Data-Driven Extraction of Vehicle States from CAN Bus Traffic for Cyber Protection and Safety

Michael R. Moore, Robert A. Bridges, Frank L. Combs, & Adam L. Anderson
Oak Ridge National Laboratory, Oak Ridge, TN

Abstract—Vehicles are increasingly cyber-physical systems, now depending on scores of networked control units. Consequently, modern transportation faces challenges to ensure autonomy, security, and safety. With this shift in focus on vehicle activities it becomes imperative to understand the complex decision making, the potential for cyber-attacks, and the vehicle state inference all based exclusively on internal electronic communications. Modern vehicles include scores of on-board electronic control units (ECUs) communicating over in-vehicle networks to control safety-critical systems. Protecting these networks is especially challenging because there is no publicly available translation of in-vehicle network data to vehicle functions. Even though packets are observable, there is no existing way to automatically know what mechanisms they control. Thus, intrusion detection systems (IDSs) for controller area networks (CAN) have been previously limited to leveraging statistical properties or protocol standards. In contrast we have developed a data driven, semi-supervised approach to learn physical relationships of CAN signals from only a limited set of CAN packets. These mappings are then used to develop a Hidden Markov Model (HMM) of the driver's actions upon which transaction analysis is performed to optimize the real-time identification of the states. Our approach builds an image from the CAN data, then trains a convolution neural network (CNN) to give emission probabilities. Initial results show that the state of the driver's actions can be predicted with approximately 90% accuracy. We believe this method will be robust to injection attacks and will provide detection of injection attacks from a vehicles network traffic even when the data formats are not known.

1 Introduction

Modern vehicles rely on communications between many small computers, called electronic control units (ECUs), which coordinate essential functions such as wheel speed and brake lights [1]. ECUs broadcast signals using a standardized protocol over one or more controller area network (CAN) buses. The CAN bus is exposed directly via the mandatory on-board diagnostic (OBD-II) port, such that attackers can inject signals if physical access is gained [2, 3]. Many wireless systems provide additional attack vectors; perhaps most notably, Miller et al. [4] caused a Jeep Cherokee to stop on the highway after compromising ECUs remotely. Regardless of the pathway into the vehicle, the end result can be treated as a network intrusion or cyber attack.

A primary roadblock to defending vehicles via the in-vehicle CAN bus is lack of standardization across makes and models for how functionality is communicated within the CAN protocol. As shown in Fig. 1, CAN data frames have two essential components for communicating messages—an arbitration ID (also known as process ID or PID) field and the data field. Yet, the mapping for how PIDs and data fields encode functionality is proprietary and varies greatly

Fig. 1. CAN 2.0 data frame depicted. Image from Cho & Shin [5]. 11 bit arbitration ID field used for prioritization of messages. Up to 64 bit (8 byte) data field used for message contents. How arbitration ID and data fields code for vehicle functionality varies wildly across models.

across models. As an example, in our experience with a particular vehicle, we have discovered a PID that uses its first two bytes of the eight-byte data field to encode running lights (on/off), while using the last two bytes to encode speedometer values, yet this PID does not occur in other makes.

As vehicles often have over 100 PIDs, each sending 64 bits of data per message, manually mapping changes in the CAN data to the myriad of potential physical consequences of the vehicle is tedious, difficult, and often unreliable. Exacerbating this problem is non-standardization in what functions are CAN-bus communicated per models; e.g., hybrid vehicles have both gas and electric motors receiving signals, while traditional vehicles only have the former.

Thus, the key challenge for state-based methods is mapping the proprietary data fields of the CAN bus packets to vehicle functions such as accelerator, wheel speed, and braking without having documentation of the data formatting. This motivates development of algorithms for automatic discovery of classes of functions (e.g., data fields linearly related to speed) that could be in turn mapped to states such as acceleration or idling.

State modeling approaches often involve the use of Hidden Markov Modeling (HMM) to infer the state of a system from observables called emissions. For this research, we have chosen to establish a methodology for discovering an HMM that will support automated decision making processes in the future. In future endeavors, we believe the best model for using HMMs to support autonomous decision making and cyber protection are Partially Observable Mark Decision Process (POMDP) and Predictive State Representations (PSR) since they have been shown to be especially pertinent for applications involving user (agent) inputs, environmental awareness and system reactions such as driver-assisted vehicles.

A relational diagram of Markov models is shown in Fig. 2 which is an adaptation from [6]. As shown, the items at the bottom left (Markov Model [MM] and Markov Deciosion

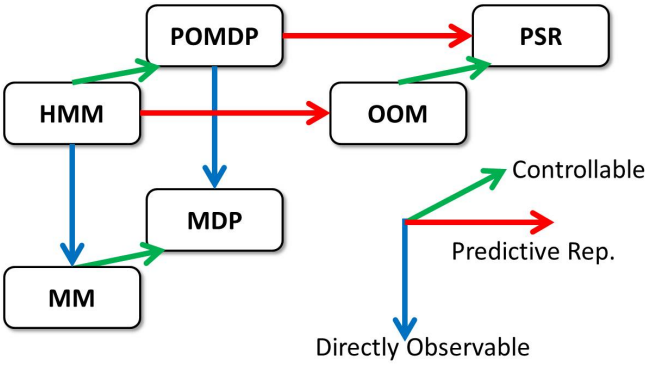

Fig. 2. Zoubin-like Cube Showing Relationships Among the Relevant State-Based Markov Models

Process [MDP]) represent directly observable states and decision processes. The upper right pair (PSR and observable operator models [OOMs]) are models that incorporate predictive outcomes.

A machine learning approach for identifying the Hierarchical HMM of an Operational Vehicle is given by Bai et al. [7]. As discussed by Song et al. [8] the first step in developing a POMDP for predicting optimum decisions is to develop multiple levels of HMMs; hence, we view this work as a progression toward autonomous decision making. To our knowledge, the problem of automatically understanding CAN messaging is first presented in the literature and thus far only discussed by Jaynes et al. [9]. Their approach requires CAN messages with ground-truth vehicle function labels (requiring tedious, manual reverse engineering) as training supervised learning techniques.

We propose a novel data-analytic approach for characterizing CAN communications based on their relationship to physical states of the vehicle. First we make the critical observation that the four sequential byte pairs that comprise each eight-byte data frame is the appropriate unit of data for analysis, and call these byte pairs *data words*. Next, we propose a way to learn speed-related data words via correlation statistics, which allows understanding of vehicle speed. This unsupervised discovery of the wheel speeds from the CAN messages is enough to understand basic states of the vehicle (e.g., idle, accelerating, etc.) but is not robust to noise, e.g, a cyber attack manipulating CAN messages; further, we seek to understand the other CAN messages in terms of the vehicle states. Consequently, we leverage this to represent each data word's values as a linear combination of speed and acceleration. Finally, the groupings inform creation of an image over small time-windows of the CAN data, which permits CNN-based image processing techniques to accurately predict the state of the vehicle even with somewhat noisy (maliciously manipulated) CAN signals. This takes advantage of work such as described in Lemley et al. [10] which supports smart city integration [11].

We present initial results of this data-analytic workflow for making sense of the CAN data (learning the vehicles speed, grouping state-related data words, and knowing the vehicles driving state). We emphasize that given a few assumptions, this is a vehicle-agnostic workflow, which directly addresses the primary roadblock of wide variation

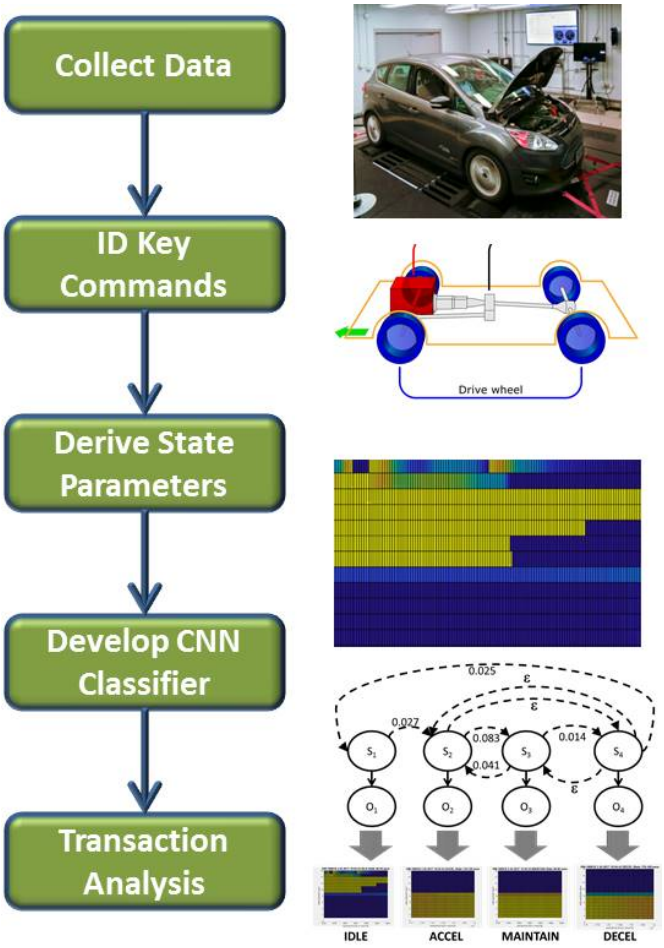

Fig. 3. Overview of Steps for Extracting States from CAN bus data packets

across models of CAN traffic-to-function mappings.

2 Multi-Stage State & System Discovery Process

We have developed a combination of supervised and unsupervised machine learning tools to extract the relationships between network data packets and the functions of the vehicle. The steps for deriving the states of the driver's actions from CAN bus traffic are described in Fig. 3. The flow begins with the collection of CAN bus traffic data from a vehicle that is being operated through a progression of system states. Then, several data driven approaches are applied to learn the attributes of a state and the sequences of states that are most likely to represent a driver's intentions. The novelty of this approach includes the use of physical system models to derive a relationship between several 1-D data streams and 2-D images that can be trained using CNN without retraining the whole system on each vehicle. The last step involves producing a trained system that provides near real-time state discovery from a vehicle's CAN bus traffic without training specifically on that make and model of vehicle.

2.1 Step 1 - Data Collection using Step Functions

The first challenge is to acquire data that is suitable for extracting useful data trends. Thus, one unique aspect

of this research involved collecting data in a way that is typically used for deriving transfer functions of physical systems. Namely, step functions (or impulse functions) are used to excite a system in order to derive its transfer function. This work is based on the idea that there is an underlying physical system that is to some extent reflected in the data carried by the CAN bus traffic. A series of abrupt speed and acceleration changes were chosen for the driver's actions in order to optimize the data driven approaches. A typical speed profile is shown in Fig.4.

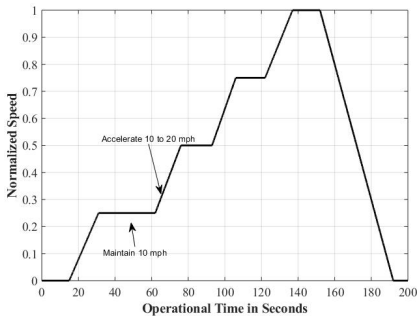

Fig. 4. A typical step function based driving profile used to collect data that is optimized for data driven analysis

Two fields of the CAN bus frame are discussed within this paper. First is the 11-bit arbitration field or PID. In order to protect the IP of the makes and models used in our research, we create pseudonyms for these values. The second field of interest is the data field which (for our purposes) always consist of 64 bits that we treat naively as four 16-bit data words. This assumption is correct for most, if not all, of the measured values being reported via the CAN bus. The authors plan to develop more sophisticated parsing of the data field in the future that would be able to automatically determine the length and number of data words contained in each PID's data field.

2.2 Step 2 - ID Wheel Speed

The second challenge was to identify key commands and data words from the hundreds of data words available using automated algorithms rather than the manual searches used by almost all other research efforts. Based on prior evidence from a small fleet of hybrid vehicles, it was noted that only one PID had four highly correlated data words and that each data word encoded a wheel's speed. This effort involved automatically determining a CAN bus data word that represented the speed in linear units. Specifically, the PID that most probably represents the wheel speed command is known to have a linear relationship with the vehicles speed. Given that each PID has 64 bits and that (at least for electric and hybrid vehicles) the 64 bits are divided into four 16-bit data words, a covariance function across all four data words should identify the key PID. Later, any inconsistencies would indicate a malfunction or a cyber-attack. This physical model and rule enabled a search algorithm to be developed that determines which of the more than 100 PIDs contains the wheel speed value by performing the covariance operation over all four data words for each PID and identifying the maximum value. Figure 5 shows the data from a hybrid vehicle representing these covariance functions and the identification of PID 37 as the wheel speed command.

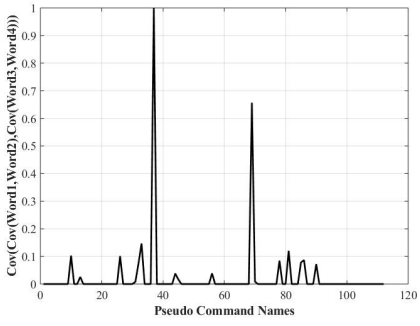

Fig. 5. A Comparison of the Covariances of All PIDs, Highlighting the Ability to Automatically Determine the Wheel Speed Command.

2.3 Step 3 - ID of Other PIDs Related to Speed and Acceleration

Although under normal conditions the states (Idle, Accel, Maintain, Decel) can be derived from the wheel speed alone, attacks or faults in the network would change the input values (CAN data) and thus necessitate additional data to handle more complex situations. So, we derive several other data trends in order to provide corroborating evidence that can be used to verify whether or not the wheel speed or related individual commands is being attacked by comparing them to each other. Similarly to the challenge of finding the wheel speed command, the problem here is that with undocumented commands and data fields, we have to extract information without knowing the exact parsing or protocol. To solve this, we developed a method that also utilizes data from a suite of related commands. Once the wheel speed command was identified, a covariance across all of the other data words was performed in order to map a collection of many CAN data words (communicating key functions of the vehicle) to the physical state (idle speed, accellerating, etc.).

The uniqueness was realized by combining features based on physical models of the system (in this case a moving passenger vehicle) with tools that are typically utilized for unsupervised machine learning in order to derive the constants for the equation describing the status of the vehicles systems.

Specifically, this algorithm involved two insights: 1) many vehicle sensors report speed and acceleration-related values via the CAN bus traffic, and 2) over a given time window, the velocity and the acceleration will be linearly independent. Therefore, the covariance of the data words versus speed and versus acceleration are plotted. Only the highest 25 values were plotted in order to identify key groupings of functions. The example from this research shown in Fig. 6 demonstrates how the data readings from all of the PIDs fall into clusters that can be grouped on the 2D plot that has speed on the X-axis and acceleration (first time derivative of speed) on the Y-axis.

The methods used to determine the optimum groupings of PID-based data sets included k-means clustering and tree diagrams. The k-means clustering with $k = 7$ results are shown in Fig. 6. Visual inspection shows that there are approximately seven clusters, five or six of which should be useful for mapping to physical systems related in a weighted fashion to speed and acceleration.

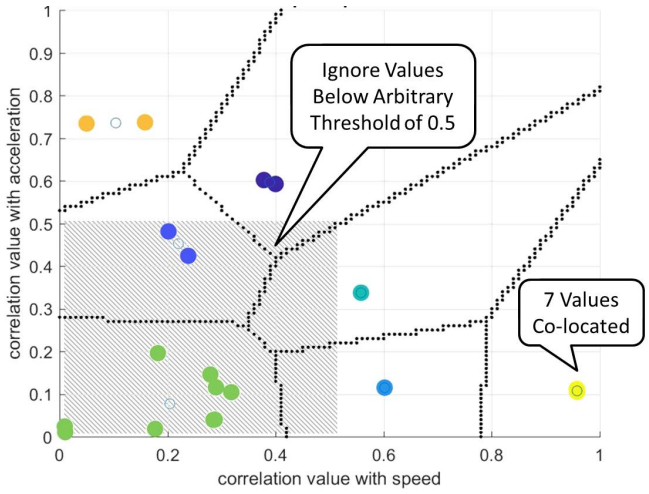

Fig. 6. Plot of the data words with the highest covariance values with the wheel speed (x-axis) versus covariance with acceleration (y-axis). Empirical analysis showed k-means with $k = 7$ clusters as optimal.

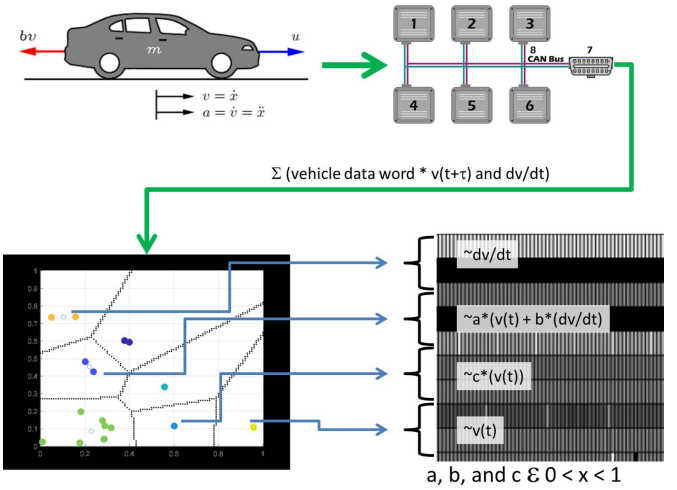

Fig. 7. This graphic shows how the previously defined features were placed on a 2D map associated with each state. The constants a, b, and c ∈ [0,1].

2.4 Step 4 - Derivation of State Hyperparameters

The next challenge addressed was that of optimizing the hyperparameters of the states. As mentioned previously, we did not assume that either the number of states or their exact boundaries were predetermined. So, we assumed four states (for this phase of the research) but explored methods for automatically determining optimal parameters to define where states stopped and started. In this case, the time duration and amplitude range of the state were intuitively chosen as the hyperparameters of interest that would be selected once and then used for all future mappings of states. For this research, the hidden states will represent the state of the accelerator pedal (increased pressure, neutral pressure or slackened pressure), and the observed emissions will be the actual acceleration, maintained speed, or deceleration of the vehicle.

Step 4 used the trend of this data word over a scripted set of driving data to optimize the time and amplitude (velocity) steps. In this case, a match score was developed that combines 1) the closeness of the match between the actual and predicted wheel speed, and 2) the accuracy of predicting the actual driver intent. The first value calculated was the covariance score

$$\text{CovScore}(\hat{V}) = \text{cov}(V, \hat{V})$$

where $V = (v_1, ..., v_n)$ is the actual velocity data and the predicted values are $\hat{V} = (\hat{v}_1, ..., \hat{v}_n)$ with v_i and $\hat{v}_i$ denoting the actual and predicted velocity at time t_i, respectively,

$$\hat{v}_i = \begin{cases} 0, & \text{if } m_i = \text{IDLE} \\ a_i + \hat{v}_{i-1}, & \text{if } m_i = \text{ACCEL} \\ \hat{v}_{i-1}, & \text{if } m_i = \text{MAINTAIN} \\ -a_i + \hat{v}_{i-1}, & \text{if } m_i = \text{DECEL.} \end{cases}$$

The second part of the score was based on the need to minimize the number of state transitions to match normal driver behavior. Thus, a penalty was added to the covariance score to eliminate rapidly changing state estimates.

Since there is a desire to have the smallest time window and amplitude value, there was a clear optimum intersection at a time window of about 1 second and an amplitude (velocity) of about one percent of the speed. Larger time windows give slight increases in velocity accuracy; however, this reduces the response time of any anomalous activity or network activity.

2.5 Step 5 - Mapping of CNN Images

One of the requirements addressed was the need for finding a way of combining all of the relevant data trends derived previously into a type of map or image that could be classified into the four states. The purpose was to combine multiple related data trends in a manner that would be both more robust to anomalies and attacks, and that would be vehicle make and model agnostic. Thus, we chose to make an image to represent the CAN data for a time window, with time increasing across the columns and data words changing from velocity-correlated to acceleration-correlated data down the rows. Then similar to Vann et al. [12] we use CNN for image classification since CNNs can learn relationships between objects in an image. Figure 7 shows the previous steps and how they map to the final state-based feature representation.

Figure 9 shows a few examples of each of the 2D images representing the four driver states. Each image was formed using the method shown in Fig. 7.

2.6 Step 6 - Transaction Analysis for Maximum Likelihood Sequences

The final challenge addressed is the fact that 100% accurate CNN image classification for the driver states is not achievable. Therefore, we employed transaction analysis using MATLAB built-in HMM functions to refine the sequence of estimated states from the CNN classifier. As discussed in several references, one cannot assume prior knowledge of the exact underlying states or numbers of states [13]. Since we are discovering states that have a time dependency, we first show an example of how states would evolve over time in the vehicle driver's mapping. In Fig. 8, adapted from a

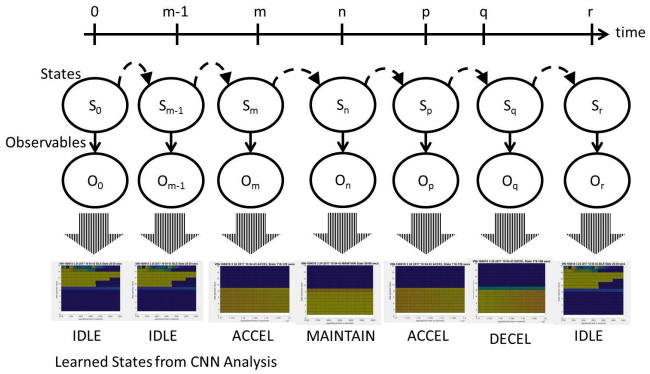

Fig. 8. The time sequence of HMM states derived from a typical set of CAN bus data.

TABLE 1
Confusion Matrix from CNN Blind Testing

True vs Est	IDLE	ACCEL	MAINTAIN	DECEL
IDLE	1	0	0	0
ACCEL	0.241	0.724	0.035	0
MAINTAIN	0	0.0385	0.9615	0
DECEL	0	0	0	1

similar model in [14], there are four states. As time values increase from 0 to m, n, p, q, and r, the states evolve through a normal driving cycle. Since the states are not directly observable, we rely on observed feature vectors represented graphically as images.

3 Hypothesis & Experiment

Here we present our first tests of the above methodology. The CNN feature images from one data capture from one vehicle were plotted and a few representative examples are shown in Fig. 9. A CNN was trained using approximately 40 images of each state. Testing was performed on the data that were held out for blind testing and the resulting confusion matrix is shown in Table 1. The methodology described above was developed for the purpose of being able to automatically build 2D maps on any vehicle. Standard MATLAB libraries were used to train on the CNN images. The resulting network was then used to classify the states of the driver's intentions of the vehicles which are depicted as the red symbols in Fig. 10. Finally, using the transition and emission matrices generated in MATLAB, the most likely states were derived which are shown as the solid line in the same figure. For this example, only one state was mislabeled in the original CNN classification. This one error was corrected by applying the likelihood algorithm in MATLAB. The overall model of the HMM, shown in Fig. 11, gives the relationship between the data trend images and the actual states and sequences used in the final analysis. As shown, certain transitions are illegal, such as IDLE to MAINTAIN, since a driver would have to go through the state of ACCEL before they could maintain a non-zero speed. Other illegal states include IDLE to DECEL, MAINTAIN to IDLE, and ACCEL to IDLE. The ϵ designation is for transitions that had very small but non-zero probabilities due to the artificial driving profile that was exercised. Once more real-to-life driving data is included these values may become more significant.

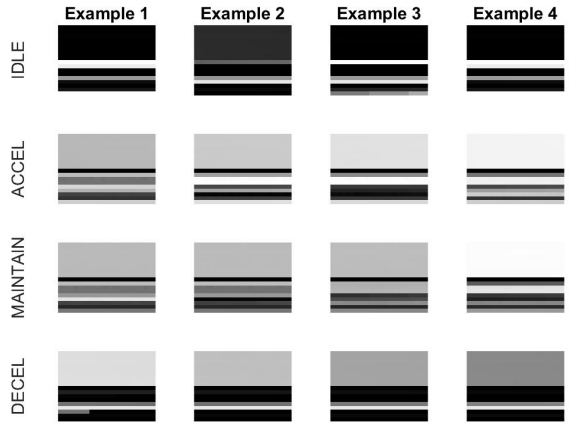

Fig. 9. Examples of CNN feature images of all four states.

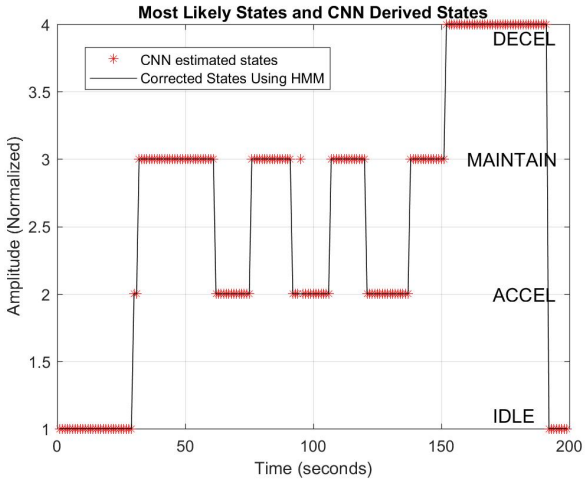

Fig. 10. The derived states are shown before and after using HMM to improve the state estimation.

4 Conclusions & Future Work

Our initial investigations of CAN bus traffic have concluded that there exists a combination of data driven methods for automatically 1) mapping data from CAN bus traffic to representative functions of the physical attributes of the vehicle, and 2) using that mapping to estimate the state of the driver's intentions. The classification outputs (IDLE, ACCEL, MAINTAIN and DECEL) provide the basis for cyber intrusion detection and establish an important first step for autonomous vehicle decision processes.

We have developed attack methods that simulate a highly sophisticated network intrusion that would not be detectable by most methods. The next phase of our research will involve applying the CNN image classification process to distinguish between normal and anomalous states or attacks. The methods described above represent several unique contributions including combining physical model

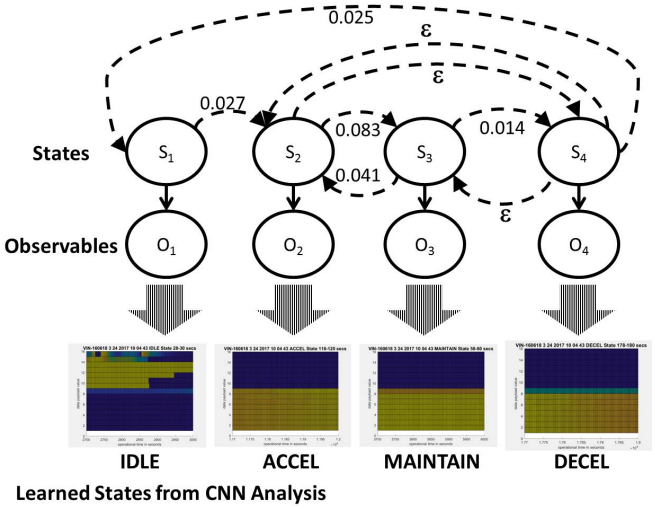

Fig. 11. The HMM model showing the relationship between driver action states and observed data trends in the form of images.

approaches with unsupervised learning as well as developing CNN-based approaches for determining the state of the driver's actions. The approach appears to be a promising avenue for accurate detection of an important class of CAN bus attacks.

Acknowledgements

This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The US government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

About the Authors

Michael Michael R. Moore (mooremr@ornl.gov) received B.S. and M.S. degrees from Mississippi State University in 1985 and 1987, respectively, in electrical engineering. He is a Senior Researcher and Program Manager in the Global Security Directorate at Oak Ridge National Laboratory (ORNL) with over 30 years experience in radio communications, machine learning and signal processing.

Robert A. Bridges (bridgesra@ornl.gov) received a BS in '05 and PhD in '12 both in mathematics from Creighton & Purdue Universities, respectively. Bobby works as an applied mathematician, developing algorithms to support information and cyber-physical security.

Frank Combs (combsfl@ornl.gov) is a research engineer that serves as the manager and experiment lead for the Vehicle Security Lab at ORNL. Frank has a BS in EE and over 40 years experience.

Adam L. Anderson (aanderson@tntech.edu) received the B.S. and M.S. degrees from Brigham Young University and the Ph.D. degree from the University of California at San Diego in 2008. He is currently a Senior Embedded Radio Frequency Systems Engineer at Oak Ridge National Laboratory with a Joint Faculty Appointment at Tennessee Technological University. Dr. Anderson was the winner of the 2014 DARPA Spectrum Challenge, received the 2014 Leighton E. Sissom Award for Creativity and Innovation, and is currently an Open Track Team lead in the 2018 DARPA Spectrum Collaboration Challenge.

References

[1] A. Munir, "Safety assessment and design of dependable cybercars: For today and the future." *IEEE Consumer Electronics Magazine*, vol. 6, no. 2, pp. 69–77, 2017.

[2] M. R. Moore, R. A. Bridges, F. L. Combs, M. S. Starr, and S. J. Prowell, "Modeling inter-signal arrival times for accurate detection of can bus signal injection attacks: a data-driven approach to in-vehicle intrusion detection," in *Proceedings of the 12th Annual Conference on Cyber and Information Security Research*. ACM, 2017, p. 11.

[3] J.-H. Lee and H. Kim, "Security and privacy challenges in the internet of things [security and privacy matters]," *IEEE Consumer Electronics Magazine*, vol. 6, no. 3, pp. 134–136, 2017.

[4] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," *Black Hat USA*, vol. 2015, 2015.

[5] K.-T. Cho and K. G. Shin, "Fingerprinting electronic control units for vehicle intrusion detection." in *USENIX Security Symposium*, 2016, pp. 911–927.

[6] R. McAllister and T. Bui, "State space abstraction for reinforcement learning," 2014. [Online]. Available: http://mlg.eng.cam.ac.uk/thang/docs/talks/rcc_rl.pdf1

[7] H. Bai, S. Cai, N. Ye, D. Hsu, and W. S. Lee, "Intention-aware online POMDP planning for autonomous driving in a crowd," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 454–460.

[8] W. Song, G. Xiong, and H. Chen, "Intention-aware autonomous driving decision-making in an uncontrolled intersection," *Mathematical Problems in Engineering*, vol. 2016, 2016.

[9] M. Jaynes, R. Dantu, R. Varriale, and N. Evans, "Automating ecu identification for vehicle security," in *Machine Learning and Applications (ICMLA), 2016 15th IEEE International Conference on*. IEEE, 2016, pp. 632–635.

[10] J. Lemley, S. Bazrafkan, and P. Corcoran, "Deep learning for consumer devices and services: Pushing the limits for machine learning, artificial intelligence, and computer vision." *IEEE Consumer Electronics Magazine*, vol. 6, no. 2, pp. 48–56, 2017.

[11] S. P. Mohanty, U. Choppali, and E. Kougianos, "Everything you wanted to know about smart cities: The internet of things is the backbone," *IEEE Consumer Electronics Magazine*, vol. 5, no. 3, pp. 60–70, 2016.

[12] J. M. Vann, T. P. Karnowski, R. Kerekes, C. D. Cooke, and A. L. Anderson, "A dimensionally aligned signal projection for classification of unintended radiated emissions," *IEEE Transactions on Electromagnetic Compatibility*, vol. 60, no. 1, pp. 122–131, 2018.

[13] C. P. Calderon and K. Bloom, "Inferring latent states and refining force estimates via hierarchical dirichlet process modeling in single particle tracking experiments," *PloS one*, vol. 10, no. 9, p. e0137633, 2015.

[14] J. Liu, L. Zhu, Y. Wang, X. Liang, J. Hyyppä, T. Chu, K. Liu, and R. Chen, "Reciprocal estimation of pedestrian location and motion state toward a smartphone geo-context computing solution," *Micromachines*, vol. 6, no. 6, pp. 699–717, 2015.