslant d \leqslant 50$, the entire dataset corresponding to driving test c is represented by $N_c = \cup_{1\leqslant d\leqslant 50} N_{c,d}$. The entire dataset corresponding to driving Tests 2 and 3 is defined as $N = N_2 \cup N_3$.

Machine learning algorithms and hardware. We use the Random Forest algorithm to train supervised PBD, DBD and PDBD detectors on a computer with Intel Core i-7 and 8GB-Ram because random forest can handle high dimensional and unbalanced data, and provide quick prediction and training speed with a high variance but low bias. We use Long Short-Term Memory (LSTM) Autoencoder [44] to train unsupervised PBD, DBD and PDBD detectors to model sequence data because LSTM is good for predicting time-series training/test datasets. We use 4 hidden layers with each layer having 4 nodes, and 16 nodes respectively for the input and output layers. These numbers are selected as they perform the best among our experiments. The ReLU activation function and the Mean Absolute Error (MAE) loss function are used to train for 100 epochs. Since normal (or abnormal) is specific to individual drivers, we create one LSTM Autoencoder-based unsupervised detector for each driver from Test 2 measurements, and use these detectors to predict driver behaviors in Test 3. Each unsupervised detector is trained on an Nvidia Tesla K80 GPU server within 2 minutes.

C. Recruiting Drivers

We recruit 50 drivers from the campus of a large university in the U.S. to conduct the IRB-approved experiment. The recruitment method is to post the project flyers on campus and send digital flyers via email and social media. Interested drivers contact the lab coordinator to make an appointment for measuring their DD rates and conducting the experiments. The 50 drivers are between 18 and 50 years old, (A mean value of 23.2 and a standard deviation of 6.6), with 34% female. Among these 50 drivers, 47.3% are Hispanic, 32% are Caucasian and another 34% are Asian, African American and others. The mean age of these drivers when they start driving is 17.5 years (with a standard deviation of 1.6). We choose to recruit college student drivers for the following reasons: (1) traffic accidents are the leading cause of deaths among college students [45]; (2) college-age drivers account for a disproportionately high percentage of traffic accident

Table I: The 20 driving tasks in driving Test 1 and Test 2 (no cyber attacks). In Test 3, cyber attacks are waged against the brake (indicated by "Brake"), the accelerator (indicated by "Accelerator"), or the steering system (identified by "Steering"), and "NA" means no attack is waged.

Task #	Test 1-2 (Normal)	Test 3 (Attack)	Task #	Test 1-2 (Normal)	Test 3 (Attack)
1	speed-change	NA	11	1-lane-right	Attack 4. Steering
2	2-lane-right	Attack 1. Steering	12	2-lane-left	
3	speed-change		13	speed-change	NA
4	speed-change	NA	14	2-lane-left	Attack 5. Brake
5	speed-change	Attack 2. Brake	15	speed-change	
6	1-lane-left		16	1-lane-right	NA
7	speed-change	NA	17	speed-change	Attack 6. Accelerator
8	2-lane-left	Attack 3. Accelerator	18	1-lane-left	
9	speed-change		19	speed-change	NA
10	2-lane-right	NA	20	2-lane-right	NA

deaths than other groups of drivers [46]; (3) alcohol and drug usage increase the risk of traffic crashes[45], and DD has been repeatedly linked to drug and alcohol abuse [47], we expect to find significantly different DD rates among the college-age group which helps us to understand whether or not DD (or impulsiveness in general) would have an impact on the driving behavior changes and detector effectiveness.

D. Measuring Drivers' DD Rates

DD is often measured through a series of choices where an individual is presented with two mutually exclusive options. One option is for a smaller but sooner reward. The other option is for a larger but delayed reward. For example, an individual is asked to choose between either 5 dollars immediately or 10 dollars after two weeks. Waiting two weeks will double the amount, but increasing the delay between when the choice is made and when the reward is received decreases (i.e., discounts) the subjective reward value. This reward discounting is quantifiable and appears to be stable for individuals.

The DD rate or k value in the psychological literature is a metric for quantifying impulsiveness. 90% individual's k value $\in [0.00016, 0.25]$, and a larger k value means the individual is more impulsive [48]. This interval is discretized into a 9-dimension vector $\vec{x} = [0.00016, 0.00040, 0.0010, 0.0025, 0.0060, 0.016, 0.041, 0.10, 0.25]$, denoted by $\vec{x} = [x_1, \ldots, x_9]$, where $x_1 = 0.00016$,..., and $x_9 = 0.25$. Correspondingly, the psychological community defined an *ordered* set of delays (of days), denoted by $\vec{g} = [7, 14, 21, 30, 60, 90, 120, 150, 180]$, where $g_1 = 7$, ..., and $g_9 = 180$.

The psychological community designed the Monetary Choice Questionnaire (MCQ) [48] to measure an individual's DD rate. The basic MCQ contains 27 questions, denoted by Q_i for $1 \leqslant i \leqslant 27$. Question Q_i ($1 \leqslant i \leqslant 27$) asks a driver to choose between (0), namely receiving a smaller but immediate reward V_i, and (1), namely receiving a larger reward A_i in some D_i days, where V_i is a discounted value of A_i. For question Q_i where $1 \leqslant i \leqslant 27$, $i = 3(\alpha - 1) + \beta$ where $1 \leqslant \alpha \leqslant 9$ and $1 \leqslant \beta \leqslant 3$; then, V_i, A_i, and D_i are determined as follows.

- $A_i \in_R [\$25, \$35]$ if $\beta = 1$, $A_i \in_R [\$50, \$60]$ if $\beta = 2$, $A_i \in_R [\$75, \$85]$ if $\beta = 3$, where "$\in_R$" means choosing a value from a set uniformly at random.
- $D_i = g_{10-\alpha}$, meaning that the three questions corresponding to the same α have the similar delay value.
- $V_i = \frac{A_i}{x_\alpha D_i + 1}$, meaning that each question has a different V_i as it depends on both α (via x_α and D_i) and β (via A_i). $V_i < A_i$ explains that V_i is a discounted value of A_i.

The 27 questions are ascending ordered according to their x_i. The three questions with the same x_i are ascending ordered according to their A_i. We call it *original* order. These questions are presented to a driver in a *random* order; after collecting the responses from a driver, these questions are re-organized into the *original* order.

For reference, the 27 questions are summarized in Table VII of the Appendix. For question Q_i (in the original order), if the driver chooses (0), meaning the driver chooses to receive a smaller but immediate reward V_i, we denote it by $r_i = 0$; if the driver chooses (1), meaning the driver chooses to receive a larger but delayed reward, we denote it by $r_i = 1$. The responses of a driver formulate a binary vector $\vec{r} = [r_1, \ldots, r_{27}]$.

Algorithm 1 that has never been explicitly written in the psychological literature for calculating DD rates based on answers to the basic MCQ 27 questions [48], is a side-product of the computer science community. It computes the DD rate of a driver by taking $\vec{x}$ and $\vec{r}$ as inputs. As shown in Lines 2-5, if $\vec{r} = \vec{1}$, meaning the driver always chooses the delayed but larger reward A_i, then the driver's DD rate is set to x_{27}, namely $\max_i x_i$; if $\vec{r} = \vec{0}$, meaning the driver always chooses the immediate but smaller reward V_i, then the driver's DD rate is set to x_1, namely $\min_i x_i$; otherwise, we need to determine if the responses of a driver are consistent (i.e., useful in deriving the driver's DD rate; see Lines 7-15). Given the response vector $\vec{r}$, for each $1 \leqslant i \leqslant 27$ the consistency score, denoted by $\mathrm{ConsistencyScore}[i]$ and defined in Line 9, reflects (i) the driver's consistency in selecting the immediate but smaller reward with respect to $Q_1, \ldots, Q_i$, and (ii) the driver's consistency in selecting the delayed but larger reward with respect to $Q_{i+1}, \ldots, Q_{27}$. The driver's responses are *consistent* if one $\mathrm{ConsistencyScore}[i]$ is greater than or equal to the consistency threshold 0.75.

This is important because low consistency scores indicate that the driver does not pay due attention to the questionnaire; as a consequence, the reliable DD rate cannot be derived

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3069951, IEEE Internet of Things Journal

for the driver. Suppose the responses of a driver are deemed consistent, there are two scenarios. If the highest consistency score S has multiple appearances at some indices i's, then the driver's DD rate is the geometric mean of the corresponding x_i's (Line 19-23); otherwise, S corresponds to a unique index i and the DD rate of the driver is defined as the geometric mean of $x_i \times x_{i-1}$ (Line 24-28). Note that in any test, the DD rate of a driver who gave consistent responses falls into the internal $[x_1, x_{27}] = [0.00016, 0.25]$.

The *DD Rate* column in Table IV summarizes driver's DD rates. Based on this measurement result, we evenly split the 50 drivers into *high DD* and *low DD* groups (i.e., top 50% vs. bottom 50%) according to the drivers' median DD rate, which is 0.00972. This means that the low-DD drivers' DD rates fall into the interval $[0.00016, 0.00971]$ and the high-DD drivers' DD rates fall into the interval $[0.00972, 0.24837]$.

E. Consequences of Vehicle Cyber Attacks

In Test 3 we observe that Attacks 1 and 4 cause fatal vehicle crashes leading to the loss of life in the real world. Vehicles unexpectedly slow down due to Attacks 2-3 and 5-6, but no crashes occur. This because no other vehicles are in the driving test, which would not be true in practice. Also, these vehicle cyber attacks may cause accidents when vehicles do not keep a safe distance between each other. Table II summarizes the 19 crashes among the 50 drivers and the time interval between when an attack is waged and when a crash occurs. The 19 crashes include 18 crashes caused by Attack 1 and 1 crash caused by Attack 4. This hints that drivers surviving a certain attack (if their vehicles are not crashed) might survive the same kind of cyber attacks in the future. A crash can occur at most 29.9 seconds after an attack is waged (8.11 seconds on average); thus there is adequate time to detect attacks and alert drivers to stop their vehicles. The 19 drivers whose vehicles crash have an average 7.37-years of driving experience, whereas the other 31 drivers (not crashing) have an average 4.81-years of driving experience. Among the 19 drivers whose vehicles crash, drivers #13, #20, #27, #32 and #38's surviving time is longer than 10 seconds. These 5 drivers have an average 3.6-years of driving experience, which contrasts with the other 14 crashing drivers' average of 8.71-years of driving experience. Among these 19 drivers, 11 are high-DD drivers and 8 are low-DD drivers; 3 of the 5 drivers who show a long surviving time are high-DD drivers and the other 2 are low-DD drivers. These suggest that the driver's experience may be a factor for surviving cyber attack-caused crashes.

Insight 1: The driver's experience might play a big role in surviving drivers from cyber attacks-caused crashes.

F. HD and Its Effectiveness

In order to see HD effectiveness, we ask each driver to answer the following post-survey questions:

- Q1 Did you experience any problems during your 3rd driving test (i.e., Test 3)?
- Q2 If yes to Q1, what do you think caused those problems?
- Q3 What would you do when the aforementioned problems occur in real-world driving?
- Q4 What is the likelihood (%) someone would hack into your vehicle to disable a critical function (e.g., brake)?

Table II: The time interval between when an attack occurs and the victim's vehicle crashes.

Attack #	Driver #	Surviving Time (Second)	Driver #	Surviving Time (Second)
Attack 1	3	3.2	28	2.1
	13	29.86	32	19.35
	16	4.35	35	5.6
	17	4.78	36	5.15
	19	6.85	38	10.34
	20	12.32	44	1.5
	25	5.75	46	7.75
	26	5.5	47	4.0
	27	13.85	48	3.05
Attack 4	12	4.05		

Table III: Summary of the 50 driver's post-survey answers

Questions	Answers		# Low-DD drivers	# High-DD drivers
Q1. Problems	Steering Inversion	Attack 1	24	22
		Attack 4	7	2
	Acceleration	Attack 2	11	8
		Attack 5	0	0
	Braking	Attack 3	9	10
		Attack 6	0	0
Q2. Reasons	Cyber Attack		1	1
	Software Bug		8	3
	Mechanical		10	10
	Human		5	13
	Road Condition		3	2
Q3. Reaction	Better Control Vehicle (Pullover)		22	22
Q4. Vehicle Cyber Attack	Likelihood (%)		28.28%	26.76 %
	Likelihood >= 60%		7	4

Table III summarizes drivers' answers to the post-test survey. In regards to Q1, we observe that attacks against the steering system are easier for a driver to detect than attacks against brakes and accelerators, perhaps because the attack to steering system changed vehicle physical behavior more significantly. In regards to Q2, most drivers are neither aware of vehicle cyber attacks, nor aware that cyber attacks can cause vehicle malfunctions. In regards to Q3, 44 (out of the 50) drivers try to maintain control of their vehicles and/or immediately stop their vehicles, highlighting that drivers desire to avoid accidents. Drivers' answers to Q4 further confirm that they are not aware of vehicle cyber attacks. In terms of the impact of a drivers' impulsiveness on HD effectiveness, we observe that low-DD drivers may be able to detect more cyber attacks than high-DD drivers. However, there is no significant difference between low-DD and high-DD drivers in terms of Q2-Q4 except that more low-DD drivers anticipate vehicle cyber attacks. Most of the 19 drivers whose vehicle crashes believe that the accidents are caused by mechanical issues.

Insight 2: HD is effective in detecting substantial vehicle behavior changes, highlighting the importance of automatic detectors to detect attack-caused small changes that cannot be observed by drivers.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3069951, IEEE Internet of Things Journal

Algorithm 1 DD_Rate($\vec{x}, \vec{r}$)

Input: $\vec{x} = [x_1, \ldots, x_{27}]$; $\vec{r} = [r_1, \ldots, r_{27}]$
Output: DD Rate of a participant (k value)

$k \leftarrow -1$ ▷ default DD rate indicating the responses of a participant are inconsistent or useless
if $\vec{r} = \vec{1}$ **then** ▷ the participant always chooses delayed, but larger, reward A_i
  $k \leftarrow x_1$ ▷ lowest DD rate $x_1 = 0.00016$; least impulsive
else if $\vec{r} = \vec{0}$ **then** ▷ the participant always chooses an immediate, but smaller, reward A_i
  $k \leftarrow x_{27}$ ▷ highest DD rate $x_{27} = 0.25$; most impulsive
else ▷ if the responses are consistent, participant DD rate belongs to $[0.00016, 0.25]$
  $S \leftarrow 0$; $P \leftarrow 1$; ConsistencyScore[1..27] $\leftarrow \vec{0}$; Index[1..27] $\leftarrow \vec{0}$; $n \leftarrow 0$ ▷ initializing intermediate variables
  ConsistencyScore[1] $\leftarrow$ (the number of 1's in $[r_1, \ldots, r_{27}]$)/27
  for $i = 2$ to 27 **do**
    ConsistencyScore[i] $\leftarrow$(the number of 0's in $[r_1, \ldots, r_{i-1}]$ + the number of 1's in $[r_i, \ldots, r_{27}]$)/27
  $S \leftarrow \max_i$ ConsistencyScore[i] ▷ the highest consistency score
  for $i = 1$ to 27 **do**
    if ConsistencyScore[i] $= S$ **then**
      Index[i] $\leftarrow 1$
      $n \leftarrow n + 1$ ▷ the number of appearances of the highest consistency score
  if $S \geqslant 0.75$ **then** ▷ the threshold indicating one's responses are consistent or not
    if $n > 1$ **then** ▷ the highest consistency score has multiple appearances
      for $i = 1$ to 27 **do**
        if ConsistencyScore[i] $= 1$ **then**
          $P \leftarrow P \times x_i$
      $k \leftarrow \sqrt[n]{P}$ ▷ geometric mean of the x_i's corresponding to the i's that lead to highest consistency score S
    else ▷ $S \geqslant 0.75$ but $n = 1$, meaning S has exactly one appearance
      for $i = 1$ to 27 **do**
        if ConsistencyScore[i] $= 1$ **then**
          $P \leftarrow x_i \times x_{i-1}$
      $k \leftarrow \sqrt{P}$ ▷ Geometric mean of x_i and x_{i-1}
return DD rate k ▷ -1 means inconsistent or useless

G. PBD and Its Effectiveness

Confirming physical behavior changes. Figure 6 plots trajectories of driver #22 in Test 2 (no attack) vs. Test 3 (under attack; no crash), where the x-axis represents the vehicle's longitudinal position (in terms of the 20 driving tasks $T_1, \ldots, T_{20}$) and the 6 attacks (Attacks 1-6) as well as their duration, the y-axis represents the vehicle lateral position in 5 lanes ($y = 0$ is the middle lane). A vehicle drives from the left to the right and is supposed to stay in the middle lane when no lane-change tasks are given to the driver. We observe that the two trajectories overlap during tasks T_1 and T_2, but exhibit large differences when Attacks 1 and 4 are waged and continue after these attacks are terminated; we also observe significant differences when the other 4 attacks are waged.

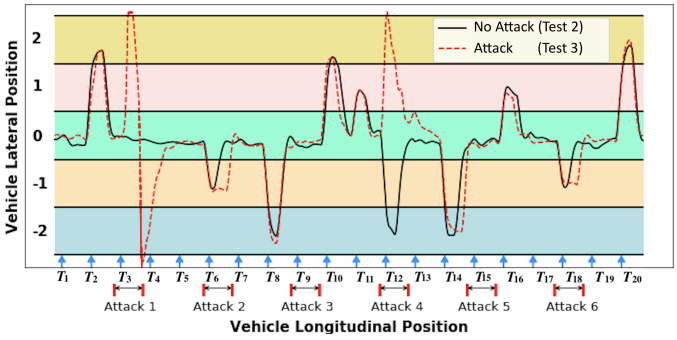


Figure 6: Trajectories of driver #22 in Test 2 (no attack) vs. Test 3 (under attack) in the simulated driving test.

In order to see the value of designing automated vehicle physical behavior-based detector (rather than human as detector), we observe that the cyber attack to disable the brake and accelerator waged at 50 units before T_6, T_9 , T_{15}, and T_{18}, which do not cause dramatic physical behavior changes, meaning that they might not be detected by the driver (i.e., failure of human as detector). It is even more interesting to see that when the same attack waged 50 units before T_3 is waged again 50 units before T_{12}, we observe that when the driver tries to finish T_{12} (i.e., changing from lane 0 to lane 2 on the left), the vehicle actually drifts to the rightmost lane owing to the attack. The abnormal behavior can be immediately recognized by the driver. Unlike the attack waged 50 units before T_3 where the driver loses the control of the vehicle and the vehicle's physical behavior changes continue until the end of T_5, the attack against the vehicle at T_{12} does not prevent the vehicle from getting back to lane 0 (i.e., getting back to no attack) at T_{13}. This highlights the importance of detecting the attack before T_{12} so that the driver can be alerted to pull over the vehicle to minimize the chance of fatal consequence of the

attack (i.e., drifting to the rightmost lane in this test).

Supervised PBD. To train a supervised PBD for all drivers, we label each measurement of (F_1, F_2, F_3) in Test 3 during an attack as "attack" and each measurement in any other case as "no attack" (i.e., every measurement in Test 2 and every measurement in Test 3 in the absence of attack). There are 627,813 total measurements for the 50 drivers: 69.63% are labeled "no attack" and 30.36% are labeled "attack". We split the dataset into two parts: 66% for training a Random Forest-based PBD and 34% for testing. Experimental results show that the detector achieves a 96.37% detection accuracy, 92.46% TPR, 1.93% FPR, 7.54% FNR, 95.4% precision, 95.26% AUC, 0.05 seconds detection delay (time spent on applying the PBD detector), 19 crashes avoided, and an average 7.71 seconds surviving time (time interval between when the first feature vector measurement corresponding to an attack is predicted as attack and when the vehicle crashes). We split the dataset in two; one corresponding to the 25 high-DD drivers and the other corresponding to the 25 low-DD drivers. For each dataset, we train a Random Forest-PBD as mentioned above. PDB learned from high-DD drivers vs. the low-DD drivers achieves the following: 97.07% vs. 96.82% in accuracy, 94.28% vs. 92.61% in TPR, 1.53% vs. 1.59% in FPR, 5.72% vs. 7.39% in FNR, 96.87% vs. 95.64% in precision, and 96.37% vs. 95.51% in AUC. PBD's effectiveness is not impacted by impulsiveness significantly.

Unsupervised PBD. An unsupervised PBD is trained by using LSTM Autoencoder with on average 5,300 vectors of a sequence of feature vectors (F_1, F_2, F_3). The predicted output is compared with the input to calculate the MAE value, where a large MAE value means a large deviation from the normal behavior and therefore is interpreted as an attack. Different drivers have MAE thresholds as their vehicle physical behaviors are different. The training data is captured when there are no attacks, and the maximum MAE value corresponding to the training data is selected as the MAE threshold to distinguish between normal behaviors and abnormal ones.

Table IV summarizes driver's MAE thresholds for anomaly detection and unsupervised PBD's effectiveness in detecting the 6 attacks. We make two observations for the 82 attacks against the steering system (i.e., 50 Attack 1 and 32 Attack 4 against the vehicles that are not crashed by Attack 1). First, 56 attacks involving 46 vehicles are detected. Among these detected attacks, 46 are Attack 1 and 10 are Attack 4. Among the 46 detected Attack 1, 10 are detected before the attack occurs, 5 are detected when it occurs, and 31 are detected after it occurs; among the 10 detected Attack 4, 2 are detected before the attack occurs and 8 are detected after it occurs. However, driver's impulsiveness is not a significant factor influencing unsupervised PBD's effectiveness. Among the 46 detected vehicles, 24 are high-DD and 22 are low-DD drivers. The # of crashes avoided by PBD is 19 with an average surviving time of 7.76 seconds.

For the 63 brake attacks against uncrashed vehicles, PBD detects 6 attacks against 6 vehicles, where 3 are Attack 2 and 3 are Attack 5. Among the 3 detected Attack 2, 1 is detected before the attack occurs, and 2 are detected after it occurs; among the 3 detected Attack 5, 2 are detected before they occur and 1 is detected after it occurs. Among the 6 vehicles that detect attacks, 4 are operated by high-DD drivers and 2 are operated by low-DD drivers, hinting that a driver's impulsiveness has no significant impact on PBD's effectiveness. Although no crashes in the tests, these attacks can cause crashes in the real world, justifying the value of PBD.

For the 63 attacks against the accelerator, PBD detects 14 attacks against 14 vehicles. Among these 14 attacks, 5 are Attack 3 and 9 are Attack 6; the 5 Attack 3 are detected after attacks occur, the 2 Attack 6 are detected before they occur, and the other 7 are detected after they occur. Among the 14 vehicles that detect attacks, 8 are high-DD drivers and 6 are low-DD drivers, also showing that driver's impulsiveness has no significant impact on PBD's effectiveness.

Insight 3: PBD is effective in detecting vehicle cyber attacks before crashes occur, and the driver's impulsiveness does not have a significant impact on PBD's effectiveness.

H. DBD and Its Effectiveness

Confirming driving behavior changes. Figure 7 plots the driving behaviors of driver #22 in Test 2 (no attacks) vs. Test 3 (under attacks but no crash), where driving behaviors are measured by features F_5 (brake position), F_6 (gas pedal position), and F_7 (reaction time). The x-axis represents the vehicle's longitudinal position (in terms of the 20 driving tasks $T_1, \ldots, T_{20}$) and the 6 attacks (Attacks 1-6); the y axis of Figure 7a represents percentage of F_5 and the y-axis of Figure 7b represents F_6. The value of F_5 and F_6 are between 0% and 100%, where 0% means brake/accelerator is inactive and 100% means the driver presses the brake/gas pedal with highest force to stop/accelerate vehicle speed as soon as possible. Table V summarizes the measurements of driver #22's feature F_7 in Test 2 vs. Test 3, where we use bold fonts to highlight the sharp contrast in the driver's reaction time in Test 2 vs. Test 3. This hints at the importance of feature F_7.
Supervised DBD. In order to train a supervised DBD for all drivers, we label the driving behavior measurements of features $(F_4, \ldots, F_8)$ during an attack as "attack" and each measurement in any other case as "no attack" (as in the case of PBD). There are 627,813 measurements in total for the 50 drivers. We split this dataset into two parts: 66% for training a Random Forest-based DBD and 34% for testing. Experimental results show that DBD achieves: 97.97% accuracy, 96.28% TPR, 1.3% FPR, 3.72% FNR, 96.99% precision, 97.49% AUC, 0.05 seconds in detection delay, 19 crashes avoided, and on average the surviving time is 8.84 seconds. In order to see if this detector is sensitive to a driver's impulsiveness, we split the datasets into two (i.e., one for the 25 high-DD drivers and the other for the 25 low-DD drivers) and train Random Forest-based DBD. Experimental results show the effectiveness of high-DD vs. low-DD drivers as follows: 98.17% vs. 97.96% in accuracy, 96.7% vs. 95.72% in TPR, 1.09% vs. 1.2% in FPR, 3.3% vs. 4.28% in FNR, 97.8% vs. 96.79% in precision, 97.8% vs. 97.26% in AUC, 0.05 seconds vs.0.05 seconds in

Table IV: The 50 drivers' DD rates and effectiveness of unsupervised PDB vs. DBD vs. PDBD, where 'NA' means a vehicle crashes because of Attack 1 (orange row) or Attack 4 (blue row), 'x' means the detector fails to detect the attack as an anomaly, a numerical value $\pm s$ represents the *detection delay* (unit: second) in detecting an anomaly as attack (a positive value means the delay, and a negative value means the forecast, of an attack with respect to an attack's start point).

Driver #		DD Rate	Unsupervised PBD, DBD, and PDBD Detector Attack Detection Time (Seconds) Negative Value: Dectecting Attack Before It Occurs; Positive Value: Detecting Attack After It Occurs; 0: Detecting Attack When It Occurs; 'x': Attacks Are Not																			
			PBD (Physical Behavior-based Detector)							DBD (Driving Behavior-based Detector)							PDBD (Physical and Driving Behavior-based Detector)						
			MAE Thre.	Attack 1. Steering Inversion	Attack 2. Breaking	Attack 3. Acceleration	Attack 4. Steering Inversion	Attack 5. Breaking	Attack 6. Acceleration	MAE Thre.	Attack 1. Steering Inversion	Attack 2. Breaking	Attack 3. Acceleration	Attack 4. Steering Inversion	Attack 5. Breaking	Attack 6. Acceleration	MAE Thre.	Attack 1. Steering Inversion	Attack 2. Breaking	Attack 3. Acceleration	Attack 4. Steering Inversion	Attack 5. Breaking	Attack 6. Acceleration
Low DD	1	0.000158128	0.34	0.25	x	x	x	x	x	0.22	0.18	0.02	0.83	0	0	x	0.28	0.2	0	0.85	-0.1	-0.05	0.8
	2	0.000398989	0.32	-0.15	x	x	x	x	x	0.21	-0.7	x	0.85	0.02	0	x	0.31	-0.75	x	0.83	0	0	0.7
	3	0.001001837	0.37	0.25	NA					0.24	0.15	NA					0.18	-0.15	NA				
	4	0.001002193	0.27	-0.05	x	x	x	x	x	0.46	0	0.9	0.1	0	-0.35	x	0.25	-0.2	x	1.1	0	-0.35	x
	5	0.001583814	0.34	0	x	x	x	x	x	0.11	0	x	0.85	0	-0.01	x	0.26	0	x	0.8	0	-0.1	0.85
	6	0.001586399	0.19	0.2	x	x	x	x	x	0.47	0	x	-0.35	0	-0.35	x	0.21	0	1	-0.35	0	-0.35	x
	7	0.001586399	0.17	0	x	x	x	x	x	0.23	0	x	x	0	x	x	0.27	0	x	x	0.15	0	x
	8	0.001586399	0.41	x	x	x	x	x	0.9	0.21	0.05	x	0.9	0	0	x	0.27	0.05	x	0.8	0	-0.1	x
	9	0.002511154	0.38	x	x	x	x	x	0.85	0.11	0.05	x	0.8	0.01	0	x	0.27	0.1	x	x	0.7	-0.1	x
	10	0.002511154	0.18	-0.2	x	0.85	x	x	x	0.19	-0.2	x	0.78	0	0	x	0.32	-0.2	x	1	0	0	x
	11	0.002511154	0.33	0.2	x	x	x	x	x	0.43	0.1	x	0.83	-0.01	0	x	0.27	0.1	x	0.8	-0.1	0	x
	12	0.002511154	0.18	0.3	x	x	0.2	NA		0.23	0.5	x	-0.3	0	NA		0.24	0.3	x	-0.6	0	NA	
	13	0.002511154	0.5	0.15	NA					0.18	0.2	NA					0.29	0.04	NA				
	14	0.002535234	0.33	0.15	x	x	x	x	x	0.22	0.1	x	0.83	x	0	0.9	0.29	0.35	x	0.83	x	-0.15	1.2
	15	0.002535234	0.19	0.05	x	x	x	x	x	0.2	0.1	0	0.85	-0.02	0	x	0.23	0.1	x	0.7	-0.05	0	x
	16	0.002535234	0.23	-0.4	NA					0.42	-0.4	NA					0.23	-0.35	NA				
	17	0.003896509	0.33	0.2	NA					0.24	0.15	NA					0.29	0.12	NA				
	18	0.003896509	0.08	-0.15	x	x	1	-0.15	-0.35	0.27	-0.25	x	x	0.05	-0.1	-0.3	0.29	-0.3	x	x	1.2	-0.25	-0.2
	19	0.004837067	0.17	0.15	NA					0.13	0.15	NA					0.19	0.05	NA				
	20	0.006003171	0.41	0.3	NA					0.24	0.25	NA					0.36	0.33	NA				
	21	0.006003171	0.31	0.2	x	x	1.6	x	-0.35	0.23	0.1	x	x	x	-0.35	0	0.25	0.2	x	x	x	-1	-0.5
	22	0.009706451	0.17	0.4	X	x	0.15	x	x	0.22	0.25	x	0.85	0	0	x	0.23	0.4	x	x	0	x	x
	23	0.009706451	0.39	0.2	x	x	-0.1	-0.35	0.3	0.26	0.15	x	x	-0.1	-0.05	0.5	0.25	0.1	x	x	-0.15	-0.5	-0.35
	24	0.009706451	0.39	x	x	x	x	x	x	0.35	0.01	0.85	0.01	0.01	0.05	0.85	0.15	0.1	x	x	0.83	1.2	0.83
	25	0.009706451	0.32	0.18	NA					0.44	0.2	NA					0.37	0.05	NA				
High DD	26	0.009723443	0.18	0.2	NA					0.24	0.1	NA					0.24	0.1	NA				
	27	0.009723443	0.18	0.3	NA					0.4	0.05	NA					0.25	0.05	NA				
	28	0.009723443	0.32	-0.2	NA					0.19	-0.4	NA					0.26	-0.2	NA				
	29	0.015745325	0.39	x	x	x	x	x	x	0.25	0	x	x	0	0.01	x	0.28	-0.1	x	0.82	0	0	x
	30	0.015745325	0.5	0	x	x	x	x	x	0.42	0	x	x	0.05	0	x	0.45	0.05	x	x	x	0	x
	31	0.015745325	0.37	0.2	x	x	x	x	0.9	0.2	0.1	0.05	x	0	0	x	0.27	0.15	0	0.9	-0.05	0	x
	32	0.01586679	0.19	0.35	NA					0.1	0.3	NA					0.21	0.35	NA				
	33	0.01586679	0.21	-0.1	x	x	0.2	x	0.9	0.24	-0.1	x	x	0.05	x	1	0.25	-0.2	-0.5	x	0.1	x	1.2
	34	0.025492269	0.16	0.1	x	0.75	0.3	x	x	0.24	0.04	x	x	x	0.05	x	0.29	0.1	x	x	x	0.01	x
	35	0.025492269	0.34	0.15	NA					0.47	0.1	NA					0.31	0.1	NA				
	36	0.025762303	0.19	0.05	NA					0.26	0.1	NA					0.37	0.2	NA				
	37	0.040956979	0.49	0.15	x	x	x	x	x	0.47	0.2	0.9	0.15	0	0	x	0.25	0.2	x	x	0	0	0.9
	38	0.040956979	0.19	0.1	NA					0.19	0.1	NA					0.23	0.01	NA				
	39	0.040956979	0.21	0.2	x	x	x	x	x	0.11	0	x	0.85	0.1	0	x	0.24	0	x	0.83	0.85	0	0.9
	40	0.040956979	0.17	0.05	x	0.85	0.38	x	x	0.12	0.05	-0.35	0.6	0	0	x	0.23	0	-0.35	0.83	0.25	-0.05	0.85
	41	0.040956979	0.49	-0.3	x	x	x	x	0.75	0.13	-0.25	x	0.9	0	0	x	0.25	-0.3	x	1.2	0	0	x
	42	0.041011431	0.33	0.1	x	x	x	x	x	0.19	0.05	-0.35	0.85	x	0	x	0.22	0.02	-0.5	0.78	0.15	0	1.1
	43	0.041011431	0.07	-0.05	x	x	x	x	x	0.22	0	0.85	0.25	x	0.05	1	0.19	-0.05	1	0.2	x	x	x
	44	0.064446792	0.48	0.3	NA					0.27	0.05	NA					0.29	0	NA				
	45	0.064446792	0.24	0	x	x	0.1	0.15	0.7	0.24	0	x	0.01	0	-0.6	-0.4	0.33	0	x	0.82	0	-0.35	-0.35
	46	0.064446792	0.26	0.3	NA					0.09	0	NA					0.23	0	NA				
	47	0.065212406	0.49	-0.2	NA					0.12	-0.2	NA					0.22	-0.15	NA				
	48	0.12667139	0.07	0.18	NA					0.1	0.1	NA					0.22	0.15	NA				
	49	0.15843801	0.18	0	x	x	x	x	x	0.25	0	x	x	0.05	x	x	0.31	0	x	x	0.01	x	x
	50	0.248371318	0.15	0.1	-0.05	0.2	-0.35	x	x	0.1	0.1	0	-0.35	-0.35	-0.3	x	0.23	0.1	-0.2	-0.75	-0.35	-0.35	1.1

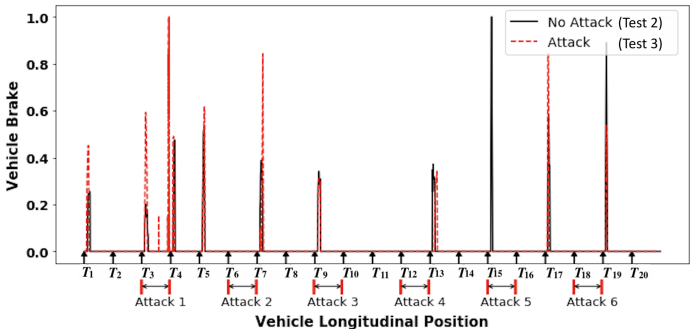
detection delay, 11 vs. 8 in # crashes avoided, and 7.05 seconds vs. 8.84 seconds in surviving time on average. This means that the driver's impulsiveness does not have a significant impact on the DBD's effectiveness.

Table V: Driver #22's driving behavior measured by F_7, where a red T_i means that the value of F_7 is affected by an attack.

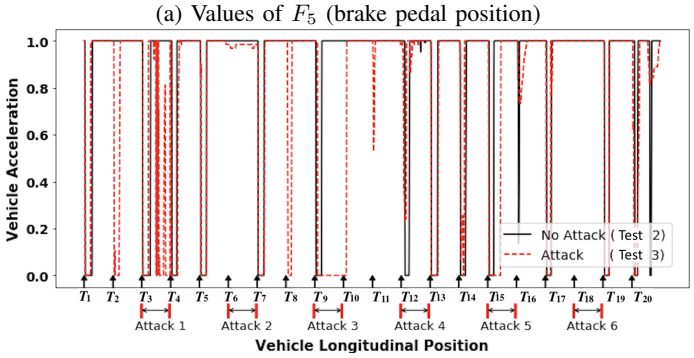
Driving Task #	F_7 Test 2 (s)	F_7 Test 3 (s)	Driving Task #	F_7 Test 2 (s)	F_7 Test 3 (s)
T_1	2.352	1.736	T_{11}	2.91	2.801
T_2	10	10	T_{12}	**3.673**	**10**
T_3	2.466	1.779	T_{13}	1.723	2.725
T_4	1.743	1.986	T_{14}	3.588	4.193
T_5	**1.873**	**10**	T_{15}	**1.659**	**10**
T_6	2.553	2.395	T_{16}	3.103	3.094
T_7	1.925	2.419	T_{17}	1.553	1.261
T_8	3.712	3.451	T_{18}	2.169	2.346
T_9	2.004	2.104	T_{19}	1.451	1.603
T_{10}	10	10	T_{20}	4.752	4.266

Unsupervised DBD. Similar to unsupervised PBD, we use LSTM Autoencoder to train 50 DBD detectors from Test 2 to predict attack-caused driving behavior changes in Test 3. Each detector is trained on average using 5,300 vectors of a sequence feature measurements $(F_4, \ldots, F_8)$. Table IV summarizes driver's MAE thresholds and the DBD's capability in detecting the 6 attacks. For the 82 attacks against the vehicle's steering system (i.e., Attack 1 against 50 drivers and Attack 4 against the 32 drivers that are not crashed by Attack 1) the detector detects 77 attacks involving 50 vehicles, and 50 are Attack 1 whereas 27 are Attack 4. Among the 50 detected Attack 1, 8 are detected before attacks occur, 11 are detected when they occur, and 31 are detected after they occur; among the 27 detected Attack 4, 4 are detected before they occur, 15 are detected when they occur, and 8 are detected after they occur. Among the 50 detected Attack 1, 25 are high-DD and 25 are low-DD drivers. Among the 27 detected Attack 4, 11 are high-DD and 16 are low-DD drivers. This hints that impulsiveness has a marginal impact on the DBD's effectiveness.In sum, DBD offers an average 7.84 seconds in surviving time to avoid 19 crashes caused by Attack 1 and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3069951, IEEE Internet of Things Journal



(a) Values of F_5 (brake pedal position)



(b) Values of F_6 (gas pedal position)

Figure 7: Driver #22's driving behaviors in features F_5 and F_6: no attack vs. attack.

Attack 4.

For the 63 attacks against the brake, DBD detects 40 attacks against 30 vehicles that do not crash. Among these 40 detected attacks, 12 are Attack 2 and 28 are Attack 5. Among the 12 detected Attack 2, 2 are detected before it occurs 2 are detected when it occurs, and 8 are detected after it occurs; among the 28 detected Attack 5, 8 are detected before it occurs, 16 are detected when it occurs, and 4 are detected after it occurs. Among the 40 vehicles that detect attacks, 14 are high-DD and 16 are low-DD drivers, hinting that impulsiveness has no significant impact on DBD.

For the 63 attacks against the accelerator, DBD detects 31 attacks against 27 vehicles. Among these 31 detected attacks, 23 are Attack 3 and 8 are Attack 6. Among the 23 detected Attack 3, 3 are detected before it occurs and 20 are detected after it occurs; among the 8 detected Attack 6, 2 are detected before it occurs, 1 is detected when it occurs, and 5 are detected after it occurs. Among the 27 vehicles that detect attacks, 10 are high-DD and 17 are low-DD drivers, hinting that low-DD drivers lead to more effective DBD.

Insight 4: DBD is effective in detecting attacks. The effectiveness of supervised DBD is not affected by a driver's impulsiveness, but the unsupervised DBD is more effective for low-DD drivers.

I. PDBD and Its Effectiveness

Having showed that DBD (using features $F_4, \ldots, F_8$) is more effective than PBD (using F_1, F_2, F_3), now we investigate the effectiveness of PDBD (using $F_1, \ldots, F_8$).

Supervised PDBD. To Train a supervised PDBD we label each measurement of $(F_1, \ldots, F_8)$ in Test 3 during an attack as "attack" and each measurement in any other case as "no attack". There are 627,813 measurements in total for the 50 drivers, where 69.63% are labeled as "no attack" and 30.36% are labeled as "attack". We split this dataset into two parts: 66% for training a Random Forest-based PDBD and 34% for testing. Table VI summarizes the effectiveness of PDBD vs. PBD vs. DBD. We observe that PDBD achieves a significantly higher effectiveness in detecting the 6 attacks (e.g., FPR is 0.02% vs. 1.93% vs. 1.2%). Supervised PDBD also detects the 19 attacks that cause crashes and offer an averaged surviving time 7.84 seconds to the drivers (over the 19 drivers) to pull over and avoid crashes. The time spent on training supervised PDBD is still less than 50 seconds and the detection time is 0.05 seconds (averaged over the 50 drivers). We split the dataset into two (i.e., one for the 25 high-DD drivers and the other for the 25 low-DD drivers) to train two Random Forest-based PDBD in order to see if PDBD is sensitive to a driver's impulsiveness. Table VI summarizes the experimental results. We do not observe substantial differences (e.g., FPR is 0.01% vs. 0.02%), except that 11 high-DD drivers vs. 8 low-DD drivers crash. This means the driver's impulsiveness does not have a significant impact on supervised PDBD's effectiveness.

Unsupervised PDBD. Similar to unsupervised PBD and DBD, we use LSTM Autoencoder and the data collected in Test 2 to train 50 unsupervised PDBD detectors, one for each driver. Each detector is trained on average by using 5,300 vectors of $(F_1, \ldots, F_8)$. Figure 8 plots the MAE loss distribution of driver #2, where the x-axis represents MAE value and the y-axis represents the number of training examples with a certain MAE. In this case, 0.31 is selected as the threshold, meaning that a measurement of $(F_1, \ldots, F_8)$ leading to a higher MAE value is detected as an anomaly and interpreted as an attack.

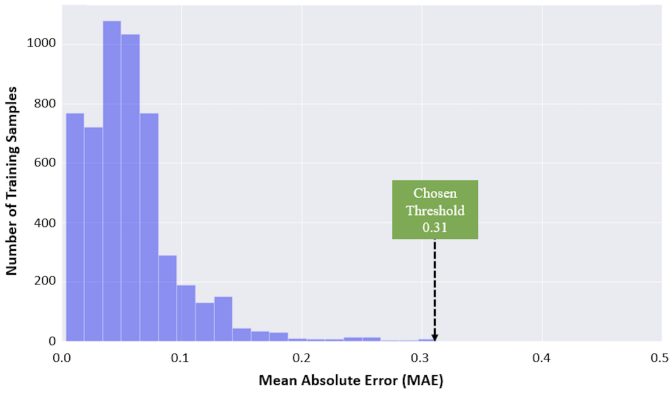


Figure 8: MAE distribution and threshold value obtained when training unsupervised PDBD for Low-DD driver #2.

Table IV summarizes driver's MAE thresholds and the effectiveness of unsupervised PDBD. For the 82 attacks against the vehicle's steering system (i.e., Attack 1 against 50 drivers and Attack 4 against the 32 drivers who did not crash during Attack 1), we make two observations. First, PDBD detects 77 attacks against 50 vehicles. Among these 77 detected attacks, 50 are Attack 1 and 27 are Attack 4. Among the 50 detected

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3069951, IEEE Internet of Things Journal

Table VI: The effectiveness of PBD, DBD, and PDBD.

Evaluation Metrics		PBD			DBD			PDBD		
		High DD	Low DD	All	High DD	Low DD	All	High DD	Low DD	All
Accuracy (%)		97.07	96.82	96.37	98.17	97.96	97.97	99.95	99.92	99.93
TPR (%)		94.28	92.61	92.46	96.7	95.72	96.28	99.88	99.78	99.84
FPR (%)		1.53	1.59	1.93	1.09	1.2	1.3	0.01	0.02	0.02
FNR (%)		5.72	7.39	7.54	3.3	4.28	3.72	0.12	0.22	0.16
Precision (%)		96.87	95.64	95.4	97.8	96.79	96.99	99.97	99.94	99.95
AUC (%)		96.37	95.51	95.26	97.8	97.26	97.49	99.93	99.88	99.91
# Crashes Avoided		11	8	19	11	8	19	11	8	19
Average Detection Delay (s)	Unsupervised Detector	0.09	0.19	0.14	0.01	0.03	0.02	0.04	0.08	0.06
	Supervised Detector	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
Average Surviving Time (s)	Unsupervised Detector	7.02	8.78	7.76	7.14	8.82	7.84	7.1	8.89	7.86
	Supervised Detector	6.97	8.73	7.71	7.09	8.77	7.79	7.05	8.84	7.81

Attack 1, 12 are detected before the attack occurs (owing to the forecasting capability of LSTM Autoencoder), 9 are detected when the attack occurs, and 29 are detected after the attack occurs; among the 27 detected Attack 4, 6 are detected before the attack occur, 12 are detected when the attack occurs, and 9 are detected after the attack occurs. Among 27 vehicles that detect Attack 4, 11 high-DD drivers with an average of 0.04 seconds in detection delay and 16 low-DD drivers with an average of 0.08 seconds in detection delay. This hints that PDBD is more effective for low-DD drivers. Second, Attack 1 causes 18 crashes and Attack 4 causes 1 crash. Figure 9 plots the *surviving time* for each of the corresponding 19 drivers, where the start time for each attack is marked as 0 (second). The '•' (▲) indicates when Attack 1 (4) is detected, a '×' (▼) indicates when a vehicle crashes, the length of the dashed (solid) lines represents the surviving time. We observe that the surviving time is 1.5 seconds for driver #44, 2.1 seconds for driver #28, 3.2 seconds for driver #3, 3.05 seconds for driver #48, at least 4 seconds for the other 15 drivers.

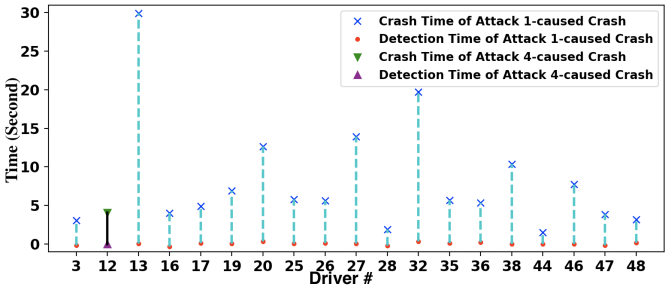


Figure 9: The surviving time offered by PDBD to the 19 drivers to pull over and avoid crashes.

For the 63 attacks against the brake system (i.e., Attack 2 against the 32 vehicles that did not crash by Attack 1 and Attack 5 against the 31 vehicles that did not crash by Attack 4), we observe that PDBD detects 37 attacks against 31 vehicles. Among these 37 detected attacks, 10 are Attack 2 and 27 are Attack 5. Among the 10 detected Attack 2, 4 are detected before the attack occurs (owing to the forecasting capability of LSTM Autoencoder), 2 are detected when the attack occurs, and 4 are detected after the attack occurs; among the 27 detected Attack 5, 13 are detected before the attack occurs, 12 are detected when the attack occurs, and 2 are detected after the attack occurs. Among the 31 vehicles detecting attacks, 15 drivers have a high-DD rate and 16 drivers have a low-DD rate, hinting that a driver's impulsiveness does not have a significant influence on the effectiveness of PDBD.

For the 63 attacks against the accelerator (i.e., Attack 3 against the 32 vehicles that did not crash by Attack 1 and Attack 6 against the 31 vehicles that did not crash by Attack 4), we observe that PDBD detects 36 attacks against 27 vehicles. Among these 36 detected attacks, 21 are Attack 3 and 15 are Attack 6. Among the 21 detected Attack 3, 3 are detected before the attack occurs and 18 are detected after the attack occurs; among the 15 detected Attack 6, 4 are detected before the attack occurs and 11 are detected after the attack occurs. Among the 27 vehicles detecting attacks, 12 drivers have a high-DD rate and 15 drivers have a low-DD rate, hinting that a driver's impulsiveness somewhat affects PDBD's effectiveness.

Insight 5: PDBD is more effective than PDB and DBD in detecting attacks and its effectiveness is somewhat affected by the driver's impulsiveness.

V. Limitations and Future Directions

A. Limitations

The present study has several limitations, which represents significant gaps between the experimental environments and real-world situations. The first gap is caused by the notion of *driving tasks*, which plays an important role in our driving tests. Although these driving tasks are meant to mimic real-world situations (e.g., response to traffic lights and/or lane changes), their counterparts in the real world are not well-defined or cannot be measured yet. The second gap is caused by the simplified conditions in our driving tests, including the use of straight roadways in the driving test and no other vehicles on the road, and the way of simulating the effect of cyber attacks on a vehicle's physical and mechanical behaviors. The third gap is determining which aspect(s) of impulsiveness or other human factors (e.g., attention) meaningfully contribute to vehicle cyber attack detection. The fourth gap is the ExHPD's scalability because of the range of different driving

behaviors which may also be impacted by different vehicle make/model. In what follows we outline research directions towards bridging these gaps.

B. Towards bridging the first gap

We observe that driving task-based features may not be easy to measure. In what follows we consider feature F_7 (reaction time) as an example for illustrating what research needs to be done in the future. The measurement of F_7 would need the support of smart transportation infrastructure (e.g., smart traffic lights/stop signs) and connected vehicles. This motivates us to investigate whether or not feature F_7 is necessary for creating highly accurate vehicle cyber attack detectors. In order to answer this question, we contrast the effectiveness of supervised and unsupervised PDBDs trained from dataset corresponding to features $(F_1, \ldots, F_8)$ vs. $(F_1, \ldots, F_6, F_8)$.

We train a Random Forest-based supervised PDBD using the dataset corresponding to $(F_1, \ldots, F_6, F_8)$, in the same fashion as using the dataset corresponding to $(F_1, \ldots, F_8)$. Experimental results show that the resulting PDBD achieves 99.34% accuracy (in contrast to 99.93% when using F_7), 98.39% TPR (in contrast to 99.84%), 0.24% FPR (in contrast to 0.02%), 1.61% FNR (in contrast to 0.16%), 99.44% precision (in contrast to 99.95%), 99.07% AUC (in contrast to 99.91%), 0.05 seconds in detection delay (same as before), 19 crashes avoided (same as before), and 7.81 seconds of average surviving time (same as before). This means that supervised PDBD without using F_7 is still reasonably effective, but not as effective as when using F_7.

We train LSTM Autoencoder-based PDBD detectors for the 50 drivers as in the case of using features $(F_1, \ldots, F_8)$. Experimental results show that the resulting PDBDs detect 100 cyber attacks in total (in contrast to 150 when using F_7). The # of detected attacks drops from 72.12% (when using F_7) to 48.08% (when not using F_7); the detection delay remains at 0.02 seconds for detecting Attack 1, increases from 0.19 to 0.5 seconds for detecting Attack 2, reduces from 0.61 to 0.47 seconds for detecting Attack 3, increases from 0.13 to 0.22 seconds for detecting Attack 4, increases from -0.09 to 0.1 seconds for detecting Attack 5, reduces from 0.6 to 0.28 seconds for detecting Attack 6; the detector still detects the 19 crashes. This means that the effectiveness of unsupervised PDBD without using F_7 is significantly reduced.

The preceding discussion suggests two future research directions. One is to investigate the feasibility of having the future intelligent transportation infrastructure to provide a driving task-like service to measure the relevant features (e.g., F_7). The other is to identify alternate features that can replace those driving task-based features before they can be measured by intelligent transportation infrastructure.

C. Towards bridging the second gap

The driving simulation environment should be enhanced to accommodate more realistic roadway situations, such as real-world traffic. Systematic cyber attacks should be conducted against vulnerable vehicles. This suggests two future research directions. One is to investigate how to leverage Augmented Reality/Virtual Reality for the driving tests. The other is to investigate realistic attack effects on vehicle physical properties.

D. Towards bridging the third gap

In order to identify the human factors about different aspects of impulsiveness and in addition to impulsiveness that would have a significant impact on the effectiveness of HD/PBD/DBD/PDBD, which leverage drivers and their driving behaviors to detect vehicle cyber attacks, we plan to measure different impulsiveness aspects via different tasks (e.g., BART [49] and Go/No Go [11]) concurrently in each driver. These measurements can be compared using the similar methodology employed in the current study. Additionally, many of these impulsiveness measures use hypothetical money as the main commodity for which drivers make their decisions. This may be a less powerful predictor for differences in driving behavior, especially during cyber attacks, compared to decisions about driving behavior (e.g., the trade-off between engaging in a risky driving behavior and getting in an accident [40]). Lastly, other human factors outside of impulsiveness, such as overall attention via eye tracking technologies, need to be measured and incorporated in to these predictions to determine the most important human factors.

E. Towards bridging the fourth gap

A much larger population of drivers should be recruited to conduct the aforementioned enhanced driving tests to identify representative driving behaviors and driver impulsiveness in their DD rates. It is important to investigate whether or not a detector learned from some vehicles' historic data can be applied to other vehicles that may or may not be manufactured by the same maker. It is interesting to investigate whether reinforcement learning (other than supervised and unsupervised learning) can help alleviate the scalability issue.

VI. Conclusion

In this study, we considered a vehicle as a cyber-physical system and treated vehicles as CPH systems. Afterward, we proposed the novel ExHPD framework for exploiting human, physical, and driving behaviors to detect vehicle cyber attacks. As proof of concept, we designed a hardware-in-the-loop driving simulation testbed and recruited college-age drivers to conduct a case study. Experimental results have validated the efficiency of ExHPD. We also discussed the limitations of the present study and future research directions.

APPENDIX A
THE 27-ITEM MONETARY CHOICE QUESTIONNAIRE AND TWO DRIVER'S RESPONSES

Table VII lists the Monetary Choice Questionnaire (MCQ) survey we give to the 50 drivers as well as two driver #22 and #35's responses (as examples). The 1st column "Random Order Q #" lists the randomized question order given to each driver. The 2nd column "Original Order Q #" presents the sorted 27 questions in ascending order. The 27 questions are first sorted based on their respective k values (one k value per question). Three questions having similar k values are grouped into one "dimension", meaning that the 27 questions are split into 9 "dimensions" (α) as presented in the 7th column. The questions within a dimension are sorted based on their respective reward magnitude (β) as shown in the 8th column (in ascending order). The 3rd, 4th and 5th columns respectively present the values of the small but immediately monetary rewards (SIR) V, the larger but delayed monetary rewards (LDR) A, and the delays (unit: day) D. The last two columns present drivers #22 and #35's response (r) to the 27 questions, where 0 means the driver selects LDR and 1 means the driver selects SIR. These two driver's DD rates (k values) are derived according to Algorithm 1 and are presented at the bottom of the table.

Table VII: 27-item Questionnaire for Measuring DD Rates and Two Examples

Random Order Q #	Original Order Q # (Q)	SIR (V)	LDR (A)	Delay Day (D)	k Value (x)	Dimension (α)	Reward Magnitude (β)	Response (r)	
								Driver #22	Driver #35
13	1	\$34	\$35	186	0.000158128	1	Small	0	0
1	2	\$54	\$55	117	0.000158278	1	Medium	0	0
9	3	\$78	\$80	162	0.000158278	1	Large	0	0
20	4	\$28	\$30	179	0.000399042	2	Small	0	0
6	5	\$47	\$50	160	0.000398936	2	Medium	0	0
17	6	\$80	\$85	157	0.000398089	2	Large	0	0
26	7	\$22	\$25	136	0.001002674	3	Small	0	0
24	8	\$54	\$60	111	0.001001001	3	Medium	0	0
12	9	\$67	\$75	119	0.001003386	3	Large	0	0
22	10	\$25	\$30	80	0.0025	4	Small	0	0
16	11	\$49	\$60	89	0.002522357	4	Medium	0	0
15	12	\$69	\$85	91	0.002548176	4	Large	0	0
3	13	\$19	\$25	53	0.005958292	5	Small	1	0
10	14	\$40	\$55	62	0.006048387	5	Medium	0	0
2	15	\$55	\$75	61	0.005961252	5	Large	0	0
18	16	\$24	\$35	29	0.015804598	6	Small	1	0
21	17	\$34	\$50	30	0.015686275	6	Medium	1	0
25	18	\$54	\$80	30	0.016049383	6	Large	1	1
5	19	\$14	\$25	19	0.041353383	7	Small	1	0
14	20	\$27	\$50	21	0.040564374	7	Medium	1	1
23	21	\$41	\$75	20	0.041463415	7	Large	1	1
7	22	\$15	\$35	13	0.102564103	8	Small	1	1
8	23	\$25	\$60	14	0.1	8	Medium	1	1
19	24	\$33	\$80	14	0.101731602	8	Large	1	1
11	25	\$11	\$30	7	0.246753247	9	Small	1	1
27	26	\$20	\$55	7	0.25	9	Medium	1	1
4	27	\$31	\$85	7	0.248847926	9	Large	1	1
							DD Rate (k)	0.009706451	0.025515352

ACKNOWLEDGMENT

This research is sponsored by the National Science Foundation under Grant No.1812599, in part by National Science Foundation Grants No. 2122631 (No. 1814825) and No. 1736209. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] "Upstream security global automotive cyber security report," 2019, https://www.upstream.auto/upstream-security-global-automotive-cybersecurity-report-2019/.

[2] Bowman and B. M. V. Group, "Vehicle safety in a connected world-nhtsa proposed rulemaking on v2v technology and oem liability," 2016, https://www.bowmanandbrooke.com/insights/nhtsa-proposed-rulemaking-v2v-communication-tech.

[3] Association for Safe International Road Travel, "Road safety facts," https://www.asirt.org/safe-travel/road-safety-facts/.

[4] NHTSA, "2017 fatal motor vehicle crashes: Overview," *U.S. Department of Transportation*, 2017, https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812603.

[5] National Highway Traffic Safety Administration, "The economic and societal impact of motor vehicle crashes, 2010 (revised)1," *Annals of Emergency Medicine*, vol. 66, 08 2015.

[6] L. Franceschi-Bicchierai, "Hacker finds he can remotely kill car engines after breaking into gps tracking apps," 04 2019, https://www.vice.com/en_us/article/zmpx4x/hacker-monitor-cars-kill-engine-gps-tracking-apps.

[7] S. Gillman, "Hackers could blackmail owners of self-driving cars," 03 2017, https://horizon-magazine.eu/article/hackers-could-blackmail-owners-self-driving-cars-dr-alexander-kr-ller-tomtom.html.

[8] B. M. Jenkins and B. R. Butterworth, "An analysis of vehicle ramming as a terrorist tactic," 2018, https://transweb.sjsu.edu/sites/default/files/SP0518%20Vehicle%20Ramming%20Terrorism.pdf.

[9] UPSTREAM, "Q1 2019 sees a rapid growth of automotive cyber incidents," 2019, https://www.upstream.auto/blog/q1-2019-sees-a-rapid-growth-of-automotive-cyber-incidents/.

[10] K. A. Abay and F. L. Mannering, "An empirical analysis of risk-taking in car driving and other aspects of life," *Accid Anal Prev*, vol. 97, pp. 57–68, Dec 2016.

[11] Y. Ba, W. Zhang, G. Salvendy, A. S. Cheng, and P. Ventsislavova, "Assessments of risky driving: a Go/No-Go simulator driving task to evaluate risky decision-making and associated behavioral patterns," *Appl Ergon*, vol. 52, pp. 265–274, Jan 2016.

[12] "Opends 3.5 - open source driving simulation," 2020, https://opends.dfki.de,.

[13] K.-T. Cho and K. G. Shin, "Fingerprinting electronic control units for vehicle intrusion detection," in *25th USENIX Security Symposium (USENIX Security 16)*. Austin, TX: USENIX Association, Aug. 2016, pp. 911–927. [Online]. Available: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/cho

[14] P. Murvay and B. Groza, "Source identification using signal characteristics in controller area networks," *IEEE Signal Processing Letters*, vol. 21, no. 4, pp. 395–399, April 2014.

[15] 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*, ser. CCS '18. New York, NY, USA: ACM, 2018, pp. 787–800. [Online]. Available: http://doi.acm.org/10.1145/3243734.3243751

[16] 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, June 2018.

[17] N. Nowdehi, W. Aoudi, M. Almgren, and T. Olovsson, "Casad: Can-aware stealthy-attack detection for in-vehicle networks," 09 2019.

[18] C. A. Kerrache, N. Lagraa, A. Benslimane, C. T. Calafate, and J.-C. Cano, "On the human factor consideration for vanets security based on social networks," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[19] C. A. Kerrache, N. Lagraa, R. Hussain, S. H. Ahmed, A. Benslimane, C. T. Calafate, J.-C. Cano, and A. M. Vegni, "Tacashi: Trust-aware communication architecture for social internet of vehicles," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 5870–5877, 2018.

[20] T. Mekki, I. Jabri, A. Rachedi, and M. B. Jemaa, "Vehicular cloud networking: evolutionary game with reinforcement learning-based access approach," *International Journal of Bio-Inspired Computation*, vol. 13, no. 1, pp. 45–58, 2019.

[21] ——, "Proactive and hybrid wireless network access strategy for vehicle cloud networks: An evolutionary game approach," in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, 2017, pp. 1108–1113.

[22] Y. Yahiatene, A. Rachedi, M. A. Riahla, D. E. Menacer, and F. Nait-Abdesselam, "A blockchain-based framework to secure vehicular social networks," *Transactions on Emerging Telecommunications Technologies*, vol. 30, no. 8, p. e3650, 2019.

[23] T. Abar, A. Rachedi, A. ben Letaifa, P. Fabian, and S. El Asmi, "Fellowme cache: Fog computing approach to enhance (qoe) in internet of vehicles," *Future Generation Computer Systems*, vol. 113, pp. 170–182, 2020.

[24] I. Jabri, T. Mekki, A. Rachedi, and M. B. Jemaa, "Vehicular fog gateways selection on the internet of vehicles: A fuzzy logic with ant colony optimization based approach," *Ad Hoc Networks*, vol. 91, p. 101879, 2019.

[25] P. Fabian, A. Rachedi, and C. Guéguen, "Programmable objective function for data transportation in the internet of vehicles," *Transactions on emerging telecommunications technologies*, vol. 31, no. 5, p. e3882, 2020.

[26] R. S. A. L. M. H. M. R. R. S. J. L. David Hallac, Abhijit Sharang, "Driver identification using automobile sensor data from a single turn," November 2016, pp. 953–958.

[27] K. Ozawa, T. Wakita, C. Miyajima, K. Itou, and K. Takeda, "Modeling of individualities in driving through spectral analysis of behavioral signals," in *Proceedings of the Eighth International Symposium on Signal Processing and Its Applications, 2005.*, August 2005.

[28] C. Miyajima, Y. Nishiwaki, K. Ozawa, T. Wakita, K. Itou, and K. Takeda, "Ceptral analysis of driving behavioral signals for driver identification," vol. 5, May 2006, pp. V–V.

[29] C. Miyajima, Y. Nishiwaki, K. Ozawa, T. Wakita, K. Itou, K. Takeda, and F. Itakura, "Driver modeling based on driving behavior and its evaluation in driver identification," *Proceedings of the IEEE*, vol. 95, no. 2, pp. 427–437, February 2007.

[30] M. V. Ly, S. Martin, and M. M. Trivedi, "Driver classification and driving style recognition using inertial sensors," *2013 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1040–1045, June 2013.

[31] S. Jafarnejad, G. Castignani, and T. Engel, "Towards a real-time driver identification mechanism based on driving sensing data," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, Oct 2017, pp. 1–7.

[32] J. E. Meseguer, C. K. Toh, C. T. Calafate, J. C. Cano, and P. Manzoni, "Drivingstyles: a mobile platform for driving styles and fuel consumption characterization," *Journal of Communications and networks*, vol. 19, no. 2, pp. 162–168, 2017.

[33] M. W. Johnson and W. K. Bickel, "Within-subject comparison of real and hypothetical money rewards in delay discounting," *J Exp Anal Behav*, vol. 77, no. 2, pp. 129–146, Mar 2002.

[34] G. J. Madden, A. M. Begotka, B. R. Raiff, and L. L. Kastern, "Delay discounting of real and hypothetical rewards," *Exp Clin Psychopharmacol*, vol. 11, no. 2, pp. 139–145, May 2003.

[35] A. L. Odum, "Delay discounting: I'm a k, you're a k," *J Exp Anal Behav*, vol. 96, no. 3, pp. 427–439, Nov 2011.

[36] G. J. Madden, N. M. Petry, G. J. Badger, and W. K. Bickel, "Impulsive and self-control choices in opioid-dependent patients and non-drug-using control participants: drug and monetary rewards," *Exp Clin Psychopharmacol*, vol. 5, no. 3, pp. 256–262, Aug 1997.

[37] M. Grabski, H. V. Curran, D. J. Nutt, S. M. Husbands, T. P. Freeman, M. Fluharty, and M. R. MunafÃ², "Behavioural tasks sensitive to acute abstinence and predictive of smoking cessation success: a systematic review and meta-analysis," *Addiction*, vol. 111, no. 12, pp. 2134–2144, 2016.

[38] J. L. Evenden, "Varieties of impulsivity," *Psychopharmacology (Berl.)*, vol. 146, no. 4, pp. 348–361, Oct 1999.

[39] M. Green, "Driver reaction time," 2013, https://www.visualexpert.com/Resources/reactiontime.html.

[40] Y. Hayashi, H. J. Fessler, J. E. Friedel, A. M. Foreman, and O. Wirth, "The roles of delay and probability discounting in texting while driving: Toward the development of a translational scientific program," *J Exp Anal Behav*, vol. 110, no. 2, pp. 229–242, 09 2018.

[41] A. Milenkoski, M. Vieira, S. Kounev, A. Avritzer, and B. D. Payne, "Evaluating computer intrusion detection systems: A survey of common practices," *ACM Comput. Surv.*, vol. 48, no. 1, pp. 12:1–12:41, Sep. 2015.

[42] M. Pendleton, R. Garcia-Lebron, J. Cho, and S. Xu, "A survey on systems security metrics," *ACM Comput. Surv.*, vol. 49, no. 4, pp. 62:1–62:35, 2017.

[43] "Logitech G920 & G 29 Driving Force Steering Wheel and Pedal," 2020, https://www.logitechg.com/en-ch/products/driving/driving-force-racing-wheel.html.

[44] N. Srivastava, E. Mansimov, and R. Salakhutdinov, "Unsupervised learning of video representations using lstms," *CoRR*, vol. abs/1502.04681, 2015. [Online]. Available: http://arxiv.org/abs/1502.04681

[45] J. C. Turner, E. V. Leno, and A. Keller, "Causes of Mortality Among American College Students: A Pilot Study," *J College Stud Psychother*, vol. 27, no. 1, pp. 31–42, Jan 2013.

[46] L. Gicquel, P. Ordonneau, E. Blot, C. Toillon, P. Ingrand, and L. Romo, "Description of Various Factors Contributing to Traffic Accidents in Youth and Measures Proposed to Alleviate Recurrence," *Front Psychiatry*, vol. 8, p. 94, 2017.

[47] J. D. Jentsch, J. R. Ashenhurst, M. C. Cervantes, S. M. Groman, A. S. James, and Z. T. Pennington, "Dissecting impulsivity and its relationships to drug addictions," *Ann. N. Y. Acad. Sci.*, vol. 1327, pp. 1–26, Oct 2014.

[48] K. N. Kirby, N. M. Petry, and W. K. Bickel, "Heroin addicts have higher discount rates for delayed rewards than non-drug-using controls," *J Exp Psychol Gen*, vol. 128, no. 1, pp. 78–87, Mar 1999.

[49] I. S. for Research on Impulsivity, "Balloon analogue risk task (bart)," 2020, http://www.impulsivity.org/measurement/BART.