Northwestern Journal of Technology and Intellectual Property

Volume 20 | Issue 3 Article 3

Spring 2023

THE EVIDENTIARY IMPLICATIONS OF INTERPRETING BLACK-BOX ALGORITHMS

Varun Bhatnagar

N O R T H W E S T E R N
JOURNAL OF TECHNOLOGY
AND
INTELLECTUAL PROPERTY

THE EVIDENTIARY IMPLICATIONS OF INTERPRETING BLACK-BOX ALGORITHMS

Varun Bhatnagar

April 2023 VOL. 20, NO. 3

Volume 20, Number 3 (2023)
Northwestern Journal of Technology and Intellectual Property

THE EVIDENTIARY IMPLICATIONS OF INTERPRETING BLACK-BOX ALGORITHMS

*Varun Bhatnagar**

I. INTRODUCTION

Biased black-box algorithms have drawn increasing levels of scrutiny from the public. This is especially true for those black-box algorithms with the potential to negatively affect protected or vulnerable populations.[1] One type of these black-box algorithms, a neural network, is both opaque and capable of high accuracy. However, neural networks do not provide insights into the relative importance, underlying relationships, structures of the predictors or covariates with the modelled outcomes.[2] There are methods to combat a neural network's lack of transparency: globally or locally interpretable post-hoc explanatory models.[3] However, the threat of such measures usually does not bar an actor from deploying a black-box algorithm that generates unfair outcomes on racial, class, or gendered lines.[4]

Fortunately, researchers have recognized this issue and developed interpretability frameworks to better understand such black-box algorithms. One of these remedies, the Shapley Additive Explanation ("SHAP") method, ranks determinative factors that led to the algorithm's final decision and measures the partial effects of the independent variables that were used in the model.[5] Another, the Local Interpretable Model-agnostic Explanations

* J.D. Candidate, Northwestern Pritzker School of Law, 2023; M.B.A. Candidate, Northwestern Kellogg School of Management, 2023; B.A., Northwestern University, 2015. I would like to thank Professor Daniel Linna for supervising my note-writing process. This note would not be possible without the edits by the JTIP Production, Notes, Associate, and Executive Editor teams. Lastly, I would like to thank my family, who supported me throughout the writing process. Any errors in this piece are my own.

1 *See* VIRGINIA EUBANKS, AUTOMATING INEQUALITY: HOW HIGH-TECH TOOLS PROFILE, POLICE, AND PUNISH THE POOR (2018).

2 *See generally* Zhongheng Zhang et al., *Opening the black-box of neural networks: methods for interpreting neural network models in clinical applications,* 66 ANNALS OF TRANSLATIONAL MED. 216, 216 (2018).

3 Mengnan Du et al., *Techniques for Interpretable Machine Learning* (2019) (unpublished manuscription file with Texas A&M University's Department of Computer Science and Engineering), https://arxiv.org/pdf/1808.00033.pdf [https://perma.cc/N4B8-59YU]. These methods include ethics reviews, proper code documentation, and the usage of clean, audited data sources.

4 EUBANKS, *supra* note 2, at 205.

5 *Interpretability,* CLOUDERA FAST FORWARD, https://ff06-2020.fastforwardlabs.com/ [https://perma.cc/GB8T-QNU5].

("LIME") method, uses a similar method to reverse-engineer the determinative factors harnessed by the algorithm.[6] Both the SHAP/LIME methods have the potential to shine light into the most accurate, precise black-box algorithms.

These black-box algorithms can harm peoples' physical being and property interests.[7] However, algorithm developers currently hide behind the nominally impenetrable nature of the algorithm to shield themselves from liability. These developers claim that black-box algorithms are the industry standard, due to the increased accuracy and precision that these algorithms typically possess. However, SHAP/LIME can ascertain which factors might be cloud the judgement of the algorithm, and therefore cause harm. As such, SHAP/LIME may lower the foreseeability threshold currently set by tort law and help consumer-rights advocates combat institutions which recklessly foist malevolent algorithms upon the public.

Part II will provide an overview of the SHAP/LIME methods, as well as applying it to a tort scenario involving a self-driving car accident. Part III will cover the potential tort claims that may arise out of the self-driving car accident, and how SHAP/LIME would advance each of these claims. SHAP/LIME's output has not yet been compared to the foreseeability threshold under negligence or product/service liability. There are numerous factors that sway SHAP/LIME both towards and against reaching that threshold. The implications of this are severe—if the foreseeability threshold is not reached, a finder of fact might not find fault with the algorithm generator. Part IV will cover the evidentiary objections that might arise when submitting SHAP/LIME-generated evidence for admission. Reverse-engineering an algorithm mirrors crime scene re-creation. Thus, the evidentiary issues involved in recreating crime scenes appear when reverse-engineering algorithms.[8] Important questions on relevance, authenticity, and accessibility to the algorithm directly affect the viability of submitting evidence derived using either the SHAP or LIME methods.[9] Part V will conclude by contextualizing the need for transparency within an increasingly algorithm-driven society.

I conclude that tort law's foreseeability threshold is currently not fit for purpose when it comes to delivering justice to victims of biased black-box

---

[6] *Id.*

[7] Joy Buolamwini, *How Do Biased Algorithms Damage Marginalized Communities?*, NPR (Feb 26, 2021), https://www.npr.org/2021/02/26/971506520/joy-buolamwini-how-do-biased-algorithms-damage-marginalized-communities [https://perma.cc/3EEX-V766].

[8] *See generally* Paul W. Grimm et al., *Artificial Intelligence as Evidence*, 19 NW. J. TECH. & INTELL. PROP. 9, 84–94 (2021).

[9] *Id.*

algorithms. As for complying with the Federal Rules of Evidence, SHAP/LIME's admissibility depends on the statistical confidence level of the method's results. I conclude that SHAP/LIME generally have been properly tested and accepted by the scientific community, so it is probable that statistically relevant SHAP/LIME-generated evidence can be admitted.[10]

II. DESCRIPTION OF SHAP/LIME ALGORITHMS

Before explaining these methods admissibility or tort law's insufficiency, one must understand the SHAP and LIME methods. As the introduction demonstrated, explaining these methods can be complicated. Thus, this paper will use a specific example: a self-driving vehicle algorithm (going forward, "SDA").

A. Legend

Part II will include lots of terms and concepts. For reference, please use this legend section.

Confounding factor- a factor in the SDA that leads to a biased real-time output. This bias might be caused by not including the factor in the training data or the factor acting in an unexpected manner during the accident. For

[10] *Interpretability*, *supra* note 6, at 42–48.

example, a self-driving car might not know how to identify a kangaroo in the road, and made an action accordingly.

End-to-end control system- an SDA that can handle any environment. It can map any real-time input provided into a real-time output action.

Environment- surrounding area within the SDA's sensors

Explanation model- the output of a SHAP/LIME analysis. It reveals which factors were most important in determining the real-time output of the SDA.

Results from the explanation model- the rankings of factors that proved to be most determinative in generating the real-time output. Each of these ranking have a statistical confidence level.

Factor- any object or third-party action or movement in the self-driving camera feed that affects the solution.

Overriding factor- the object in the image that determines the solution. Its appearance completely affects the prescribed output or real-time output. For example, a pedestrian crossing in a cross-walk when they have the walk sign will override the original action and cause the car to stop.

Prescribed output- the best action that the car should take given the sample input image. This is typically set by a human.

Example of a prescribed output:

Prescribed output:
Accelerate from the traffic light to 35 mph

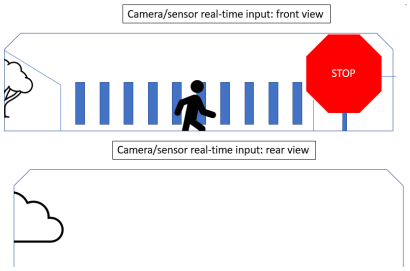
Prescribed complete output- the best action that the car should take given a stream of sample input images.

Example of a prescribed complete output:

Prescribed complete output:
Accelerate from the traffic light to 35 mph. Follow the car in front leaving 20 meters of space. If a pedestrian enters the street, immediately apply the emergency brake.

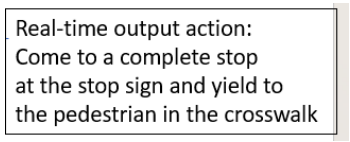
Real-time input- In real time, the images that are being filmed by cameras facing outside the car. These images are then stitched together to form a 360-degree external view of the car.

Example of a real-time input:



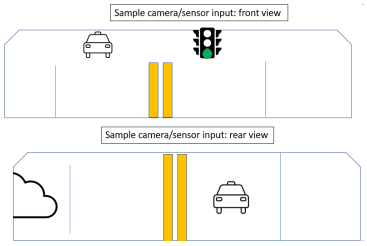
Real-time output- In real time, the action that the car takes based on the input image and SDA.

Example of a real-time output:



Sample input image- a sample image of any environment. Imagine a photographer in your passenger seat taking photos for future use. It is meant to emulate an environment that an SDA might encounter.

Example of a sample input image:



SDA- Self-driving algorithm

Statistical technique- A method to analyze an algorithm. In this case, SHAP/LIME are statistical techniques that will be used to analyze an SDA.

Training data set- a database of sample input images tagged with a prescribed output. Think of the data stored as ordered pairs: (sample_input_image_1, prescribed_output_1). Each sample input image mimics real-time input. Each prescribed output mirrors the format of real-time output.

SHAP/LIME variable effect- the quantifiable effect that the presence of an object/factor has on the real-time output.

B. Self-driving algorithm generation and training

How does a self-driving algorithm (SDA) work? The SDA receives a real-time input of exterior traffic scenes and generates a real-time output for the car to take. The algorithm maps the environment (via input images) into real-time prescribed complete output action.[11] This real-time prescribed complete output action could be the algorithm directing the car to change one lane to the right, stop at the stop sign, or take the second exit at the roundabout. An end-to-end control system can map a relevant action for

[11] SHAHIN ATAKISHIYEV et al., *Explainable Artificial Intelligence for Autonomous Driving: A Comprehensive Overview and Field Guide for Future Research Directions* at 20 (Dec. 21, 2021), https://arxiv.org/pdf/2112.11561.pdf [https://perma.cc/7ZET-Q7KG]. This process is termed a control system.

every possible environment; hence, SDAs that can account for *any* type of scenario is an end-to-end control system.[12]

Say that an important object is detected in the image feed, such as a pedestrian walking across a crosswalk. This will be the overriding factor in determining the real-time prescribed complete output. The pedestrian serves as an example of an "overriding factor." Overriding factors are parameters or objects that affect the algorithm. All these deterministic images as "factors," and the various factors that a SDA utilizes in making its real-time output action can be listed as {F1, F2, . . . }. For example, a real-time input image might contain factors such as two pedestrians {F1, F2}, a crosswalk {F3}, and a sign denoting a school zone {F4}. These factors {F1, F2, F3, F4} might meld together to create a single overriding factor.

Admittedly, this process is a simplified version of a multi-step algorithm that goes into generating a self-driving algorithm but will suffice for this paper's purposes.[13]

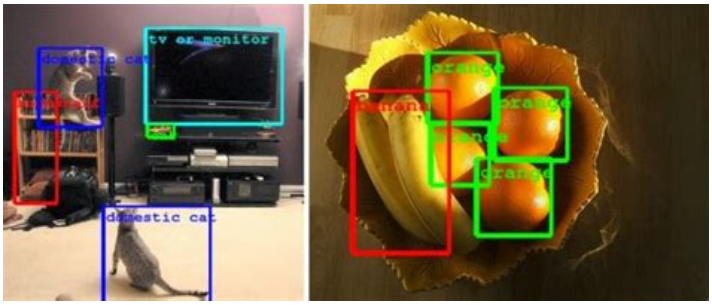


Figure 1: An example of objection detection and identification.[14]

Now that we know how the SDA operates in the real world, let's see how the SDA gets "trained." Start with a batch of ten million sample input images. These images are images that the SDA might encounter in any environment. For example, there might be hundreds of images of being

[12] *Id.* at 11.

[13] Savaram Ravindra, *The Machine Learning Algorithms Used in Self-Driving Cars*, KDNUGGETS (June 2017), https://www.kdnuggets.com/2017/06/machine-learning-algorithms-used-self-driving-cars.html [https://perma.cc/LXS2-AR9M]. The SDA's output generation process can be de-constructed into four-step processes for each environment: object detection, object identification, object localization, and movement prediction. Object detection entails detecting the pattern and borders of various objects within the image. Object identification entails determining the type of object, whether it be a pedestrian, stop sign, or cloud. There are many opinions as to how to best deploy object identification. Object localization entails determining the distance of the object from the car. Movement prediction entails determining the oncoming speed, and if relevant, the expected reaction of the object.

[14] Alex Chitu, *Google's Object Recognition Technology*, GOOGLE OPERATING SYS. (Sept. 11, 2014 2:39 PM), https://googlesystem.blogspot.com/2014/09/googles-object-recognition-technology.html [https://perma.cc/MJV2-JVLT].

stopped at a traffic light. Each image might have a slightly different scenario of the traffic light intersection environment (one with a car with its blinker on, one with a car cutting the driver off, etc.). Each image is tagged with a "prescribed output" telling the algorithm what to do. This prescribed output is set by a human manually. For example, an image with a stop sign in 100 feet may have the prescribed output of "start braking." Similarly, there might be another 1,000 images with a "speed limit 35 mph" sign. The prescribed output would instruct the car to drive at a maximum of 35 mph. Almost all images' solutions entail keeping the car at the center of the road/lane.[15] The individual prescribed outputs, importantly, merge over each other to form a prescribed complete output, e.g. "merge one lane left into the highway at a maximum speed of 65 mph."

The algorithm "learns" by determining differences between images and classifying these differences as factors. For example, if image 1 is completely identical to image 2 except for the color of the light, the algorithm will rationalize which differences cause different prescribed solutions.[16] For example, take two images in the training data set: one with a stop sign and one without a stop sign. The algorithm would link the appearance of the stop sign with the "stop at the stop sign" solution. Therefore, almost all future image inputs with a red stop sign will return a real-time output of "stop at the stop sign."

Putting all this together, the SDA uses statistical pattern recognition to generate real-time outputs for every real-time input. These solutions are overlaid with each other in an end-to-end control system. The real-time prescribed complete output always has a confidence level between 0-1. Say the image is blurry (because of a foggy day) and the algorithm isn't sure; it might decrease the confidence of its prescription from 0.99 to 0.95 as a result. However, its output would still stand. To accurately direct a self-driving car, this process is repeated up to 2300 frame per second.[17]

C. A simplified example to explain interpretability

An SDA is admittedly very complicated, but there are still a few more technical aspects. The next concept to incorporate is that of interpretable vs.

---

[15] *See generally* Manajit Pal, *Deep Learning for Self-Driving Cars*, TOWARDS DATA SCI. (Jan. 6, 2019), https://towardsdatascience.com/deep-learning-for-self-driving-cars-7f198ef4cfa2 [https://perma.cc/RQD9-UUWP]. The picture focuses on keeping the car in the center of the road, and an error occurs when the car veers outside the assigned path.

[16] *See generally* Mohit Tripathi, *Image Processing using CNN: A beginners guide*, ANALYTICS VIDHYA (June 21, 2021), https://www.analyticsvidhya.com/blog/2021/06/image-processing-using-cnn-a-beginners-guide/ [https://perma.cc/9WK5-9UFC].

[17] Emil Talpes et al., *Compute Solution for Tesla's Full Self-Driving Computer*, 40.2 IEEE Micro 25, 25 (2020).

uninterpretable SDAs. Most SDAs are uninterpretable so it is crucial to understand the difference. Such a distinction is important for the legal use case we will explore later: a car accident involving an SDA that detected conflicting factors. For this section, we will use a simplified version of the SDA. The scenario is broken down into its parts below:

Environment: a traffic intersection with four all-way stop signs and four crosswalks. There is an ambulance flashing its lights 500 feet behind the car, and two pedestrians about to enter the crosswalk directly in front of the car driven by the SDA. There is a mannequin on the sidewalk not moving. In the middle of the intersection, there is a hopscotch game made from chalk.

Five factors that affect the real-time output {F1, F2, F3, F4, F5} that dictate the real-time output.

F1 and F2 are not interpretable to a human because they would not affect the car's actions. F1 is a mannequin that the SDA misidentifies as a person and F2 is chalk hopscotch on the road that the SDA misidentified as a crosswalk.

F3, F4, and F5 are factors recognizable to a human. F3 is {a stop sign 20 feet ahead} and F4 is {the two pedestrians about to enter the crosswalk}. F5 is {the emergency lights 500 feet behind}. The algorithm uses image detection to detect F1-F5. F5 is an interpretable factor detected by SDA 1 but not by SDA 2. (Reflective of the greater accuracy that is typically achieved by a black-box neural net algorithm)

Two factors (F6, F7) that do not affect the real-time output. For example, a puffy cloud in the sky or a McDonalds in the far distance.

Two different SDAs are being used:

SDA 1 is an interpretable model. This means that its real-time output can be readily tied back to individual factors. For example, for this scenario, F3 can be attributed to the SDA's real-time output to pull over short of the stop sign. F3 is the overriding factor.

SDA 2 is an uninterpretable model. This means that while it still uses the factor approach, it is not clear how the factors were weighed in determining the real-time output.

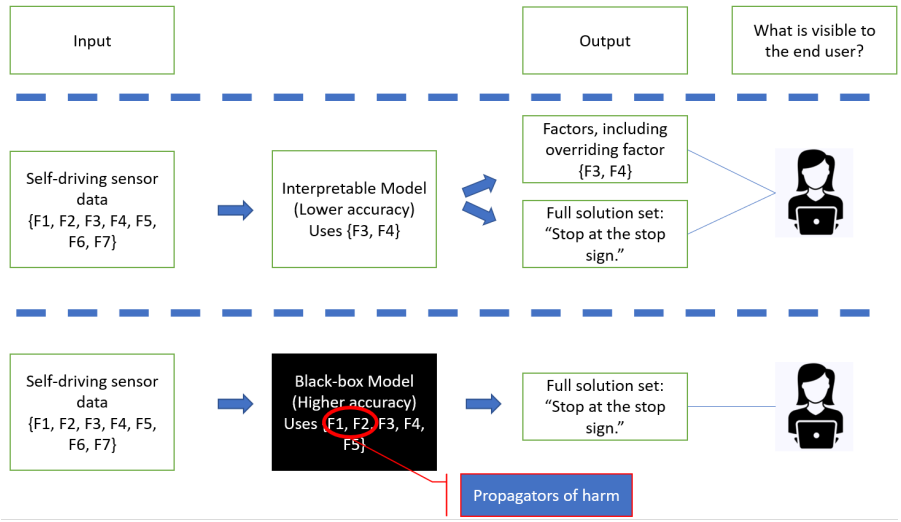


Figure 2: A visualization of the SDA interpreting the real-time input image, breaking it down into relevant factors, generating a real-time output. interpretable and uninterpretable "black-box" AI algorithms.

Note that this difference in interpretability does not always lead to different real-time outputs for SDA 1 and SDA 2; however, it does in this case. The uninterpretable SDA 2 is confounded by the presence of F1 and F2. Both SDAs detect the appearance of factor F3, which cause an initial prescription to stop in 100 feet. However, F1 (the presence of the mannequin) and F2 (the presence of the chalk hopscotch) are not factored into SDA 1. The outcome of this scenario is that the car driven by SDA 1 will stop at the stop sign, but will interfere with the ambulance's path. The car driven by SDA 2 will stop short of the intersection to let the ambulance pass but will not be able to deal with the hopscotch or mannequin.

Factors F1 and F2 confound SDA 2 but will not be readily interpretable to algorithm auditors. That means investigators of the accident will not know their effects on the real-time output decision that was made.[18] What does it mean to be "uninterpretable?" In short, it means that the real-time output is not retroactively attributable to any factors. The uninterpretable SDA instead identifies a mixture of patterns within the image that it deems important

[18] *See generally* Quanshi Zhang et al., Interpretable Convolutional Neural Networks 1 (Feb. 14, 2018) (unpublished) (on file with the University of California, Los Angeles).

to/relevant to the real-time output. The factors are put into a figurative black-box and transforms into a real-time output without any relative weighting or ranking as to which factors are most important.[19] Refer to the image below to view an example of F1/F2 vs F3/F4.

Figure 3: The top CNN filter clearly identifies the cat's face as the rationale for detecting a cat. The bottom CNN filter has a less clear identification strategy, akin to F1 and F2.[20]

D. Using SHAP/LIME to reverse-engineer an uninterpretable SDA to get to the root cause of a car accident

The perils of using an uninterpretable SDA are laid out above. Given that most SDAs are uninterpretable, it is likely that these are the versions that will be involved in car accidents. Let's use the example above: a car using an uninterpretable SDA stopped in the middle of an intersection because it thought that a hopscotch was a crosswalk was rear-ended. There were many factors that may or may not have been detected and factored into the accident: two pedestrians at the crosswalk, a stop sign, an ambulance's emergency lights that were 500 feet back, and crucially the hopscotch set up in the middle of the intersection.

LIME and SHAP are two statistical techniques that partially reverse-engineer the SDA to rank which interpretable factors drove the real-time

[19] *See id.* at 4.

[20] Zhang, *supra* note 19, at 1.

output.[21] By identifying the root factor that caused the SDA to return a bad real-time output, liability for the car accident can be more accurately placed.

Here are the concepts that will be discussed in this section as well as during Section III:

The accident- a car being operated by an SDA in the middle of an intersection and was rear-ended. There were emergency lights 500 meters behind the car. At face value, it seems that it misidentified a mannequin as a pedestrian and hopscotch as a crosswalk. However, that is not confirmed.

SDA- the original SDA that was used and led to the accident.

Relevant factors used by the SDA:

F1, F2- the mannequin and the chalk hopscotch. It is not clear that they were misidentified, nor is it clear that they were factored into the SDA.

F3, F4- the stop sign and the emergency lights 500 feet behind.

F5- the two pedestrians crossing at the intersection.

LIME model- the statistical technique whose analysis yields:

The estimated quantifiable effect of F1-F5 on the SDA

The statistical significance/accuracy of the quantifiable effects of F1-F4

There are two inherent risks in deploying an uninterpretable SDA: F1 and F2 are not just unrecognizable (and thus may propagate bias/harm) but a black-box algorithm does not reveal any F1-F5 that went into the decision-making process.[22] If the passenger in the car driven by the SDA is harmed, recourse and interpretability could be potentially achieved by SHAP/LIME by illuminating gaps in object identification and uncoded biases. These gaps, in turn, could uncover deficiencies when the SDA is supposed to detect overriding factors. For example, if the existence of a stop sign doesn't significantly affect the model's prescribed solution, then there likely wasn't enough training data with stop signs.

Before digging into SHAP/LIME, we will look at crucial terminology that applies to both sets of algorithms. We will differentiate between the original SDA and the results from the SHAP/LIME statistical technique by denoting the output from the latter as the "explanation model." This explanation model, in turn, produces "SHAP/LIME-generated evidence." Much like how the SDA produced real-time outputs from real-time input images, the explanation model estimates which deterministic variables caused the real-time output from the SDA. The term "explanation model" is

[21] *See generally* Marco Tulio Ribeiro et al., *"Why Should I Trust You?": Explaining the Predictions of Any Classifier*, 2 (Aug. 9, 2016) (unpublished) (on file with the University of Washington).

[22] Ribiero, *supra* note 22, at 2.

ascribed by the authors behind SHAP to the SHAP/LIME analyses because the analysis itself is not a perfect mirror of the original SDA.[23] Rather, it draws statistical conclusions from the SDA inputs and outputs, and thereby has statistical uncertainties built into the framework. For example, the explanation model may rank emergency vehicles as bearing a higher weight/effect on the output decision than the presence of an approaching stop sign. However, such a ranking may have a low statistical significance.

Let's start with Local Interpretable Model-agnostic Explanations (LIME). LIME measures the partial effect of individual factors that are detectable by humans by perturbing features of the example.[24] In the above example, they would consist of not just the presence of the hopscotch and emergency lights, but also visible factors which the model did not deem significant- a puffy cloud in the sky, yellow leaves on a tree, or a McDonalds in the far background of the image. The presence of the hopscotch obviously affects the real-time output. LIME perturbs the images to gauge whether or not the existence of the hopscotch (a "superpixel") affects the SDA output.[25] This would change not just the real-time output ("keep driving straight at 35 mph" to "stop for the perceived pedestrian in the hopscotch") but also the confidence level with which the solution is prescribed. This shift in real-time output (attributable to the correlative effects of the confidence level) is the quantifiable effect of the presence of the hopscotch. It is also the slope of the line in Figure 4. This "hopscotch effect" will be referred as the LIME variable effect, or the quantifiable effect that the presence of an object/factor has on the real-time prescribed output.

However, this "hopscotch effect" upon the SDA would not always exert such a deterministic effect. For example, say that the hopscotch is present but stopping would obstruct the ambulance. The hopscotch is no longer as deterministic, and the ambulance is the overriding factor. That is why LIME then assigns a linear regression to the effect of a change in probability on the prescribed solution. Let's break this down. A linear regression (as seen in Figure 4) entails drawing a line-of-best-fit across points on a scatter plot. The x-axis, in this case, is the likelihood that a stop sign is detected in the photo. Remember that the original algorithm assigns a statistical significance to the stop sign's presence; there is a probability between 0 and 1 that the stop sign is found in the input photo based on image recognition confidence. The y-axis is the probability that the answer ["stop at the stop sign"] is returned. As the probability of the SDA detecting a stop sign increases, the probability

[23] *See generally* Scott M. Lundberg & Su-In Lee, *A unified approach to interpreting model predictions*, 31 ADVANCES NEURAL INFO. PROCESSING SYS. 1, 1 (2017).

[24] *Interpretability*, *supra* note 6, at 44.

[25] *Id.*

that it returns the solution ["stop at the stop sign"] increases. The linear regression looks like this:

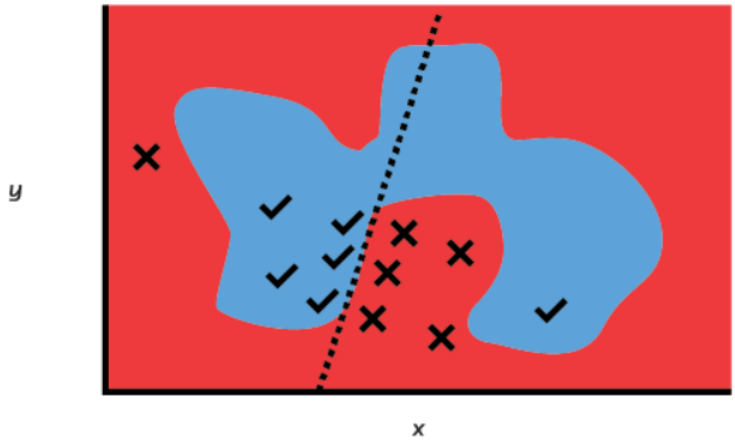


Figure 4: A visualization of the testing process for creating a linear likelihood indicator. The dashed line is the output of the LIME model: its positive slope indicates the direct relationship between this variable and the black-box model's output. [26]

It is important to note the limits of the LIME framework, specifically the L- "local." This perturbation-driven framework only measures shifts within a relatively small range.[27] Thus, this model would not be able to capture the "hopscotch effect" if the hopscotch's probability shifted from 0% to 100%. Practically, that means such an analysis must use an image with largely the same surrounding scenery. The stop sign confidence level must be toggled with the same foreground, middle ground, and background. It must also be toggled with the same image focus and definition. In Figure 4 above, note the relatively small portion of the x-axis within which the regression line sits. LIME also analyzes factors individually, not in comparison to other factors.

In total, the LIME model creates a local (small x-axis range) linear regression. The slope of this linear regression is an estimate of the partial effect of a known factor within the original algorithm. This has been proven to work on image classification.[28]

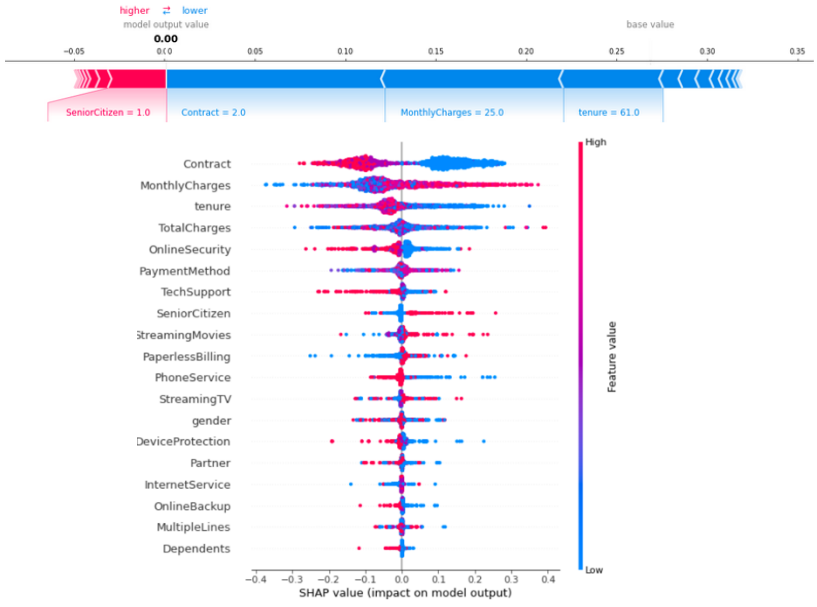
How does these results from the LIME explanation model affect the culpability of an SDA in an accident? It might demonstrate that the SDA

[26] *Interpretability*, *supra* note 6, at 44.
[27] *Id.* at 48.
[28] Lundberg, *supra* note 24, at 25.

didn't properly account for the ambulance, or unduly account for the hopscotch. Say that increasing the likelihood of a hopscotch's presence being detected (from 0% detection to 10% detection) to an input image increases the likelihood the car stopping in the middle of the intersection by 15%. This real-time output shift, in turn, might not affect whether the car stopped *enough*. The jury might deem the presence of the hopscotch as an overriding factor, compelling the car to stop in every instance. In other words, the jury may find that the algorithm should be more sensitive to even the vague possibility of a hopscotch existing.

Now let's look at Shapley Additive Explanations, or SHAP. In an algorithm, some factors have a larger partial effect than others upon the result. For example, a sign denoting a rest area will affect the algorithm less than a stop sign. Shapley values are numbers that estimate the relative magnitude of a variable upon the model output.[29] These values can be used to rank the factors that affected the SDA's real-time output.

Because the results from the SHAP explanation is relative, it can supplement the LIME output (which only looks at variables one-by-one). This is best seen through the visualizations of the *shap* Python library. In the top diagram of Figure 4, each variable is represented by a long bar. The blue segments of the bar indicate that the presence of the variable had a positive effect on the outcome while the pink segments of the bar have a negative effect on the outcome.



[29] *Interpretability*, *supra* note 6, at 4–13.

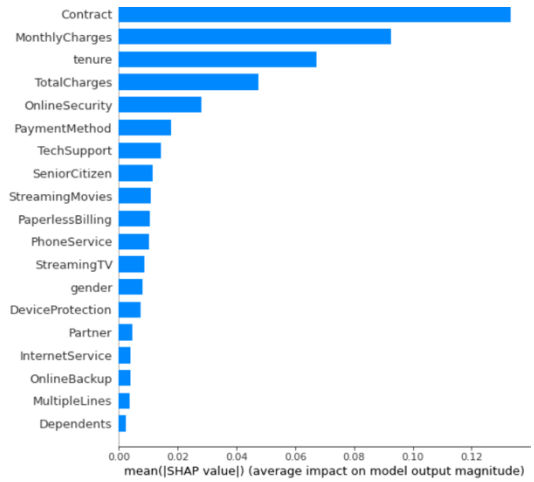


Figure 4: the charts above show the output of the *shap* python library. These in turn demonstrate the magnitude of the relationship between pre-defined, interpretable independent variables and the outcome (dependent) variable of the black-box algorithm.[30]

While the SHAP framework hypothetically can be applied to a self-driving algorithm driven by image recognition, such a model has yet to be proven in concept. The mechanism would use the same numeric values harnessed in the LIME example: the probability that a factor is detected by the image detection mechanism.

E. E. Considerations in algorithm generation

With this in mind, we should temporarily take a step back and ask: why would a company use a black-box SDA when an interpretable option is available? Because the most common type of SDA (neural networks, a type of black-box algorithm) often has a higher accuracy rate.[31] Given the latitude to detect shapes and patterns in the training data and link it to the output actions, the algorithm is able to provide more accurate output in line with the prescribed solution set.

[30] *Interpretability*, *supra* note 6, at 14–33. The Total Charges had a positive effect when the variable was positive and had a mixed effect when the variable was negative. While statistical significance is not calculated since this does not mimic the effects of a t-test/F-test, the variables' respective impact helps to convey the truly deterministic variables in the original self-driving algorithm.

[31] *Id.* at 4–13. It should be noted that the example SHAP framework above was used on tabular data. Therefore, data was much more easily quantified (set number of samples without statistical guesswork whether a variable actually existed in an image) and was complete (each sample had an entry for the variable; there were no samples without, say, a stop sign present)

For SDAs, neural networks' higher accuracy rate could be the difference between being only driving on closed single-lane highways and driving on open streets with pedestrians and construction barriers.

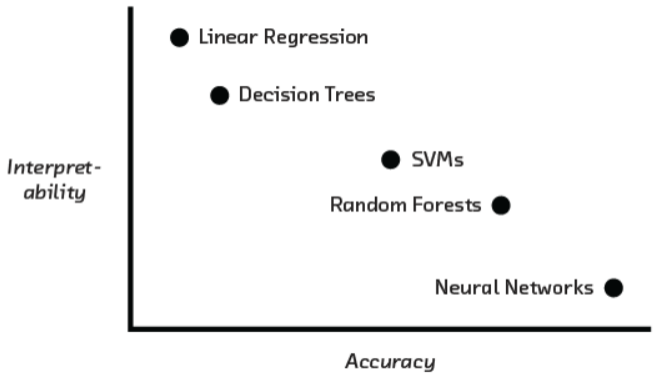


Figure 5: A chart describing the trade-off between accuracy and interpretability in AI.[32]

There are plenty of negative outcomes that can result from the usage of such a black-box algorithm. Say an unknown variable somehow incorporates decision-making for an SDA involved in a trolley problem. The car is driving at an unstoppable speed and must choose lane 1 or lane 2. A crowd of pedestrians is detected in lane 1 (impact would kill the pedestrians) and a barrier is in lane 2 (impact would kill the car passengers). A black-box algorithm hides these determinative factors and thereby lends an air of plausible deniability to the whole algorithm development process. Even worse, the algorithm could be programmed to prioritize the health and safety of the car occupants over the lives of the pedestrians. Such an algorithm, if it could be reverse-engineered, could shed light on the programmers' decision-making in these scenarios.

[32] *Interpretability*, *supra* note 6, at 14–33.

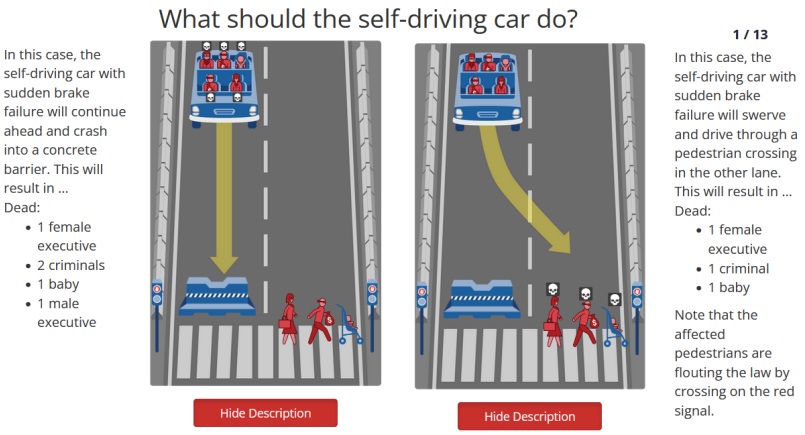


Figure 6: A "moral machine" that replicates the trolley problem but replaces the trolley with a self-driving car.[33] It is hosted by MIT.

This serves as an effective transition into the next portion of the paper, where tort liability may or may not be established using reverse-engineered algorithms. By statistically determining the latent variables that drove a black-box algorithm (self-driving or otherwise) to make a harmful decision, could we find the algorithm developers liable for harm?

III. RELEVANT TORT LAW

There are three general spheres of tort law under which an algorithm programmer might be held liable for its results. These are product liability, service liability, and general negligence. The first of these is the most unlikely sphere in which to successfully prosecute an algorithm development company, because products that are made bespoke for a purpose are not deemed to bear a defective product design.[34] The second sphere, service liability, is comparatively more likely to foster a successful lawsuit using evidence from SHAP/LIME because of courts' greater preclusion for applying strict liability.[35] Negligence, the third sphere, is best poised to hold SDA-caused harms accountable.[36]

To make these concepts easier to grasp, let's re-introduce a scenario that applies across both Sections II and III. Party 1 ("plaintiff") was driving in her SDA-driven car when it suddenly stopped in the middle of an intersection without notice. She was promptly rear-ended by the car behind her. There were emergency lights detected 500 meters behind the car. The

[33] MORAL MACHINE, https://www.moralmachine.net/ [https://perma.cc/A6BX-L5DT].

[34] *See infra* Section III.B.

[35] *See infra* Section III.C.

[36] *See infra* Section III.C.

accident was deemed to be the fault of the SDA-instructed car. The car was manufactured jointly by Parties 2 ("carmaker") and 3 (self-driving algorithm generator). Party 4, the driver who rear-ended the driver and suffered injuries, is deciding what charges to bring and how to prove their case. Party 4 chooses to sue for tort damages and must bring evidence to prove their claim.

A. Product Liability

Product liability is used to address injuries stemming from product defects, or from the misrepresentation of product usage.[37] One benefit to the plaintiff is that product liability can yield strict liability, thereby dodging the intent issue that would confound an intentional tort framework. There are many tort claims (such as malpractice and manufacturing defects) that product liability can generate, but the ones that are relevant to our case are negligence in design and misrepresentation.[38]

To set the context for how self-driving product capabilities are currently organized, it is useful to know the current standards for levels of self-driving used by the SDA industry. It is set by the Society of Automotive Engineers (SAE).[39] Currently, the self-driving program product is governed by six levels of self-driving capability, titled SAE Levels 0 through 5.

Product liability includes several elements that tie to SDAs. The elements to determine liability include whether: (1) the SDA is a "product;" (2) the defendant must be a seller of the SDA in question; (3) the SDA must reach the injured party without substantive alteration; (4) the SDA must be defective; and (5) the defect must be the source of the injury.[40]

The first element is whether the SDA involved in this accident is a product. Using the definition of "product," the fact-finder must determine whether the SDA meets certain sub-elements. Is the SDA mass-marketed and

---

[37] John Villasenor, *Products liability law as a way to address AI harms*, BROOKINGS (Oct. 31, 2019), https://www.brookings.edu/research/products-liability-law-as-a-way-to-address-ai-harms/ [https://perma.cc/EKU2-TUZC].

[38] *Id.*

[39] *SAE Levels of Driving Automation Refined for Clarity and International Audience*, SAE BLOG (May 3, 2021), https://www.sae.org/blog/sae-j3016-update [https://perma.cc/D5NZ-RKL4]. SAE Level 0 entails the human driver being provided with warnings and momentary assistance. SAE Level 1 is when an automated system provides steering or brake/acceleration support (e.g. lane centering), and SAE Level 2 is when an automated system provides both steering and brake/acceleration support (lane centering and adaptive cruise control at the same time). SAE Level 3 has the SDA taking over the driving role under limited conditions, but the driver must take back control when the feature requests. SAE Level 4 will not require the driver to take over driving at any point but will not take control unless certain conditions are met. SAE Level 5 can perform all driving tasks equivalent to a human in any condition.

[40] George Cole, *Tort Liability for Artificial Intelligence and Expert Systems*, 10 UIC J. MARSHALL J. PRIV. & TECH. L. 127, 159 (1990).

sold at-large, is the SDA an artifact, and has a large customer base?[41] The fact-finder will likely determine that SDAs are mass-marketed and sold at-large. Over thirty self-driving software start-ups and corporate ventures have $5-10 billion between 2016-2020.[42] This competition indicates that while the Level 5 self-driving capability is not commoditized yet, certain self-driving algorithms have been. No cars have SAE Level 4-5 capabilities and only one brand has Level 3 capability, so most lawsuits will emerge from SDAs that fall under SAE Level 1-2, which are widespread.[43] The next sub-element as to determining whether an SDA is a product is: is this SDA an "artifact?"[44] The definition of artifact is whether the product at fault is part of a larger supply chain with an untraceable customer base.[45] In this case, the SDA developer is a software company (a professional service) without a mass production of other SDAs. Moreover, SDA-producing companies have an easily traceable customer base and supply chain. SDAs are tailor-made for a car-maker client based on the car's braking and acceleration capabilities.[46] This swings the balance against an SDA being deemed an artifact and thus a product that falls under product liability. However, an SDA has a foreseeably large customer base; this swings in the other direction towards being considered a product. Hence, this element will likely be debated and decided in court based on the individual facts of the case.

The next element is whether the defendant sold the SDA. Given the original scenario consisting of two parties (algorithm developer and carmaker), this requirement will be met as the AI developer either sold or licensed the software package to the carmaker in a discrete transaction. The carmaker then sold the car possessing the SDA-driven capability. This sale directly caused the accident, and in turn injured P4. Therefore, this element swings firmly in favor of the SDA being considered a product.

The third element is whether the AI reached the injured party without substantive alteration. This will likely be met via the original P2 carmaker – P3 SDA generator relationship.[47] More generally, if the SDA developer

---

[41] *Id*. at 154.

[42] Arthur Zuckerman, *45 Self-Driving Cars Statistics: 2020/21 Market Data, Industry Growth, & Trends,* COMPARE CAMP (May 15, 2020), https://comparecamp.com/self-driving-cars-statistics/ [https://perma.cc/6HFG-VK9U].

[43] James Gilboy, *Honda's Now Selling the World's First Production Car with Level 3 Self-Driving Tech*, THE DRIVE (Mar. 5, 2021), https://www.thedrive.com/news/39609/hondas-now-selling-the-worlds-first-production-car-with-level-3-self-driving-tech [https://perma.cc/KH9B-NPZE].

[44] Cole, *supra* note 41, at 163.

[45] La Rossa v. Scientific Design Co., 402 F.2d 937, 941–42 (3rd Cir. 1968).

[46] *Software for Self-Driving Cars*, NVIDIA, https://www.nvidia.com/en-us/self-driving-cars/drive-platform/software/ [https://perma.cc/F5UC-X558].

[47] William Dixon, *What is adversarial artificial intelligence and why does it matter?*, WORLD ECON. F. (Nov. 21, 2018), https://www.weforum.org/agenda/2018/11/what-is-adversarial-artificial-

explicitly states the range of robustness (say, Level 3 autonomy on clear highways) and the user doesn't follow this range, then the complaint's viability is in doubt: the user has a duty to constrain his inputs to the domain in which the AI is designed to operate.[48] However, if a user tries to apply the above Level 3 autonomy for, say, offroad use, then that would not qualify. This will likely entail substantive alteration of intended use. Similarly, if the algorithm is being subjected to a modified speed limit sign or adversarial AI that purposefully confounds the AI, that scenario does not qualify for product liability. (Refer to Figure 7 for an example) Rather, the harmed party would instead have to pursue claims against the party who purposefully confounded the AI. Thus, the factfinders would determine this element.

Figure 7: A form of adversarial AI. By purposefully distorting the "3" in the speed limit, this sign is now read by self-driving algorithms as "Speed Limit 85." This misinterpretation could lead to a crash, but is not the fault of the SDA developer.[49]

The fourth element is whether the AI is defective. This is, in turn, dependent on where in the economic world the algorithm is manufactured, sold, and used. Was the SDA deployed in a defective manner, or did sensor or input detector malfunction? The algorithm developer generally makes the same algorithm for each customer; remember the end-to-end control system? For such a defect to occur, the SDA-generating Party 2 would have to mess up the code in a small percentage of customers. This erroneous bug would,

intelligence-is-and-why-does-it-matter/ [https://perma.cc/YEH8-S9H5]. Adversarial AI is the intentional confounding of an algorithm, taking advantage of its internal programming mechanism to yield bad outcomes. Thus, an altered input created by adversarial AI would complicate the viability. Say an assassin hacked the SDA and created the hopscotch manually!

[48] Cole, *supra* note 41, at 165.

[49] Alex Woodie, *Hacking AI: Exposing Vulnerabilities in Machine Learning*, DATANAMI (July 28, 2020), https://www.datanami.com/2020/07/28/hacking-ai-exposing-vulnerabilities-in-machine-learning/ [https://perma.cc/U8DT-SNYL].

in turn, lead to harm. This is unlikely and is complemented by another sub-element that will likely doom this cause of action: if the product's design features are the reason for the purchase, the injured party cannot raise a product liability cause of action.[50] In most cases, the self-driving software and capability is a significant reason for the purchase of a car. The injured Party 4 can rebut this by arguing that driving is inherently dangerous and cannot be safe. However, Party 2 and 3 will contest this by arguing that the crash rate of autonomous vehicles is small, particularly those which cause severe crashes, despite the sample size being comparatively small.[51] Such an analysis of SDAs lies is direct contrast to the conclusion of *Blevins v. Cushman Motors*, which implicated car parts as products: "[motor vehicles] are not incapable of being made safe for their intended and ordinary use."[52] Because self-driving cars are already safer than normal cars, courts will likely not seek an impetus to make them even safer.[53] Therefore, this element will not be met except in exceedingly rare circumstances.

The fifth element is whether the algorithm is the source of injury. This is the portion that will be heavily contested in court by the P2 SDA developer and the P3 carmaker, and therefore doesn't need to be discussed at length. One critical consideration of all these elements is that courts pursue equitable rulings when weighing these elements.[54] Therefore, the customer should be framed as powerless and lacking agency.

B. Finding Strict Liability within Product Liability

Strict liability is the level of strictness that a court will hold product companies if the factfinders determine that a company is guilty of product liability.[55] Without an intent requirement, the company will be found guilty of the charges as long as four key principles compel such a decision.

There are four principles that a court of equity will balance when evaluating whether to apply strict liability in a negligence case: stream of commerce, control of risks, risk cost-spreading, and deep pockets. We will discuss the SDA-driven accident as it relates to each of these principles.

---

[50] Cole, *supra* note 41, at 168.

[51] Myra Blanco et. al., *Automated Vehicle Crash Rate Comparison Using Naturalistic Data*, VIRGINIA TECH TRANSP. INST. (Jan. 8, 2016), https://featured.vtti.vt.edu/?p=422 [https://perma.cc/7ZHC-W897].

[52] Blevins v. Cushman Motors, 551 S.W.2d 602, 608 (Mo. 1977).

[53] *Tesla Safety Report*, TESLA, https://www.tesla.com/VehicleSafetyReport [https://perma.cc/2S9P-4459].

[54] Cole, *supra* note 41, at 174.

[55] Epstein Becker Green, *The Product Liability Dilemma: Product vs. Service,* LEXOLOGY (Dec. 18, 2012), https://www.lexology.com/library/detail.aspx?g=22f43645-a095-429c-8a41-d0dbf12146d4 [perma.cc/67R2-SHFP].

The first principle through which to further analyze the claim against the SDA is: whether the SDA's public benefit is outweighed by its unforeseen hazards. This cost-benefit analysis does not absolve the company from recompensing those injured by the SDA, even if the SDA generator provides implicit assurance the product is safe.[56] The benefits of self-driving cars are numerous: there are 1,300,000 car accident deaths worldwide and 98% are due to human error.[57] According to Tesla's Q4 2021 Safety Report, Tesla's operating under the Level 3 Autopilot technology recorded an accident only every 4.31 million miles, making it almost 10x safer than unaided human driving.[58] The costs are slightly more complex accidents, such as the one that we are analyzing in this paper. The benefits do significantly outweigh the costs, indicating that the courts will lean against apply strict liability due to the stream of commerce principle.

The next principle to analyze whether the self-driving algorithm generator (P2) or carmaker (P3) is in the best position to control the risk, or whether the responsibility should be attributed to the government. [59] Obviously, SDA generators have the power to withhold irresponsible software. They are also well-placed to advise the government on setting standards for deployment. By eschewing an adversarial approach and instead promoting public policy in a manner that makes the benefits of self-driving more transparent, SDAs and carmakers could shift the risk to a regulatory arm of the government. This in turn, would be based on clearly delimiting the capabilities of self-driving cars, and goes beyond the scope of this paper. The implication for finding an SDA generator strictly liable will depend on the factfinder's vision for how responsibility should be distributed.

Another influential principle for determining strict liability is that of risk cost-spreading. While we know technology applications will always fail at some point, the question becomes: how to balance risk between the buyer and seller of the self-driving car? The lack of a mass market for self-driving algorithms means that the software is bespoke, likely shielding the developers from harm.[60] Insurance companies and government regulators are best positioned to wage this debate of how to efficiently distribute risk. The result of this debate will determine whether the P2 SDA generator will be

---

[56] *Id.*

[57] *USDOT Releases New Data Showing That Road Fatalities Spiked in First Half of 2021*, US DEP'T OF TRANSP. (Oct. 28, 2021), https://www.nhtsa.gov/press-releases/usdot-releases-new-data-showing-road-fatalities-spiked-first-half-2021 [https://perma.cc/BF5E-W9EJ].

[58] *Tesla Vehicle Safety Report*, TESLA (Dec. 2021), https://www.tesla.com/VehicleSafetyReport [https://perma.cc/Q4YB-QX38].

[59] *See* Cole, *supra* note 41, at 177.

[60] *Id.* at 164.

approached by a government insurance regulator or private insurance company when a claim is brought.

Lastly, we evaluate whether this is a deep-pocketed industry or a budding industry. Since the world of self-driving cars is a budding industry, courts will likely eschew the strict liability imposition to promote innovation. Thus, these principles overall will likely balance against imposing strict liability upon the SDA generating P2, making the product liability case even more difficult. While these principles might sway the factfinders into assigning strict liability in a small subset of cases, service liability provides a broader case.

C. Service Liability

The growth of "Software as a Service" (SaaS) gives rise to the view that the output of a self-driving algorithm (SDA) is a service as opposed to a product. This is particularly convincing given the UCC definition of a good: "[tangible] things which are movable at the time of identification to the contract . . . or other identified things attached to realty."[61] Few would classify an algorithm as a "good/product" under this definition.

The case of *Barbee v. Rogers* illustrates how the court resolves conflicts as to whether an offering should fall under product or service liability. The plaintiff in the case alleged improper fitting and instructions for contact lenses.[62] The court first denied product liability because of the professional nature of the relationship between the patient and doctor. The court then denied assigning strict liability to the lens manufacturer (associated with product liability) because the contact lenses were not a finished product in regular channels of trade.[63]

D. Finding Strict Liability within Service Liability

As established above, SDA generation maps closely to the service portion of service liability. Service liability can also lead to strict liability via the case *Newmark v. Gimbel's Inc*.[64] Strict liability was instituted in this case despite there not being a professional standard of care. That is because there was an implied warranty provided by the service provider, in this case a beauty parlor operator. Strict liability was deemed necessary only in cases where a warranty was provided[65]; professional services (such as dentistry,

---

[61] UNIFORM COMMERCIAL CODE § 2-105 (2021).

[62] Barbree v. Rogers, 425 S.W.2d 342, 342–43 (Tex. 1968).

[63] *Id.* at 344.

[64] Newmark v. Gimbel's, 258 A.2d 697, 703 (N.J. 1969).

[65] *See id.* at 703.

etc.) avoid this warranty obligation by nature of their client-specific work.[66] Certain professions change their client-specific nature (and thus warranty obligation) over time, including software developers.[67] Software engineers work is growing more commoditized as the education pipeline and licensing barriers for software engineering is democratized. With the rapid onset of programming bootcamps and online tutorials, coding is becoming a rapidly standardized market.

Within the realm of service liability, strict liability can be assigned to professions that provide a warranty. If the factfinder deems the client-specific warranty obligation provided by software engineers as closer to hairdressers than dentists, the degree of care required of and proffered by the SDA developer is not the only factor in determining strict liability. Rather it is whether they belong to a "recognized socially warranted profession."[68] (If they do not belong to such a profession, then strict liability is imposed) Currently, the software engineering industry is currently safely embedded in professional services (a la dentistry and doctors) but is slowly shifting and may incur a higher standard of care.

Given this trend of SDA developers gravitating towards being found strictly liable for demonstrating an improperly low standard of care, we will analyze this trend using four policy factors prescribed by Cole. They are: "(1) the service is marketed to a large number of individuals; (2) the service is identical across distinguishable classes or individuals, rather than requiring and reflecting specific circumstances to be performed for each purchaser; (3) the service is of such a definable and delimitable nature that, given the circumstances of the purchaser, human experts reasonably would not produce different services; and (4) the service is a voluntary interaction whose principle motivation is economic."[69] An SDA that meets these four criteria might give rise to imposing strict liability against the SDA developer on the grounds of service liability.

Let's start with the first policy factor: whether the SDA service is marketed to many individuals. Self-driving cars are marketed across nearly the entire population of the United States, considering that 91.1% of households own a car and all of them would benefit from self-driving capabilities.[70] This increases the need for strict liability; such a diverse mass

---

[66] *Id.* at 702.

[67] *See* Cole, *supra* note 41, at 208.

[68] *See* Cole, *supra* note 41, at 191.

[69] *Id.* at 192.

[70] Mathilde Carlier, *Number of households with one or more vehicles in U.S. 2017*, STATISTA (Aug. 4, 2021), https://www.statista.com/statistics/184082/vehicles-per-household-in-the-usa-in-2001/ [https://perma.cc/5GMA-FPZ6].

market means that some consumers will be vulnerable to puffery made by SDA programmers and carmakers. Since the service expectations of a self-driving algorithm are the same across the country, this element points towards instituting strict liability.

The next factor is whether the SDA-provided service is certain across all populations. In other words, will all end users apply the SDA in the same manner? That is certainly the case here; consumers purchase cars with one form of self-driving capabilities. No self-driving capability customization exists on a consumer-by-consumer basis.[71] On the other hand, doctors, lawyers, and other members of professional services provide inherently unsure outcomes.[72] The work of SDA is one-size-fits-all, as it is marketed. Thus, the SDAs are sufficiently uniform a certain outcome (self-driving) is essentially guaranteed and standardized. Therefore, this factor works towards the imposition of strict liability.

The third factor is most relevant to self-driving cars: whether SDA service is clearly defined such that a competitor would not produce an SDA with different capabilities and range. In other words, does the SDA claim to be "complete" (i.e. able to handle a clearly defined set of scenarios)? The completeness of the SDA in question hinges on the definitions set by the SAE (Levels 0-5, defined above). This depends on how the P2 SDA generator represents the SDA to the P3 car-maker as well as the public. Critically, the SDA is mathematically incapable of perfection because of the sheer number of *impossible* scenarios to decide when piloting a car. Advertising as Level 5, therefore, is terribly risky. On the other hand, if a limited representation is made and "[a] human actor could not possibly further circumscribe the potential for injury by any better or more complete representation," then strict liability is not feasible.[73] Thus, self-driving cars will likely under-sell their capabilities to avoid lawsuits. If under-selling (i.e. claiming SAE Level 2 while actually capable of Level 3) doesn't occur, then there are more grounds for a successful claim by the harmed party P4.

The last factor is essentially dependent on whether self-driving cars constitute a want or a need. If it is a want, then strict liability is in order. A strict liability for a need is bad because it disincentivizes practitioners who are required to make society run. For example, imposing strict liability on workers in the medical profession would lead to lower employment in a field that qualifies as a societal need. Self-driving cars are decidedly a luxury that no one will feel obligated to get in the near future, and thus is a want.

---

[71] *Innovating the Future of Driving. Again*., CADILLAC, https://www.cadillac.com/world-of-cadillac/innovation/super-cruise [https://perma.cc/P6UP-JYWM].

[72] Cole, *supra* note 41, at 193.

[73] *Id.* at 196.

There are a few additional policy considerations that swing the needle against imposing strict liability for SDA generators. The first of these is the overall purpose of strict liability- to prevent or correct defects, whether in products or in services.[74] In this case, since an SDA is not perfectible (remember the confidence level?), it would be somewhat unfair to hold it to a strict liability standard. Unless the software bug is tied directly to the harmful outcome, it cannot be pinned back as a manufacturing defect. Similarly, if the algorithm is deemed to be in "beta-testing" (tech-speak for "in development"), it cannot yet be held strictly liable for harm because the general purpose hasn't yet been achieved.

A synthesis of these factors indicates that a court, when deciding whether to assign strict liability under service liability upon a harmful SDA generator P2, is unlikely to do so. However, it may still sustain a general service liability claim. While these circumstances are slightly broader than those that would sustain a successful product liability suit, it is still narrower than general negligence.

E. General Negligence

General negligence is the most promising form of pursuing liability because it does not bear an intentionality standard, and falls nicely within service liability.[75] Three elements of general negligence need to be evaluated to determine the viability of suing the SDA developer P3. They are: whether a duty exists for the programmer, whether the damages in question is limited to the risky behavior of the SDA developer, and whether the SDA actually falls within the scope of liability.

The first of these is whether the duty exists for the programmer. As defined in the Third Restatement, an actor ordinarily has a duty to exercise reasonable care when the actor's conduct creates a risk of physical harm.[76] However, this is countered by "exceptional cases" where courts deem no duty because of policy.[77] Courts are often hesitant to stifle innovation, so the imposition of a duty upon a company to claim responsibility for the future uses of its algorithm falls apart on policy grounds.[78] An SDA would likely qualify as an exceptional case because of its revolutionary innovative potential. Thus, this element factors against a successful suit against an SDA generator P3.

---

[74] *Id.* at 206.

[75] Cole, *supra* note 41, at 214.

[76] RESTATEMENT (THIRD) of TORTS § 7(a) (AM. L. INST. 2010).

[77] RESTATEMENT (THIRD) of TORTS § 7(b) (AM. L. INST. 2010).

[78] Anna Laakmaan, *When Should Physicians Be Liable for Innovation?*, 36 CARDOZO L. REV. 101 (2015).

The next element to explore is the harm-based limitations of the tort. Specifically, according to the Third Restatement, "An actor's liability is limited to those harms that result from the risks that made the actor's conduct tortious."[79] Let's go back to the SDA-driven car accident. Assuming that no driver error or other factors are to blame, can the algorithm be implicated? Yes—through the differential tort liability framework. There are three reasons that the plaintiff could claim a tort cause of action: (1) the essence of the injury lies in tort, (2) there is a need to avoid a statute of limitations, or (3) there is a concern that the court will uphold a "lack of privity" defense.[80] The first reason applies to the culprit of the accident- the SDA.[81] In the example in section II.B, the SDA caused the car to brake in the middle of the intersection without a clear reason.

Let's say that the courts don't seek to pin full blame on the SDA developers; what are the joint-and-several liability breakdown of the algorithm? Within the complex ecosystem of a self-driving car (or any other algorithm-governed process), the decision-making may not be entirely driven by the algorithm. There are constraints and governance systems elsewhere, such as emergency brakes and cameras implemented by the carmaker P2. If the judge holds both companies (the carmaker P2 and the SDA programmer P3) jointly liable for injury caused by the algorithm, liability depends whether the court follows a comparative negligence system or traditional contributory negligence system.[82] The resulting assignment of blame will likely be driven by the adversary system of law, and there is little point in us speculating.

The last element that affects the viability of a general negligence case regarding a harmful algorithm is scope of liability. The Third Restatement defines the scope of liability for reckless (not negligent) tortfeasors as: "an actor who intentionally or recklessly causes harm is subject to liability for a broader range of harms than the harms for which that actor would be liable if only acting negligently."[83] This implies that negligent tortfeasors are liable solely for the range of harms that they caused directly. In general, the important factors in determining the scope of liability are the moral culpability of the actor, the seriousness of harm intended and threatened by

[79] RESTATEMENT (THIRD) of TORTS § 29 (AM. L. INST. 2010).

[80] Cole, *supra* note 51, at 156.

[81] *See supra* Section II.B.

[82] Cole, *supra* note 51, at 156.

[83] RESTATEMENT (THIRD) of TORTS: PHYS & EMOT. HARM § 33(b) (AM. L. INST. 2010).

those acts, and the degree to which the actor's conduct deviated from appropriate care.[84]

In determining this element, the court will debate whether an SDA is an actor. Most fields of law indicate that this is not the case: an actor must be human, rather than an artificial intelligence bit of code.[85] Therefore, this sub-element will likely swing against the plaintiff P4 in bringing a successful suit.

There are two additional sub-elements of Element 3 that would make the case against a self-driving car's decision-making algorithm difficult to sustain. The first of these is the need to find an expert witness that can construct a self-driving algorithm without the flaws of the defendant.[86] This is particularly difficult within the self-driving data science space, as immense amounts of training and testing have been invested in the space. This includes applying training data from CAPTCHA images that require human users to identify driving-centric objects (motorcycles, traffic lights, etc.) to prove that they're not a robot.[87] Replicating such a data set and having an algorithm software developer create an algorithm that doesn't have the harmful error in such a fledgling industry is unlikely.

The next sub-element of scope of liability for the plaintiff P4 to answer then becomes: "If a program was tested for two years without finding a condition which later damages a plaintiff, and if the plaintiff can prove that a certain flaw could have been detected by testing the program for an additional twenty-four hours, would this constitute a proof of breach of duty of care?"[88] The modern-day unit testing and QA techniques are excellent, but self-driving errors are innately unpredictable yet inevitable. Thus, Cole's prescription for constant algorithmic testing and verification is wise. The court's judgement will determine whether the testing conducted was adequate. Critically, if the court determines the self-driving task itself to be impracticable, the claim of negligence will be unlikely to prevail. Until courts define the duty taken on by algorithm developers (while they are not

---

[84] *Id.* Note that the Restatement adds: "an actor who intentionally or recklessly causes harm is not subject to liability for harm the risk of which was not increased by the actor's intentional or reckless conduct."

[85] Sarah S. Landau, *Copyright Office Review Board Reaffirms that Human Authorship is a Prerequisite for Copyright Protection*, COWAN LIEBOWITZ & LATMAN (Mar. 22, 2022), https://www.cll.com/CopyrightDevelopmentsBlog/copyright-office-review-board-reaffirms-that-human-authorship-is [https://perma.cc/M5Q2-Y2Z4].

[86] Cole, *supra* note 41, at 214.

[87] Milla Henson, *How Security Captchas Crowdsourced Self-Driving Car Technology*, MOTOR BISCUIT (Apr. 16, 2020), https://www.motorbiscuit.com/how-security-captchas-crowdsourced-self-driving-car-technology/ [https://perma.cc/K4ZK-4UTH].

[88] Cole, *supra* note 41, at 214 (citing Gemignani, *Product Liability and Software*, 8 RUTGERS COMPUT. & TECH. L.J. 173, 214 (1981)).

insurers, their products do possess a warranty), this question will remain one for each factfinder.

This creates a somewhat bleak picture for those pursuing general negligence claims against SDA developers, despite being rid of the intentionality element/strict liability. However, SHAP/LIME can point out gaps in testing that lead to harm being inflicted.[89] Given the open-source nature of the SHAP/LIME techniques, harmed parties could generate algorithmic audits, replicating the scenarios that gave rise to the harm and determine exactly where the algorithm's judgement lapsed.[90] Moreover, given that the plaintiff parties don't have to prove intentionality or the professionalism standard associated with data science, there are fewer blockers associated with this path.[91] That being said, filing both a negligence and a strict liability under service liability claim will likely yield the optimal mix of supported claims.

There are some positive society-wide benefits which this sphere of negligence will generate. Innovation around self-driving safeguards (such as such as blind spot monitoring, emergency braking, and lane centering features) will continue to grow. It will also ensure both carmakers and SDA generators post adequate warnings of the risk of misuse. Similarly, it will prevent fraudulent representations of SDA (including over-stating the SDA's capabilities).

Evaluating these factors holistically, it is unlikely that any lawsuit following a general negligence liability framework will result in a successful claim. Between the policy implications and negative effect on innovation, the fiddly nature of assigning joint-and-several liability, and the lack of duty that was assigned to the algorithm generating party in the first place, such a lawsuit may be dismissed. Laws should aim to keep pace with technology, and the current body of negligence/product liability law has fallen behind.

IV. RELEVANT RULES OF EVIDENCE

However, before dismissal, the court will have to weigh the evidence provided by the SHAP/LIME algorithms, which may sway the balance. Let's review the pertinent rules of evidence that will determine whether SHAP/LIME evidence is even admissible.

Grimm, Grossman, and Cormack's recently published work on the applicability of evidence law to AI is the framework upon which I conduct

---

[89] *See supra* Section II.D.

[90] Muhammad Fawi, *Explain Python Machine Learning Models with SHAP Library*, MINIMATECH (Sept. 11, 2021), https://minimatech.org/explain-python-machine-learning-models-with-shap-library/ [https://perma.cc/ND6N-QP3J]

[91] *See supra* Section III.A–D.

my analysis.[92] It is made a tiny bit more complicated by the fact that SHAP/LIME is an AI-driven analysis of an AI self-driving system (rather than the AI mechanism itself), but the results are still just as applicable. To keep things straight, please continue to refer to the scenario in II.D and legend in II.A.

A. Relevance

The first consideration is that of relevance. Under Federal Rules of Evidence 401-403, SHAP/LIME-generated evidence stands a good chance of being admitted. Rule 401 states: "Evidence is relevant if: (a) it has any tendency to make a fact more or less probable than it would be without the evidence; and (b) the fact is of consequence in determining the action."[93] Since SHAP/LIME increases the probative base upon which the self-driving algorithm was made, it complies with Rule 401.

Rule 402 states: "Relevant evidence is admissible unless any of the following provides otherwise: the United States Constitution; a federal statute; these rules [of evidence]; or other rules prescribed by the Supreme Court. Irrelevant evidence is not admissible."[94] Crucially, "while the first part of Rule 402 is flexible, the second part is immutable: [i]rrelevant evidence is never admissible."[95] SHAP/LIME withstand this test as well; as above, the causative factors which will be extracted by the algorithm are critical to the decision and are therefore not irrelevant.

Rule 403 has three important dimensions across which SHAP/LIME should be analyzed. The rule itself states: "The court may exclude relevant evidence if its probative value is substantially outweighed by a danger of one or more of the following: unfair prejudice, confusing the issues, misleading the jury, undue delay, wasting time or needlessly presenting cumulative evidence."[96]

The first dimension establishes a "balancing test" for determining whether relevant evidence may be considered by the judge or jury.[97] In the case of SHAP/LIME-generated evidence, this balance favors admissibility despite potentially adverse consequences of its introduction, namely its difficult explanation. The pros are established in section I; the cons that judges must consider include unfairness, awareness that confusion may

[92] Grimm, *supra* note 9, at 1.
[93] FED. R. EVID. 401.
[94] FED. R. EVID. 402.
[95] Grimm, *supra* note 9, at 87.
[96] FED. R. EVID. 403.
[97] Grimm, *supra* note 9, at 87.

result, and judgment as to whether the jury may be misled.[98] SHAP/LIME will likely result in confusion and a potential misled jury, especially if presented by a party with limited technical literacy. Specifically, confusion will result because of the technical intricacy of the model. More importantly, the jury may be misled if the SHAP/LIME evidence is presented as infallible. Rather, it should be presented as an interpretable, less accurate re-creation of the self-driving car algorithm with statistical margin for error.[99] However, given the potential for illustrating the critical factors in the algorithm's decision-making just before the harm took place, the evidence will likely pass this test.

The next dimension is whether the trial judge will permit the jury to hear the SHAP/LIME-generated evidence. This is a separate decision from the balancing test above and does not technically require compliance with the Federal Rules of Evidence; it depends on whether the plaintiff and defendant provide enough information to the judge for her to make this admittance decision. This includes whether the attorneys raise well in advance that they intend to bring SHAP/LIME-generated evidence.[100] Such best practices will help increase the likelihood of admittance.

The last dimension, and most substantive, includes disclosure of all underlying code and relevant training data used by the plaintiff to generate the SHAP/LIME output. This will provide the judge with adequate information to decide whether the evidence was generated in an unbiased manner. The judge will seek insight into the SHAP/LIME's output's *validity* (i.e., the degree of accuracy with which the AI tool measures what it purports to measure), and the *reliability* (i.e., the consistency with which the AI algorithm correctly measures what it purports to measure), of the evidence being generated.[101] More importantly, given the adversarial nature of the law, it will give the opposing party a chance to view how it was derived.[102] The opposing party will likely criticize the algorithm for statistical uncertainty, the inherently imperfect re-creation of the self-driving algorithm, and the choice of training data upon which the SHAP/LIME algorithm was built to generate the overriding factors.

Between all these relevance rules, a judge is likely to admit SHAP/LIME-generated evidence assuming it was generated in an ethical manner using training data as closely representative of the original scenario and self-driving algorithm as possible.

---

[98] *Id.* at 62.

[99] *Interpretability*, *supra* note 6, at 4–13.

[100] Grimm, *supra* note 9, at 89.

[101] *Id.*

[102] *Id.*

B. Authentication of SHAP/LIME-generated evidence

The next consideration is the evidence's authentication, which means that the evidence is what it claims to be. To be clear, we are seeking confirmation that the SHAP/LIME-generated evidence does in fact mirror the self-driving algorithm as closely as possible. The evidence of this is Federal Rule of Evidence 901(a), which states: "To satisfy the requirement of authenticating . . . an item of evidence, the proponent must produce evidence sufficient to support a finding that the item is what the proponent claims it is."[103] Rule 901(b) then lists ten non-exclusive ways in which a party can accomplish this. The examples that most readily lend themselves to authenticating SHAP/LIME-generated evidence are: Rule 901(b)(1), which discusses testimony of a witness with knowledge that an item is what it is claimed to be, and Rule 901(b)(9), which refers to evidence describing a process or system and showing that it produces an accurate result.[104] Let's analyze both of these mechanisms.

Bringing in any single expert to explain the self-driving algorithm is nigh on impossible. All types of algorithms, ranging from the simple to the complex, invariably lean on the talent of data scientists, software engineers, project managers, and more.[105] In order to minimize the total number of witnesses that might pull from all of these teams, parties can lean on expert witnesses, as provided by Rule 702:

A witness who is qualified as an expert by knowledge, skill, experience training or education may testify in the form of an opinion or otherwise if: (a) the expert's scientific, technical, or other specialized knowledge will help the trier of fact to understand the evidence or to determine a fact in issue; (b) the testimony is based on sufficient facts or data; (c) the testimony is the product of reliable principles and methods; and (d) the expert has reliably applied the principles and methods to the facts of the case.[106]

Expert witnesses, however, must stick to their personal experience of the facts, per Rule 703:

An expert may base an opinion on facts or data in the case that the expert has been made aware of or personally observed. If experts in the particular field would reasonably rely on those kinds of facts or data in forming an opinion on the subject, they need not be admissible for the opinion to be admitted.[107]

---

[103] FED. R. EVID. 901(a).

[104] FED. R. EVID. 901(b).

[105] Grimm, *supra* note 9, at 86.

[106] FED. R. EVID. 702.

[107] FED. R. EVID. 703.

The party which generated the SHAP/LIME evidence will also likely be questioned. In testing the responses of the self-driving car, they will likely have to simulate the SDA in a similar environment as when the accident occurred. They will have to simulate the emergency lights, hopscotch, and weather conditions as close to the original scenario as possible. Ideally, the tested scenarios will take place at the same intersection to remove any location-driven decision-making variability.

These rules ensure that all expert witnesses must "either have personal knowledge of the authenticating facts or qualify as an expert that is permitted to incorporate into their testimony information from sources beyond their own personal knowledge, provided it is sufficiently reliable."[108] In the case of SHAP/LIME-generated evidence, this will require facts specific to the case by the plaintiff. In-depth questions pertaining to the interpretability and prioritization of the SDA will be required. For example, data scientists may be asked whether the SDA contains any overriding factors that change the decision-making process. These answers will help to confirm or deny the estimated SDA responses' accuracy, generated by the SHAP/LIME algorithm.

Now let's look at Rule 901(b)(9). It permits authentication demonstrating "[e]vidence describing a process or system and showing that it produces an accurate result."[109] The critical factor is whether SHAP/LIME-generation is sufficient for a "reasonable factfinder," which is equivalent to a 51% threshold.[110] It will be up to the judge determine whether SHAP/LIME-generated evidence meets this threshold.[111] There is no trade secrets or proprietary information impeding the open-source Python library for either mechanism.[112,113]

In the scenario presented with Parties 1-4, there are some negative effects of the SHAP/LIME-generated evidence being used or interpreted

---

[108] Grimm, *supra* note 9, at 93.

[109] FED. R. EVID. 901.

[110] *See, e.g.*, Lorraine v. Markel Am. Ins. Co., 241 F.R.D. 534, 542–43 (D. Md. 2007); United States v. Safavian, 435 F.Supp.2d. 36, 38 (D.D.C. 2006); United States v. Holmquist, 36 F.3d 154, 168 (1st Cir. 1994) ("[T]he standard for authentication, and hence for admissibility, is one of reasonable likelihood.").

[111] FED. R. EVID. 901(b) Subdivision (3) Example (3).

[112] Muhammad Fawi, *Explain Python Machine Learning Models with SHAP Library*, MINIMATECH (Sept. 11, 2021), https://minimatech.org/explain-python-machine-learning-models-with-shap-library/ [https://perma.cc/QD9D-ME6D]. Indeed, the only trade secret likely to impede the investigation is that surrounding the SDA, but that's a topic for another paper.

[113] Lundberg, *supra* note 24, at 1. *See also Interpretability, supra* note 6, at chs. 2–5. For further reference, I would recommend looking at the original paper on SHAP as well as explainers. The key factor will be the confidence level of the SHAP/LIME coefficient estimates, and their respective statistical significance.

improperly. These negative effects need to be balanced by the judge. The testing process for SHAP/LIME is scenario-specific to the algorithm that is being recreated. Thus, the statistical uncertainty will be dependent on the number of images, number of conflicting factors, and complexity of underlying algorithm in the testing set. For example, in trying to test whether an SDA prioritizes a hopscotch or emergency lights, the SDA may be fed images with one of the two factors, both, and neither. Images may be sourced from the location of crash or elsewhere, tweaking other variables to determine their local importance on the SDA's output. However, it will be impossible to come close to the original SDA's training data set- some are estimated to have over 100 billion annotated images.[114] An improperly-drawn conclusion from the evidence may generate a negative judgment for the incorrect party. Thankfully, the SHAP/LIME-generated evidence is necessarily specific to this case; therefore, it can't be used as negative precedent for other cases. Because of this, a trial judge, under Rule 403, would authenticate this evidence by a mere preponderance, despite the danger of unfair prejudice to the adverse party, or the possibility of a either a confused or misled jury on the basis of authenticity.

C. Applying Daubert factors to SHAP/LIME to confirm authentication

The Daubert factors stand alongside Rule 702 and are particularly relevant to the admission of technology-generated evidence. They instruct judges, when admitting evidence, to evaluate: "(1) whether the expert's technique or theory can be or has been tested . . . ; (2) whether the technique or theory has been subject to peer review and publication; (3) the known or potential rate of error of the technique or theory when applied; (4) the existence and maintenance of standards and controls; and (5) whether the technique or theory has been generally accepted in the scientific [or technical] community."[115]

Meant to complement the Rule 702 and reinforce the rigor of Rule 403, the Daubert factors both hurt and help the admissibility of SHAP/LIME-generated evidence. The first two factors, replicability and precedence, support its usage. The techniques of SHAP/LIME are replicable because the package is publicly available, and both the SHAP/LIME-generated evidence's training data and results can be readily viewed and re-tested. The SHAP/LIME technique has been peer reviewed and published, per the resources listed in section III-B. Importantly, it's been used to analyze and re-create image classification convolutional neural networks that were

[114] *Tesla Safety Report*, *supra* note 59.

[115] Advisory Committee Note, FED. R. EVID. 702 (2000).

previously black-boxes.[116] That being said, it has not been used specifically for self-driving algorithms to my knowledge. Therefore, it might give a judge pause to have this use be the first-of-a-kind application.

There are three additional factors (accuracy, standardization, acceptance by the scientific community) weighed into the Daubert balancing test. They are inconclusive, and depend on the future trajectory of SHAP/LIME usage. As for accuracy, the potential rate of error is dependent on the input data for the SHAP/LIME-generated evidence.[117] If more data with diverse factors F7-F10, such as multiple configurations of signs, traffic lights, etc., are used to generate local variable coefficients, these coefficients will have a lower variance.[118] On the other hand, a small training data set can lead to estimates being provided for a small set of factors and background variables.[119]

Currently, there are few standards for establishing model transparency aside from the GitHub description, commit/in-line code comments, and some peer-reviewed frameworks.[120] The National Institute for Standards and Technology (NIST) is currently working on an AI Risk Management Framework that will hopefully assign more rigor to this space.[121] Lastly, SHAP/LIME is steadily gaining acceptance but has yet to go beyond the data science community.[122] While it is not a guarantee that it will gain further acceptance, the number of citations to the original SHAP paper by Lundberg and Lee has been cited 6,280 times.[123]

These factors generally point towards this evidence being accepted. However, the scenario involving Parties 1-4 is not going to be readily solved with the sole introduction of SHAP/LIME-generated evidence.[124] Rather, this type of evidence will move the needle towards more accurate outcomes and counter the string of patent-protected profiteering on AI trade secrets. Most importantly, tort law is clearly not fit for the purpose of rendering

---

116 *Interpretability*, *supra* note 6, at 14–33.

117 *See supra* Section III.C.

118 *See* Fawi, *supra* note 113.

119 *Id.*

120 Brian Hannaway, *The Importance of Comments for Maintainable Code*, DZONE (Sept. 5, 2019), https://dzone.com/articles/the-importance-of-comments-for-maintainable-code [https://perma.cc/P6ZW-2BF3].

121 *AI Risk Management Framework*, NAT'L INST. OF STANDARDS & TECH., https://www.nist.gov/itl/ai-risk-management-framework [https://perma.cc/9T7M-TFB5].

122 Dario Radečić, *LIME vs. SHAP: Which is Better for Explaining Machine Learning Models*, TOWARDS DATA SCI. (Dec. 14, 2020) https://towardsdatascience.com/lime-vs-shap-which-is-better-for-explaining-machine-learning-models-d68d8290bb16 [https://perma.cc/HPM4-AHPD].

123 Lundberg, *supra* note 24, at 4.

124 *See supra* Section II.C.

justice, but evidence law is closer to facilitating this type of claim. Could this set the stage for a separate private right of action for those harmed by AI?[125]

V. CONCLUSION

We are transitioning from discussions on computer ethics (human's ethical use of computers) to machine ethics (ensuring the behavior of machines towards human users- and perhaps other machines as well) is ethically acceptable.[126] There are many challenges that this transition poses- namely, how to implement ethical principles in machines, and which of these ethical principles to choose.[127] To answer this question, we lean on the differentiation between an implicit ethical agent versus an explicit ethical agent.[128]

According to James Moor, "A full ethical agent can make explicit ethical judgments and generally is competent to reasonably justify them. An average adult human is a full ethical agent. We typically regard humans as having consciousness, intentionality, and free will. Can a machine be a full ethical agent?"[129] Neural nets (with prescribed correct actions and the ability to learn/draw inferences regarding other scenarios) begin to blur the lines of "full ethical agent." Since technical decisions are consciously implemented by programmers (a full ethical agent), any resulting regulatory or evidentiary approach should approach the algorithm as an implicitly full ethical agent. As an implicitly full ethical agent, the SDA-driven car poses significant challenges to the regulatory system because such an algorithm entails "public risk" but is otherwise difficult to control without infringing upon innovation.[130]

However, given the lack of recourse that is currently being provided to victims of technology-driven recklessness,[131] the rules of evidence should permit SHAP/LIME to add some bite to the current scope of tort law. The

---

[125] Sebastian K. Skelton, *EU Act 'Must Empower Those Affected by AI Systems to Take Action'*, COMPUT. WEEKLY (Mar. 31, 2022 2:58 PM), https://www.computerweekly.com/news/252515378/EU-Act-must-empower-those-affected-by-AI-systems-to-take-action [https://perma.cc/HS64-NR7F].

[126] J.H. Moor, *The Nature, Importance, and Difficulty of Machine Ethics*, IEEE INTELLIGENT SYS. (2006) http://www.psy.vanderbilt.edu/courses/hon182/The_Nature_Importance_and_Difficulty_of_Machine_Ethics.pdf [https://perma.cc/YJ3A-XFGU].

[127] *Id.*

[128] *Id.*

[129] *Id.*

[130] Matthew U. Scherer, *Regulating Artificial Intelligence Systems: Risks, Challenges, Competencies, And Strategies*, 29 HARV. J.L. & TECH. 353, 373 (2016).

[131] W. Nicholson Price, *Can You Sue An Algorithm For Malpractice?*, FORBES (Feb. 11, 2019), https://www.forbes.com/sites/insights-intelai/2019/02/11/can-you-sue-an-algorithm-for-malpractice/?sh=2a8cc2957013 [https://perma.cc/6HCH-D4RW].

notion of data science is facially neutral, but the various biases that enter the process (historic, representation, measurement, aggregation, evaluation, and deployment) reinforce existing patterns of harm.[132] A prominent example is COMP-STAT, a "predictive policing" staffing algorithm which further entrenched over-policing of under-resourced communities of color.[133] The harms of such an algorithm are best described by Corlann Gee Bush as possessing a "valence," essentially a predisposition for perpetuating harm through secondary systems.[134]

An algorithm-driven self-driving car is slightly different; it does not necessarily bear a "valence" towards harm. To be clear, it *may* still bear such a valence. The algorithm may, for example, be unable to detect pedestrians of color at the same rate as white pedestrians, thereby endangering pedestrians of color at an undue rate. Such a scenario draws upon Joy Buolamwini's study of image detection.[135] However, this valence is not necessary to prove that an algorithm inflicted harm upon a party. This harm, when elucidated by SHAP/LIME, can help facilitate justice for parties wronged by carelessly assembled algorithms.

---

[132] Harini Suresh & John Guttag, *A Framework for Understanding Unintended Consequences of Machine Learning*, (Jan. 28, 2019) (unpublished) (on file with Mass. Inst. of Tech.).

[133] Cedric L. Alexander, *Community Policing as a Counter to Bias in Policing: A Personal Perspective*, 126 YALE L.J. F. 381 (2017).

[134] Corlann G. Bush, *Women and the Assessment of Technology*, *in* TECHNOLOGY AND THE FUTURE (Albert H. Teich ed. 1993).

[135] Buolamwini, *supra* note 8.