Solar Car Telemetry System
(Technical Topic)

Analyzing the Ethical Issues Surrounding Autonomous Vehicles From a Data Feminism Standpoint
(STS Topic)

A Thesis Prospectus
In STS 4500
Presented to
The Faculty of the
School of Engineering and Applied Science
University of Virginia
In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science in Your Major

By

Joon Kim

November 1, 2021

Technical Team Members:
Victor Pham
Vinay Bhaip

On my honor as a University student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments.

ADVISORS

Sean Ferguson, Department of Engineering and Society

Rosanne Vrugtman, Department of Computer Science

Daniel G. Graham, Department of Computer Science

Introduction

Any technological device that relies on human computer interaction must collect, process, and present data to the user in order for them to make informed decisions and ensure the device's success. Take the example of a car; the user must constantly be monitoring certain dynamic pieces of information such as speed, gas level, and revolutions per minute to adjust their driving habits accordingly. More importantly, it eliminates the blindness between the user and any potential life threatening issues that may arise. This is part of the reason why my team has initiated the development of a solar car telemetry system for the Solar Car Team at UVA. This solar car telemetry system is a web application that receives and displays data that is collected from the solar car in real time. It displays a variety of data points concerning the safety of operation as well as the performance of the vehicle's various components.

Recent technological advancements, most notably with autonomous vehicles, have attempted to shift the responsibility of making informed decisions away from humans and to the vehicles. The STS topic of this paper will use the framework of data feminism to uncover some of the ethical issues associated with the training and operation of autonomous vehicles. Many who favor the use of autonomous vehicles argue that autonomous vehicles will significantly reduce the number of vehicle related accidents caused by human error, but autonomous vehicles are prone to unique errors of their own. Autonomous vehicles are trained using machine learning techniques, resulting in a heavy reliance on data quality. A particularly problematic way in which data quality can be hindered is when biases are introduced to the datasets they algorithms train upon or the algorithms themselves, resulting in models that may end up favoring these biases. A data feministic approach at analyzing practices associated with machine learning in autonomous vehicles will allow me to explore the ways in which these systems may not be truly neutral when placed in society, as well as potentially shed light on the ways this issue may be addressed.

Solar Car Telemetry System

A dashboard developed for the Solar Car Team at UVA

Joon Kim
Computer Science
University of Virginia School of Engineering and Applied Sciences
Charlottesville, VA
jtk2rw@virginia.edu

Victor Pham
Computer Science
University of Virginia School of Engineering and Applied Sciences
Charlottesville, VA
vbp8vgc@virginia.edu

Vinay Bhaip
Computer Science
University of Virginia School of Engineering and Applied Sciences
Charlottesville, VA
vb4ztv@virginia.edu

ABSTRACT

The goal of the Solar Car Team at the University of Virginia is to design, implement, and maintain efficient and effective solar powered cars that can compete with other solar cars from across the nation. To keep the cars competitive, designers and engineers must constantly improve and maintain the car, but such changes cannot be proven as reasonable without data that would warrant them. Additionally, it is important for drivers to monitor vehicle performance so they are able to make informed judgements on potential dangers to the vehicle and/or driver. To address this issue, I worked with a team that created a web based application that displays data collected from the Solar Car. We created the application using asynchronous websockets in the frontend and a Python centered backend. We also collected and saved data for future viewing and analysis. The web application consists of a few pages, each dedicated to a different component of the car (solar panels, motors, battery), where various data are displayed and updated in real time. The application gives members of the team a means of monitoring and judging the performance of various parts of the car as it's on the road to assist the team with future design changes and improvements. Currently, the system is still a work in progress,

INTRODUCTION

Creating a piece of technology or software that would allow for use without difficulty is only half the effort to creating a successful long term product. To conduct maintenance and additional research in order to improve upon the product, the performance of the product must be monitored through data, where potential faults and shortcomings may be identified. In addition, the safety of the product may not be guaranteed without proper monitoring of certain components that may pose any danger to the user. The importance of tracking data for these purposes are amplified when put into the context of the goal of the Solar Car Team at the University of Virginia. The mechanical components of the car must be improved and reworked every year in order to stay competitive with other solar cars racing against it. In addition, while gas provides a stable reliable source of fuel to cars, solar power must be constantly monitored to make sure enough power is flowing through the car for the duration of the drive. It is for these reasons that spurred the motivation to develop a telemetry system for the solar car.

Telemetry is defined as "the process of recording and transmitting the readings of an instrument [1]. The solar car telemetry system in particular handles the displaying of collected data through a web application. Data is collected from the car and is serialized and transmitted/received wirelessly through the car and computer. The reception of asynchronous and continuous data is made possible using Socket.IO, a javascript library that allows for the establishment of websockets (bidirectional communication channel) between the web application and the computer that is receiving the data [2]. Each individual data value is assigned a key, and the javascript uses this to display the data to their respective positions on the UI as the web socket detects new incoming data. The design of the dashboard UI was implemented using the frontend HTML and CSS framework called Bootstrap.

RELATED WORKS

There have existed many similar past projects, each of which have attempted to create efficient telemetry communication systems. Hackystat, starting development in 2001, is a tool to provide "automated collection and analysis of software engineering process and product data", with an aim to simulate a "telemetry-based approach to software measurement trend definition and display" [3]. It was initially developed with a focus on communication between client and server, but interestingly was later reworked completely to use a service oriented architecture to process more types of data and stakeholders. The server oriented architecture uses distributed computation, caching, and pre-fetching to preserve the quicker asynchronous performance that comes with client-server architecture [3].

Many similar telemetry systems for solar cars have been implemented by other teams with similar motives and functionality. The telemetry system developed by the Missouri S&T Solar Car Team is one such example. They similarly utilize a Controller Area Network (CAN) and broadcast data through a CAN to Ethernet bridge using the User Data Protocol (UDP) [4]. The team uses a chase vehicle that will stay within close proximity to the solar car “running the telemetry software layer full time logging data from the vehicle and reading data off to the team” [4]. The goals of our team may prove more difficult compared to this approach, as we hoped to be able to receive data with a strong enough connection such that the host computer running the software would be able to remain in a stationary position. However, this goal may be unrealistic considering the current budget of the team and vast size of the race track.

SYSTEM DESIGN

4.1 Review of System Architecture

The architecture of the solar car telemetry system can be divided into 3 different tiers: the vehicle layer, the server, and the client.

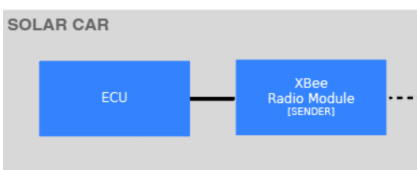

Figure 1. Vehicle Layer

The vehicle layer exists inside of the solar car itself and holds the electronic control unit (ECU) which collects data and sends it to the sender radio module. The sender radio module in turn transmits the data to the server layer.

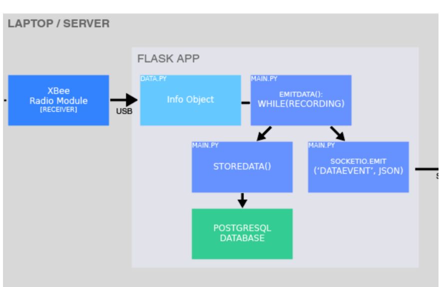

Figure 2. Server side architecture

The server layer is connected to the vehicle layer via a radio module, which acts as the receiver for the data transmitted from the radio module on the solar car which acts as the sender. This data is then transferred to the server via a simple USB connection. The server side of the application contains a Postgresql database for the storing of incoming data, and creates the websockets for the transmission and communication with the client.

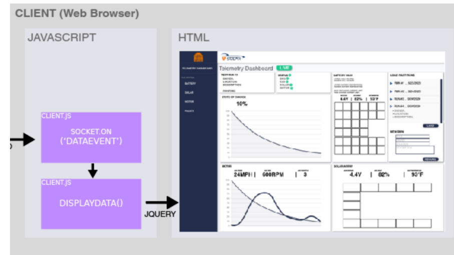

Figure 3. Client side architecture

Lastly, the client side layer receives and displays the processed data sent by the server through a websocket. This layer contains the front-end code for creating the dashboard UI by rendering HTML templates. It also contains javascript code for the rendering and updating of the graphs on the dashboard.

4.2 Web Page Requirements

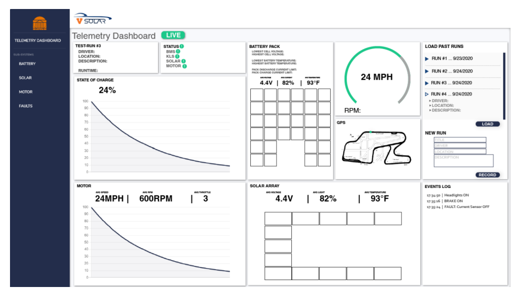

Figure 4. Finished dashboard concept

Ideally, the client side of the web application should be usable to all members of the team. Thus, it is important that the web application displays the appropriate data that each sub team may find useful for analysis. For these reasons the team has decided to divide the web application into different pages for each subteam (battery, solar, and motor). The battery page should display a diagram of the layout of all battery cells, along with their respective temperatures within each cell in the diagram. General statistics such as average battery temperature, lowest/highest cell voltage, pack amp hours, etc will be displayed as well in a seperate table. Similarly, the solar page will also display a diagram showing the arrangement of solar panels on the car. Each cell in the diagram will display the respective solar panel’s generated voltage, light level, and temperature. The averages of these statistics across solar panels will also be displayed. The motor page will display relevant statistics corresponding to each motor on the car such as revolutions per minute (RPM), temperature, throttle, min/max voltage, and min/max temperature.
The web application will also have a general “homepage” that will display a broad overview of all areas of the car. From this screen, users will also be able to initiate

recordings of data as well as see the physical location of the car through GPS.

4.3 Key Components

4.3.1 Socket.IO

As mentioned earlier, the real time displayal of data as it is transmitted from the solar car is made possible with websockets. We have specifically used a javascript framework called Socket.IO for our websocket implementation. Socket.IO allows for the establishment of a "full-duplex communication channel" that allows for "real-time, streaming updates between a web server and a browser client" [5]. This communication between server and client can be represented through "emit events" and "listener" methods from either side. Mainly, the "socket.send method will send the message on the socket", classified by an event name, while socket.on listener methods calls corresponding event handlers and "is triggered when a message sent with socket.send is received [5].

Figure 5. Emitting and listening for Socket.IO events

Our web application uses the socketio.emit method instead, which is identical to socket.send but used to trigger custom events. When the server receives data from the solar car, it immediately emits it as a event named "dataevent" with the data formatted using javascript object notation (JSON). On the client side, the socket.on method listens for this "dataevent" from the server. As soon as it receives the dataevent, it executes the javascript to display the JSON data onto the dashboard.

4.3.2 Database and Runs System

A database proves essential for any task that requires data to be stored and retrieved for future use. A database is "an organized collection of structured information, or data, typically stored electronically in a computer system" [6]. For our web application, we used a free and open source data management service named PostgreSQL to manage our tables in which we would be storing data. Data was stored and retrieved primarily through our "runs" system. A user can initiate a run on the main dashboard, which will set a boolean variable to true. While this variable is true, data that is emitted from the socket.emit method will be formatted as JSON before being sent to a method named storeData. The storeData method itself will then unpack the JSON and store each attributed key value pair into their respective columns in the database. Each entry in the table will also be associated with the ID of the current run that is determined by user input.

Queries are made to the database as the user requests runs to be loaded onto the web application. When the user clicks a button to load the run , an event is emitted with the corresponding run id that is listened for by the server. When the server catches this event, it initiates the query and returns all data associated with the run id, and emits this back to the client as another event. Lasty, the client catches this method, loops through the data entries, and displays the data accordingly.

4.3 Challenges

We as a team have encountered many challenges since the beginning of development. One such challenge that still stands today is the need to optimize the performance of the dashboard. The application must handle and process a large amount of incoming data that is sent at a very quick rate, and must constantly update graphs and values on the dashboard in order to display the most up to date statistics about the components of the car. The team has considered a few solutions to address this. Socket.IO comes with a sleep() method that will pause the channel for a set amount of time. We have placed an instance of this method inside of our method that will emit data, in order to slow down the rate at which data is being sent to the client. We currently have the channel to sleep in 1 second intervals between emissions of data [7]. The team has also discovered that dividing the application to separate pages based on different areas of the car (see section 4.2) would help in that less data would need to be updated on the current page view. Another challenge encountered more early on in development was the process of being able to associate incoming data with correct corresponding parts of the dashboard. Data came into the server without any specific label that would help in being able to identify values apart from each other, but did come in a certain sequence. With this, we were able to associate each value with a specific key in a python dictionary using labels that we defined. With this, the javascript could pull values from using the labels of key names from the dictionary.

RESULTS

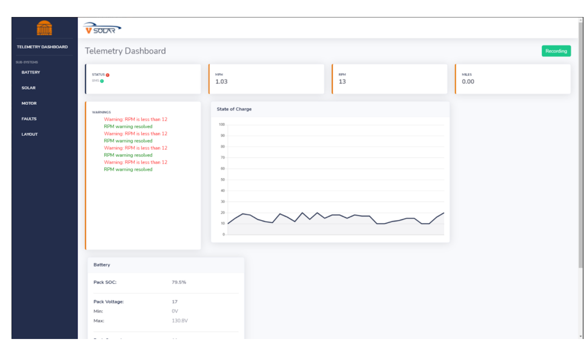

Figure 6. Current dashboard implementation

Although not yet in widespread use by the solar car team due to incomplete development of the web application and of the solar car, the web application was able to simulate the receiving, processing, and displaying of data with a class that would generate random values and send the data through a fake serial port. These random values were able to be successfully displayed onto the dashboard in an asynchronous manner. Teammates were successfully able to access the dashboard by connecting to the host computer's public IP address, which was running the application on a local server.

COURSEWORK RELEVANCE

Although this project was conducted during my first year at the university, I found the knowledge acquired by CS2150 (Program and Data Representation) to be helpful for the context of this project. Particularly, acquired knowledge about hashmaps proved essential at optimizing the speed of the application. As hash maps provide for constant time access for values [8], our web application was able to access and display values passed in by the server at a desirable pace. The convenience of being able to access values by their associated keys eliminated the need to display values on the dashboard in a specific sequence according to value positions in the data, as well as the need for searches in the data.

FUTURE WORK

There are many ways in which this web application can be expanded to provide more use for the team. Currently, we don't utilize the database in any other way other than storing data associated with runs. We have agreed that more meaningful graphs and statistics could be produced by associating data values together and creating new values through calculations between values, but lack of knowledge with database management and querying languages such as SQL has prevented us achieving any desirable results with this. There are many more optional features that may be added to the web application that may be useful for the team. We have started looking at developing a GPS system using MapBox's API but have only gone so far as to displaying/updating the current position of the car on the map. The team has yet to find a way to transmit GPS coordinates to the web application, but some possibilities have been explored such as a physical GPS module or a phone that will transmit the coordinates with the rest of the data.

CONCLUSION

Overall, the telemetry web application would be a very useful tool for the team as it would allow members of the team to make more informed design decisions based on observations in data displayed by the dashboard. The asynchronous nature of the web application will always keep the team up to date with the performance of the car, and will allow team members to help the driver make more informed decisions on driving. The database and runs system will allow team members to look back on collected data to help identify strong and weak elements of the car. Although development is yet to be complete, the web application has much potential to provide much more benefit to the team. It can be expanded to include many more features and applications in data analysis, as well as more types of charts that may be more insightful than simple digits. As how the possibilities with computer science are limitless, so can this web application given enough research and development.

REFERENCES

[1] Lexico. Telemetry English definition and meaning. Retrieved October 2021 from https://www.lexico.com/en/definition/telemetry

[2] Socket.IO. Introduction: What Socket.IO is. Retrieved 24, 2021 from https://socket.io/docs/v4/

[3] Johnson, Philip & Zhang, Shaoxuan & Senin, Pavel. (2009). Experiences with hackystat as a service-oriented architecture.

[4] Eric Walter, Nicholas Glover, Jesse Cureton, and Kurt Kosbar. Telemetry System Architecture for a Solar Car. *International Telemetering Conference Proceedings*.

[5] Rohit Rai. 2013. *Socket.io real-time web application development build modern real-time web applications powered by Socket.io*, Birmingham: Packt Pub.

[6] Oracle, "What is a database?" Retrieved October 2021 from https://www.oracle.com/database/what-is-database/

[7] Socket.io. API. Retrieved October 2021 from https://python-socketio.readthedocs.io/en/latest/api.html

[8] Oracle, "HashMap" Retrieved October, 2021 from https://docs.oracle.com/javase/8/docs/api/java/util/HashMap.html

STS Topic

With the introduction of autonomous vehicles, there has been a shift in this responsibility away from the driver and to the machine, taking the “human” out of human computer interaction. However, even with the mitigation of human error with widespread adoption of autonomous vehicles, autonomous vehicles may still be presented with the same dilemma that humans cannot avoid. For example, in the event of an unavoidable accident, the machine must make a decision about who will be harmed, including potentially the driver. This issue is more complex with machines, because although we as humans each have our own set of morals, Technology must follow a universal moral code that is predetermined by industry experts responsible for programming and training the cars (Zhou 2019). From the initial data used to train them, to the algorithmic decisions made, there is always the possibility of some bias being introduced that may shape and skew this moral code machines will adopt. More specifically, underrepresentation, overrepresentation, and misrepresentation of demographic populations in data and algorithms all introduce potential for biases in autonomous vehicle performance. Using D’Ignazio and Klein’s framework of data feminism, I would like to explore the ethical challenges associated with the operation of autonomous vehicles and the complexities of addressing these challenges using the principles of data feminism.

Section 1: Representation of demographic groups on algorithmic training and performance for pedestrian recognition

One task an autonomous vehicle must be able to perform is to be able to identify pedestrians by training upon large amounts of images consisting of pedestrians and objects. This data can become biased in many ways. One such way is representation bias where “the development sample under-represents some part of the population, and subsequently fails to generalize well for a subset of the use population” (Suresh and Guttag 2021). Representation bias can stem from existing sociocultural differences between demographic populations, and can be difficult to address without an awareness and understanding of these differences. For example, “women in the United Arab Emirates are socially stigmatized against photographing their faces, skewing the availability of this type of data to be lower than the true distribution” (Jo and Gebru 2020). The importance of addressing the mere existence of these sociocultural differences is further emphasized by the variations in volumes of demographic populations between regions and countries. To mitigate the possibility for these algorithms to be influenced by such biases, it is important to understand the context in which data comes from and what the data is composed of and ensure equal representation of demographic groups. Data collection should be enforced for equal representation of all demographic populations in order to maintain applicability in the realistically diverse social environments these vehicles will operate in for any social context.

Much like humans, machine-learning algorithms are able to classify demographic groups according to certain stereotypical features after sufficient training. These stereotypical features include but are not limited to skin tone, stature, and the color, length, and texture of hair. Determining fair and equal Image classification algorithms have remained a challenging problem for machine learning algorithms for a long period of time, with facial and entity recognition tests conducted by the National Institute of Standards and Technology (NIST) determining that these algorithms have “a harder time recognizing people with darker skin” and “consistently found that they perform less well for women than men" (Simonite 2019). Additional research efforts have determined many possible distinct facial attributes as possible contributors for these biases, with features such as "faces with No-Beard perform worse than faces with a beard, such as a 5 o Clock Shadow" and "a strong correlation between Square Face and Male" (Terhorst et al 2015). This problem encompasses the issue that autonomous vehicles encounter when performing various object recognition algorithms on the road between people and objects scanned by the vehicle. Misrepresentation and unequal representation of physical characteristics introduces room for unforeseen algorithmic biases which may lead to misclassification of pedestrians. It is important that these gender and demographical stereotypical differences are carefully considered and accounted for in the data used to train the algorithms, and that the algorithms are tuned accordingly based on these differences. However, determining how exactly to tune the algorithms to account for this may be a challenge in and of itself.

Section 2: Algorithm biases from gender characteristics

Unconscious biases may also be introduced to machine learning datasets and algorithmic decisions from the ways in which different gender characteristics may be reflected upon the data. According to data collected from the Insurance Institute for Highway Safety, many more male drivers in the U.S have been killed than female drivers (NHTSA). This may be correlated to gender stereotypical behaviors. Research findings have shown that "male drivers are more likely – compared to female drivers – to exhibit various patterns of aggressive driving, such as cutting another vehicle, honking the horn, or exhibiting road rage" (Fountas et al 2019), and such characteristics may be in turn adopted to some extent by algorithms trained upon it. Emerging research which aims to train autonomous vehicles to exhibit more human driving behavior should be weary of these stereotypical differences. In one specific study, researchers have utilized data features such as “deceleration rates, stopping/slowing speeds, stopping/slowing durations and acceleration rates while participants drove specific routes in Los Angeles.” (Leavy 2018). These characteristics reflected upon collected data may have profound impacts especially on neural network algorithms used by autonomous vehicles, where “even the slightest bias present into the initial set of data points will be ‘ingrained’, or ‘encoded’, in resulting neural network models” (Cheong et al 2015). This goes to suggest that an imbalance in representation as well as an algorithmic design that fails to account for these characteristic differences can lead to biases in the driving techniques these autonomous vehicles end up utilizing. With a more heavy representation of male driving patterns in data, autonomous vehicles may learn to adapt more aggressive techniques that may put the general consumer base more at risk for accidents.

Next Steps

In this thesis, I will examine the ways in which various demographic groups may be excluded from data collection and implementation for artificial intelligence/machine learning, and the discriminatory effects it may show reflected upon the algorithmic performance. Due to established industry practices without a strong emphasis on pluralism and equal representation from demographic groups in data, I would like to elaborate on the many challenges on implementing the principles of data feminism as explained by D'Ignazio and Klein to this particular field. Some changes can be enforced by simple changes in policy, while some will prove more difficult and require a much deeper reworking of deeply rooted industry norms and the status quo. Finally, I would like to explore the communities of mobilized publics, social movements, activist groups, and their responses to this lack of representation in this industry. More specifically, what sorts of radical actions are being conducted by these groups to advocate for changes in these data collection/ algorithm implementation practices.

References

Simonite, T. (2019, July 22). *The best algorithms still struggle to recognize Black Faces*. Wired. https://www.wired.com/story/best-algorithms-struggle-recognize-black-faces-equally/.

Terhorst, P., Kolf, J. N., Huber, M., Kirchbuchner, F., Damer, N., Morales, A., Fierrez, J., & Kuijper, A. (2021). A comprehensive study on face recognition biases beyond demographics. *IEEE Transactions on Technology and Society*, *14*. https://doi.org/10.1109/tts.2021.3111823

Cheong, M., Lederman, R., McLoughney, A., Njoto, S., Ruppanner, L., & Wirth, A. (2015). *Ethical implications of AI bias as a result of workforce gender ... - teachers mutual bank*. https://www.tmbank.com.au/-/media/unibank/about-us/member-news/report-ai-bias-as-a-result-of-workforce-gender-imbalance.ashx.

Buolamwini, J., Gebru, T., & Wilson, C. (n.d.). *Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification*. http://proceedings.mlr.press/v81/buolamwini18a/buolamwini18a.pdf.

Fountas, G., Pantangi, S. S., Hulme, K. F., & Anastasopoulos, P. C. (2019). The effects of driver fatigue, gender, and distracted driving on perceived and observed aggressive driving behavior: A correlated grouped random parameters bivariate Probit Approach. *Analytic Methods in Accident Research*, *22*, 100091. https://doi.org/10.1016/j.amar.2019.100091

Automated vehicles for safety. NHTSA. (n.d.). https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety.

Zhou, A. (2019). The intersection of ethics and ai. *AI Matters*, *5*(3), 64–69. https://doi.org/10.1145/3362077.3362087

Suresh, H., & Guttag, J. (2021). Understanding potential sources of harm throughout the machine learning life cycle. *MIT Case Studies in Social and Ethical Responsibilities of Computing*. https://doi.org/10.21428/2c646de5.c16a07bb

Leavy, S. (2018). Gender bias in artificial intelligence. *Proceedings of the 1st International Workshop on Gender Equality in Software Engineering*. https://doi.org/10.1145/3195570.3195580

Tavassoli, A., Cymbalist, N., Dunning, A., & Krauss, D. (2019). Learning from human naturalistic driving behavior at stop signs for Autonomous Vehicles. *SAE Technical Paper Series*. https://doi.org/10.4271/2019-01-1021