



University Transportation Research Center - Region 2

Final Report



Secure and Private Sensing for Driver Authentication and Transportation Safety

Performing Organization: New York Institute of Technology



August 2017



Sponsor:
University Transportation Research Center - Region 2

University Transportation Research Center - Region 2

The Region 2 University Transportation Research Center (UTRC) is one of ten original University Transportation Centers established in 1987 by the U.S. Congress. These Centers were established with the recognition that transportation plays a key role in the nation's economy and the quality of life of its citizens. University faculty members provide a critical link in resolving our national and regional transportation problems while training the professionals who address our transportation systems and their customers on a daily basis.

The UTRC was established in order to support research, education and the transfer of technology in the field of transportation. The theme of the Center is "Planning and Managing Regional Transportation Systems in a Changing World." Presently, under the direction of Dr. Camille Kamga, the UTRC represents USDOT Region II, including New York, New Jersey, Puerto Rico and the U.S. Virgin Islands. Functioning as a consortium of twelve major Universities throughout the region, UTRC is located at the CUNY Institute for Transportation Systems at The City College of New York, the lead institution of the consortium. The Center, through its consortium, an Agency-Industry Council and its Director and Staff, supports research, education, and technology transfer under its theme. UTRC's three main goals are:

Research

The research program objectives are (1) to develop a theme based transportation research program that is responsive to the needs of regional transportation organizations and stakeholders, and (2) to conduct that program in cooperation with the partners. The program includes both studies that are identified with research partners of projects targeted to the theme, and targeted, short-term projects. The program develops competitive proposals, which are evaluated to insure the most responsive UTRC team conducts the work. The research program is responsive to the UTRC theme: "Planning and Managing Regional Transportation Systems in a Changing World." The complex transportation system of transit and infrastructure, and the rapidly changing environment impacts the nation's largest city and metropolitan area. The New York/New Jersey Metropolitan has over 19 million people, 600,000 businesses and 9 million workers. The Region's intermodal and multimodal systems must serve all customers and stakeholders within the region and globally. Under the current grant, the new research projects and the ongoing research projects concentrate the program efforts on the categories of Transportation Systems Performance and Information Infrastructure to provide needed services to the New Jersey Department of Transportation, New York City Department of Transportation, New York Metropolitan Transportation Council, New York State Department of Transportation, and the New York State Energy and Research Development Authority and others, all while enhancing the center's theme.

Education and Workforce Development

The modern professional must combine the technical skills of engineering and planning with knowledge of economics, environmental science, management, finance, and law as well as negotiation skills, psychology and sociology. And, she/he must be computer literate, wired to the web, and knowledgeable about advances in information technology. UTRC's education and training efforts provide a multidisciplinary program of course work and experiential learning to train students and provide advanced training or retraining of practitioners to plan and manage regional transportation systems. UTRC must meet the need to educate the undergraduate and graduate student with a foundation of transportation fundamentals that allows for solving complex problems in a world much more dynamic than even a decade ago. Simultaneously, the demand for continuing education is growing – either because of professional license requirements or because the workplace demands it – and provides the opportunity to combine State of Practice education with tailored ways of delivering content.

Technology Transfer

UTRC's Technology Transfer Program goes beyond what might be considered "traditional" technology transfer activities. Its main objectives are (1) to increase the awareness and level of information concerning transportation issues facing Region 2; (2) to improve the knowledge base and approach to problem solving of the region's transportation workforce, from those operating the systems to those at the most senior level of managing the system; and by doing so, to improve the overall professional capability of the transportation workforce; (3) to stimulate discussion and debate concerning the integration of new technologies into our culture, our work and our transportation systems; (4) to provide the more traditional but extremely important job of disseminating research and project reports, studies, analysis and use of tools to the education, research and practicing community both nationally and internationally; and (5) to provide unbiased information and testimony to decision-makers concerning regional transportation issues consistent with the UTRC theme.

Project No(s):

UTRC/RF Grant No: 49198-33-27

Project Date: August 2017

Project Title: Secure and Private Sensing for Driver Authentication and Transportation Safety

Project's Website:

<http://www.utrc2.org/research/projects/secure-and-private-sensing-driver-authentication>

Principal Investigator(s):

Jonathan Voris, Ph.D.

Assistant Professor

Department of Computer Science

New York Institute of Technology

New York, NY 10023

Tel: (212) 261-1734

Email: jvoris@nyit.edu

Co Author(s):

Sertac Artan, Ph.D.

Assistant Professor

Department of Electrical and Computer Engineering

New York Institute of Technology

New York, NY 10023

Tel: (212) 261-1732

Email: nartan@nyit.edu

Wenjia Li, Ph.D.

Assistant Professor

Department of Computer Science

New York Institute of Technology

New York, NY 10023

Tel: (646) 273-6016

Email: wli20@nyit.edu

Performing Organization(s):

New York Institute of Technology

Sponsor(s):

University Transportation Research Center (UTRC)

To request a hard copy of our final reports, please send us an email at utrc@utrc2.org

Mailing Address:

University Transportation Research Center

The City College of New York

Marshak Hall, Suite 910

160 Convent Avenue

New York, NY 10031

Tel: 212-650-8051

Fax: 212-650-8374

Web: www.utrc2.org

Board of Directors

The UTRC Board of Directors consists of one or two members from each Consortium school (each school receives two votes regardless of the number of representatives on the board). The Center Director is an ex-officio member of the Board and The Center management team serves as staff to the Board.

City University of New York

Dr. Robert E. Paaswell - Director Emeritus of UTRC
Dr. Hongmian Gong - Geography/Hunter College

Clarkson University

Dr. Kerop D. Janoyan - Civil Engineering

Columbia University

Dr. Raimondo Betti - Civil Engineering
Dr. Elliott Sclar - Urban and Regional Planning

Cornell University

Dr. Huaizhu (Oliver) Gao - Civil Engineering

Hofstra University

Dr. Jean-Paul Rodrigue - Global Studies and Geography

Manhattan College

Dr. Anirban De - Civil & Environmental Engineering
Dr. Matthew Volovski - Civil & Environmental Engineering

New Jersey Institute of Technology

Dr. Steven I-Jy Chien - Civil Engineering
Dr. Joyoung Lee - Civil & Environmental Engineering

New York Institute of Technology

Dr. Marta Panero - Director, Strategic Partnerships
Nada Marie Anid - Professor & Dean of the School of Engineering & Computing Sciences

New York University

Dr. Mitchell L. Moss - Urban Policy and Planning
Dr. Rae Zimmerman - Planning and Public Administration
Dr. Kaan Ozbay - Civil Engineering
Dr. John C. Falcocchio - Civil Engineering
Dr. Elena Prassas - Civil Engineering

Rensselaer Polytechnic Institute

Dr. José Holguín-Veras - Civil Engineering
Dr. William "Al" Wallace - Systems Engineering

Rochester Institute of Technology

Dr. James Winebrake - Science, Technology and Society/Public Policy
Dr. J. Scott Hawker - Software Engineering

Rowan University

Dr. Yusuf Mehta - Civil Engineering
Dr. Beena Sukumaran - Civil Engineering

State University of New York

Michael M. Fancher - Nanoscience
Dr. Catherine T. Lawson - City & Regional Planning
Dr. Adel W. Sadek - Transportation Systems Engineering
Dr. Shmuel Yahalom - Economics

Stevens Institute of Technology

Dr. Sophia Hassiotis - Civil Engineering
Dr. Thomas H. Wakeman III - Civil Engineering

Syracuse University

Dr. Riyadh S. Aboutaha - Civil Engineering
Dr. O. Sam Salem - Construction Engineering and Management

The College of New Jersey

Dr. Thomas M. Brennan Jr - Civil Engineering

University of Puerto Rico - Mayagüez

Dr. Ismael Pagán-Trinidad - Civil Engineering
Dr. Didier M. Valdés-Díaz - Civil Engineering

UTRC Consortium Universities

The following universities/colleges are members of the UTRC consortium.

City University of New York (CUNY)
Clarkson University (Clarkson)
Columbia University (Columbia)
Cornell University (Cornell)
Hofstra University (Hofstra)
Manhattan College (MC)
New Jersey Institute of Technology (NJIT)
New York Institute of Technology (NYIT)
New York University (NYU)
Rensselaer Polytechnic Institute (RPI)
Rochester Institute of Technology (RIT)
Rowan University (Rowan)
State University of New York (SUNY)
Stevens Institute of Technology (Stevens)
Syracuse University (SU)
The College of New Jersey (TCNJ)
University of Puerto Rico - Mayagüez (UPRM)

UTRC Key Staff

Dr. Camille Kamga: *Director, UTRC*
Assistant Professor of Civil Engineering, CCNY

Dr. Robert E. Paaswell: *Director Emeritus of UTRC and Distinguished Professor of Civil Engineering, The City College of New York*

Dr. Ellen Thorson: *Senior Research Fellow*

Penny Eickemeyer: *Associate Director for Research, UTRC*

Dr. Alison Conway: *Associate Director for Education*

Nadia Aslam: *Assistant Director for Technology Transfer*

Dr. Wei Hao: *Post-doc/ Researcher*

Dr. Sandeep Mudigonda: *Postdoctoral Research Associate*

Nathalie Martinez: *Research Associate/Budget Analyst*

Tierra Fisher: *Office Assistant*

Andriy Blagay: *Graphic Intern*

Disclaimer

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. The contents do not necessarily reflect the official views or policies of the UTRC, New York Institute of Technology, or the Federal Highway Administration. This report does not constitute a standard, specification, or regulation. This document is disseminated under the sponsorship of the Department of Transportation, University Transportation Centers Program, in the interest of information exchange. The U.S. Government and New York Institute of Technology assume no liability for the contents or use thereof.

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle Secure and Private Sensing for Driver Authentication and Transportation Safety		5. Report Date August 25 th . 2017	6. Performing Organization Code
7. Author(s) Jonathan Voris N. Sertac Artan Wenjia Li		8. Performing Organization Report No.	
9. Performing Organization Name and Address New York Institute of Technology 1855 Broadway New York, NY 10023		10. Work Unit No.	11. Contract or Grant No. 49198-33-27
12. Sponsoring Agency Name and Address University Transportation Research Center The City College of New York 137 th Street and Convent Ave, New York, NY 10031		13. Type of Report and Period Covered Final Report June 1 st 2015 - June 30 th , 2017	
15. Supplementary Notes		14. Sponsoring Agency Code	
<p>16. Abstract</p> <p>Recent technology trends have allowed affordable and efficient collection of driver data. This has enabled a variety of potential applications, including more accurate pricing determinations for insurance and finer grained traffic planning for improved public safety. Although this technological growth provides for a wealth of new opportunities, given the safety implications of driving, there are many security and privacy issues that must be considered for their deployment. For instance, some applications require access to a vehicle's engine via a debug interface, known as On-Board Diagnostics (OBD-II), which may provide a vector for attack. Other systems may involve GPS tracking, which can potentially violate a driver's privacy. Our research seeks to find solutions to these shortcomings by using local sensing and monitoring to support the development of new driver devices and applications, such as driver authentication, while preserving vehicular security and privacy.</p> <p>We propose a novel approach to data collection for commercial driving applications and vehicle safety that puts users in control of how their information is used. By collecting local driving data in a manner that is decoupled from critical car components and Internet connections, our system can support transportation applications, such as driver authentication, without sacrificing vehicle security or driver privacy. The legitimate driver of a vehicle traditionally gains authorization to access their vehicle via tokens such as ignition keys, some modern versions of which feature RFID tags. However, this token-based approach is not capable of detecting all instances of vehicle misuse. Technology trends have allowed for affordable and efficient collection of various sensor data in real time from the vehicle, its surroundings, and devices carried by the driver, such as smartphones.</p> <p>This report describes the result of our research effort investigating the use of this sensory data to actively identify and authenticate the driver of a vehicle by determining characteristics which uniquely categorize individuals' driving behavior. Our approach is capable of continuously authenticating a driver throughout a driving session, as opposed to alternative approaches which are either performed offline or as a session starts. This means our modeling approach can be used to detect mid-session driving attacks, such as carjacking, which are beyond the scope of alternative driver authentication solutions. A simulated driving environment was used to collect sensory data of driver habits including steering wheel position and pedal pressure. These features are classified using a Support Vector Machine (SVM) learning algorithm. Our results show that our approach is capable of using various aspects of how a vehicle is operated to successfully identify a driver under 2.5 minutes with a 95% confidence interval and with at most one false positive per driving day.</p>			
17. Key Words Vehicular security, intelligent transportation systems, smart cities, machine learning, driver authentication		18. Distribution Statement	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No of Pages 43	22. Price

Executive Summary

The legitimate driver of a vehicle traditionally gains authorization to access their vehicle via tokens such as ignition keys, some modern versions of which feature RFID tags. However, this token-based approach is not capable of detecting all instances of vehicle misuse. Technology trends have allowed for affordable and efficient collection of various sensor data in real time from the vehicle, its surroundings, and devices carried by the driver, such as smartphones. The goal of this project was to study the use of sensory data to actively identify and authenticate the driver of a vehicle by determining characteristics which uniquely categorize individuals' driving behavior. Our approach is capable of continuously authenticating a driver throughout a driving session, as opposed to alternative approaches which are either performed offline or as a session starts. This means our modeling approach can be used to detect mid-session driving attacks, such as carjacking, which are beyond the scope of alternative driver authentication solutions. A simulated driving environment was used to collect sensory data of driver habits including steering wheel position and pedal pressure. These features are classified using a Support Vector Machine (SVM) learning algorithm. Our pilot study with 10 human subjects shows that we can use various aspects of how a vehicle is operated to successfully identify a driver in less than 2.5 minutes with a 95% confidence interval and with at most one false positive per driving day.

Background

Data on driving habits is being used in a wide variety of applications and impacts stakeholders with a broad range of interests; such as transportation departments in the governments, private entities, individual drivers, and the public in general.

Transportation departments can utilize this information to anticipate future driver needs and problematic road safety areas and for optimizing infrastructure investments which render the most long-term benefits. *Private entities*, such as insurance companies and car-share programs, can use this information to better understand driver behavior and set product prices accordingly; *drivers* can use the data to improve their driving habits, to reach their destinations faster and tap into new safety features such as collision avoidance systems. The general public will use this information to get safer roads, and lower taxes ensuring from better planning.

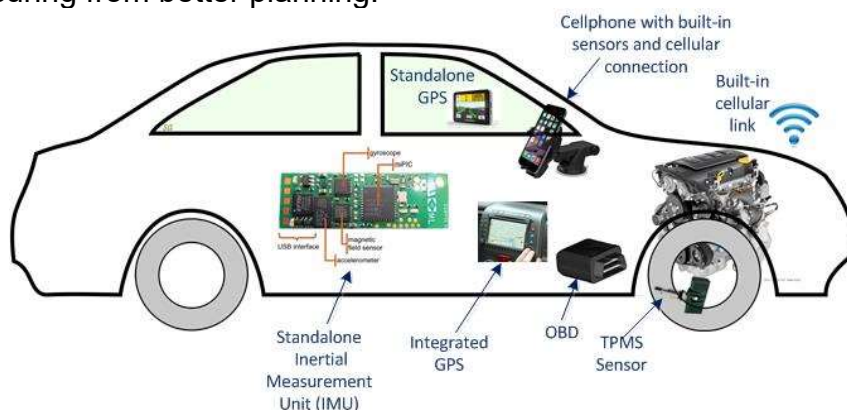


Figure 1: Sensors and Communication Interfaces Available on a Modern Vehicle

Traditionally, measuring driver habits involved conducting costly traffic surveys which took time and human effort and yielded results with limited accuracy. Recently, however, several technological trends have converged to allow affordable and efficient collection of driver data. The cost and availability of wireless communication and sensing hardware has allowed for easy collection of data, often in real time via ubiquitous devices installed in vehicles. This has enabled a variety of potential applications, including more accurate pricing determinations for insurance (Pay as you drive (PAYD) or Usage-Based Insurance (UBI)) and finer grained traffic planning for improved public safety.

Although this technological growth provides for a wealth of new opportunities, given the safety implications of driving, there are many security and privacy issues that must be considered for their deployment. Devices that utilize the same networks as vehicle control components may increase the attack surface of the car's system, making it more vulnerable to cyber-attacks. Other devices involve the collection of sensitive information which users would prefer to keep private, such as GPS coordinates or photographs of the drivers. The goal of our research is to design local sensor systems which protect driver privacy and vehicle security while supporting existing data collection applications. The data we will collect could potentially benefit broader vehicular networks if shared in a privacy-preserving fashion. Furthermore, in some cases our proposed technology may be able to provide additional details about the state of a vehicle, such as whether a driver's behavior resembles their past habits or violates safety regulations. This potentially enables several new services, such as driver authentication for fraud detection and enhanced public safety monitoring.

Objectives

This project can be broken down into three core goals. Firstly, we desired to identify distinctive underlying characteristics of individual's driving habits which could potentially be used to construct a model of driving behavior, which could in turn be used to classify drivers, allowing vehicles to confirm the identity of their current driver by comparing current usage to patterns of past behavior. Secondly, we sought to identify driving simulation software appropriate to the task of collecting these features as part of a human subject study of driving behavior. Once selected, this simulation software would be used to design scenarios for users to navigate which would realistically simulate real world driving situations. Next, this project aimed to use this simulation to collect data from a pool of volunteers. Finally, the data collected from this study would be analyzed in order to draw conclusions regarding the viability of behavior based driver authentication.

Introduction

Ignition keys have served as authentication tokens for vehicle drivers for decades. More recently, traditional keys based on physical shape have been augmented with embedded Radio Frequency Identification (RFID) tokens to provide an additional layer of protection against theft. Unfortunately, such keys are susceptible to theft, cloning, forgery, and relay attacks. However, RFID enabled steering columns represent only a small portion of the sensing hardware available on modern vehicles.

Traditionally, measuring driver habits involved conducting costly traffic surveys which take a large amount of time and human effort yet yielded results with limited accuracy [1]–[3]. Recently, however, several technological trends have converged to allow affordable and efficient collection of driver data [4], [5]. The cost and availability of wireless communication and sensing hardware has allowed for easy collection of data, often in real time via ubiquitous devices installed in vehicles or worn by drivers. This has enabled a variety of potential applications, including more accurate pricing determinations for insurance (Pay as you drive (PAYD) or Usage-Based Insurance (UBI)) [6], [7] and finer grained traffic planning for improved public safety [8].

This project seeks to solve the problem of unsafe and untrustworthy transportation systems caused by vehicle misuse by authenticating drivers according to the manner in which a vehicle is operated. To this end, we conducted a study with 10 human subjects to assess the efficacy of using the data collected from the vehicle sensors on identifying the driver.

We propose to authenticate drivers based on a variety of data that is available via common onboard vehicular sensors and systems. There are a variety of stakeholders involved in the operation of transportation systems for which a more thorough guarantee of a driver's identity would be of interest. For example, municipal governments may wish to ensure that buses are being operated by a predetermined employee. Similarly, car sharing service providers may want to confirm that a member has picked up the correct vehicle, and owners of taxi fleets may wish to ensure their vehicles have not been operated without permission. Insurance providers may wish to verify that only drivers listed on a particular policy are allowed access to a covered car. Finally, recognition that a vehicle is being operated by someone other than the vehicle's typical owner may allow for advanced notice in the event of vehicle theft.

Summary of the Literature Review

In a recent issue of IEEE Transactions on Intelligent Transportation Systems, Woo, Jo and, Lee performed an experiment which demonstrated the possibility of attacking a vehicle's CAN bus through an attached diagnostic device [9]. This work focuses on appliances which are connected with a vehicle's CAN bus via an OBD-II connection and transmit data to a cell phone application. The authors show that a malicious diagnostic application can be used to intercept and inject their own CAN bus data, effectively gaining remote control over a vehicle's ECUs. This is demonstrated by performing a variety of actions which affect the vehicle including accelerating and turning off the engine. The authors propose a protocol which addresses these vulnerabilities by encrypting and authenticating CAN bus data.

Foster et. al investigated the security of an OBD-II device for collecting vehicle usage information for insurance pricing [10]. This group discovered a variety of security issues in the device they analyzed, including credential reuse among devices, the ability to output the device's RAM state over a local USB connection, and the ability to update the device's configuration via SMS.

The authors concluded that this leaves any vehicle which the device was connected to vulnerable to both local and remote attacks [10]. They concurred with the assessment provided by Woo, Jo, and Lee that a successful attack on the OBD-II device would leave the entire compromised vehicle's systems open to malicious control. Foster et. al suggest several solutions to assuage these issues, including stronger authentication and key management.

In addition, He et al. recently studied a conditional privacy-preserving authentication (CPPA) scheme for vehicular ad hoc networks (VANETs), which is able to address both security and privacy-preserving challenges in VANETs, because the CPPA scheme can support both mutual authentication and privacy protection simultaneously [11]. In recent years, many identity-based CPPA schemes for VANETs have been proposed, in which bilinear pairing technology is used to enhance security or to improve performance. However, it is widely accepted that the bilinear pairing operation generally introduces a large amount of additional computational overhead, which makes it infeasible to be deployed to VANETs because of the constrained resources. Thus, the authors in [11] proposed a CPPA scheme for VANETs that does not use bilinear pairing technology and the authors also showed in this paper that the proposed scheme supports both the mutual authentication and the privacy protection at the same time. The proposed CPPA scheme can produce a better performance in terms of computational cost and communication cost when compared to existing CPPA schemes.

Alternatively, Bittl et al. studied a GPS time spoofing attack that aims to cause denial of service (DoS) in VANETs [12]. In recent years, Car2X (such as Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), etc.) communication has become a key enabling technology for vehicular networks. However, most of the existing Car2X communication protocols significantly rely on the global positioning system (GPS) for providing location information and time synchronization, which is vulnerable to both location and time spoofing attacks. In this work, the authors primarily focus on time spoofing attack for VANETs, and they show that this type of attack can lead to severe denial of service attacks. In addition, the non-repudiation feature of the security system can also be violated by offering the possibility to misuse authentication features. The authors also discussed a potential Sybil attack which can severely influence the reliability of the basic time and location data sets inside VANET messages. To cope with the time spoofing attack, the authors also depicted some mechanisms, which performance has been studied as well.

Summary of the Work Performed

This report summarizes the research efforts performed as part of our UTRC supported project entitled "Secure and Private Sensing for Driver Authentication and Transportation Safety" which has recently been brought to a successful conclusion. The main goal of this project was to assess a driver's behavior while driving with an aim of identifying driver identity and other characteristics, such as drowsiness, to support transportation safety and security.

Our approach to this problem consisted of minimizing the set of sensors required to make assessments of driving behavior for two core reasons. Firstly, we sought to make the driving data collection as privacy-conscious as possible by collecting data which would not be problematic from a privacy perspective if released. Secondly, a goal of our project was to decouple sensing from the vehicle itself to avoid any potential connection, and thus attack vector, to critical vehicular systems. We accomplished this goal by performing two distinct but related studies of user behavior with a simulated driving task. Our results indicate that driver identification is possible with a minimally invasive set of monitoring sensors.

An outline which briefly summarizes each of the research tasks we completed during the course of this project is included below:

- Obtaining IRB approval for the first phase of our human subject study with a simulated driving environment
- Investigating potential biometric modalities for driver identification
- Identifying potential simulator software to utilize in our human subject study
- Surveying sensor hardware for components which would potentially be used to monitor various driving modalities, such as pressure sensors
- Assessing potential driving simulation software
- Initial development of a preliminary driving scenario
- Performing a small scale human user study in which 12 participants performed our preliminary driving task
- Cleaning, analyzing, and modeling our preliminary dataset, the results of which were presented as a paper at the Workshop on Green ICT and Smart Networking (GISN 2016), co-located with the International Conference on Network and Service Management (CNSM 2016)
- Developing as detailed a driving task and environment as possible with the capabilities of our simulation software
- Augmenting our study simulation hardware and software to allow for the collection of additional features
- Completing a large scale human subject study of driving habits with more features, a post-conditional survey, and emotional preconditioning
- Performing a thorough review of academic literature on transportation security

The remainder of this section of the report will provide details on each of these accomplishments.

Institutional Review Board Approval:

One of our earliest project tasks was the pursuit of Institutional Review Board (IRB) approval for our human subject studies. Our protocol was approved on 9/22/15. Our IRB protocol materials and approval letter were submitted as supplemental materials.

Survey of Smart Transportation Security

As noted in the introduction, our research team also completed a thorough survey of the security of current transportation technology as well as the state-of-the-art in proposed

future transit systems. Our research found a troubling lack of security in today's typical vehicles. The development of in-vehicle networks in many ways mirrors that of the early Internet, which was initially conceived as a network of friendly collaborators. Without having to worry about the presence of malicious actors or vulnerable assets, early network designers were free to focus on performance and efficiency. Since everything was intended for sharing, there was no need for attribution or access control.

As the Internet scaled and commercialized, a patchwork of network security solutions eventually emerged, yet the lack of foresight lingers in the vulnerabilities of today's computer networks. Similarly, in-vehicle networks were designed under the assumption of a completely closed system comprised of trustworthy components with no external connections. Modern cars are controlled by dozens of Electronic Control Units (ECUs) which communicate via the Controller Area Network (CAN) standard. Before the widespread availability of wireless networks and mobile devices for personal use, gaining access to a vehicle's network was considered impractical due to their physical security. Cars are either moving at high speed or are physically secured against theft.

As a result, intra-vehicular networks are designed with even fewer security considerations than the early Internet. Much like the Internet, there is no built-in access control mechanism in the CAN standard. The Internet Protocol requires source and destination addresses at a minimum, however, which provides a crude form of attribution. CAN systems, on the other hand, are strictly broadcast, making attribution all but impossible once access to the network has been gained. Vehicles have long been viewed as valuable assets for their material worth, but until recently little attention was paid to enhancing the security of vehicles at an information level. Most early forms of protection focused on preventing physical theft or tampering, including electronic theft protection systems and mileage counter protection measures. This attitude has gradually shifted as more and more computerized components have been added to cars.

The first microcontroller was added to a vehicle in 1977 when General Motors utilized one for spark plug timing in its Oldsmobile Toronado models. Since then, the expansion of processors into other aspects of vehicular systems has transpired at a precipitous pace, culminating in today's incredibly complex intra-vehicular systems. Modern cars of all price levels currently require tens of millions of code and dozens of microcontrollers to function. This means that it is critical to provide information security as well as physical security to today's automobiles. Just as with traditional computer systems, attacks against vehicle data networks can be classified as local or remote. Local attacks are those that require physical access to a vehicle's network. Due to the emphasis placed on the physical security of vehicles and the fact that they are often in rapid motion, launching a direct physical attack is typically considered to be outside of the scope of vehicle threat models. However, most realistic models allow for the possibility of indirect physical access, often assumed to be made through a third party. Such potential avenues of access include devices which connect to a vehicle via its On-Board Diagnostics (OBD-II) port, CD player, or media center USB input, or even an audio jack.

Although originally intended for trusted access by first party devices given to specialized technicians and mechanics, the market for OBD-II tools has opened to include many third party devices. This means that the attack surface of the average vehicle must also be extended to include many third party devices, making the coordination of security audits for all potential hardware and software involved highly impractical. Furthermore, any vehicle or device with a network connection introduces the potential for remote system exploitation. According to a 2015 congressional report [13], nearly all vehicles currently on the road feature a wireless connection which could serve as a potential avenue of attack.

Remote attacks against vehicular systems can be further divided into the range of access provided; clearly, the further away an attack can be carried out from, the stronger the adversary. Short range access media include Bluetooth phone and music systems, remote start and entry systems, RFID access tokens, and 802.11 network connections. Longer range connections are typically established using cellular network connections. Note that some attacks may be hybrid forms which cross these classification boundaries. For instance, an attack launched remotely across a 4G cellular network against a device connected to a vehicle's OBD-II port would involve both indirect physical access and a long range wireless connection.

Unfortunately, manufacturers of vehicles and OBD-II devices are not always forthcoming regarding the wireless connections which a vehicle may be equipped with. It therefore may be possible for data transmissions to occur without the vehicle owner's permission or knowledge. Even if they are aware of such transmissions, it may be impossible to disable them without also turning off desirable functionality.

It is difficult to gauge the number of incidents in which attacks have occurred against vehicular networks in the wild due to a dearth of data in part because of manufacturers' reluctance to share such information. However, such attacks are no longer theoretical in nature, having been demonstrated against actual vehicles being driven on public roadways. In 2013, Miller and Valasek demonstrated that attacks could be launched against various ECUs over a car's CAN bus with direct physical access via an OBD-II interface [14]. This included control over the speedometer, odometer, onboard navigation, steering, braking, and acceleration. The authors were also able to modify the firmware of some ECUs, raising the possibility of advanced persistent threats which reside on an intra-vehicular network for a prolonged period of time.

Threats which apply to firmware in general also apply to vehicular systems [15], with a heightened risk due to safety issues associated with transportation systems. Initially, the automotive industry downplayed the impact of security research by emphasizing the implausibility of the direct physical access which was required. In 2015, Miller and Valasek challenged these claims when they extended their attacks by carrying them out remotely [16]. In the specific instance tested, access was gained over a long range cellular network to a vehicle's entertainment system, granting them the same control over critical car systems. The results of this research were widespread, including a recall of over a million vehicles, lawsuits against cellular carriers, and the

aforementioned congressional report on the current state of insecurity in vehicular systems.

This only represents the beginning of wireless deployments in transportation systems, however. Future deployments will cause the handful of wireless channels on today's vehicles to seem quaint by comparison. Researchers are currently envisioning future transportation systems in which all vehicles and infrastructure are connected via pervasive wireless links; such environments are typically referred to as "V2X" systems. This proliferation of wireless links implies a many-fold increase in the attack surface of a typical vehicle, as every new connection could potentially be utilized as an attack vector. Furthermore, many aspects of V2X systems differ from typical computer networks in ways which add unique challenges. For instance, in order for roadside units and other forms of infrastructure to communicate useful information they must be equipped with micro-controllers, sensors, and wireless chips. To see widespread deployment, these components will have to be as small and resource-efficient as possible. From this perspective, V2X systems are subject to many of the same constraints which "Internet-of-Things" (IoT) security solutions are subject to, including small amounts of memory and processing power, limited form-factors, low-power radio standards, and limited energy availability.

Another unique requirement of V2X is their extreme flexibility with respect to speed and movement. Transportation systems are by nature dynamic and mobile, while traditional network protocols are not designed with mobility in mind; indeed, an initial design assumption of the early Internet was that nodes would not move. The speed at which modern transportation systems are expected to move may complicate handoffs between different network cells and limit opportunities for executing protocols such as key exchange and session establishment. This, combined with the aforementioned IoT device constraints, necessitates very low overhead protocols for V2X, which is a barrier to utilizing the kinds of cryptographic solutions to security found on traditional computer systems.

The benefits of a smart transportation system can only be realized if the data collected and shared by participants is authentic. Establishing trust in a TCPS is therefore paramount. For existing vehicular networks this requires the difficult task of designing secure solutions while retaining compatibility with existing deployments. Fortunately, since V2X systems are currently emerging as more and more connections are established between vehicles and roadside assets, there is more room to "bake-in" security into these systems by applying the lessons learned in other domains. Trust is an issue for both in-vehicle and V2X communication with any potential TCPS cloud services. However, without being able to judge the veracity of messages originating from within a vehicle or other transportation asset, it is very difficult to establish trust in the data the node relays to other parties. Most V2X trust solutions thus begin with mechanisms for enforcing trust by adding cryptographic authentication components to in-vehicle networks. An alternative approach is to modify the CAN standard by adding new fields which can be used as indicators of intrusion.

Design of Driving Features:

We have initiated our investigation into potentially viable behavioral characteristics which could be used to profile driving behavior, and what sensors could potentially be used to capture them in a real world driving situation; this is one of our M2 tasks. As we continued to refine our set of potentially discriminative driving characteristics, we focused on traits which can be monitored in a way that is non-invasive both for the vehicle's computer system as well as the driver. A list of modalities and, where applicable, supporting references, is included in Table 1.

Hardware Assessment for In-Vehicle Data Collection:

Our research team assessed sensing hardware components which could potentially be used to observe the behavioral modalities identified as part of M2; this information is included in Table 1.

Figure 2 depicts pressure sensors which were considered based on their suitability as steering wheel hand position monitors. Figure 3 shows the PIs testing one of the pressure sensors; the pressure activity can be observed on the oscilloscope in the background of the figure. Figure 4 shows the same pressure sensor deployed in position on the Logitech G27 racing wheel.

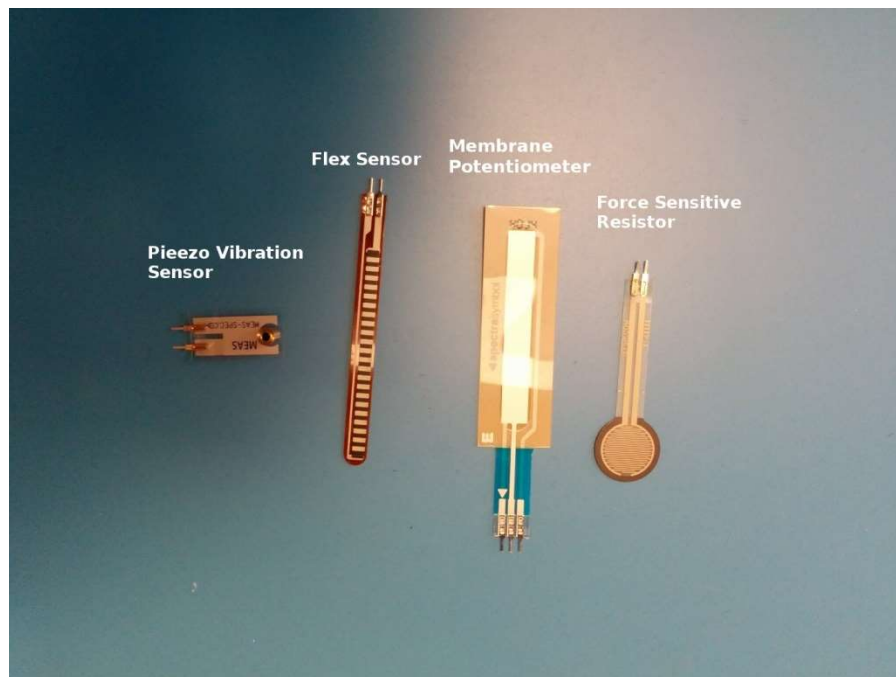


Figure 2: Various Sensors under Consideration for Measuring the Amount of Pressure Drivers Induce on Various Contact Points (e.g. on the Steering Wheel).

Modality	Description	Sensor	References
Blinker	For how long was the blinker on	A pair of gyroscopes for relative angle of the blinker or a stretch sensor	[17] [18]
Seat Belt	When does the driver put the seat belt on? How much pressure is exerted on the restraint?	Proximity Sensing with Reed Switch Sensor	[19]
Hand Position	Angle between the driver's hands	Switch array embedded into the wheel	N/A
Braking	Hard or Soft braking; emergency break usage	Force sensors placed on the pedals	[20]
Seating Position	Pressure points between the driver body and the seat	Tactilus® Automotive Occupant Pressure Measurement System	[21]
Speed	Mostly driver will have some specific destination and almost the same speed everyday	Accelerometer & GPS	[22]
Lane Position	To the extreme left, extreme right or middle. To see where the white line is.	Camera	[23]
Distance From the car	To see how much distance is there between our car and the car in front.	Proximity Sensors	[22]
Turns	How sharp the turns are	Inertial Sensors	[22]
EEG	Driver's brainwave activity while controlling the vehicle.	EEG Monitoring System	[24]
ECG	Driver's heartbeat patterns while controlling the vehicle.	ECG Monitoring System	[25]
Video-oculography	Driver's eye movement activity while controlling the vehicle.	Camera	
Operating procedure sequence	Order in which a driver performs actions and the relative timing between these events. This can be measured in a variety of circumstances, i.e., when preparing to start driving, while turning, exiting the vehicle, etc.	See above, plus timing.	N/A

Table 1: Potential Driving Modalities and Associated Sensing Hardware

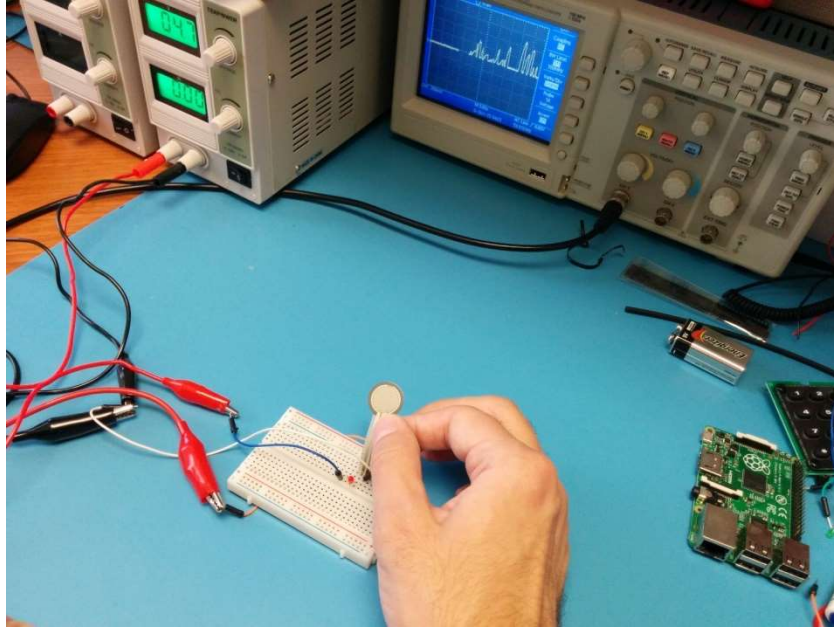


Figure 3: A Pressure Sensor Being Tested with an Oscilloscope



Figure 4: A Pressure Sensor Deployed on a Logitech G27 Racing Wheel Controller

Simulation Study Setup:

We made use of a Logitech G27 steering controller, pictured below, to present our volunteers with as realistic of a driving experience as possible.



Figure 5: The Logitech “G27 Racing Wheel” Controller

We then considered various driving simulators to determine if any present a realistic enough driving experience to derive meaningful modeling data from. A secondary goal was to assess the ease at which such simulations can be modified to extract behavioral data.

We identified the driving simulations listed in Table 2 and tested their suitability for our experimental goals.

Simulator	Website
OpenDS	http://www.opens.de/
TORCS	http://torcs.sourceforge.net/index.php?name=Sections&op=viewarticle&artid=3
VDRIFT	http://vdrift.net
Speed Dreams	http://www.speed-dreams.org
Racer	http://inkeepr.com/updates/?gclid=CN6V17OzicYCFY2RHwodd4YAqQ
CARS	http://cars.pcuie.uni-due.de/index.php?id=6

Table 2: Driving Simulation Software

Figure 6 shows a graduate research assistant in the process of testing the OpenDS driving simulator [26].

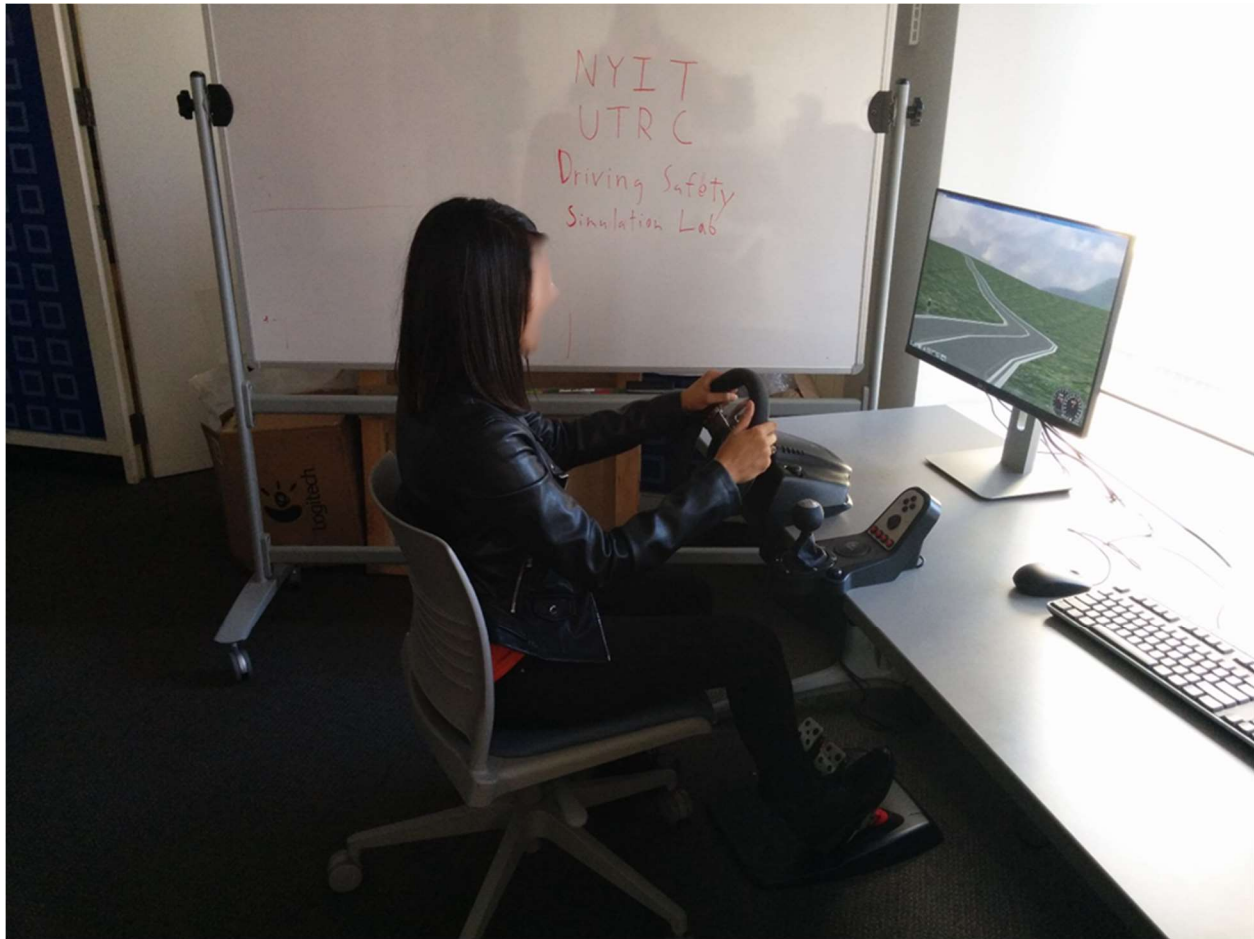


Figure 6: A Research Assistant Testing the OpenDS Driving Simulator with the Logitech G27 Controller.

Each driving simulator was assessed along five metrics: ease of setup, modifiability with respect to software instrumentation, scenario flexibility, computer resource load, and the realism of the driving experience. The results of our experience working with each simulator to this point is detailed below.

OpenDS

Website: <http://www.opens.de/>

OpenDS is compatible with Windows and our steering controller hardware. The project is open source, so its code was easily available and easy to modify. OpenDS is built on a JAVA platform. Proper documentation is available for modifying its source code to better suit the needs of our study; this extensive documentation includes video tutorials and webinars for deeper understanding of

the simulator. Its source code is well structured and organized, which facilitated our modification of its logging capabilities to output additional relevant data on user interactions.

The OpenDS maintainers have integrated third party tools to develop unique feature like scenario recording and replay. OpenDS offers several preconstructed scenarios. These are created using version 3 of the JavaMonkey Engine. This framework makes it easy to add 3D objects such as stationary cars and road signs. Adding objects which interact with their environment in a realistic manner, such as more traffic, has proven to be relatively difficult, however.

Performance snapshot: Memory consumption - 4.51 GB, CPU Utilization - 45%.

TORCS

Website:

<http://torcs.sourceforge.net/index.php?name=Sections&op=viewarticle&artid=3>

TORCS is a 3D racing car simulator. Although it is compatible with Windows as well as Linux, TORCS has many libraries and dependencies that need to be installed. We were nonetheless able to compile and execute this software without encountering any issues. The limited documentation provided by TORCS made modifying its code difficult, however.

TORCS offers one default scenario and instructions for creating new ones were also lacking. Although there are many different kinds of cars offered and an engine for creating new racing tracks, there did not seem to be a mechanism for adding traffic, pedestrians, or other everyday driving elements. Overall this program seemed more suitable as a “racing” game rather than a platform for studying realistic driving behavior.

Performance snapshot: Memory consumption - 3.42 GB, CPU Usage - 36%.

VDRIFT

Website: <http://vdrift.net>

VDrift is an open source driving simulation which focuses on drift racing. It supports Linux, MacOs and Windows. It written in C++ and inspired by the vamous physics engine. We found this software to be difficult to work with due to incomplete documentation.

VDRIFT comes with one predesigned scenario which mainly focuses on a drifting style racing game. There is no documentation provided regarding how to create new scenarios. Much like TORCS, VDRIFT seems better suited to racing usage rather than real world driving.

Performance snapshot: Memory Usage - 3.16 GB, CPU Usage - 34% .

Speed Dreams

Website: <http://www.speed-dreams.org>

Speed dreams is an open source motorsport driving simulation. It is compatible with both Windows and Linux. We could not locate documentation for scenario creation. Various stock scenarios are offered, but none corresponded to everyday driving.

Memory Performance snapshot: 4.32 GB, CPU Usage - 42%

CARS

Website: <http://cars.pcuie.uni-due.de/index.php?id=6>

CARS is an open source driving simulator which is can be compiled for Windows. It uses a Java and C++ based platform which is easy to manipulate. CARS consists of a driving simulation tool, map editor, and an analysis tool. The map editor allows to new scenarios to be created easily with the JavaMonkey Engine. Unfortunately documentation was sparse and we encountered several hardware compatibility issues. For instance, the code does not run on 64-bit environments by default.

Performance snapshot: Memory Usage - 3.30 GB, CPU Usage - 47%

Driving Study Simulation Task Development:

Having concluded, based on aforementioned driving software assessment, that the most realistic and easily utilized option for an initial study is the OpenDS project, we proceeded to create a scenario that was appropriate for collecting data on people's driving behavior. This scenario was intended to group drivers into behavior clusters (such as cautious and aggressive) and identify the driving habits of an individual.

OpenDS provides choices to control the weather and it's intensity in a scenario. Options include rain, snowfall, and fog. OpenDS also allows scenario designers to control the friction between the wheel and the road. Traffic vehicles in the scenario are modeled to keep a safe distance from the driving vehicle.

The maintainers of OpenDS provide several default scenarios with the program's source code, including a countryside and city environment. In OpenDS, scenario creation is handled using version 3 of the Java Monkey Engine. As a first step, we imported the provided "j3o" scenario files and modified them. We reviewed the scenes, models, textures, and materials used in the provided scenarios. We made us of the Blender 2.7X 3D modelling software to generate these files.

We inserted new objects in our scenario to create more "events of interest" which were intended to induce responses which may involve discriminative driving behavior. For example, signs were inserted so that we could analyze the responses that they induced when encountered by our study participants.

Additional Python scripts were needed to generate an ORGE Mesh file and a corresponding scene file. This required the implementation of a Java class to convert the mesh.xml file to a J3O file [27]. Once this was completed we were able to add new objects to driving environments. For example, Figure 7 shows a car object placed on a blank field. OpenDS allows stationary objects to be positioned in a scenario without having to import it in Java Monkey Engine by using the object locator. With the help of the object locator we can insert stationary object in the scenario at various positions.



Figure 7: A Screenshot of an Unpopulated OpenDS Map

Inserting traffic into a scenarios requires using an OpenDS concept know as “waypoints.” Waypoints are the coordinates that are given to the 3D object in order for it to traverse a scenario map. Since it is a 3D plane the waypoints are provided along three (x,y,z) planes. Various parameters for the object like rotation, scaling, translation are available to position the object accordingly in the scenario. Rotation allows for the placement of objects according to different viewpoints. Scaling allows the sizes of objects to be manipulated. Lastly, translation allows objects to be positioned within a scenario environment.

Next, we added many objects to our OpenDS-based driving simulation in order to improve its realism. We have successfully added the traffic in the form of cars and pedestrians as well as road signs to our testing scenario. We have also inserted traffic lights in our scenario to make the scenario more realistic by providing drivers with another roadside component to react to. “Waypoint” coordinates in 3D space must be assigned to objects to move them through a specific path. We also used OpenDS’s object locator functionality to calculate object waypoints.

Preliminary Human Subject Study with Driving Simulation

Using the features identified in the previous section, we created a scenario for preliminary data collection where each user can complete the whole driving task in 5 minutes by following signs and traffic lights. We recruited 10 test subjects from the NYIT community and asked them each to perform 4 5-minute laps, therefore we collected approximately 20 minutes of driving data per subject. The most critical decision of the data collection is the parameters to be considered to discriminate each user's behavior from one another. We have collected 6 parameters of each user's driving activity, namely a timestamp, Position (X, Y, and Z coordinates), speed (km/hr), steering wheel position, gas pedal position, and brake pedal position. As an example, a snapshot of approximately 1-sec driving data collected from a single subject is shown in Figure 8.

Time (minutes:seconds:milliseconds)	X Position	Y Position	Z Position	X Rotation	Y Rotation	Z Rotation	W Rotation	Speed (km/h)	Steering Wheel Position	Gas Pedal Position	Brake Pedal Position
0:0:0	-135.421	-0.47	-51.307	0.0009	-0.7142	0.0009	0.6999	53.04	0.00186	1	0
0:0:48	-134.948	-0.47	-51.297	0.0009	-0.7141	0.0009	0.7	53.39	0.00186	1	0
0:0:96	-134.471	-0.47	-51.288	0.0009	-0.714	0.0009	0.7001	53.74	0.00186	1	0
0:0:150	-133.992	-0.47	-51.279	0.0009	-0.7139	0.0009	0.7002	54.1	0.00266	1	0
0:0:185	-133.509	-0.47	-51.27	0.0009	-0.7138	0.0009	0.7004	54.46	0.00266	1	0
0:0:221	-133.023	-0.47	-51.262	0.0009	-0.7136	0.0009	0.7005	54.82	0.00266	1	0
0:0:258	-132.538	-0.47	-51.253	0.0009	-0.7135	0.0009	0.7007	55.18	0.00266	1	0
0:0:290	-132.041	-0.47	-51.245	0.0009	-0.7133	0.0009	0.7009	55.54	0.00266	1	0
0:0:334	-131.545	-0.47	-51.237	0.0009	-0.7131	0.0009	0.701	55.91	0.00266	1	0
0:0:370	-131.046	-0.47	-51.229	0.0009	-0.713	0.0009	0.7012	56.27	0.00266	1	0
0:0:412	-130.544	-0.47	-51.221	0.0009	-0.7128	0.0009	0.7014	56.64	0.00266	1	0
0:0:454	-130.039	-0.47	-51.213	0.0009	-0.7126	0.0009	0.7015	57	0.00351	1	0
0:0:495	-129.53	-0.47	-51.206	0.001	-0.7125	0.0009	0.7017	57.37	0.00351	1	0
0:0:536	-129.018	-0.47	-51.199	0.001	-0.7122	0.0009	0.7019	57.74	0.00351	1	0
0:0:626	-128.503	-0.47	-51.192	0.001	-0.712	0.0009	0.7022	58.11	0.00351	1	0
0:0:659	-127.984	-0.47	-51.186	0.001	-0.7118	0.0009	0.7024	58.48	0.00351	1	0
0:0:694	-127.462	-0.47	-51.18	0.001	-0.7116	0.0009	0.7026	58.85	0.00431	1	0
0:0:732	-126.937	-0.47	-51.174	0.001	-0.7113	0.0009	0.7029	59.22	0.00431	1	0
0:0:773	-126.408	-0.47	-51.169	0.001	-0.711	0.0009	0.7032	59.59	0.00431	1	0
0:0:813	-125.877	-0.47	-51.164	0.001	-0.7107	0.0009	0.7034	59.96	0.00431	1	0

Figure 8: “Raw” or unprocessed data collected from the OpenDS driving simulator

Exploratory Modeling and Analysis of Preliminary Data

With the groundwork laid by the efforts described in the previous sections, we were able to concentrate on getting the most meaningful data from our study. The core goal of our experiments was to classify users based on their driving behaviors. We developed a feature extraction and representation methodology which distills the data that was captured by the simulator into statistical measurements that are optimized for behavior classification. Our preliminary analysis included an unsupervised learning algorithm, namely K-means clustering, and supervised learning algorithms, such as the naïve bayes and Support Vector Machines (SVM) classification algorithms.

After completing the data collection on our participants' driving habits, the data is stored in a MySQL database. Next, pertinent information is extracted and discriminative features with a potential to distinguish drivers are identified. The original or “raw” data output from the simulator is shown in Figure 8. Note that this snapshot in Figure 8 shows only a brief recording for illustrative purposes and it is not representative of the discriminative properties of the collected data for distinguishing drivers. Furthermore, we hypothesized that higher-level features would be more representative of driving behavior and thus more discriminative when used as a classifier.

Driver ID	Distance Travelled	Average Speed	Standard Deviation of Steering Wheel Position	Average Change in Brake Position	Average Change in Accelerator Position	Average X Position	Average Y Position	Average Z Position	Average X Rotation	Average Y Rotation	Z Rotation	Average W Rotation
0	55.9785	52.92033	0.02713	0.01667	0.01667	-103.422	-0.46965	-50.9377	0.0006	-0.71631	0.0007	0.69744
0	13.41303	12.51967	0.02112	0.01639	0.01639	-71.5894	-0.46974	-50.5204	0.00084	-0.69838	0.00078	0.71564
0	39.65633	36.54194	0.18131	0	0	-58.6749	-0.46982	-40.745	0.00046	0.4187	-0.00047	0.19952
0	82.61378	76.58033	0.01895	0.00318	0.00813	-57.3336	-0.46999	19.03712	-0.00001	0.99983	-0.00129	-0.00626
0	38.49689	35.62184	0.17445	0.01281	0.008	-52.3857	-0.46954	84.15809	0.00002	0.39968	-0.00014	0.15095
0	63.49857	59.2778	0.0218	0	0	-11.3237	-0.47	87.98444	0.00089	-0.70819	0.00093	0.70582
0	84.44627	85.19876	0.00381	0.01424	0.00885	68.94776	-0.46965	88.15843	0.00068	-0.70582	0.0007	0.70838
0	35.66643	33.4048	0.01876	0.0155	0.02071	134.038	-0.46941	87.6901	0.00063	-0.70695	0.00064	0.70714
0	26.4544	24.1868	0.0901	0	0.00169	152.6123	-0.46994	84.00558	0.00131	-0.49125	-0.00002	0.82333
0	70.33792	65.61598	0.01619	0	0	159.7704	-0.47	38.86499	0.0013	-0.0031	0.00004	0.99982
0	36.66649	34.17333	0.01725	0.01626	0.01626	160.5755	-0.46949	-28.3427	0.00081	0.01643	-0.00006	0.99978
0	40.80579	37.27605	0.16628	0	0	167.9782	-0.46987	-47.8694	0.00083	-0.38068	0.00117	0.84754
0	79.5926	74.78779	0.01049	0.00359	0.0082	226.1545	-0.46981	-51.7381	0.0008	-0.70338	0.00091	0.71073
0	21.21062	19.49246	0.02658	0.01147	0.00794	283.1357	-0.46952	-51.9179	0.00071	-0.69072	0.00056	0.72274
0	49.1395	45.80902	0.10694	0	0	298.4059	-0.46981	-69.6901	0.00143	-0.1602	-0.00044	0.95415
0	87.05216	80.32339	0.03376	0.00416	0.00806	298.937	-0.46969	-139.834	0.00114	0.01058	-0.0003	0.99954
0	96.84792	89.50371	0.08935	0	0.01613	255.9343	-0.46977	-194.037	0.00016	0.63957	-0.00206	0.74803
0	35.20955	32.29992	0.02205	0.03094	0.008	177.4186	-0.46926	-194.245	0.00062	0.71522	-0.0004	0.69881
0	33.09369	30.78959	0.10547	0	0	160.5121	-0.46992	-187.013	-0.00015	0.87965	-0.00137	0.36175
0	78.15263	71.23448	0.00664	0.00406	0.008	155.7826	-0.46993	-133.395	0.00004	0.99999	-0.00127	0.00456
0	26.33133	24.2232	0.01842	0.0225	0.008	155.3375	-0.46908	-68.8631	-0.00005	0.99971	-0.00076	-0.01834
0	29.44231	26.53437	0.01637	0	0	155.6814	-0.47	-53.1641	-0.00001	0.99988	-0.0013	-0.01313
0	74.50925	68.59847	0.0023	0	0.00265	156.2514	-0.47	-0.32367	0	1	-0.0013	-0.00157
0	26.03831	23.96864	0.00157	0.016	0.01188	156.3982	-0.46942	61.37842	-0.00001	0.99998	-0.00081	-0.00799
0	43.69866	39.90484	0.09736	0	0	164.2882	-0.46988	81.49371	-0.00002	0.29683	-0.00044	0.18845

Figure 9: Features extracted by processing the driving data collected from the OpenDS driving simulator

Initially, we chose five features as potential discriminative characteristics to capture each subject’s unique driving pattern (1) Euclidean distance travelled, (2) average vehicle speed, (3) the standard deviation of the steering wheel position, (4) the average change of brake position, and (5) the average change of acceleration position. To measure how well these features captured patterns specific to a driver, we calculated Fisher’s Score [fisher] based on the five high-level features. We use one-dimensional variation evaluating the value of each feature independently via the ratio of the inter-class and intra-class variance as shown below:

$$S = \frac{\sigma_b}{\sum_{i=1}^k \sigma_i}$$

$$\sigma_b = \sqrt{\sum_{i=1}^k (m_i - m_G)^2}$$

The Fisher scores for both the average values of our “raw” driving data and our derived features is shown in Table 1. Based on these results we will focus our attention on steering, both pedals, X and Z position, and Y and W rotation. This is valuable information both for guiding our future modeling efforts and updated driving scenario for a broader data collection effort.

Feature	Fisher Score
Distance Travelled	0.0042
Average Speed	0.0035
Standard Deviation of Steering Position	0.7556
Average Change in Brake Pressure	8.2819
Average Change in Accelerator Pressure	13.809
Average X Axis Position	13.809
Average Y Axis Position	0.0003

Average Z Axis Position	73.1598
Average X Axis Rotation	0.0002
Average Y Axis Rotation	27.0113
Average Z Axis Rotation	0.02838
Average W Axis Rotation	14.0953

Table 3: Fisher scores for driving features

We initially experimented with unsupervised learning to determine if this technique would be sufficient to derive any insights from our preliminary dataset. We first attempted to partition our user data into three clusters. The main idea is to define k-centroids around which each centroid's data is to be clustered. The next step is to take each point belonging to given data set and associate it to a nearest center. The best result occurs when each cluster is far away from other clusters.

```

Cluster centroids:
Attribute          Full Data          Cluster#
                   (79541)           (40785)    (33183)    (5573)
-----
Speed (km/h)      31.4051           29.5678    35.4504    20.7652
Steering Wheel Position [-1,1]  0.0163           0.0112    0.0238    0.0091
Gas Pedal Position    0.3896           0.0589    0.8609    0.0037
Brake Pedal Position  0.0643           0.0149    0.0001    0.8077

Time taken to build model (full training data) : 1.17 seconds

=== Model and evaluation on training set ===

Clustered Instances
0      40785 ( 51%)
1      33183 ( 42%)
2       5573 (  7%)

```

Figure 10: The result of using k-means clustering to divide the dataset into 3 clusters

From Figure 11 we can conclude that 0.82% of our sample data was incorrectly classified. Figure 12 displays a graphical representation of the clustering process, showing how much data from each user fell into each cluster.


```

=== Model and evaluation on training set ===

Clustered Instances

0      40785 ( 51%)
1      33183 ( 42%)
2       5573 (  7%)

Class attribute: cluster
Classes to Clusters:

    0    1    2 <-- assigned to cluster
40583  358   89 | cluster1
  175 32825   0 | cluster2
    27   0 5484 | cluster3

Cluster 0 <-- cluster1
Cluster 1 <-- cluster2
Cluster 2 <-- cluster3

Incorrectly clustered instances :    649.0    0.8159 %

```

Figure 11: Evaluation of Clustering Accuracy

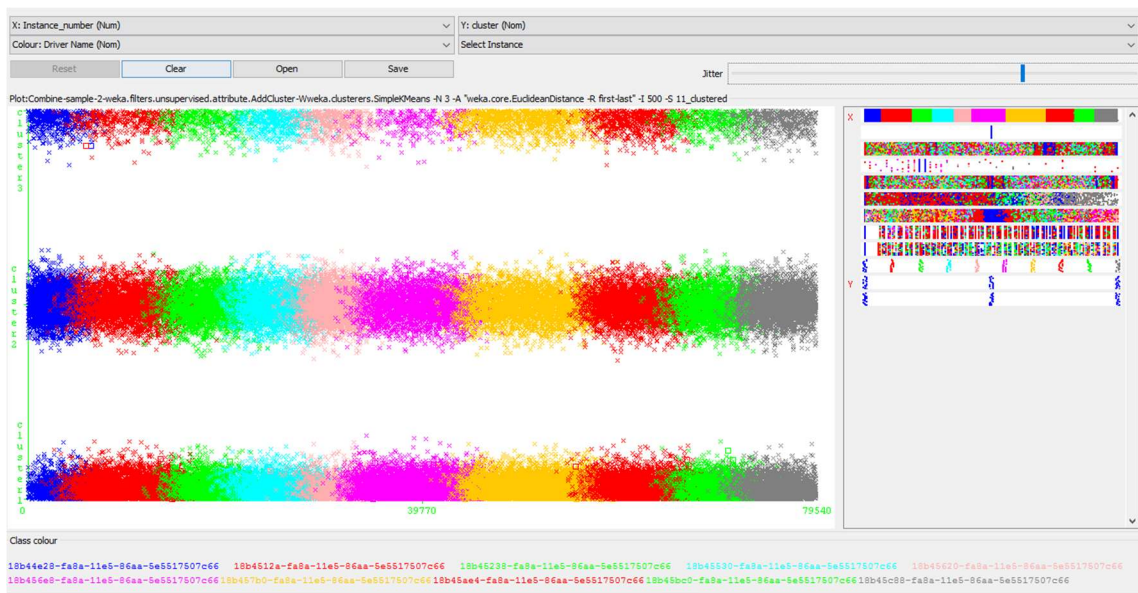


Figure 12: Clustering Visualization

Another clustering approach we experimented with is known as Expectation Maximization (EM). EM assigns a probability distribution to each sample which indicates the probability of it belonging to each of the clusters. EM can decide how many clusters to create by cross validation, or we can specify how many clusters to generate. We used 10 Fold Cross Validation and Number of clusters equal to 3, yielding the following results:

Clustered Instances

0 553 (43%)
 1 90 (7%)
 2 658 (51%)

Log likelihood: 0.37577

Cluster 0 <-- E94F0128-FBFB-40EC-A5EC-9005FDF49178
 Cluster 1 <-- 817B7B48-FDC8-4BFB-ADE7-2EC8B273F025
 Cluster 2 <-- A46B927F-3A8B-4DF7-BEA5-09362CF84DE2

Incorrectly clustered instances: 1032.0 79.3236 %

We can see that User 3, 5 and 8 are uniquely assigned to different clusters. Also, the correctly clustered instances in this case correspond to about 20.68%. Figure 13 indicates the average speed in the of the cluster assignments. Average Speed is plotted on the X axis with Class (user) on the Y axis. The coloring is based on what cluster each sample was assigned to.

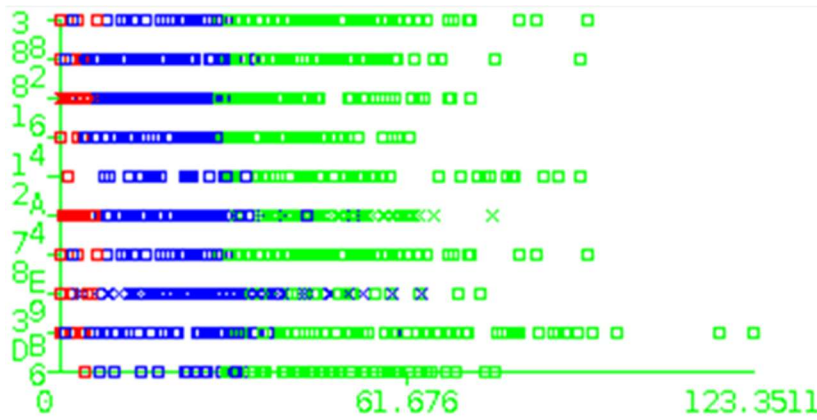


Figure 13: EM Clustering Visualization

```
ClassificationViaClustering
=====

kMeans
=====

Number of iterations: 3
Within cluster sum of squared errors: 21608.122688506595
Missing values globally replaced with mean/mode

Cluster centroids:
Attribute                Full Data          Cluster#
                          (79541)            (41030)            1                2
                          (79541)            (41030)            (33000)          (5511)
-----
Used Format = Time (ms)   1460000000000      1460000000000      1460000000000    1460000000000
Position_X                112.3664           104.4649            120.8488           120.4018
Position_Y                -0.4694            -0.4691              -0.4698            -0.4693
Position_z               -58.0421            -61.4598             -54.3679            -54.5984
Speed (km/h)              31.4051            29.5703              35.4662             20.748
Steering Wheel Position [-1,1] 0.0163             0.011                0.0241              0.0093
Gas Pedal Position         0.3896             0.0607                0.863                0.0038
Brake Pedal Position       0.0643             0.0155                0.0001                0.8119
cluster                   cluster1            cluster1              cluster2             cluster3
```

Figure 14: Classification via Clustering Results

```

Clusters to classes mapping:
1. Cluster: 18b457b0-fa8a-11e5-86aa-5e5517507c66 (7)
2. Cluster: 18b45c88-fa8a-11e5-86aa-5e5517507c66 (10)
3. Cluster: 18b45ae4-fa8a-11e5-86aa-5e5517507c66 (8)

Classes to clusters mapping:
1. Class (18b44e28-fa8a-11e5-86aa-5e5517507c66): no cluster
2. Class (18b4512a-fa8a-11e5-86aa-5e5517507c66): no cluster
3. Class (18b45238-fa8a-11e5-86aa-5e5517507c66): no cluster
4. Class (18b45530-fa8a-11e5-86aa-5e5517507c66): no cluster
5. Class (18b45620-fa8a-11e5-86aa-5e5517507c66): no cluster
6. Class (18b456e8-fa8a-11e5-86aa-5e5517507c66): no cluster
7. Class (18b457b0-fa8a-11e5-86aa-5e5517507c66): 1. Cluster
8. Class (18b45ae4-fa8a-11e5-86aa-5e5517507c66): 3. Cluster
9. Class (18b45bc0-fa8a-11e5-86aa-5e5517507c66): no cluster
10. Class (18b45c88-fa8a-11e5-86aa-5e5517507c66): 2. Cluster

Time taken to build model: 1.04 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      12187          15.3217 %
Incorrectly Classified Instances    67354          84.6783 %
Kappa statistic                    0.0315
Mean absolute error                 0.1694
Root mean squared error             0.4115
Relative absolute error             94.9745 %
Root relative squared error         137.8221 %
Total Number of Instances          79541

```

Figure 15: Cluster and Class Mapping Output

An alternative modeling approach is using clustering as a rough form of a classifier; this process is known as classification via clustering. This method uses clustering result for the classification. Figures 14 and 15 show the results of using clustering in order to classify our collected driving data. As per Figure 14, the class 7, 8 and 10 falls in cluster 1, 3 and 2 respectively. The proportion of correctly classified instances is quite small, which we took as an indication that classification via clustering should not be pursued further.

```

=== Detailed Accuracy By Class ===

TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
0.105    0.063    0.104     0.105  0.104     0.521    18b44e28-fa8a-11e5-86aa-5e5517507c66
0         0         0         0       0         0.5      18b4512a-fa8a-11e5-86aa-5e5517507c66
0         0         0         0       0         0.5      18b45238-fa8a-11e5-86aa-5e5517507c66
0         0         0         0       0         0.5      18b45530-fa8a-11e5-86aa-5e5517507c66
0.075    0.045    0.11      0.075  0.089     0.515    18b45620-fa8a-11e5-86aa-5e5517507c66
0.193    0.154    0.164     0.193  0.178     0.52     18b456e8-fa8a-11e5-86aa-5e5517507c66
0.52     0.408    0.194     0.52   0.283     0.556    18b457b0-fa8a-11e5-86aa-5e5517507c66
0.079    0.076    0.114     0.079  0.093     0.501    18b45ae4-fa8a-11e5-86aa-5e5517507c66
0         0         0         0       0         0.5      18b45bc0-fa8a-11e5-86aa-5e5517507c66
0.262    0.221    0.105     0.262  0.15      0.52     18b45c88-fa8a-11e5-86aa-5e5517507c66
Weighted Avg.  0.153    0.121    0.09      0.153  0.106     0.516

=== Confusion Matrix ===

 a  b  c  d  e  f  g  h  i  j  <-- classified as
545 0  0  0  413 479 1399 575 0 1796 | a = 18b44e28-fa8a-11e5-86aa-5e5517507c66
542 0  0  0  378 1773 4429 637 0 1887 | b = 18b4512a-fa8a-11e5-86aa-5e5517507c66
474 0  0  0  334 925 2543 572 0 1616 | c = 18b45238-fa8a-11e5-86aa-5e5517507c66
485 0  0  0  360 1138 2671 536 0 1658 | d = 18b45530-fa8a-11e5-86aa-5e5517507c66
545 0  0  0  413 560 1643 589 0 1751 | e = 18b45620-fa8a-11e5-86aa-5e5517507c66
525 0  0  0  418 2083 5294 601 0 1847 | f = 18b456e8-fa8a-11e5-86aa-5e5517507c66
580 0  0  0  421 2378 6569 676 0 2009 | g = 18b457b0-fa8a-11e5-86aa-5e5517507c66
547 0  0  0  314 1616 3827 694 0 1815 | h = 18b45ae4-fa8a-11e5-86aa-5e5517507c66
485 0  0  0  320 868 2600 555 0 1636 | i = 18b45bc0-fa8a-11e5-86aa-5e5517507c66
523 0  0  0  377 848 2898 668 0 1883 | j = 18b45c88-fa8a-11e5-86aa-5e5517507c66

```

Figure 16: Confusion Matrix for Classification via Clustering

As shown in Figure 16, classification via clustering results in a 0% true positive rate for some classes, which underscores that this combination of algorithm, features, and application are not a viable combination.

Since clustering resulted in limited classification quality, we moved on to more powerful supervised learning techniques. We were successfully able to collect basic results from Classification using SVM (SMO, libsvm – cSVC and One Class) and Clustering using the EM algorithm. All tests are performed in WEKA [weka], a machine learning software suite written in Java.

The first supervised classification approach we tried was to classify the data using Support Vector Machines (SVM). SVMs infer a function based on trained labeled data and uses it to map new data. SVM performs classification by constructing hyperplanes in a multidimensional space that separates cases of different class labels. In Weka, Class SMO implements John Platt's sequential minimal optimization algorithm for training a support vector classifier. It solves the multi class problem using a pairwise classification. For example, it first compares and solves User 1 and User 2, then User 1 and User 3 and so on for User 1 and all other Users.

We plotted ROC and Detection Error Tradeoff (DET) curves to provide a fair comparison between these disparate classification techniques. An ROC curve is a plot of a classifier's true positive rate, or sensitivity, against its false positive rate as the threshold for classification is altered. Values to the lower left of the ROC curve represent more conservative threshold values, with less false positives (i.e., false alarms about an authentic driver's identity) but also less true positives (i.e., a less successful unauthorized driver detection rate). The upper right of the ROC curve, on the other hand, shows less conservative thresholds where attacker detection is maximized at the cost of increased false positives. Because the goal of driver classification is to maximize the true positive rate of detection while minimizing the number of false alarms raised during regular driving activities, the goal is to maximize the area under the ROC curve (AUC).

DET curves are very similar to ROC curves in that both plot classifier performance as a function of threshold adjustment. A DET curve plots a classifier's true positive rate against its false positive rate, however, while a DET curve instead plots a classifier's false rejection rate against its false positive rate. DET curves are useful for visualizing the relationship between these error rates. The point at which both error rates equal each other is known as the Equal Error Rate (EER).

The default implementation of SMO is not probabilistic and only a single optimal value of threshold is provided. This basic test with a Linear Kernel and 10 Cross Validation resulted in an Area Under the Receiver Operating Characteristic curve (AUC) area of 0.684. The correctly classified Instances were 30.82%. SMO also normalizes the input data and requires the class attribute to be nominal.

To improve the AUC of our classification results, we need to assign probability values to SVM predictions. In Weka, we do this using Logistic Models. We conducted our test with, setting this attribute to True and C to 10.0.

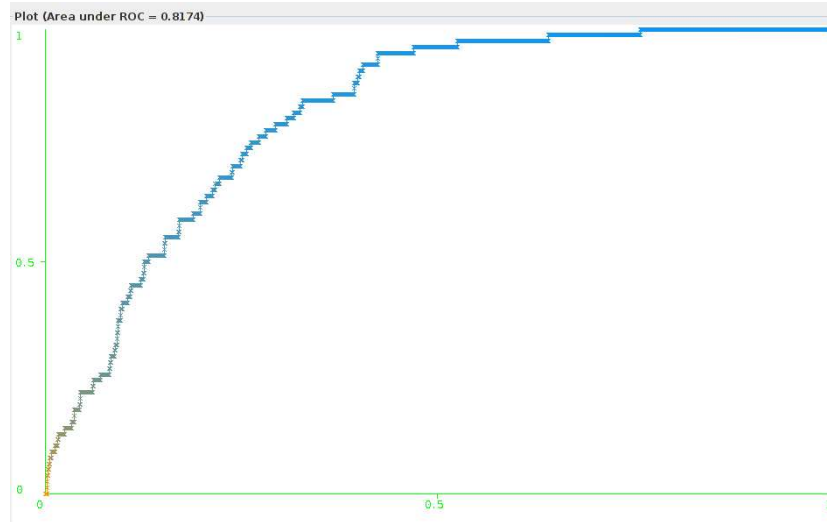


Figure 17: A Receiver operating characteristic curve for one study participant.

Figure 17 shows the Receiver Operating Characteristic (ROC) curve for user 1. It is plotted with X axis as False Positive Rate and Y axis as True Positive Rate. The coloring in the graph is based on the Threshold. The blue part represents lower threshold. Figure 18 shows the ROC area obtained for the all the users. Users 1, 3, 5 and 8 were better classified than other users. With LogisticModels set to True and C = 10.0, the average AUC went as high as 0.812 and correctly classified instances improved to 42.20%. The Kappa Statistic (analog to Correlation Co-efficient) is 0.3399.

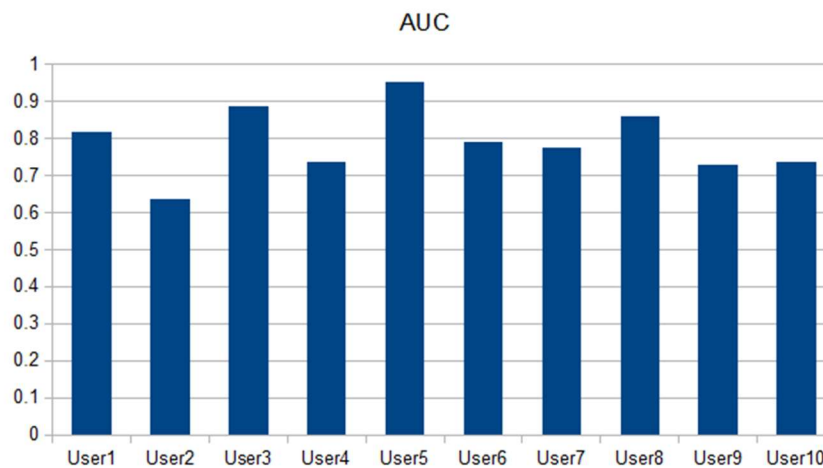


Figure 18: The area under the curve values for each participant in our user study

In addition to assigning probabilities to SVM, we tried the effects of parameter C and different kernel Types on the ROC area. Figure 19 shows the effect of C on the average ROC area. C is a budget for the slack variables which allows some instances to be on the wrong side of the hyperplane. Higher C means a smaller budget and thus a more stringent hyperplane which aims to misclassify fewer training examples. Default value of C is 1.0. We see that for C = 1.0, we get a lower ROC area (nearly 0.74). In other words, the higher the value of C, the higher is the ROC area.

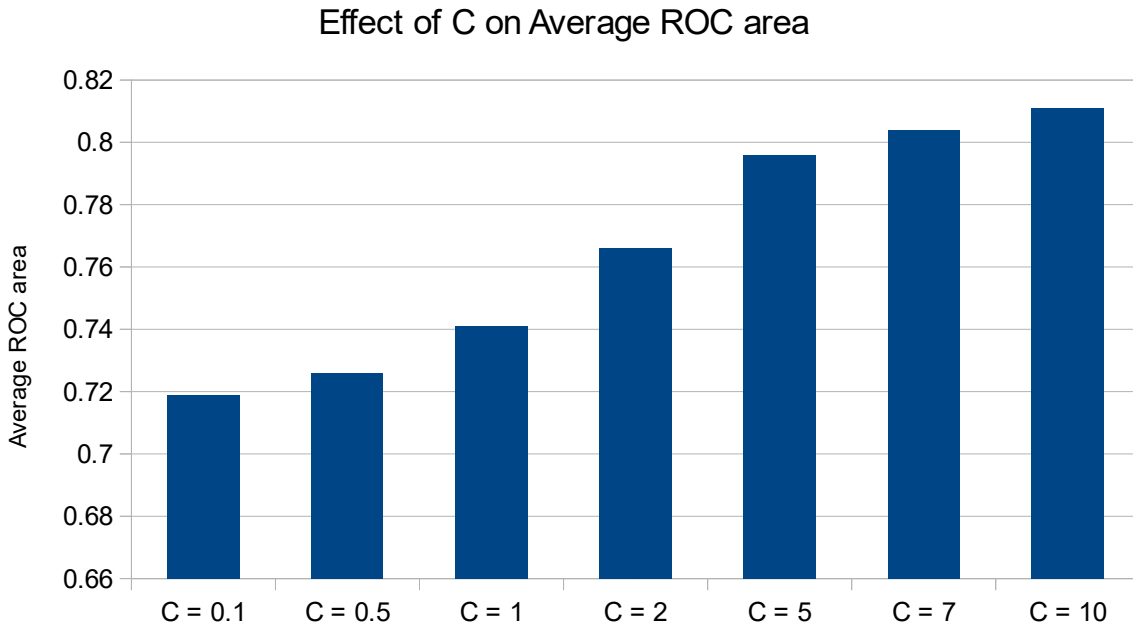


Figure 19: The effect of C on average user AUC

To test the effect of different kernels, we can set the required exponent value for the kernel in Weka. Linear (Exponent -1.0), Quadratic (Exponent- 2.0), Polynomial(Exponent- 3) or higher. Figure 20 shows that the linear kernel is the best as it gives the maximum ROC area. Tests with 5 fold cross validation results in the almost the same percentage of correctly & incorrectly instances as 10 fold with slight difference in decimal points. Also, increasing the folds to 15 results in higher value of incorrectly classified instances.

With an objective of further improving the ROC area – and hence improving the accuracy of classification, we then conducted our test using LIBSVM [28]. LIBSVM is a wrapper class for libsvm tools. It is a free package and can be added to Weka. LIBSVM is faster than SMO as it uses libSVM to build the SVM classifier. But LIBSVM uses a similar algorithm as SMO. Among 5 SVM types of LibSVM, we conducted our basic test with C-SVC and One Class. C-Support Vector Classification(C-SVC) performs classification for two class or multi- class problem. We used the default parameters to

run the test, SVM Type 0 – c-SVC : A Radial basis function as kernel type, nu as 0.5 and 10 folds Cross Validation. C-SVC doesn't normalize the data as in SMO. We get correctly classified instances of nearly 35% and the average ROC area is 0.628.

To improve the ROC area here, we programmed LIBSVM to assign probability values, a linear kernel, and a cost value of 10.0. For this test, we get an average ROC area of 0.812 (SMO - 0.811) and correctly classified instances of 41.35% (SMO – 42.20%). Users 1, 3, 5 and 8 are better classified than the others. The Kappa statistic is 0.327(SMO – 0.339). Figure 21 shows the effect of different kernel types on Average ROC area. Linear kernel gives the best ROC area. The Polynomial, Radial Basis function and Sigmoid kernel types depend on the value of gamma (default – 0.0), coef0(default -1.0) and degree(default -3). We used the default values for comparison below.

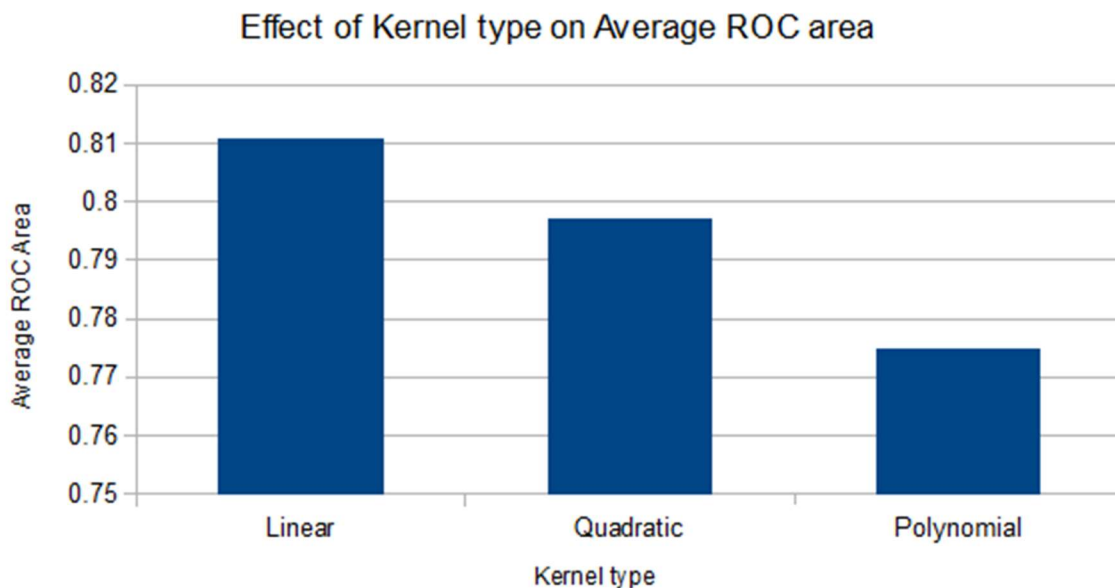


Figure 20: Effect of kernel type on AUC

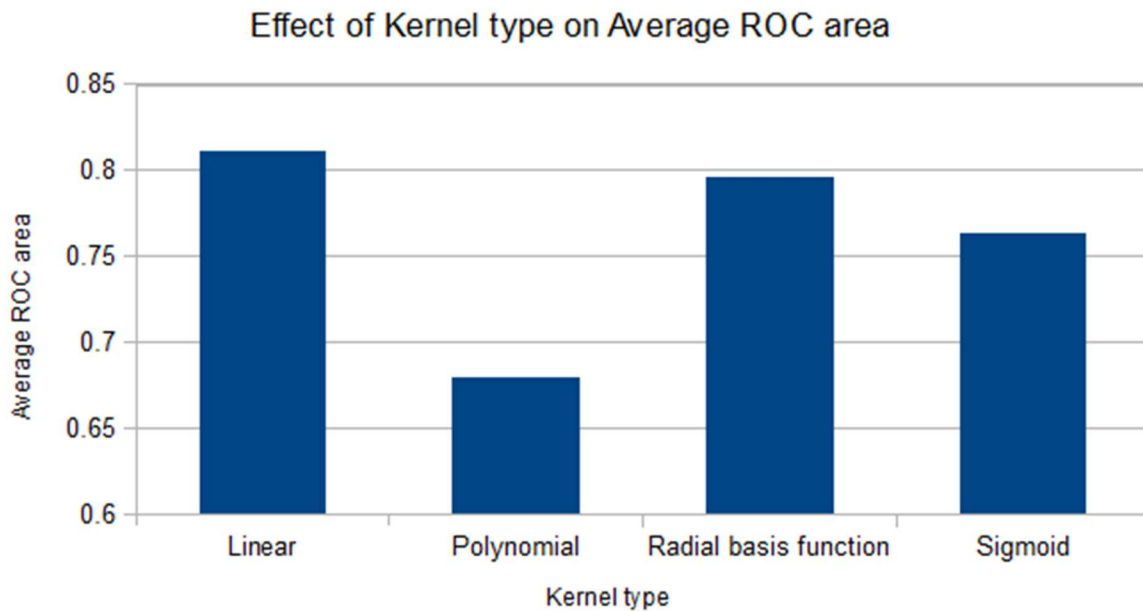


Figure 21: Effect of kernel type on LIBSVM classification

We experimented with other types of classification kernels in order to maximize our classification performance. There was a slight improvement in AUC when we used a linear kernel instead of the Weka-default polynomial kernel. With this type of kernel, the maximum true positive rate obtained for the 10 user dataset is 38.58%, which is less than the 42.2% obtained in with a polynomial kernel. However, the AUC increases to 0.816, which is slightly more than the 0.812 value achieved using a polynomial kernel. The Pearson VII function based universal kernel showed even better performance, with a TPR of 43.5% and an AUC of 0.842.

Another useful metric of comparison for classifiers is the equal error rate (EER). This is the threshold point at which both acceptance and rejection errors are equal. We wrote a scikit-learn python script to perform the libsvm test and output the classification prediction values. A machine learning toolbox known as Bob [Bob] was utilized in order to find the EER for our classifier. The measure module in Bob has a function to calculate the EER and threshold values. The script considers 10% as testing set, with probability estimates set to true and C=10.0.

We also sought to model our driving data as a one class problem instead of a multiclass one, which more accurately captures the task of driver identification. For these tests we organized each user's driving sessions into three laps of training data and one lap of test data. Training was done on a per-user basis, while testing involved all users' test sets. We performed the libsvm one class classification test for a single user and achieved a 49% TPR.

```
shannon@shannon-SATELLITE-L755:~/Desktop Matter$ python two.py
None
(1301, 5)
roc_auc = 0.68
eer_value = 0.30
eer_th = 0.36
roc_auc = 0.89
eer_value = 0.18
eer_th = 0.61
roc_auc = 0.78
eer_value = 0.34
eer_th = 0.73
roc_auc = 0.71
eer_value = 0.31
eer_th = 0.35
roc_auc = 0.94
eer_value = 0.13
eer_th = 0.20
roc_auc = 0.61
eer_value = 0.38
eer_th = 0.49
roc_auc = 0.68
eer_value = 0.34
eer_th = 0.42
roc_auc = 0.88
eer_value = 0.17
eer_th = 0.42
roc_auc = 0.70
eer_value = 0.33
eer_th = 0.36
roc_auc = 0.68
eer_value = 0.35
eer_th = 0.37
```

Figure 22: EER and Corresponding Threshold Values

We performed further analysis on the Fisher score of some features which were producing result which were contradictory our expectations. After reviewing our data processing scripts and manually calculating Fisher score, we found the problems lies in the formatting of OpenDS's output data. Therefore, a script was written to preprocess the data with changes such as truncating certain features to 2-decimal points, changing data scope which is negative to the positive one, as well as coordinate transformation.

```
the fish score of distance travelled is: 0.41870411543
the fish score of average speed is: 0.438899784274
the fish score of std deviation of steer position is: 0.268141796166
the fish score of avg change of brake position is; 0.245608006765
the fish score of avg change of accelerate position is: 0.643654106055
the fish score of avg position_x is: 0.0706782383254
the fish score of avg position_y is: 0.459305279739
the fish score of avg position_z is: 0.079019943169
the fish score of avg rotation_x is: 0.372824539807
the fish score of avg rotation_y is: 0.0628615987508
the fish score of avg rotation_z is: 0.300275450167
the fish score of avg rotation_w is: 0.0600377180026
```

Figure 23: Updated Fisher Scores

As an example of another data processing step, we plotted our drivers' X and Z coordinates to ensure they met our expectations. As shown in Figure 24, the plot of driving coordinates closely matched our designated route in a majority of cases.

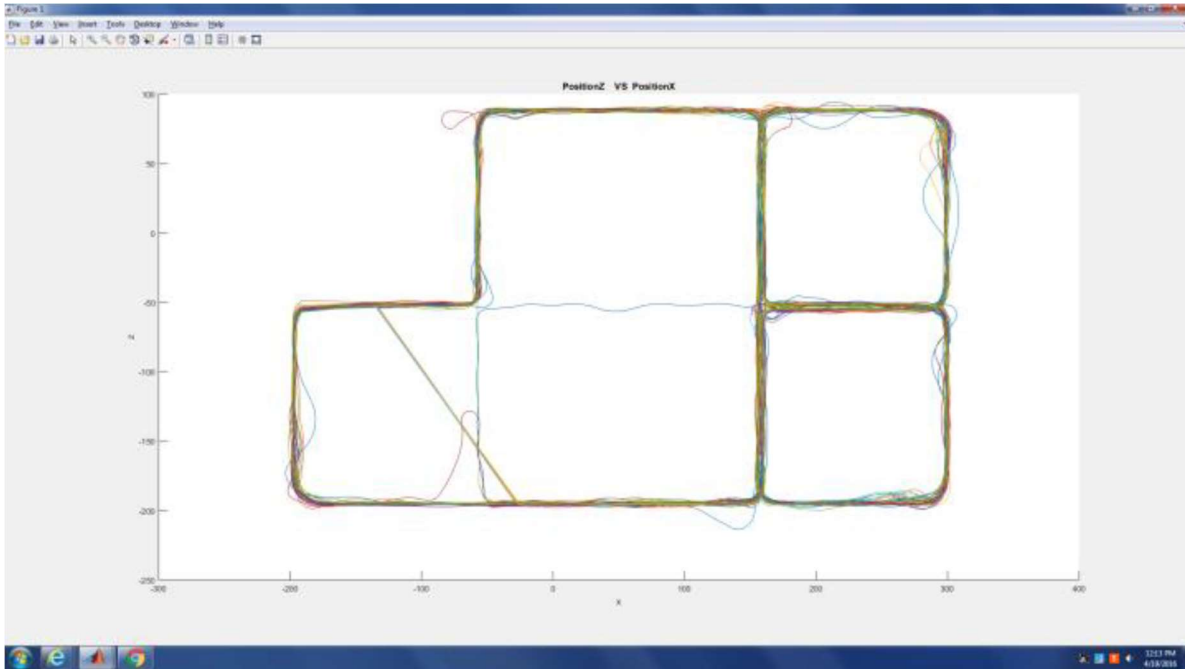


Figure 24: Plot of Position Coordinates for Study Test Runs

Finalized Analysis of Preliminary Study Data

The exploratory analysis presented in the previous section allowed us to derive a deeper understanding of driving behavior. We utilized this experience to determine how the collected data should be sanitized, pre-processed, organized into high level features, and modeled in order to accurately derive conclusions about driver identity from a set of non-intrusive features. This section describes the finalized results we were able to derive from our observations of users' driving traits. The analysis we performed on our preliminary simulated driving dataset concludes that it is possible to authenticate a driver in less than 2.5 minutes with 95% confidence and a false positive rate of less than once per driving day using only non-invasive sensors.

We chose five non-invasive features as potentially discriminative characteristics extracted from measurements to capture each subject's unique driving pattern: (1) Euclidean distance traveled, (2) average vehicle speed, (3) the standard deviation of the steering wheel position, (4) the average change of brake pedal position, and (5) the average change of gas pedal position. These features were selected for a combination of practical and theoretical considerations. The OpenDS driving simulation software's logging functionality allowed easy access to low level driver tracking details from which each of these features could be derived. Additionally, we felt that these features would be good candidates for capturing driving activity because they covered a wide range of the various controls one must utilize in order to drive proficiently. We also included the vehicle's location coordinates and rotation in order to provide a basis for comparison with our derived features.

We applied a variety of different machine learning algorithms to our collected feature set in order to assess their ability to discern between individuals as they operated a vehicle. We implemented Matlab scripts to apply 3 different supervised learning algorithms to our data: Decision Trees, Support Vector Machine (SVM), and k-Nearest Neighbor (kNN). We also attempted to apply a boosting to increase our classification accuracy: instead of using all features for classification, various subsets of features are used and classification is determined by which grouping is indicated by a majority of the learners. In the case of k-Nearest Neighbor, this is referred to as the random subspace method [29]. For decision trees, this results in an approach known as Random Forests [30].

We explored the application of multiclass modeling processes to the task of driver identification in order to perform a comparison of alternative modeling techniques. In practice, however, a particular driver's vehicle would not have access to information regarding how other drivers operate their vehicles. Furthermore, even if this information was available, it would be very difficult to scale to all users in a busy driving area. For this reason, one class models, which require only positive samples of an authentic driver's behavior patterns, are much better suited to the context of driver authentication. To see how a one-class model would perform with respect to our driving features, we applied a one-class Support Vector Machine (oc-SVM) to our data to create a separate model for each user. Each user's model was trained with 80% of their driving samples, while the remaining 20% of driving logs were reserved for testing. Each user's model

was trained only using their driving data, but driving data from all ten subjects was used to test the classification accuracy.

Since ROC curves express binary classification information, our curve plots were performed on a user by user basis. Figure 25 shows ROC curves which result from multiclass SVM classification for all ten participants. In addition to the ROC curves, Figure 25 also provides AUC values for each study participant that resulted from training an SVM on our driving features with a Polynomial kernel and applying 10-fold cross validation; the average AUC across all users is 0.8138.

Figure 26 presents a DET curve for multiclass SVM classification averaged across all users. Our multiclass driver detection SVM was capable of authenticating drivers with an EER of 24.9%.

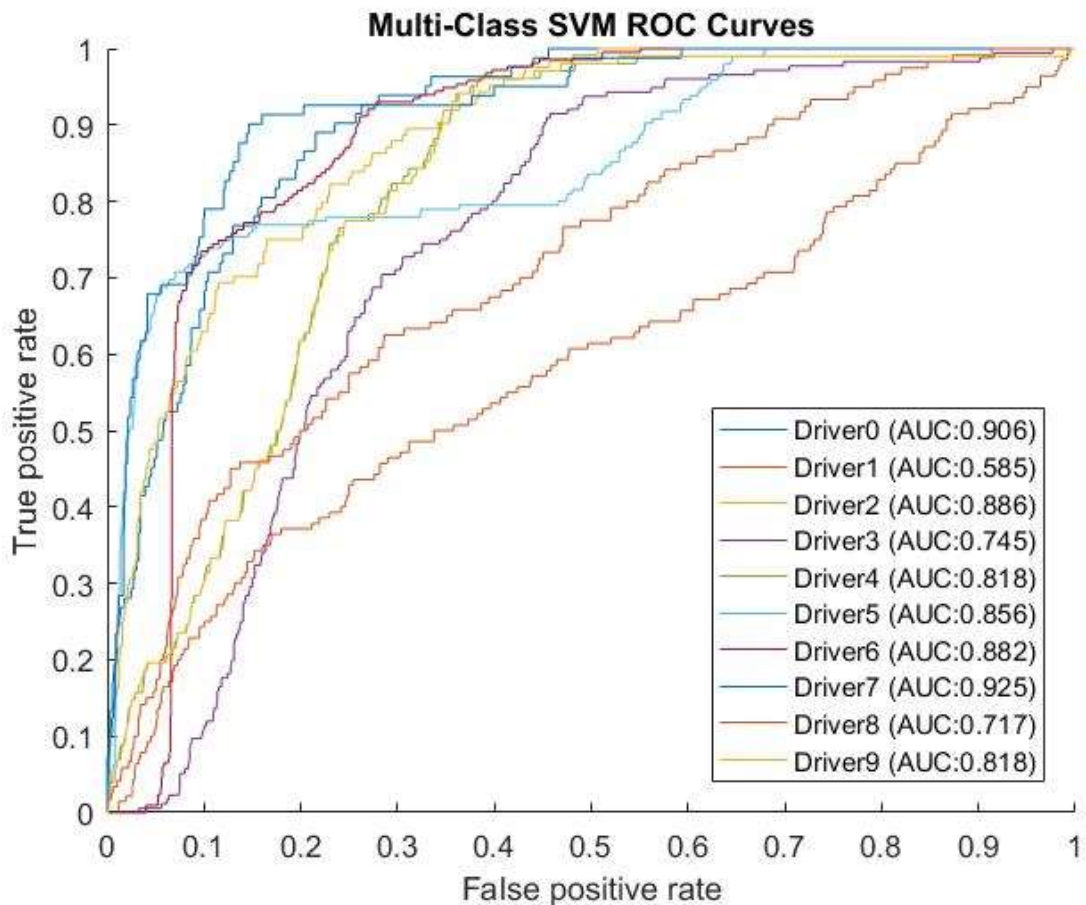


Figure 25: ROC Curves for Multi-Class SVM Classification of All Study Participants

The per-user AUC values were averaged together to produce an AUC for each classifier, which is displayed in Figure 27 as a box-and-whisker plot. Decision trees displayed a similar classification performance to SVMs on average, resulting in AUC values of 0.902 and 0.91 respectively. However, decision

trees also fell into the lower quartile for a larger portion of users. The kNN approach displayed the worst overall performance, with an average AUC of 0.781. Though boosting did increase the average AUC to 0.8 for kNN, it also resulted in very poor performance for some users, with an AUC value as low as 0.35. Though boosting decision trees to produce a random forest ensemble learner increased the average AUC to 0.932, we believe this gain may be due in part to overfitting on our relatively small sample size.

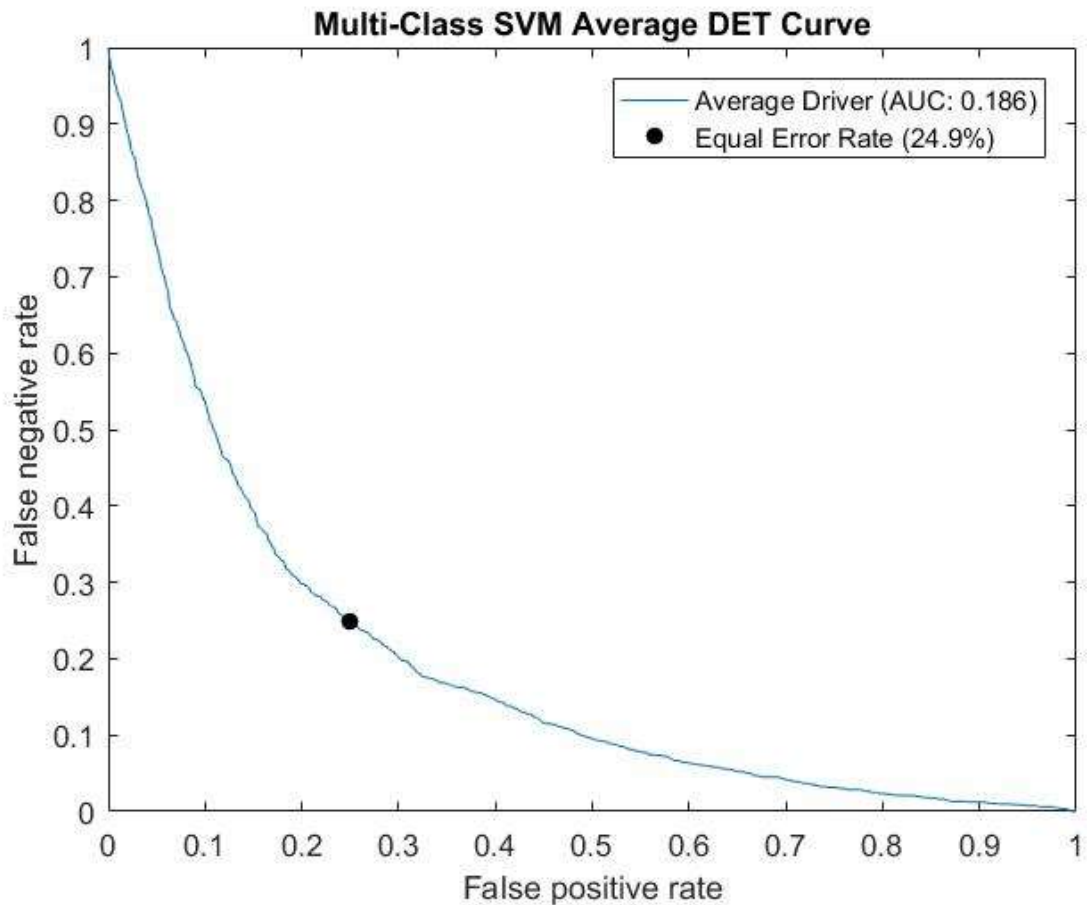


Figure 26: Average DET Curve for Multi-Class SVM Classification

The Fisher scores for both the average values of our raw driving data and our derived features are shown in Table 4. The features are listed in ascending order by their Fisher score. These scores capture the ratio of between-class and within class variance, which essentially means that higher ranked features are more consistent for a particular driver over time, and more unique between different drivers. From Table I, it is easy to see that our derived values have more discriminative power than the “raw” rotational and coordinate values collected from the simulator; recall that the coordinates are roughly equivalent to geolocation information. The third column of Table 4, labeled “Classification Contribution,” contains another measurement of the suitability of each feature to

the task of driver modeling. This value is obtained by removing the feature from our SVM modeling process and observing the new true positive classification accuracy. The new TPR is subtracted from the original to obtain the classification contribution.

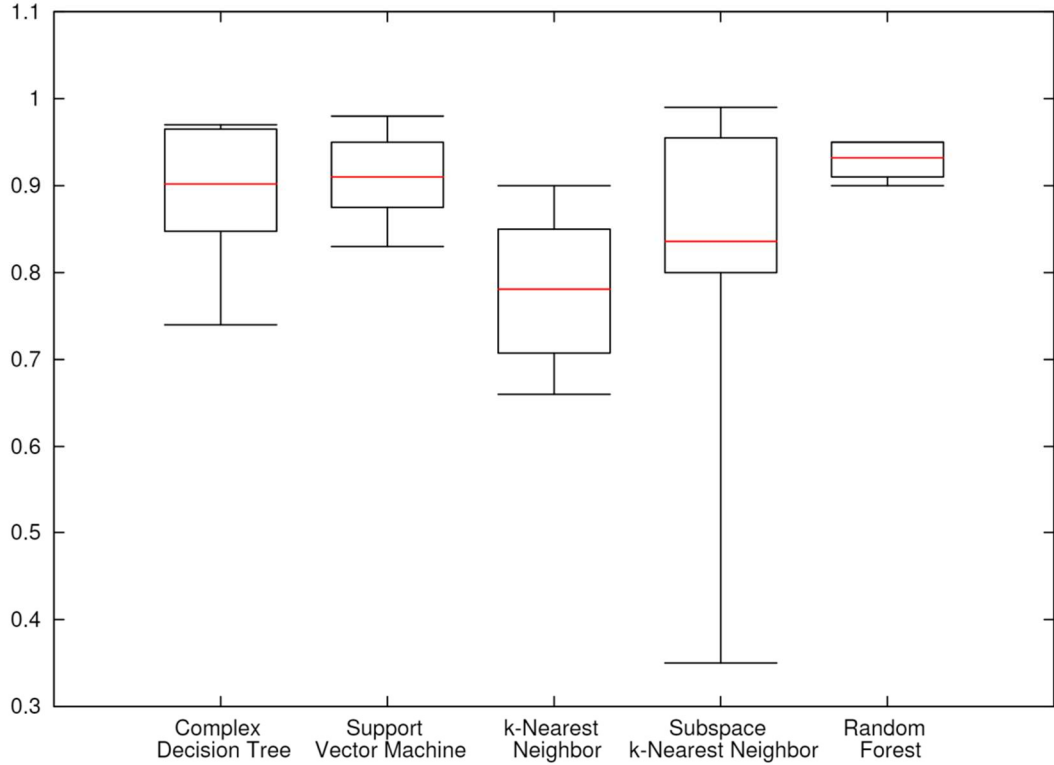


Figure 27: Multi-Class Model AUC Comparison

Feature	Fisher Score	Classification Contribution
Average Change in Accelerator Pressure	0.122	3.84%
Distance Traveled	0.101	0.23%
Average Speed	0.082	0.26%
Average Change in Brake Pressure	0.052	1.76%
Standard Deviation of Steering Position	0.039	0.60%
Average X Axis Position	0.037	0.46%
Average Z Axis Position	0.022	1.32%
Average Y Axis Position	0.020	0.00%
Average Z Axis Rotation	0.019	0.00%
Average Y Axis Rotation	0.018	-0.46%
Average X Axis Rotation	0.017	-0.03%
Average W Axis Rotation	0.014	0.07%

Table 4: Fisher Scores for Driving Features

As shown in Table 4, the classification contributions are correlated with Fisher values, with higher valued features having larger contributions to the overall modeling process. A notable exception is the Distance Traveled and Average Speed. The reason why SVM modeling retains its classification accuracy when either of these features is removed is due to the fact that they are highly correlated, thus removing one or the other only removes a small amount of information from our models due to the redundancy in these features. Some measurements, particularly the Y axis position, which represents elevation, are consistent across all users. The Fisher score and classification contributions confirm that these features are not discriminative. Including the Y and W rotational axes in our model even turned out to be detrimental to classification. We believe the reason for these features to have non-zero Fisher scores is due to noise in the underlying data introduced by very small variations in the data logged by the simulation.

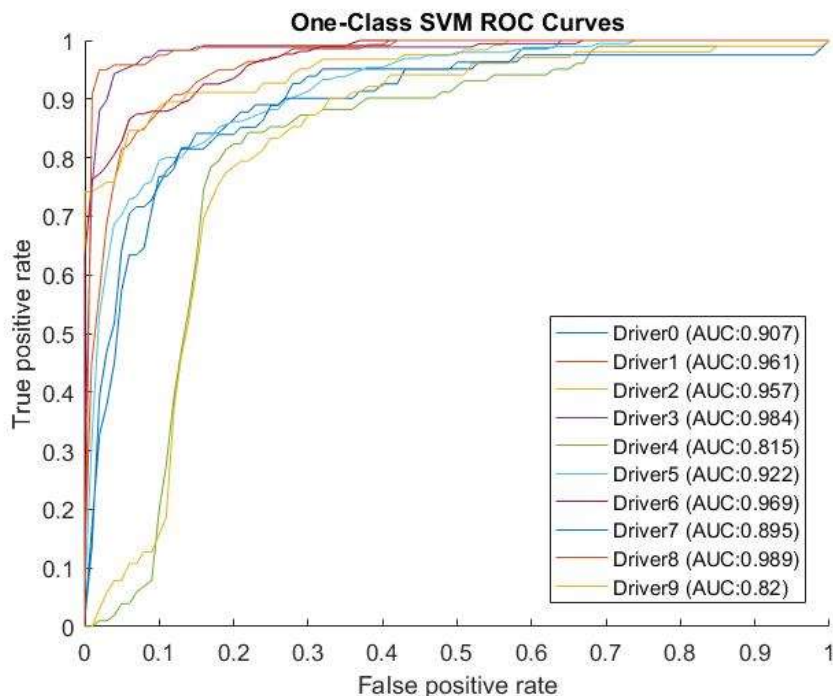


Figure 28: ROC Curves for One Class SVM Classification of All Study Participants

Figure 28 shows the ROC curves achieved for each driver using the per-user oc-SVM modeling process, while Figure 29 displays the average DET curve for all users. The oc-SVM achieved an average AUC value of 0.9219 and an EER of 14.7%, which represents an improvement over multi-class modeling in terms of both metrics. This is due in part to the fact that the one-class modeling process is asking a less specific question than the multi-class example. The multiclass model is essentially asking “Are you driver X or driver Y,” while the one-class model asks “Are you driver X or a different driver?”

We selected one false positive per driving day as a reasonable performance target. According to a recent study, the typical driver spends 46 minutes in his or her vehicle per day on average. Our model works using a 10 second sampling interval. Thus, to achieve a false alarm occurrence rate of one per 46 driving minutes would require a FPR of 0.362%.

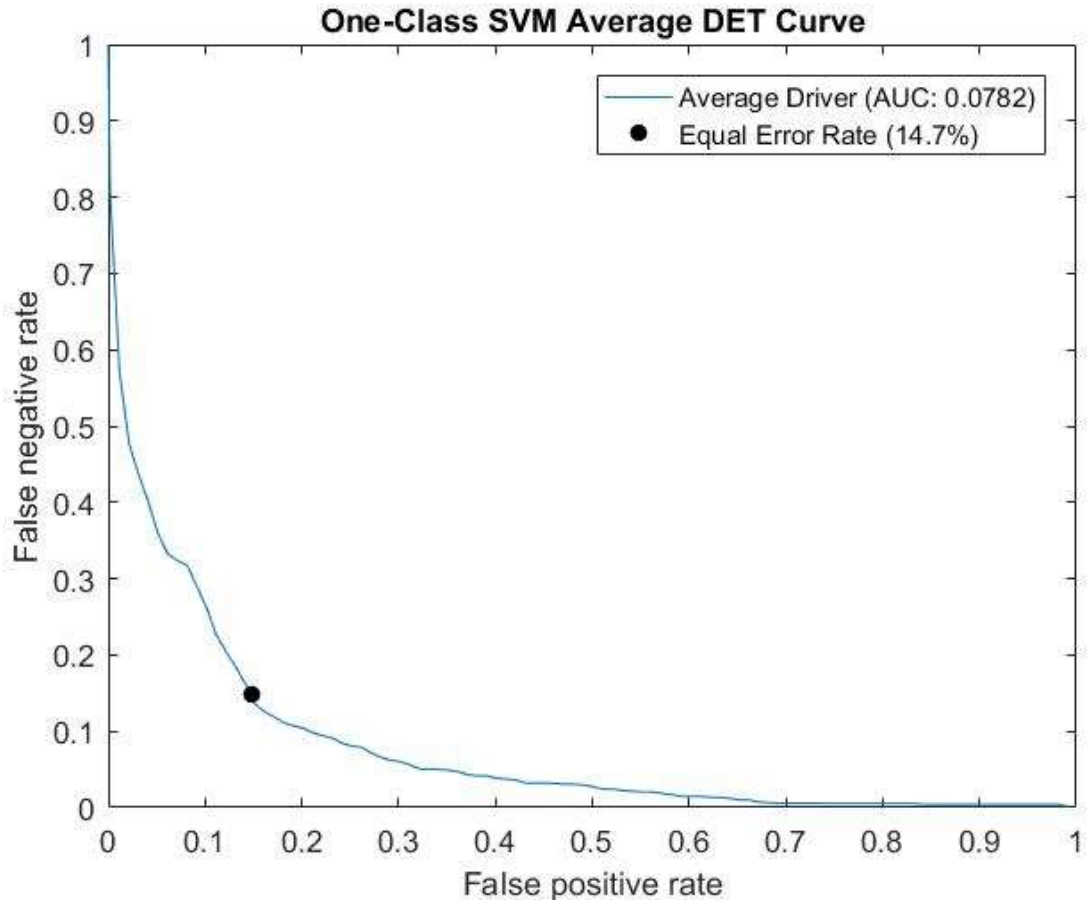


Figure 29: Average DET Curve for One Class SVM Classification

At this very restrictive FP rate, our oc-SVM model of driving behavior is capable of performing driver classification with a detection rate of 19.5%. This means that in any one particular time window there is a 80.5% change of an illicit driver avoiding detection. The following equations calculate how many 10 second samples would be required to ensure that any unauthorized drivers are detected with at least 95% confidence:

$$\begin{aligned}
 0.805^x &< 0.05 \\
 x < \log(0.05) / \log(0.805) \\
 x &< 13.81
 \end{aligned}$$

From an operational perspective our oc-SVM model of driving behavior can successfully detect illicit vehicle usage with 95% accuracy after 14 samples, or 2 minutes and 20 seconds of driver data collection, while keeping false alerts to once per driving day at

most. These results show that identifying drivers is feasible in practice with active behavior modeling without incurring any significant computation or high false positives.

Simulation Enhancements for Broader Driving Behavior Study:

We applied lessons learned from our preliminary data collection to determine how to improve our experiment prior to running it with a broader population. Our simulation was enhanced with additional driving challenges which we felt would elicit potentially discriminative responses from users.

We were able to successfully manipulated the scenario in order to add objects and make our driving tasks as realistic as possible. Our simulation now allows us to measure user behavior in the presence of interaction models and triggered events such as additional traffic, vehicles, pedestrians, and traffic lights.

We attempted to add complexity to our simulation's driving course of map by using the build-it "City" model which is included with OpenDS. Unfortunately this model caused performance issues in OpenDS which affected its responsiveness, making it inappropriate for use in our studies. Efforts were also put into updating the model structure but this did not result in a substantial performance improvement either.



Figure 30: OpenDS Screenshot with Side and Rearview Mirrors Included

Additional simulation changes which we added in preparation for our next round of data collection include blinkers, side and back mirrors. Blinker usage timing is another feature for potential modeling. We believe that both changes will help provide participants with a more realistic driving experience.



Figure 31: OpenDS Screenshot with Left Blinker Enabled

New Features for Extended Study:

Extracting new attributes can clearly have a significant impact on analysis and modeling. Recall that the core aim of our project is to provide security and privacy to users via a sensing system that is decoupled from a vehicle's critical networks. To this end, we altered the code of the OpenDS driving simulation software to allow for the collection of new attributes including lane position, blinker usage, and distance between cars. Collecting lane position data allows information on how often a driver deviates from his or her lane, which could be a potentially discriminative feature of driving behavior.

The following diagram shows the position of the entire lane ($\text{currentLane} = 0$). On the left hand side, the driver is driving on the $\text{LaneID} = 2$ side of the lane. In the right hand diagram, we see that the min is actually the center of the whole lane and the max represents the edge of the lane. Drivers are only driving in the correct lane when they are driving in the positive section of the laneID and under the lane width.

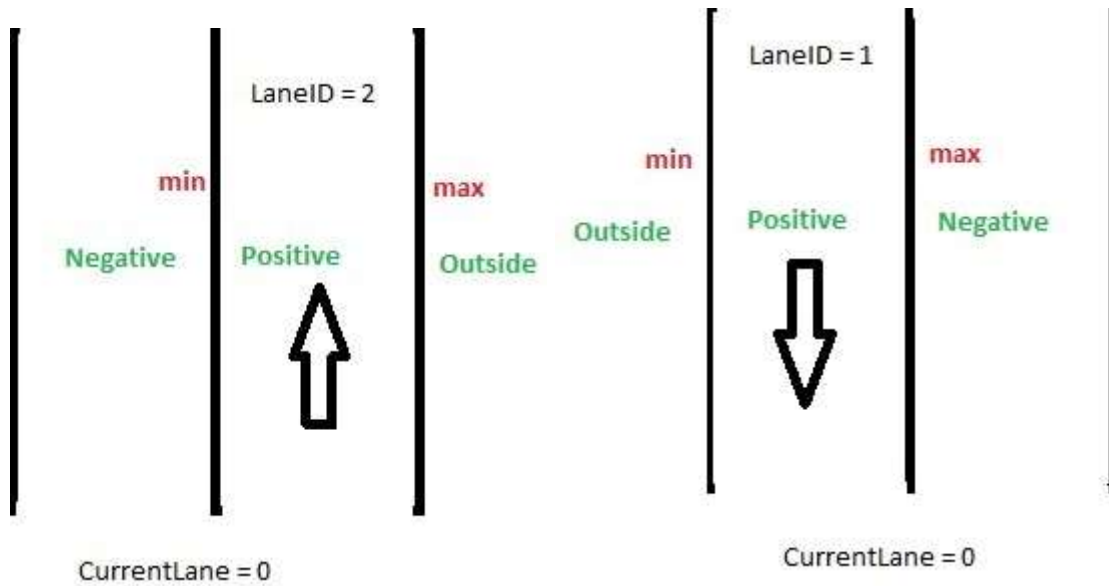


Figure 32: Measuring a Driver's Lane Position

Another driving attribute which we are considering is the distance between vehicles. This property is typically used to help determine if a driver is safe or unsafe. If driver is close to leading vehicles, the driver is in rush. If the driver is close to trailing cars, on the other hand, this implies that he or she is progressing slowly compared to surrounding traffic. Figure 33 shows a raw driving log sample which includes our new features of potential interest.

Used	Forn	Position_	Position_	Position_	Rotation_	Rotation_	Rotation_	Rotation_	Speed (kr	Steering \	Gas Pedal	Brake Pec	Engine Ru	Distance	Distance	Current R	LaneId	Min Lane	Max Lane Point
146369355	-135.93	-0.469	-51.611	9.00E-04	-0.7031	8.00E-04	0.7111	42.13	0.00186	0.718746	0	TRUE	-1	-1	0	2	-53.54	-48.35	
146369355	-135.555	-0.469	-51.616	9.00E-04	-0.703	8.00E-04	0.7112	42.31	0.00186	0.734371	0	TRUE	-1	-1	0	2	-53.54	-48.35	
146369355	-135.178	-0.469	-51.62	9.00E-04	-0.7029	8.00E-04	0.7113	42.49	2.20E-04	0.734371	0	TRUE	-1	-1	0	2	-53.54	-48.35	
146369355	-134.799	-0.469	-51.625	9.00E-04	-0.7028	9.00E-04	0.7114	42.67	2.20E-04	0.742184	0	TRUE	-1	-1	0	2	-53.54	-48.35	
146369355	-134.419	-0.469	-51.63	9.00E-04	-0.7028	9.00E-04	0.7114	42.86	-0.00133	0.742184	0	TRUE	-1	-1	0	2	-53.54	-48.35	
146369355	-134.037	-0.469	-51.634	8.00E-04	-0.7028	9.00E-04	0.7114	43.05	-0.00133	0.749996	0	TRUE	-1	-1	0	2	-53.54	-48.35	
146369355	-133.653	-0.469	-51.639	8.00E-04	-0.7029	9.00E-04	0.7113	43.24	-0.00133	0.749996	0	TRUE	-1	-1	0	2	-53.54	-48.35	
146369355	-133.268	-0.469	-51.643	8.00E-04	-0.7029	9.00E-04	0.7113	43.43	-0.00133	0.773434	0	TRUE	-1	-1	0	2	-53.54	-48.35	
146369355	-132.88	-0.469	-51.647	9.00E-04	-0.703	9.00E-04	0.7112	43.63	-0.00218	0.773434	0	TRUE	-1	-1	0	2	-53.54	-48.35	
146369355	-132.589	-0.469	-51.651	9.00E-04	-0.7031	9.00E-04	0.7111	43.78	-0.00298	0.781247	0	TRUE	-1	-1	0	2	-53.54	-48.35	
146369355	-132.199	-0.469	-51.655	8.00E-04	-0.7032	9.00E-04	0.711	43.99	-0.00298	0.789059	0	TRUE	-1	-1	0	2	-53.54	-48.35	
146369355	-131.806	-0.469	-51.658	8.00E-04	-0.7033	9.00E-04	0.7109	44.2	-0.00298	0.789059	0	TRUE	-1	-1	0	2	-53.54	-48.35	
146369355	-131.412	-0.469	-51.662	8.00E-04	-0.7035	9.00E-04	0.7107	44.42	-0.00462	0.789059	0	TRUE	-1	-1	0	2	-53.54	-48.35	
146369355	-131.115	-0.469	-51.665	8.00E-04	-0.7036	9.00E-04	0.7106	44.58	-0.00462	0.804685	0	TRUE	-1	-1	0	2	-53.54	-48.35	
146369355	-130.817	-0.469	-51.667	8.00E-04	-0.7038	9.00E-04	0.7104	44.74	-0.00542	0.804685	0	TRUE	-1	-1	0	2	-53.54	-48.35	
146369355	-130.518	-0.469	-51.669	8.00E-04	-0.7039	9.00E-04	0.7103	44.91	-0.00707	0.804685	0	TRUE	-1	-1	0	2	-53.54	-48.35	
146369355	-130.118	-0.469	-51.672	8.00E-04	-0.7042	9.00E-04	0.71	45.13	-0.00786	0.812497	0	TRUE	-1	-1	0	2	-53.54	-48.35	
146369355	-129.715	-0.469	-51.674	8.00E-04	-0.7046	9.00E-04	0.7096	45.36	-0.00951	0.82031	0	TRUE	-1	-1	0	2	-53.54	-48.35	
146369355	-129.311	-0.47	-51.675	8.00E-04	-0.705	0.001	0.7092	45.59	-0.00951	0.812497	0	TRUE	-1	-1	0	2	-53.54	-48.35	

Figure 33: Raw OpenDS Data Sample including New Features

Large Scale Study of Simulated Driving Behavior

Following our successful preliminary data collection with 10 users, our aim was to gather data of almost from a larger pool of participants with more features as they perform a more realistic driving task. We secured gift cards to provide to participants as reimbursement. Furthermore, we prepared several instruments to aid in our study and data collection efforts. These include a survey to capture demographic information and

driving experience and framing materials to provide extrinsic motivation and emotionally prime our users for their driving task to provide more depth to the data we collect during our more full-featured efforts.

We were successful in meeting our targeted pool size of 50 participants. Aside from enlarging our sample size to provide more generalizable results, we also collected data from new modalities which we felt may help capture unique attributes of an individual's driving behavior. Our research objective in doing so was to identify salient features which would improve upon the accuracy of the driver modeling, and therefore authentication, process.

Our previous study was smaller in scope and collected a set of features which were designed to be minimally invasive. For our larger scale study, we decided to broaden our collection efforts to include measurements which were not directly connected to vehicle controls.

Conclusions and Recommendations

This project resulted in a novel approach to driver authentication based on users' innate driving habits which are observable as vehicles are in operation. Two studies were performed, one with 10 human subjects and one with 50, in which they completed a simulated driving task while recording their activity. We successfully constructed models of driving activity via extracted features, namely pedal control, steering, speed, and distance traveled. The results of our experiment and modeling effort yield an average EER of 14.7%, implying a time-to-detection of 2 minutes and 20 seconds at 95% confidence with at most one false alert per day of driving. These results provide evidence in support of our hypothesis that drivers can be identified by observing the manner in which a vehicle is operated.

Implementation and Training

The results of this research have resulted in the following three publications:

1. Security, Trust, and Privacy for Cloud Computing in Transportation Cyber-Physical Systems

Wenjia Li, Jonathan Voris, and N. Sertac Artan. Data Security in Cloud Computing, The Institution of Engineering and Technology (IET), 2016.

2. Driver Identification and Authentication with Active Behavior Modeling

Angela Burton, Tapan Parikh, Shannon Mascarenhas, Jue Zhang, Jonathan Voris, N. Sertac Artan, and Wenjia Li. 1st International Workshop on Green ICT and Smart Networking (GISN) co-located with the 12th International Conference on Network and Service Management (CNSM), 2016.

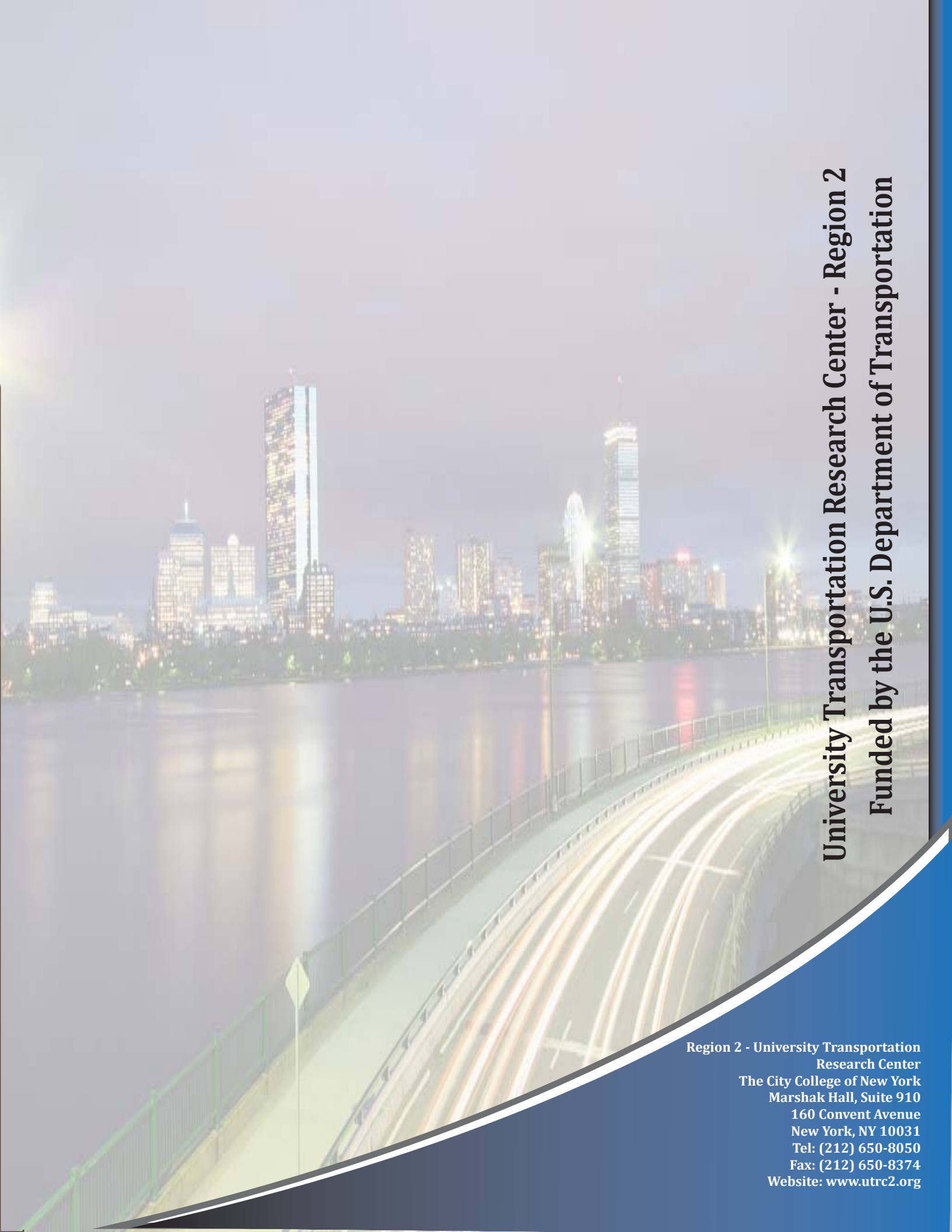
3. Utilizing Behind-the-Wheel Behavior for Driver Authentication

Jonathan Voris, N. Sertac Artan, and Wenjia Li. Transportation Technology Symposium: Innovative Mobility Solutions, 2016.

References:

- [1] A. Schmitt, "Report: Traffic studies systematically overstate benefits of road projects," <http://usa.streetsblog.org/2012/07/06/report-trafficstudies-systematically-overstate-the-benefits-of-road-projects>, 2012.
- [2] A. for Toll-Free Interstates, "Studying and forecasting tolls is inefficient, unproductive and expensive," <http://www.tollfreeinterstates.com/resources>, 2016.
- [3] M. S. Nicolaisen and P. A. Driscoll, "Ex-post evaluations of demand forecast accuracy: A literature review," *Transport Reviews*, vol. 34, no. 4, 2014.
- [4] R. Bishop, *Intelligent vehicle technology and trends*, 2005.
- [5] E. C. for Transportation, "Emerging technology trends in transportation," <https://www.enotrans.org/wp-content/uploads/EmergingTech.v13.pdf>, 2016.
- [6] Progressive Corporation, "Snapshot Common Questions," <http://www.progressive.com/auto/snapshot-common-questions>, 2014.
- [7] C. Troncoso, G. Danezis, E. Kosta, J. Balasch, and B. Preneel, "Pripayd: Privacy-friendly pay-as-you-drive insurance," *Dependable and Secure Computing, IEEE Transactions on*, vol. 8, no. 5, pp. 742–755, 2011.
- [8] K. Ozbay, "Using big data to identify hotspots of pedestrian crashes in manhattan," in *UTRC Ground Transportation Technology Symposium*, 2014
- [9] S. Woo, J. Hyo, and D. Lee. "A Practical Wireless Attack on the Connected Car and Security Protocol for In-Vehicle CAN." *IEEE Transactions on Intelligent Transportation Systems*, 2015.
- [10] Ian Foster, Andrew Prudhomme, Karl Koscher and Stefan Savage. "Fast and Vulnerable: A Story of Telematic Failures." 9th USENIX Workshop on Offensive Technologies (WOOT 15), 2015.
- [11] Debiao He; Zeadally, S.; Baowen Xu; Xinyi Huang, "An Efficient Identity-Based Conditional Privacy-Preserving Authentication Scheme for Vehicular Ad Hoc Networks," *IEEE Transactions on Information Forensics and Security*, vol.10, no.12, pp.2681-2691, Dec. 2015.
- [12] Bittl, Sebastian; Gonzalez, Arturo A.; Myrtus, Matthias; Beckmann, Hanno; Sailer, Stefan; Eissfeller, Bernd, "Emerging attacks on VANET security based on GPS Time Spoofing," in *Proceedings of 2015 IEEE Conference on Communications and Network Security (IEEE CNS 2015)*, vol., no., pp.344-352, 28-30 Sept. 2015.
- [13] E. Markey, "Tracking & Hacking: Security & Privacy Gaps Put American Drivers at Risk," US Senate, 2015.
- [14] C. Miller and C. Valasek, "Adventures in Automotive Networks and Control Units," DEFCON, 2013.
- [15] A. Cui, M. Costello, and S. J. Stolfo, "When Firmware Modifications Attack: A Case Study of Embedded Exploitation," NDSS, 2013.
- [16] C. Miller and C. Valasek, "Remote Exploitation of an Unaltered Passenger Vehicle," Black Hat USA, 2015.
- [17] XSens MTi-1 Series Module [Available Online:] <https://www.xsens.com/products/mti-1-series/>
- [18] Conductive Rubber Cord Stretch Sensor + extras! [Available online:] <https://www.adafruit.com/products/519>

- [19] Standex Meder Electronics Seat Belt Sensor. [Available online:] <https://standexelectronics.com/applications-markets/seat-belt-sensor/>
- [20] Michigan Scientific Corporation Model BPFT2 Low Profile Brake Pedal Force Transducer. [Available online:] <http://www.michsci.com/Products/transducers/bpft2.htm>
- [21] Tactilus® Automotive Occupant Pressure Measurement System. [Available Online:] <http://www.sensorprod.com/automotive-pressure-mapping.php>
- [22] GTech Pro SS Fanatic Pressure Series. [Available online:] <http://www.gtechprostore.com/cgi-bin/shopper.cgi?preadd=action&key=0500501>
- [23] M. Aly. "Real time Detection of Lane Markers in Urban Streets." *IEEE Intelligent Vehicles Symposium, Eindhoven, The Netherlands*, 2008.
- [24] Emotiv EPOC Headset: High resolution, multi-channel, portable EEG system. [Available online:] <https://emotiv.com/store/epoc-detail/>
- [25] QardioCore Continuous EKG/ECG Monitor [Available online:] https://www.getqardio.com/store/index.php#category_22
- [26] OpenDS Community Video Tutorials [Available online:] <https://opends.de/community/community-videos/118-openstreetmap-to-ogre-using-blender-to-be-used-in-opends>
- [27] JMonkeyEngine Forums: Can I convert ogremesh xml to j3o WITHOUT using the SDK? [Available online:] <http://hub.jmonkeyengine.org/t/can-i-convert-ogremesh-xml-to-j3o-without-using-the-sdk/20937>
- [28] C. Chang and C. Lin. "LIBSVM: A Library for Support Vector Machines." [Available online:] <https://www.csie.ntu.edu.tw/~cjlin/libsvm>, 2016.
- [29] T. Ho. "The random subspace method for constructing decision forests." *IEEE transactions on pattern analysis and machine intelligence*, 1998.
- [30] L. Breiman. "Random forests." *Machine learning*, 2001.

A long-exposure photograph of a city skyline at night, viewed from across a body of water. The skyline is filled with illuminated skyscrapers, including a prominent one with a blue facade. In the foreground, a bridge or highway spans the water, with light trails from moving vehicles creating a sense of motion. The overall scene is a blend of urban architecture and transportation infrastructure.

University Transportation Research Center - Region 2
Funded by the U.S. Department of Transportation

**Region 2 - University Transportation
Research Center**
The City College of New York
Marshak Hall, Suite 910
160 Convent Avenue
New York, NY 10031
Tel: (212) 650-8050
Fax: (212) 650-8374
Website: www.utrc2.org