Vehicle Self-Surveillance: Sensor-Enabled Automatic Driver Recognition

Ian D. Markwood
University of South Florida
4202 East Fowler Avenue
Tampa, FL 33620
imarkwood@mail.usf.edu

Yao Liu
University of South Florida
4202 East Fowler Avenue
Tampa, FL 33620
yliu@cse.usf.edu

ABSTRACT

Motor vehicles are widely used, quite valuable, and often targeted for theft. Preventive measures include car alarms, proximity control, and physical locks, which can be bypassed if the car is left unlocked, or if the thief obtains the keys. Reactive strategies like cameras, motion detectors, human patrolling, and GPS tracking can monitor a vehicle, but may not detect car thefts in a timely manner. We propose a fast automatic driver recognition system that identifies unauthorized drivers while overcoming the drawbacks of previous approaches. We factor drivers' trips into elemental driving events, from which we extract their driving preference features that cannot be exactly reproduced by a thief driving away in the stolen car. We performed real world evaluation using the driving data collected from 31 volunteers. Experiment results show we can distinguish the current driver as the owner with 97% accuracy, while preventing impersonation 91% of the time.

CCS Concepts

•**Security and privacy** → **Authentication;** *Access control; Authorization; Usability in security and privacy;*

Keywords

behavioral biometrics; authentication; driving behavior modeling

1. INTRODUCTION

Motor vehicles are an integral part of modern society, providing for the largest portion of transportation enjoyed by individuals in developed countries. As such, they are also quite valuable and the target of theft. In 2012, the Federal Bureau of Investigations reported an estimated 721,053 vehicles were stolen in the United States, 73.9% of them are automobiles, costing more than $4.3 billion. Indeed, only 11.9% of motor vehicle theft was cleared that year [1]. Losing a car not only causes property loss for the car owner, but could also trigger lawsuits against the car owner. For example, a legal case was reported by the media in 2009 where the owner of a stolen car was sued for the deaths of two teenagers in a fatal hit-and-run accident [2].

In dealing with car thefts, both preventative and reactive strategies have been used. Preventative measures like car alarms, proximity control, and physical locking devices intend to prevent access to a vehicle by unauthorized users. Typical preventative methods can alert passers-by or the car owner to a break-in, prevent ignition if the owner's key fob is not close to the vehicle, or deter the theft by adding extra facilities like wheel locks. However, these methods can be bypassed if circumstances or human error result in the car being left unlocked, or the car keys are obtained by the thief. Moreover, preventative methods are entirely ineffectual once the car is stolen.

Reactive strategies monitor a vehicle or item of interest but do not interact directly in its security. For example, camera feeds must be continuously observed by humans to detect theft, which is impractical for private security, so they are mostly reviewed after the theft is discovered to identify the thief. It is normally infeasible for security guards in apartment complexes, gated communities, or college or business campuses to survey the entire grounds at once or recognize a driver as having unauthorized access to one of many vehicles there. Other monitoring concepts such as motion sensors can suffer high false alarm rates for vehicles parked outside amidst high human traffic. Finally, GPS-based car services such as OnStar [3] or LoJack [4] may be used to track a vehicle but again require the knowledge that it is stolen. These assorted weaknesses of reactive strategies can result in thefts that go unnoticed for hours. For example, cars parked in neighborhoods at night may not be identified as stolen until the next morning. Allowing the thief control of a vehicle for a long time may significantly reduce the chance of recovering it due to an increase in the risk of the car being involved in a crime or disassembled in chop shops.

The aforementioned approaches have the common underlying limitation of reliance on the infallibility of the user in handling the vehicle's security (never leaving the car unlocked and unattended, for example) and the timely discovery of its potential theft. If both of these dependencies are violated, these measures are completely defeated. In fact, the United States National Highway Traffic Safety Administration reports "40-50% of vehicle theft is due to driver error" [5]. The ideal design of a security mechanism must forgo assuming these impossibilities and instead detect and notify the owner of the car theft as early as possible.

ASIA CCS '16, May 30-June 03, 2016, Xi'an, China

DOI: http://dx.doi.org/10.1145/2897845.2897917

An intuitive way to achieve this goal is to enforce passwords in car systems, with incorrect password inputs triggering an alert to the car owner. However, this strategy has its own drawbacks. First, the dashboards and control panels of all existing car models are located in the front of a car. It is therefore easy to carry out shoulder surfing attacks where passengers may purposefully or inadvertently catch the password typed by the driver. Secondly, as indicated in [6], passwords are susceptible to smudge attacks, wherein the attacker may put together the oil residue of recent finger smudges left on the touch screen to infer the password. A third issue is that passwords can reduce the system's usability, because the car owner has to memorize a password and change it periodically to maintain security. Then, whenever the user needs to drive the car, he has to provide a correct password, which, because inconvenient, can motivate a user to disable the password function.

In contrast, we propose an automatic driver recognition system that does not require any interaction between the user and the car after the initial setup. Our basic idea is to utilize driving behavior, which cannot be precisely reproduced by a thief driving away in the stolen car. This precludes the necessity for human discovery of theft, instead monitoring for unauthorized use continually. Furthermore, authorization is passive, so a user is not required to perform any additional, potentially annoying action upon car entry before every drive. This is important due to the propensity of users to brush aside new security measures which require additional effort.

We identified two guiding principles upon which we built our system. First, we should observe the existence of personal and quantifiable driving preferences. But more importantly, drivers have varying amounts of control over how much of their own driving preferences they can apply to each of the several driving events involved with any use of a car. Secondly, drivers behave differently when traveling at different magnitudes of speed. These principles will be explained in further detail later, as we evaluate the efficacy of potential behavioral features, which said principles inspired. We use the resulting effective features to classify between users, testing periodically over the beginning of each drive until a prediction is made authorizing or un-authorizing the driver. Unobservable, efficient, and easily applied to existing vehicles, this system can identify whether or not the driver of a car is its owner, so that the owner may be quickly alerted to a theft.

We performed real world evaluation using the driving data collected from 31 volunteers. Our experiment results show that the proposed system is suitable for driver identification and thereby authentication. It is capable of self-identification, that is, successfully distinguishing that the current driver is the car owner, with 97% accuracy, while also preventing impersonation 91% of the time. We show the effects of a varying training dataset size, finding that at minimum 25 minutes of city driving time is necessary in training to provide desirable accuracy in driver recognition. Likewise, the required testing time is demonstrated to be within 25 minutes of city driving.

This paper has the following contributions: (1) we propose an automatic driver recognition system for the fast detection of car thefts; (2) we identify the effective features that reflect the unique driving preferences of a driver; (3) we propose an online testing algorithm that accepts input data as it is collected continuously and outputs a decision quickly; and (4) we implement the proposed system and evaluate the performance on a real-world data set collected from 31 volunteer drivers.

2. RELATED WORK

Related work falls in the following areas.

2.1 Behavioral Authentication of Mobile Devices

The system proposed herein pertains to the field of behavioral biometrics, inspired by similar concepts applied to mobile device security to identify devices as stolen before their owners notice their absence. Integrating physiological biometric sensing technology into phones is expensive and consumes space - both important optimization factors - without providing functionality improvements the typical user would enjoy or wish to pay for, when compared with mobile GPUs enabling more interesting video games. Furthermore, the physical tests required to pass biometric scans can provide good accuracy while being inconvenient and potentially still spoofed. For example, iris scans can be very accurate [7] but require good lighting and can even be spoofed by a high quality picture [8]. Meanwhile, fingerprint scanners are popular and generally effective [9], but the Apple iPhone 5S fingerprint scanner has proved erratic [10], leading users to turn off the feature. Similar trade offs and problems are encountered with applying physiological biometrics to vehicles, so behavioral biometrics employing existing sensors and new software is an attractive alternative for both applications.

Accordingly, Shahzad et al. distinguish several features of swiping gestures on touch screens which allowed them to differentiate between 50 individuals with high accuracy as a protocol for unlocking devices [6]. With a similar method and similar high accuracy, but different application, Li et al. use gesture features to continuously re-authenticate throughout usage of the device, denying further access if the gesture behaviors change suddenly [11]. Several efforts have also been made in recognizing phone owners using their walking signature [12] or by their typing habits [13] as measured by the device's accelerometer.

2.2 Modeling Driving Behavior

For safety applications, recent research has explored the idea of driving behavior models, with individual endeavors focusing on particular portions of driving habits. General human decision making at unsigned intersections has been simulated using hidden Markov models (HMMs), to predict vehicle movements through those intersections and any conflict involved [14]. Typical highway driving has been modeled with probabilistic networks based on relative positions, velocities, and accelerations of surrounding vehicles and also some environmental variables, to inform decision making in autonomous vehicles driving in human traffic [15]. Sathyanarayana et al. illustrate distraction detection applications to their work in drivers' unique route recognition also using HMMs built around traces of their typical maneuvers [16].

Behavior modeling and driver identification in particular has been achieved with a 76.8% rate of correct driver selection [17], for the purpose of tailoring an intelligent transportation system (ITS) to augment the user's driving with adaptive cruise control or lane keeping, depending on how much the user typically needs such assistance. Identifica-

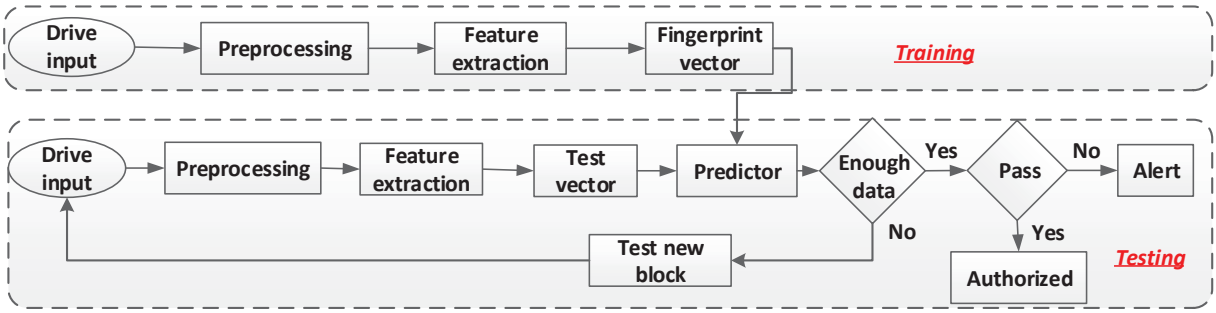

Figure 1: System Design

tion helps here to avoid annoying the driver with too much intervention or, by too little, allowing the driver to incur danger. This is accomplished by performing spectral analysis of Gaussian mixture models consisting of gas pedal depression statistics [17]. This research has been built upon in expanding the application of ITSs in vehicles, as well as work in autonomous vehicle safety, but no such driver identification schemes have been applied to authentication and car theft detection. Moreover the system we discuss involves additional driving features and enjoys higher accuracy.

3. ATTACK MODEL

We assume an attacker has physical access to the vehicle and the ability to start it and drive away. This means the attacker has bypassed any car alarms and physical impediments to accessing the vehicle and has not alerted anyone to the theft. It is then possible to relocate the car to any location to sell, scrap, or use in a crime. Selling and scrapping are similar in that they typically involve anonymizing the car's hardware to prevent its identity being discoverable in the future. Cars used in crimes are often subsequently destroyed to erase any physical evidence the perpetrators would leave inside, leading to insurance expenditure on the part of the owner. Our system is designed to recognize its host vehicle has been stolen, before such events can take place, for a much higher likelihood of retrieval. Without requiring user input or otherwise making itself known, it continuously authenticates the driver as the vehicle is operated. Should the driver fail authentication, the appropriate notifications are made to inform the owner of the car's theft.

4. DESIGN GOALS

The following factors must be in place to ensure the authentication system is effective and usable.

- **Persistence:** Feature data should be collected and analyzed throughout drive time, every instance the vehicle is driven, to consistently protect against theft.
- **Efficiency:** A verdict on the legitimacy of the user should be reached in a short period of time to ensure recovery.
- **Unobservability:** Authentication should not require conscious input from the driver nor should its operation be visible, to avoid annoying legitimate users or alerting thieves.
- **Practicality:** Integration of this system should not require extensive hardware or intensive labor so that it may be employed cheaply and easily.

5. APPROACH

To accomplish these goals, we offer the system design described herein, which makes use of the understanding that driving styles are unique to each person. Social preferences, natural talent and interests, and economic constraints all work to mold the set of places a particular person lives, works, and recreates, and travel amongst these locations forms a body of driving experience unique to that person. This, coupled with the person's typical mental focus while driving and specific thresholds for risk tolerance and patience, results in a large variety of driving styles from which any specific person can be distinguished. For example, every driver has some magnitude of acceleration they typically employ, as well as preferences on braking speed, cornering speed, turn signal use, and coasting. Several of these are encompassed in acceleration (negative and positive) data which we use to create features for analysis.

With this concept, a simple extension of our system allows for multiple legitimate users of one particular car. That is, instead of comparing the current driver's behavior to that of a singular owner, it will compare to each of the owners, to identify a match with any of the authorized users. While not tested in this paper, we can extend this further to address singular users whose behavior may be altered due to extenuating circumstances. Individuals may behave differently while accompanied by certain passengers or while experiencing affecting weather patterns. For example, a teenager may drive more cautiously and conservatively with parents than with friends, or an experienced driver may behave more cautiously during snowy weather than the summer. Of course we expect the former situation to be infrequent as the teenage years comprise a very small portion of a person's driving career, but many drivers do live in locations geographically predisposed to inclement weather. In this case, a single user may train profiles respective to, say, summer (normal) and winter (snow), in order to prevent excessive false alarms. In general most driving is done as a daily routine between the person's home, occupation, and

home again, with static passengers or usually none at all, so for the scope of this research we focus on this scenario. And in this scope, we prove it possible to perform driver identification for authentication.

Our system contains Training and Testing modules and its architecture is presented in Figure 1.

5.1 Training Phase

We must first gather some data of typical driving behavior for a vehicle owner, to be ultimately compared to new data for authentication purposes in future trips. This is the training stage of our system, as illustrated in Figure 1, and requires a minimum driving time to produce good accuracy in authentication. In our experiment, we found that users with up to two hours of driving time entertained the best accuracy. In terms of convenience, the training data does not need to be accumulated in one continuous drive, but training should be accomplished in a short time frame after installation for obvious security reasons. It should also include both city and highway driving for the most accurate cross-section of the driver's behavior. A specific discussion on training time and its effects on system accuracy is presented in Section 7: Evaluation.

The training phase begins by preprocessing all the data collected by the user. This includes removing extreme (noise-induced) values, normalization, and isolating important information that will drive the owner's features. Preprocessing steps are detailed in Section 6 on Driving Events and Metrics. The features are then computed into the set of probability distributions that make up the driver's *fingerprint vector*. This serves as a basis for comparison when testing future trips.

A driver's habits should stay relatively static, but there may be some few cases where they change over time. This change could be catalyzed by an accident or close call for example, where a driver might attempt to conform to a safer driving paradigm thereafter which might have different characteristics compared to his previous behavior. A change could also be a very slow unconscious one, where after some time an impermissible frequency of false alarms could occur. Accordingly, the driver can reset and redo his training to represent his current behavior.

5.2 Testing Phase

Intuitively, driver identification accuracy will benefit from having as much available data as possible for comparison against the fingerprint vector. We use this type of retroactive method in our offline testing to determine which features should be included in our fingerprint and test vectors. In practice, an online driver identification system should actively process an incoming continuous data source, and waiting until a trip is complete defeats the purpose. To detect an unauthorized driver and recover a stolen car quickly, we need to compare a test vector to the driver's fingerprint vector as early as an appropriate amount of data is available. Accordingly we design an algorithm that accepts the input data as it is collected continuously and outputs a decision quickly. This is accomplished by partitioning the data stream into blocks to be tested sequentially as they arrive. This block size should be long enough to provide suitable accuracy but short enough to quickly return a result. A suitable block size can be found empirically as we show in Section 7 on Evaluation.

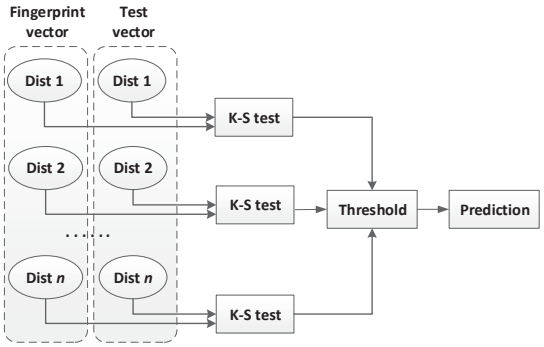

Figure 2: Predictor Component

Online testing: The online testing phase for a block of data from an unauthenticated user begins much in the same way as the training phase. It undergoes the same preprocessing, with important data selected and parsed into features making up a *test vector* $\mathcal{D}$. It then approaches the Predictor, wherein the following algorithm takes place (Figure 2). Let $\mathcal{D}_i$ denote the current test vector generated by the driver identification system. Further let $\mathcal{F}$ denote the fingerprint vector. Upon obtaining $\mathcal{D}_i$, the predictor estimates whether or not the user is legitimate by comparing $\mathcal{F}$ with an *augment test vector* $\mathcal{D}_i \bigcup \left(\bigcup_{j\in\{1,2,\ldots,i-1\}} \mathcal{D}_j\right)$, where $\mathcal{D}_1$, $\mathcal{D}_2$,...,$\mathcal{D}_{i-1}$ are the previously collected test vectors.

Three metrics are examined after the comparison. The first and second metrics are the similarity calculations d_1 and d_2 between $\mathcal{F}$ and the augment test vector, as measured by our two comparison tools. In both cases, lower numbers indicate more similarity. The third metric is the length l of the augment test vector, i.e., the number of test vectors that form the augment test vector. For the test vector $\mathcal{D}_i$, the length of the augment test vector is i.

For these three metrics we employ six thresholds. Metric d_1 is compared to thresholds t_{1_low} and t_{1_high}, while d_1 is compared to t_{2_low} and t_{2_high}. Threshold t_{both} is compared with the sum of d_1 and d_2. Finally, threshold t_l is used to limit l. Based on these metrics and thresholds, the prediction generates the following decisions:

- If both $d_1 \leq t_{1\_low}$ and $d_2 \leq t_{2\_low}$ are true, the user is temporarily authorized.
- If one of $d_1 \leq t_{1\_low}$ and $d_2 \leq t_{2\_low}$ is true, and $d_1 + d_2 \leq t_3$, the user is temporarily authorized.
- In any other case, the user's identity remains unknown.
- If either $d_1 > t_{1\_high}$ or $d_2 > t_{2\_high}$ at any time, then the user is unauthorized.

If $l < t_l$ at this time, and the current driver has been temporarily authorized or remains unknown, then the predictor continues authentication. The next block of input data is processed when it arrives to generate the new test vector $\mathcal{D}_{i+1}$, and the predictor repeats this process to continue authentication, i.e., comparing $\mathcal{F}$ to $\mathcal{D}_{i+1} \bigcup \left(\bigcup_{j\in\{1,2,\ldots,i\}} \mathcal{D}_j\right)$.

If $l \geq t_l$, and the above rules resulted in authorization, the user is identified as the owner. Otherwise if the user

remains unknown, authorization fails and an alert is made. Appropriate follow-up actions may be taken by the legitimate user.

Comparison tools: We use the well-known Kolmogorov-Smirnov (K-S) statistical test [18] as well as the total variation distance [19] to compare the fingerprint and the augment test vectors. Specifically, both vectors are made up of empirical probability distributions in the form of frequency data stemming from the collected acceleration data. First, each distribution from the augment test vector undergoes the K-S test with the respective distribution from the fingerprint vector, returning a conclusion as to whether or not the two portions of data are from the same distribution. Second, the distance between the distributions is measured, which provides an additional metric for their variation. The difference between the vectors is finally calculated using the total number of features (distributions) in the test vector which fail the K-S test as well as the sum of their variations. The specifics of this calculation and usage of both tests is covered in 7.3.

5.3 Testing Logistics

We gathered our testing data on a few mobile devices with the Android operating system, using as previously stated the acceleration information of the vehicle, to generate our features. Using this type of device and these features we were able to identify users based on their driving data with good accuracy, as detailed in Section 7 on Evaluation. One strategy for obtaining a GPS location of one's vehicle is suggested in an article [20] on the Internet site "wikiHow" and involves purchasing a smart phone with an elementary cell plan, and installing it unobtrusively in the car. The idea is that upon discovering a car theft, the phone's information can be used to access its GPS remotely to locate the car [20]. We propose that these two concepts can be combined: install a smart phone in the vehicle with our system implemented on it, and have it text an alert to the owner's personal phone when we detect unauthorized use. This provides prompt theft detection, owner notification, and tracking capability for recovery by the police. Also, this requires but the most elemental of smart phones, which are steadily decreasing in price, so this will be just as affordable as other leading protection methods, with added benefits.

A second strategy is to implement our theft detection system into the car's computer. While we carried out our testing by using Android devices' accelerometer and GPS data to create our features, so we can only comment on the accuracy of this platform, we believe an implementation in the car's computer operating system will perform similarly well. In fact, it will allow for more direct measurement of the driver's acceleration preferences by monitoring gas and brake pedal depression statistics, so similar or greater accuracy is likely. We leave this implementation to future work as we found programming in an Android device was much more practical than in a car's computer for this initial research. However, this will very likely change in the near future as vehicles become smarter, so we are optimistic that our system can be applied easily to cars in the future. An exciting indication of this opportunity is the news that vehicles from Audi, Honda, General Motors, and other companies are beginning to support the Android operating system as the car's OS [21].

5.4 False Alarm Handling

Biometric security systems which interface directly with the owner of the protected system have the unique advantage that the annoyance of false alarms can be minimized entirely. As our proposed system notifies its owner in the case of an alarm, false alarms may occur without any impact other than, say, an unnecessary text message received by the owner. While driving, the owner should ignore text messages regardless. This is just one possibility for an alert delivery system; conceptually, a more appropriate alert method may be discovered. More importantly, we provide the system which determines an alert should be made to begin with, and remark that any kind of alert generated by our system is more effective than no alert at all.

We strive to maintain low false alarm rates, and the accuracy we achieve reflects that. We also note that some users may desire an even more stringent and secure authentication test. In that context, we emphasize that the chance that the system does not identify a thief can be minimized by manipulating the false alarm vs. mis-detection rates. This is discussed in further detail in Section 7.3 on Threshold Size. Similarly, we may lower our test thresholds, making the tests more stringent, and decrease the time required to identify theft. The resulting false alarm increase, with its very small increase in annoyance, may be worth the added protection to some users.

5.5 Mimicry Attacks

To successfully mimic another driver, avoiding detection by our algorithm, requires a constant attention to several complex factors. One must regulate each distribution in the test vector to be the same as the corresponding one in the fingerprint vector. These distributions come from different sources and apply at different velocity ranges. Furthermore, the measure of "sameness" is made according to two different tests combined through several different rules. Each of these factors compounds on each other, making it utterly implausible to launch such attacks, even disregarding the difficulty of accessing the data on the owner's system to study.

6. DRIVING EVENTS AND METRICS

To illustrate the methodology used by our system to classify users, we begin by describing the collection of data which we used to prove the concept viable, followed by the processing of this data into elemental driving events.

6.1 Driving Event Types

We identified six general driving events encountered in a typical drive: increasing speed, maintaining speed (cruising), coasting, braking, turning, and changing lanes. These are essentially self-explanatory for the high percentage of the developed world familiar with driving or riding in vehicles. We do however note a key observation that these events are all types of acceleration, as shown in Figure 3. Increasing speed is achieved by depressing the gas pedal and causing positive acceleration along the y-axis of the vehicle. To maintain speed, the driver keeps a constant depression of the gas pedal or uses Cruise Control, which keeps a steady zero acceleration. Coasting involves slight negative acceleration from release of the gas pedal and no application of the brakes. In contrast to coasting, braking invokes a strong negative acceleration with the car's brakes. Turning encounters angular acceleration as force is applied along the x-axis

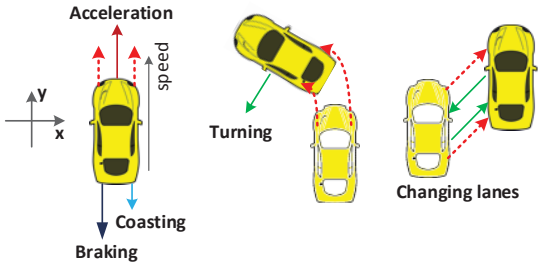

Figure 3: Forces caused by each type of acceleration. Not shown is cruising, which is defined by a lack of force.

of the car (perpendicular to the car's facing direction). Finally, lane changes encounter, in comparison to full turning, a slight angular acceleration at the beginning and end of the movement.

With this in mind, human preferences in performing these various events are a function of force tolerance, as well as reasoning and sometimes necessity. For example, approaching a red light with no traffic around, an individual will brake according to comfort, perception of safety, and care for vehicle integrity, all functions of force tolerance and reason. Similarly, a driver may prefer to stop cruising and start coasting some distance out from an impasse, such as a stoplight or wall of traffic, to avoid wasting gas, while another may continue at high speed until braking is obligatory. In the event of traffic, outside influence is also a factor, causing some of these events to be more or less effective than others for identifying personal behavior patterns. It is important, then, to select those events for which there is the least outside influence causing variation in a driver's behavior. Those events will have the features most unique to each person. Refer to *Feature Selection* in Section 7 where this process is exhibited.

6.2 Speed Effects on Events

We also expect variations in users' behavior at different speeds. A mistake at high speed, for example, is more dangerous than at low speed, and with this knowledge some drivers may accept more risk at some speeds than others. Also, environmental factors such as traffic, road conditions, or weather could force a user to drive at a different speed than preferred. With a single partition holding all data, a user might appear to favor the habits from the range of speed most often traveled in, and potentially different behavior from other speeds will be ignored. With multiple partitions, the user's behavior can be determined for each speed range regardless of the fraction of time spent driving in that range. We evaluate using speed ranges as sub-features for our driving events, ultimately assembling our fingerprint and test vectors from the data on the events for each speed range. This reduces the impact of the aforementioned environmental factors for a more robust system.

As an example, consider Figure 4. Here, the top plot shows the positive acceleration distributions for two drivers, from data measured by an accelerometer. Their difference is visible, but small. The distributions are farthest apart at an acceleration of roughly 1.6 m/s^2, where 92% of data for

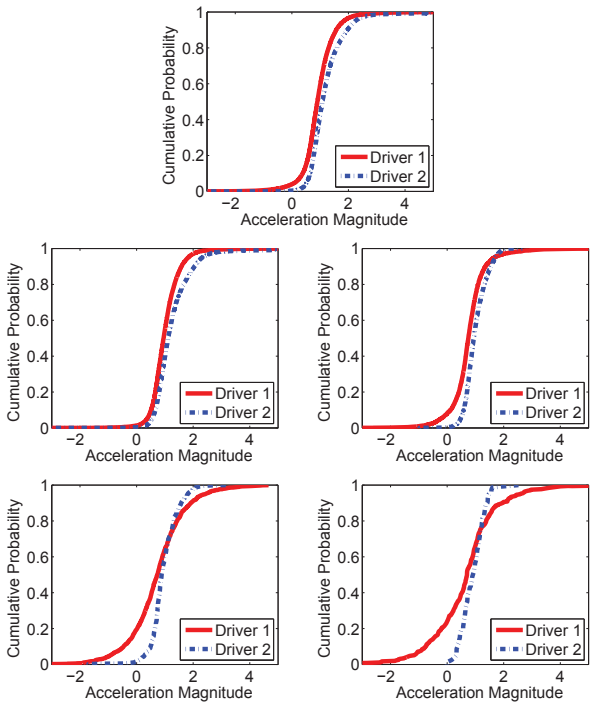

Figure 4: Empirical Cumulative Distribution Functions (ECDFs) of two users' positive acceleration data (top), followed by ECDFs of that data partitioned by four velocity ranges (lower four charts)

Driver 1 is below this value. Compared to 82% for Driver 2, there is a 10% difference here. The four plots below this show the same two distributions separated by four velocity ranges (0-20, 20-40, 40-60, and 60+ mph). The first partition looks similar to the full data plot, but the next three show marked differences between their respective data and the full data plots. Their largest separations are, in order, 12%, 19%, 26%, and 28%. These large differences combined together can form a cascade filter that helps us differentiate more effectively between these two users who at first glance appear somewhat alike.

6.3 Data Collection

As mentioned, the six overarching driving events are fundamentally different forms of acceleration. Increasing speed, cruising, coasting, and breaking are all acceleration at varying magnitudes along the y-axis of the vehicle as illustrated in Figure 3. From the frame of reference of the car, turning and lane changes create force along the x-axis. For this reason we chose the accelerometer hardware in mobile devices to collect data on acceleration in each dimension. Also necessary is velocity information so that the acceleration can be partitioned according to speed ranges as just discussed, so we included GPS hardware as well, which can provide velocity as the time-derivative of its gathered position data. As acceleration is the time-derivative of velocity, the GPS data can also provide acceleration data, but along the car's y-axis only. Having two sources of acceleration data, we can use each to verify the other for the sake of accuracy.

We developed an Android application on the Nexus 5 to collect and archive driving trace data, and distributed this

among our volunteers so they could gather data while driving for a short time. This application records continuously the global position of the device and the forces acting upon it in each direction. GPS data was queried as frequently as possible, which was roughly once per second; accelerometer data was gathered at the device's specified "normal" rate of 50 Hz. Both sets of data were recorded with their respective timestamps to be later retrieved from the device for analysis.

6.4 Driving Event Extraction

The two sets of data aggregated from GPS and accelerometer offer two sources of acceleration information. First, acceleration is directly measured by the accelerometer, and second, it can be calculated from the GPS position data. The difference between these two sources require different processing to produce driving events, which is detailed in their respective sections below.

6.4.1 Acceleration Events Sensed by Accelerometer

No calculation is required to render acceleration values from the accelerometer data, but due to variations in participants' positioning of the device collecting data, some normalization is required. For example, we need to ensure the average of the y-component of acceleration is close to zero, as it should be when the beginning and ending velocities are zero. Almost all data was collected with the device sitting in the passenger seat, which in modern cars has a very small incline when looking forward, so the y-acceleration trace would commonly be reduced by a scalar constant so that its average would be close to zero. Such a calibration is acceptable, but too large of an incline spreads the force over more than just the y-axis, so we ignore any such data to avoid invalid comparison with properly collected data.

Noting again that the y-axis acceleration trace should average zero, *positive acceleration* events are identified as beginning at a point whose moving average is larger than a standard deviation more than the overall average of zero. They continue while the acceleration remains above zero. We discovered that some noise caused by the vibration of the car (which was uniform amongst all users' data) prevented separation of coasting and braking, so we treated an overall *deceleration* event type as the combination of the two events. These deceleration events are thematically the same as positive acceleration events, but start at the point whose moving average is more than a standard deviation *below* the average of zero. Following these specifications, all points in the y-axis acceleration trace which are not identified as positive acceleration or deceleration events represent *cruising* events, because speed is maintained when acceleration is near zero. If the velocity is zero as calculated from the GPS data, the accelerometer data is discarded to avoid including time spent stationary at stoplights for example.

Turning and lane changes involve perpendicular force, as mentioned, so we look to the x-axis data for their detection. We found that lane changes could provide indiscernible x-acceleration if the user made the change slowly, so we ignored them in favor of *turning events*, which show forces of similar magnitude to acceleration or deceleration events. The same general strategy was used, to pick out left and right turns, dependent upon positive or negative direction of force on the x-axis. Finally, any data not involved in turning events is discarded as it is redundant to the cruising event data.

6.4.2 Acceleration Events Calculated from GPS Position Data

To find users' acceleration and velocity from the GPS position data, the distance between each point is first calculated. This is an approximation derived from the points' latitudes and longitudes. With distance calculated in this manner, velocity and acceleration are simple time derivatives of the position data. Acceleration data is only along the y-axis here as the distance calculation loses any direction data, and calculating direction for use in turning acceleration is prohibitive with the infrequency of data collection. Consequently, turning and lane change events are not considered for GPS data.

Though the overall data is accurate, at times in all users' data there would be a delay followed by a receipt of an old data point, now incorrectly timestamped, along with a new one. Due to the complexity and noteworthy delay involved with communicating with satellites, this issue would result in some oscillatory noise. For this reason and the normally infrequent collection of data, acceleration events are not so nicely shaped as they are for the data acquired by the accelerometer, so these events are parsed differently. Here, a data point is considered part of a *positive acceleration* or a *deceleration* event only if is further than a standard deviation away from the average zero acceleration. This ignores the tails of acceleration events, which do not usually appear tapered out in this data. Coasting is again absorbed into deceleration events, and as before any remaining points with nonzero velocity are classified as *cruising* events.

7. EVALUATION

7.1 Experiment Framework

We had 31 volunteers participate in this project. These volunteers represent ages ranging from late teens to early 60s and included students, faculty, office workers, and self-employed. They were given our application to run on an Android device. The application was designed such that at the beginning of a trip, the user would place the device in a horizontal position facing forward, and press a button to start data collection. No further input was required until reaching the destination, at which point the user would press a second button to stop data collection. All laws pertaining to phone usage in vehicles were thereby satisfied and driving safety upheld. These volunteers were tasked with accumulating at least 30 minutes to an hour of driving data, and some collected up to two hours.

To quantify the accuracy of our system, we examine its rates of false alarm and mis-detection as we vary the system parameters. We define the rate of false alarm as the fraction of users who are not correctly identified as themselves. In this case, the system would render an alarm indicating the legitimate user is unauthorized. The rate of mis-detection includes users who are incorrectly identified as other users. This case involves no alarm raised over the current driver being someone other than the owner. These values are our criteria in the following sections.

7.2 Feature Selection

We are able to extract positive acceleration, deceleration, and cruising events from both the accelerometer data and the GPS data, as well as turning events from the accelerometer data. We now show the performance of each of these

metrics in a comprehensive test of our datasets and decide whether or not to include them in our feature vectors.

As discussed earlier, we use the K-S test to compare between two users' distributions from a particular event and velocity range. The test measures the distance between the empirical distribution functions of the two distributions. If the distances are all small, the null hypothesis that they are from the same probability distribution is decided to be correct. If the largest distance is greater than a threshold dependent upon a specified significance level and the size of the distributions, the null hypothesis is rejected, indicating they are from different probability distributions. A smaller significance level results in a larger threshold value and thereby a larger tolerance for variation in two distributions ultimately identified as belonging to the same distribution. We find that among our useful event types, a significance level of 0.005 allowed most users to be identified as themselves without others also appearing the same. Consequently this value is used as we exhibit the performance of each event type.

The Total Variation Distance is similar to the K-S test, but instead of examining the largest distance between two distributions, it takes a sum of the distances between every point in the distributions. A smaller total variation distance can indicate two distributions are similar, while a large distance can imply difference. We find that the K-S test is stronger for comparing the shapes of the distributions, while the total variation distance is stronger in comparing their overall sizes. The two tests are combined as discussed in Section 5.2 detailing our testing process, with a series of thresholds we optimize later in this Evaluation section.

For the following several features, figures will be shown to illustrate the performance of each feature in comparing every driver to every other, for GPS and accelerometer data. Specifically, with 31 available datasets, there are 930 trials which involve attempting to differentiate one user from another, and 31 trials where each driver should be identified as themselves. In the Self bar, the Correct portion shows the number of users (out of 31) correctly identified as themselves, while the Incorrect portion shows the number of trials where users did not appear like themselves. The Correct portion of the Other bar shows the number of trials (out of 930) where our tests correctly differentiate between different users, while the Incorrect portion represents users who appear similar to other users.

In short, the Incorrect component to the Self and Other bars represent the false alarm and theft mis-detection rates, respectively. Good features will have few mis-detections and as few as possible false alarms, so a perfect figure would show 31 correctly authorized trials and 930 correctly denied trials. Thresholds vary for each type of data, and are selected to provide the best results and show the potential efficacy of the acceleration type and source.

7.2.1 Positive Acceleration

For both the accelerometer and GPS data, positive acceleration events present effective divergence between users. Figure 5 shows the results. For the GPS, requiring all velocity ranges to have nearly equivalent distributions properly identified all 31 users as themselves, but also allowed users to impersonate others in 116 of 930 such comparisons. The accelerometer data allows fewer impersonations, 32 of 930 possible, but fails to identify 17 of 31 users as themselves. As the two sources have different strengths, including both

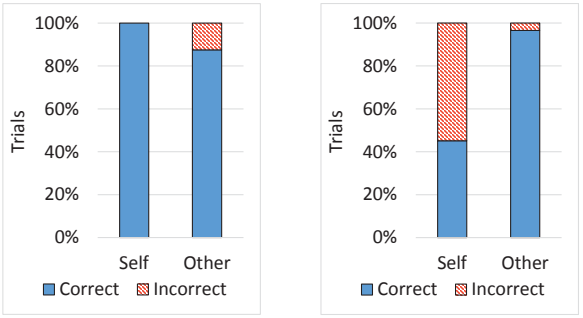

Figure 5: Positive acceleration trial statistics from GPS (left) and accelerometer (right) datasets

can potentially (and does, as shown later in this section) produce good results.

The usefulness of these features is logical: a driver's preferred positive acceleration is only ever limited with an upper bound by traffic (at stoplights with long lines, for example), so for most places and times, drivers are able to accelerate as they please. In the case of traffic, the data used in this experiment was amassed in a city consistently named among the worst cities in the United States for traffic, so whatever inaccuracy traffic can impose is present in our results. By a lower bound, positive acceleration is never limited except in the cases of risky merging or exceptionally impatient followers. The latter case is an out-lier, and the former incorporates an individual's particular acceptance of risk into the data which can increase rather than limit the accuracy in discerning between users.

7.2.2 Negative Acceleration

As seen in Figure 6, deceleration events from the GPS data prove roughly as effective as positive acceleration events. Indeed, 30 of 31 users are correctly self-identified, and 124 of 930 possible impersonations are allowed using only this data. In contrast, the accelerometer data does not function well for this feature, with 176 impersonations and only half of the possible self-identifications.

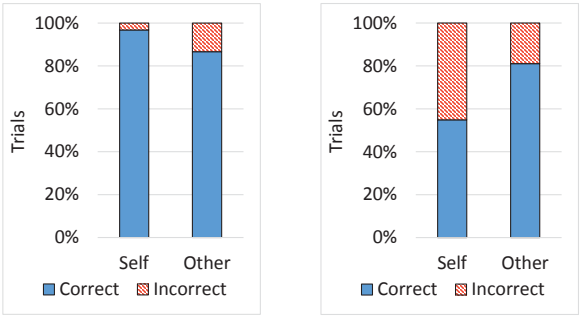

Figure 6: Negative acceleration trial statistics from GPS (left) and accelerometer (right) datasets

It does make sense that deceleration might be less useful than acceleration events. Negative acceleration events have more potential limitations on driver freedom and variation, as they are more a function of necessity than positive acceleration events. Other drivers' behavior can induce braking, by merging or pulling out in front of the subject, or in traffic by simply being numerous. Stoplights can change unexpectedly, causing attenuated available stopping time such that drivers have to brake differently than they would otherwise.

The difference in efficacy between the two sources is attributed to the more coarse-grained GPS data failing to capture brief rapid braking events caused by reactions to the environment instead of driver preference. The accelerometer includes these un-useful reactions along with the useful information on driver preference, and suffers accordingly.

7.2.3 *Cruising*

Figure 7 shows the poor results of identification using cruising data. While only one false alarm occurred with the GPS data, 198 trials allowed impersonation to take place. Accelerometer data was considerably worse, with 324 impersonations and just a little more than half of the users correctly self-identified. This is rather unsurprising, because cruise time depends almost entirely on the driving locale. Also, variation in speed while maintaining a mostly constant cruising speed can depend on the car's cruise control system and changes in terrain as much as it might depend on a user manually maintaining speed with measurable fluctuation.

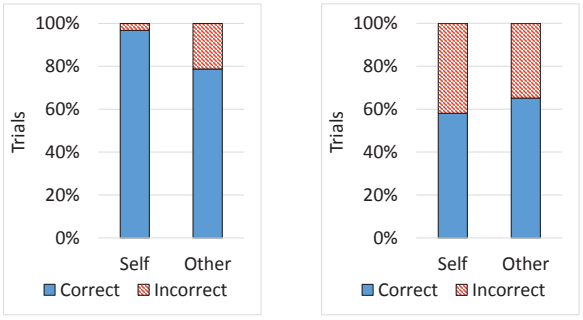

Figure 7: Zero acceleration (cruising) trial statistics from GPS (left) and accelerometer (right) datasets

7.2.4 *Turning*

Turning data from the accelerometer produces the results shown in Figure 8. Intuitively, drivers handle cornering according to their force tolerance and perceived safety, as mentioned before. In practice, this data is not very effective. At best, only 3 of 31 self-identification tests resulted in false alarms, but about half of the possible 930 impersonations were allowed. The infrequency of turning events in comparison with positive or negative acceleration events likely reduces the usefulness of this data, and more data could result in better performance. If we had enough data to properly train the system on turning events, it would still however require too much data-gathering time in testing the identity of new users to efficiently render a decision, which is the overarching concern.

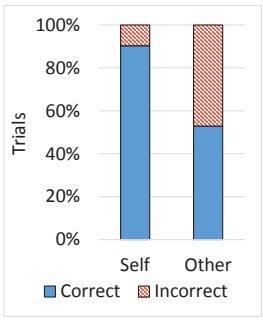

Figure 8: Turning trial statistics from accelerometer dataset

7.2.5 *Velocity Range Partitioning*

Finally, Figure 9 illustrates the benefits afforded by splitting the training and testing datasets by the velocity of each data point. At left is the best result obtainable without the use of speed ranges, while the right chart shows the better result from their use. While the mis-detection rate holds static, the false alarm rate collapses from six to one, providing more confidence in our system's ability to correctly identify users as themselves. This equates to 97% self-identification and 91% differentiation rate.

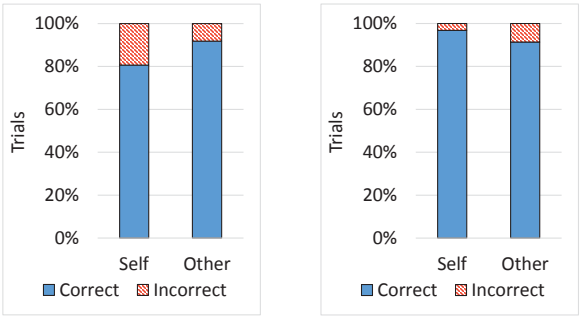

Figure 9: Trial statistics not using (left) and using (right) velocity range partitioning of driving data

The velocity ranges used here are chosen such that the following concerns are satisfied:

- The divisions should be numerous enough to allow for all the behavioral differences a person could make based on speed.
- The divisions should be small enough such that the lower bound does not "feel" significantly different than the upper bound.
- The divisions should be few enough and sufficiently large to accumulate adequate data for comparison in a timely manner, so decisions can be made efficiently.

Empirical testing evinced a set of ranges most appropriate for our experiment comprised of 0-20, 20-40, 40-60, and 60+ mph.

7.2.6 *Feature Selection Summary*

We found that both sources of acceleration data were useful as they both had strengths and weaknesses. The data from the accelerometer was very fine-grained, being measured fifty times per second, and was able to be parsed into acceleration events quite well. However the magnitudes of the forces upon the accelerometer did not show enough variation to identify unique users with high accuracy by itself. This may be due to the device's placement on the cushioned passenger seat, which likely dulled the force somewhat. The acceleration as calculated from position data, in contrast, had more accurate magnitudes due to its direct calculation from the location trace. GPS is accurate to within 3-7 meters with 95% confidence [22], so apart from some local inaccuracies the overall traces were highly accurate. Nevertheless those local inaccuracies were large, so the GPS data was also insufficient for identification on its own. Finally, previously stated as a benefit to the negative acceleration feature, GPS data contributes its ability to often ignore reactive (non-preferential) actions that are too brief for the frequency of data collection.

Taking from the useful events, we have features including (1) positive acceleration measured by the accelerometer and (2) combined positive and negative acceleration calculated from the GPS position measurement. Both features are finally subdivided into four velocity ranges, for a total of eight features, ultimately resulting in 97% self-identification and 91% differentiation rates. Our Predictor component's set of thresholds is set to accommodate the total number of feature tests in its classification of drivers.

7.3 Threshold Size

The number of features required to pass authentication has an effect on the rate at which a car owner is authorized as well as the rate at which illegitimate users are authorized. We have eight features and two tests performed on each feature. With the data split into ten blocks, there are thus a total of 160 feature tests, 80 for each test type. To use both test types, we combine them according to several rules and thresholds as discussed in Section 5.2. For the K-S test, we have thresholds specifying the maximum number of tests which may be failed while still resulting in authorization. For the total variation distance, we specify the maximum variation distance to which the tests may be summed. Lower thresholds will prevent more users from being erroneously authorized, but will reduce the number of rightful owners being authorized as well. Likewise, high thresholds can ensure all legitimate users are authorized, but several "thieves" will also be accepted.

For this portion of the experiment, full training and testing datasets are used, for the broadest applicable view of the data. These datasets are tested sequentially block by block. To find the optimal set of thresholds, we find the point where the false alarm and mis-detection curves intersect in Figure 10. In this figure, we search the possible thresholds to find the mis-detection rate associated with a specific false alarm rate. We find an intersection with lowest total error nearest 3 false alarms, identifying all but roughly a tenth of illegitimate users.

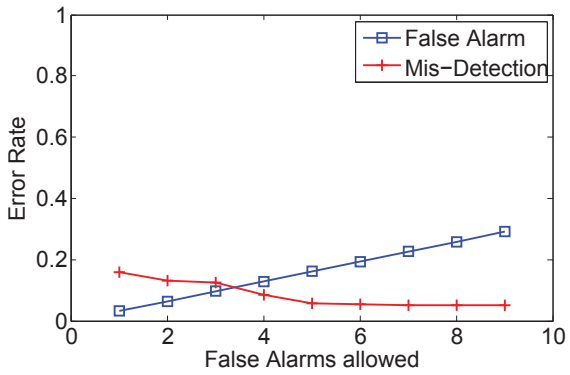

Figure 10: Finding an optimized threshold

We also note that with a different threshold selection, we can reduce illegitimate access to the neighborhood of 6% with 16% false alarm rate. This boost to the false alarm rate might be worth the added security depending on the individual. Because alarms only notify the car owner (depending on the setup), users may find this false alarm rate unobtrusive enough to choose the very stringent and consequently secure threshold set. In practice, users could be given the option to specify their own mis-detection threshold based on perceived risk and comfort level.

7.4 Training Data Size

The training process for a classifier benefits from an abundance of data (though sometimes too much data can make a classifier too specific and not applicable to new data). If the application of the classifier is to prevent irretrievable vehicle theft, the user would prefer to have this protection as soon as possible. We examine the impact of different training sizes here, to find out how much data collection is necessary before a user can begin protecting the car.

As our volunteers collected differing amounts of data, we base our training data size on the small datasets, those supplied by our volunteers who drove for smaller amounts of time. This size is broken into ten blocks to see the effects of increasing the data size on our false alarm and mis-detection rates for all drivers. The maximum training data allowed here is roughly 450 seconds (450 points) worth of positive and negative acceleration from the GPS data, and 180 seconds (9000 points) of positive acceleration measured by the accelerometer. For reference, our users accomplished this amount in 25-45 minutes depending on how much time was spent cruising or at stop lights. All available testing data is used, and the threshold set used for authentication is that arrived at in the previous section on Threshold Size. The results are shown in Figure 11.

As expected, the best accuracy is attained with all available blocks of data included. These error rates are higher than those we measure in other tests, because we are restricted to the small dataset size. Additionally, because the threshold set is constant, optimized for the full available training size, the false alarms are slightly erratic before settling to lower values in later blocks. Of largest impact here is the fall of the mis-detection rate showing better theft detection with larger training data. We recommend 1.5-2 hours of training time, because in perusing the results, we find those

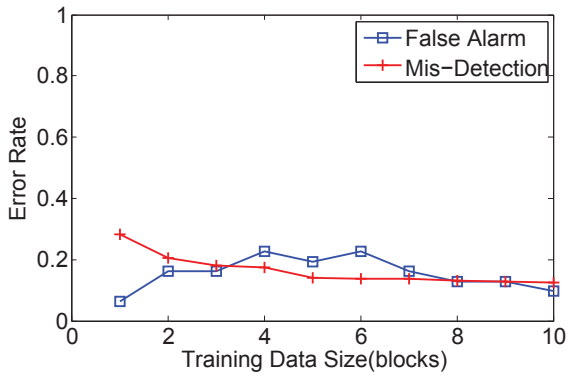

Figure 11: Effects of training dataset size on error rates

users who provided that amount of data almost universally were identified as themselves with the least number of failed feature tests, and were rarely confused with other drivers (mis-detected).

7.5 Testing Data Size

The amount of data available for testing unsurprisingly has effects on accuracy similar to the training data. The desire is again to require only a small amount of data, this time in order to perform accurate authentication in time to quickly identify a theft. We therefore analyze the effect of differing testing sizes to ascertain the appropriate length of testing data collection.

As with the training size experiment, we restrict the maximum amount of data allowed to be 450 seconds of GPS sourced positive and negative acceleration alongside 180 seconds of accelerometer sourced positive acceleration. Again, this equates to 25-45 minutes of driving to ensure the larger datasets do not skew the results to a higher accuracy and a uniform amount of data is used. We allow full training datasets, however, as users in the real world will have full control of their training phase. The threshold set is again held constant, using the results of the Threshold Size study. Results appear below in Figure 12:

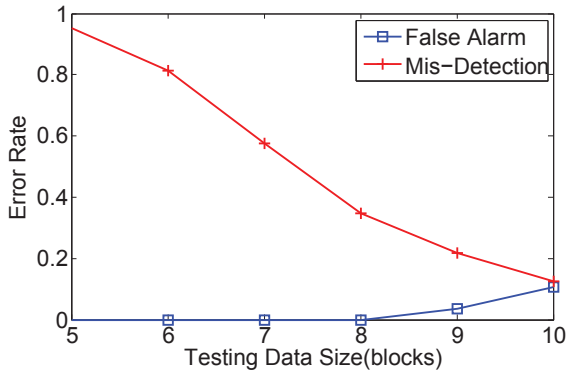

Figure 12: Effects of testing dataset size on error rates

It is important to note again that the thresholds used for this examination of testing dtaa size were optimized for all data available. This threshold set works quite well for the proper test size, and this emphasizes the importance of the minimum length of time used for gathering testing data. It also displays the effect of increasing test data on the testing stringency. To begin, all users are permitted access, resulting in no false alarms but 100% mis-detection, and by the time all 10 blocks of data are used, most unauthorized users are detected and some legitimate users begin to generate false alarms. It is therefore important also to prevent testing data size to grow too large and too specific.

As further evidence of the effect of additional data, consider Figure 13. As discussed, the K-S test returns a decision on whether or not two datasets are from the same distribution, and it does so by estimating the probability that the two datasets' empirical distributions would be the measured distance apart while still being part of the same overall distribution. This probability is referred to as the p-value, and points on this figure are the average p-values for all feature tests between users as additional data is included. The line labeled "Self" refers to the average of self-identification tests, and by the criteria above, the p-values for these tests should be large. The line labeled "Others" refers to the average of those tests between distinct users, and these p-values should be small. The figure shows self-identification tests remaining at a flat rate, indicating that it takes very little time to match one's testing data with one's own training data. As desired, the p-values for the Others line are below those for the Self line. Furthermore, while it takes longer to rule out other users than it does to self-identify, our results show that the p-values diverge quickly after a few final data blocks are added to the testing data.

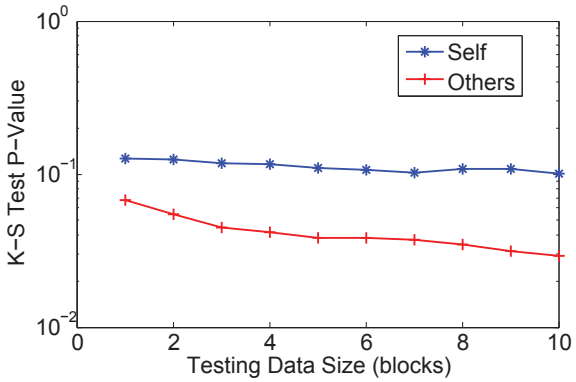

Figure 13: Effects of testing dataset size on K-S test p-values

7.6 Evaluation Summary

To conclude this evaluation, we present the following successes:

- We found effective features including positive acceleration measured by the accelerometer as well as positive and negative acceleration measured by the GPS.
- We found further effective features in partitioning the above according to specific velocity ranges.

- We showed this collection of features capable of attaining 97% self-identification and 91% differentiation accuracies.
- We illustrated the effects of varying our testing thresholds and the ability to attain very low mis-detection (around 7%) by allowing slightly higher false alarm rates.
- We analyzed the size of training and testing data to determine the requirements for robust accuracy.

8. CONCLUSION

In this paper, we proposed a fast automatic driver recognition system that continuously authenticates the driver as the vehicle is operated. Our basic idea is to extract unique features from the driving behavior, which cannot be exactly reproduced by a thief driving away in the stolen car. Through an in-depth investigation of the typical driving events, we identified effective driving features (i.e., positive and negative accelerations, at multiple speed ranges) to distinguish between the car owner and any unauthorized users. We performed extensive experimental evaluation using the driving data collected from 31 volunteers. Our experiment results show that the proposed system can successfully distinguish that the current driver is the car owner, with 97% accuracy, while also preventing impersonation 91% of the time.

Acknowledgments

This work is supported by the National Science Foundation under grants 1527144 and 1553304, and the Army Research Office under grant W911NF-14-1-0324.

9. REFERENCES

[1] Federal Bureau of Investigation, "Motor vehicle theft," http://www.fbi.gov/about-us/cjis/ucr/crime-in-the-u.s/2012/crime-in-the-u.s.-2012/property-crime/motor-vehicle-theft, 2012.

[2] Fox News, "Owner of stolen car sued in deadly Hit-and-Run," http://www.foxnews.com/story/2009/05/15/owner-stolen-car-sued-in-deadly-hit-and-run/, 2009.

[3] "Onstar," https://www.onstar.com/web/portal/home?g=1, 2014.

[4] "Lojack," http://www.lojack.com/Home, 2014.

[5] N. H. T. S. Administration, "Vehicle theft prevention: What consumers should know," http://www.safercar.gov/Vehicle+Owners/Resources/Theft+Prevention.

[6] M. Shahzad, A. X. Liu, and A. Samuel, "Secure unlocking of mobile touch screen devices by simple gestures: You can see it but you can not do it," in *Proc. of the Annual International Conference on Mobile Computing and Networking (Mobicom)*, 2013.

[7] M. Qi, Y. Lu, J. Li, X. Li, and J. Kong, "User-specific iris authentication based on feature selection," in *Proc. of International Conference on Computer Science and Software Engineering*, vol. 1, 2008, pp. 1040–1043.

[8] R. Bowe, "Red flag on biometrics: Iris scanners can be tricked," https://www.eff.org/deeplinks/2012/07/red-flag-biometrics-iris-scanner-vulnerability-revealed, 2012.

[9] T. C. Clancy, N. Kiyavash, and D. J. Lin, "Secure smartcard-based fingerprint authentication," in *Proc. of the 2003 ACM SIGMM Workshop on Biometrics Methods and Applications*, 2003, pp. 45–52.

[10] A. W. Kosner, "iphone 5s touch id fingerprint scanner is a fail," http://www.forbes.com/sites/anthonykosner/2013/10/15/iphone-5s-touch-id-fingerprint-scanner-is-a-fail-for-20-of-users-heres-what-to-do/, 2013.

[11] L. Li, X. Zhao, and G. Xue, "Unobservable re-authentication for smartphones," in *Proc. of the Network and Distributed System Security (NDSS) Symposium*, 2013.

[12] C. Nickel, T. Wirtl, and C. Busch, "Authentication of smartphone users based on the way they walk using k-NN algorithm," in *Proc. of International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, July 2012.

[13] N. Clarke and S. Furnell, "Authenticating mobile phone users using keystroke analysis," *International Journal of Information Security*, vol. 6, no. 1, pp. 1–14, 2007.

[14] X. Zou and D. M. Levinson, "Modeling pipeline driving behaviors: Hidden markov model approach," *Journal of the Transportation Research Board*, vol. 1980, no. 1, pp. 16–23, 2006.

[15] N. Oza, "Probabilistic models of driver behavior," in *Proc. of Spatial Cognition Conference*, 1999.

[16] A. Sathyanarayana, P. Boyraz, and J. Hansen, "Driver behavior analysis and route recognition by hidden markov models," in *Proc. of International Conference on Vehicular Electronics and Safety*, Sept 2008.

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

[18] P. Olofsson, *Probability, Statistics, and Stochastic Processes 2nd edition.* John Wiley, 2012.

[19] J. A. Adell and P. Jodrá, "Exact kolmogorov and total variation distances between some familiar discrete distributions," *Journal of Inequalities and Applications*, vol. 2006, no. 1, p. 64307, 2006.

[20] "How to make a smart car surveillance system using a mobile phone," http://www.wikihow.com/Discussion:Make-a-Smart-Car-Surveillance-System-Using-a-Mobile-Phone, 2005.

[21] "Open automotive alliance," http://www.openautoalliance.net/, 2014.

[22] National Coordination Office for Space-Based Positioning, Navigation, and Timing, "GPS accuracy," http://www.gps.gov/systems/gps/performance/accuracy/, 2014.