Vicarious Liability for AI

Mihailis Diamantis, Professor of Law

September 2021

Cambridge Handbook of AI and Law (Kristin Johnson & Carla Reyes eds., forthcoming 2022)
Iowa Legal Studies Research Paper No. 2021-27
When an algorithm harms someone—say by discriminating against her, exposing her personal data, or buying her stock using inside information—who should pay? If that harm is criminal, who deserves punishment? In ordinary cases, when A harms B, the first step in the liability analysis turns on what sort of thing A is. If A is a natural phenomenon, like a typhoon or mudslide, B pays, and no one is punished. If A is a person, then A might be liable for damages and sanction. The trouble with algorithms is that neither paradigm fits. Algorithms are trainable artifacts with “off” switches, not natural phenomena. They are not people either, as a matter of law or metaphysics.

An appealing way out of this dilemma would start by complicating the standard A-harms-B scenario. It would recognize that a third party, C, is usually lurking nearby when an algorithm causes harm, and that third party is a person (legal or natural). By holding third parties vicariously accountable for what their algorithms do, the law could promote efficient incentives for people who develop or deploy algorithms and secure just outcomes for victims.

The challenge is to find a model of vicarious liability that is up to the task. This chapter provides a set of criteria that any model of vicarious liability for algorithmic harms should satisfy. The criteria cover a range of desiderata: from ensuring good outcomes, to maximizing realistic prospects for implementation, to advancing programming values such as explainability. Though relatively few in number, the criteria are demanding. Most available models of vicarious liability fail them. Nonetheless, the chapter ends on an optimistic note. The shortcomings of the models considered below hold important lessons for uncovering a more promising alternative.

I. The Necessity of Vicarious Liability .......................................................... 2
II. Criteria for a Successful Model of Vicarious Liability ................................... 3
   A. Identify Whom to Hold Accountable ...................................................... 4
   B. Avoid the Possibility of Gaming ............................................................. 5
   C. Set Efficient Incentives .......................................................................... 6
   D. Secure Fair Outcomes ........................................................................... 8
   E. Keep Barriers to Implementation Low .................................................... 8
   F. Advance Programming Values ............................................................... 9
III. The Criteria Have Teeth ........................................................................... 12
   A. Evaluating a Solicitation-Inspired Approach ....................................... 12

* Professor of Law, The University of Iowa, College of Law and Department of Philosophy.
The Second Principle of Robotics: “Humans, not robots, are responsible agents.”

I. The Necessity of Vicarious Liability

If robots and algorithms are not responsible agents, who should pay when one of them hurts somebody? The Second Principle of Robotics offers an answer: some human. However, there is no obvious human to hold accountable for many algorithmic injuries today. To illustrate the difficulty, pick your favorite modern program—an operating system, a video game, a self-driving protocol, whatever. Very likely, no human owns, or even could afford, the algorithm you chose. There is also a good chance that no human wrote, or even could write, all of it. Most significant algorithms today have many millions of lines of code that would take several lifetimes for any human to punch in. Algorithms themselves do a lot of the coding, and vast teams of humans distributed across isolated modules do the rest.

One common (but ultimately unworkable) proposal acknowledges the force of the Second Principle so far as moral responsibility is concerned but argues that legal responsibility is a more flexible concept. Maybe the law could simply deign that algorithms are responsible, i.e. liable, for the injuries they inflict. There is historical precedent for this maneuver. Consider corporations. The scope of their legal responsibility was relatively limited early in corporate history, in part because of the widespread perception that they are not morally responsible entities. As corporations became increasingly important technologies with a far-reaching power to hurt people, something needed to

---

2 I use “robot” and “algorithm” interchangeably.
change. Lawmakers resolved that they law should *pretend* that corporations are people. As a result, corporations could be sued and ordered to pay when they broke the law. Some modern commentators propose a similar legal development for algorithms. It is an obvious nonstarter. Unlike corporations, algorithms own nothing. Consequently, they cannot pay for harms they cause, no matter how loudly a court’s gavel falls.

If algorithms cannot pay for themselves, any payment for algorithmic harms must come from some third party. Transferring legal responsibility from the direct source of a harm to another entity is called “vicarious liability.” Vicarious liability occurs throughout the law. Co-conspirators are liable for each other’s offenses. Parents pay when their children hurt others. Car owners fork out for accidents caused by uninsured drivers in their cars. There are many models of vicarious liability, each with different rules on when and to whom liability transfers. Co-conspirators are vicariously liable only for each other’s foreseeable offenses. Parents often only when they fail to supervise their children. Car owners only when they authorize uninsured drivers. Each model is tailored to its context.

This chapter offers a roadmap for understanding how vicarious liability for algorithmic harms should work. It offers several criteria that any model should satisfy (Part II). The criteria address the types of liability determinations that the model should generate, pragmatic issues of political economy, and programming-specific values. Collectively, the criteria set a demanding bar that most available models of vicarious liability cannot clear (Part III). The objective of the chapter is not to induce despair. Rather, it clarifies the legal challenge posed by algorithmic harms and provides some guideposts for creative solution building. The chapter concludes by drawing lessons from the shortcomings of the failed models to start paving a path forward (Part IV).

II. *Criteria for a Successful Model of Vicarious Liability*

Though I suggested above that vicarious liability is a necessary paradigm for addressing algorithmic harms, much more is needed to translate vicarious liability into legal doctrine. Most basically, when a victim comes to court with a

---


*See generally Joanna J. Bryson, Mihailis E. Diamantis & Thomas D. Grant, *Of, For, and By the People: The Legal Lacuna of Synthetic Persons*, 25 ARTIF. INTELLIGENCE & L. 273, 275 (2017).*

*Pinkerton v. United States, 328 U.S. 640, 647 (1946).*

*See, e.g.,* CONN. GEN. STAT. § 52–572; KAN. STAT. ANN. § 38–120; OR. REV. STAT. § 30.765(1); 4 MODERN TORT LAW: LIABILITY AND LITIGATION § 28:40 (2d ed. 2020).*

*16 AM. JUR. PROOF OF FACTS 3d 271 (Originally published in 1992).*
harm traceable to an algorithm, the doctrine say whether the victim is out of luck or whether some third party will pay. Beyond the basics, there are some optional features that the doctrine would ideally have. For example, a fully general doctrine would cut across types of liability, both civil and criminal, but the fundamental differences between liability regimes may make that difficult to achieve. There are also several mandatory features that that any promising doctrine of vicarious liability for algorithmic harms must satisfy. I present them below, so that they may serve both as ex ante guideposts for developing doctrine and as ex post evaluative metrics.

A. Identify Whom to Hold Accountable

It is not enough simply to say that some third party will be liable for an algorithmic harm; the doctrine must identify who that third party is. This criterion may sound obvious but making good on it is no simple task. As the opening paragraph to this chapter suggested, for most impactful algorithms, there will be no obvious human third-party to hold accountable. Modern algorithms can be enormously complex and expensive. Many are beyond the capability of any individual human being to own, code, or manage. The law would do better to set its sights instead on another type of legally responsible person that is associated with most important algorithms: the business corporation. Unlike individual humans, individual corporations do design, own, or manage today’s most important algorithms. Additionally, corporations are more likely than individual humans to have the necessary assets to pay for algorithmic harms.

Identifying “corporations” as the vicariously liable party is just the start of a solution. While there are far fewer corporations than natural people in the world, the United States still has millions of them. Plaintiffs and prosecutors cannot sue them all. A doctrine of vicarious liability must tell complainants which to name as defendant.

The economics of algorithms complicate the matter. It may seem easy enough to say that corporations should pay for the injuries of “their” algorithms. Where just one corporation bears any significant relationship to an algorithm, that straightforward approach may be adequate. However, as the economic arrangements around algorithms become increasingly complex, the one-corporation-per-algorithm scenario is quickly becoming a quaint anachronism. Corporations can and do bear any number of different relationships to algorithms that the simple grammatical possessive cannot disambiguate. While there is usually at least one corporation behind most important algorithms, there
are often many.\textsuperscript{10} One corporation may have designed a module for an algorithm that a second assembled. A third corporation may have tested the algorithm. A fourth may have marketed it to a fifth that owned and licensed it to a sixth that operated it on hardware owned by a seventh. Any doctrine for holding corporations vicariously liable for algorithmic harms must say in any circumstance which of these is on the hook. Having a predictable answer is important not only for letting victims know where to look for satisfaction, but also for providing businesses with ex ante incentives to take appropriate care.

The task is made no simpler by the fact that harmful defects in algorithms can arise at any link in the chain of development and deployment, or even from interactions between links.\textsuperscript{11} For example, there may be a problem with the algorithm itself (e.g., it does not direct self-driving cars to apply enough brake pressure in certain situations), the hardware running the algorithm (e.g., the brakes lack sufficient stopping power), or some interaction between the two (e.g. the algorithm and the brakes may both be adequate but not properly calibrated to each other). Simply proclaiming that one link in the chain (e.g. the developer of the algorithm, or its owner, or the hardware manufacturer) will always be liable will more often than not impose liability on the wrong party from the perspectives of Fairness and Efficiency (see below for these criteria).

\textbf{B. Avoid Opportunities for Gaming}

As a corollary to the first criterion, vicarious liability should not be manipulable. Businesses are masters at managing liability. Indeed, risk management is one of corporations’ defining purposes.\textsuperscript{12} Corporate leaders have a duty to manage business-level liability risk as one tool for maximizing profit.\textsuperscript{13} Failing to do so could subject them to personal suit by shareholders.

If there is a loophole in a regime of vicarious liability, we should expect that corporations (or their savvy attorneys) will find it. Through licensing agreements and creative business arrangements, motivated corporations can formally substitute an underfunded entity at any single point in the chain of production for any algorithm. For example, if the rule of vicarious liability is simply that owners of harmful algorithms will be held liable, sophisticated corporations will transfer ownership to underfunded shells, subsidiaries, or partners. Or if the rule is that operators of algorithms are responsible for

\textsuperscript{10} See Marta Infantino & Weiwei Wang, \textit{Algorithmic Torts: A Prospective Comparative Overview}, 29 TRANSNAT’L L. & CONTEMP. PROBS. 309, 353 (2019).

\textsuperscript{11} Id. at 353-54.

\textsuperscript{12} Douglas C. Michael, \textit{To Know A Veil}, 26 J. CORP. L. 41, 55 (2000) ("[O]ne of the corporation’s central purposes is the limitation of liability.").

\textsuperscript{13} In re Caremark Int’l Inc. Derivative Litig., 698 A.2d 959, 970 (Del. Ch. 1996).
injuries, we should expect to see a series of carefully worded corporate licensing agreements such that customers “license” and become “operators” of the algorithms they use on websites, phones, cars, etc. A mechanism for identifying liable corporations that is susceptible to such manipulation risks allowing corporations to undermine whatever purposes it set out to achieve.

C. Set Efficient Incentives

Beyond saying which corporations are potentially liable for algorithmic harms, a model of vicarious liability should transmit liability neither too conservatively nor too generously. One way to gauge whether a model of liability does too much or too little is to evaluate the incentives it sets. The threat of sanction—whether civil or criminal—affects how corporations run their businesses. Inducing them act in socially efficient ways is a balancing act between encouraging them to take too many and too few risks.

As a lower threshold, vicarious liability must be sufficient to induce corporations to develop and utilize algorithms in ways that are less likely to cause harm. “[T]he safest way to secure care is to throw the risk upon the person who decides what precautions shall be taken.”14 While nothing can guarantee that a sophisticated algorithm will never hurt anyone, software engineers and operators can take steps to reduce the probability that any algorithm will misbehave.15 These steps include: diversifying the body of engineers writing algorithms,16 more careful initial programming,17 more mindful selection of training data sets,18 more extensive pre-roll-out testing,19 regular post-roll-out quality audits,20 routine

14 OLIVER WENDELL HOLMES, JR., THE COMMON LAW 117 (1881).
run-time compliance layers,\textsuperscript{a} effective monitoring,\textsuperscript{b} and continuous software updates to address problems as they arise.\textsuperscript{a} Programmers also have tools they can use to prove (to themselves or to others) that an algorithm has been applying its rules consistently.\textsuperscript{a} Each of these precautions entail costs that, all things considered, profit-focused corporations would rather avoid. Through the threat of sanction, the law can make precaution cheaper than the risk of violation.

Where corporate incentives to minimize harm are concerned, more is not always better. If a model of vicarious liability exposes corporations to too much legal risk, it might over-deter them from investing in technology in the first place. This would lead to a significant social loss. Some algorithms take lives, but they have the capacity to save many more.\textsuperscript{a} Some may discriminate in lending or hiring, but they have the potential to make decisions more objective.\textsuperscript{b} Some manipulate markets, but they can also make markets more efficient.\textsuperscript{b} Pursuing prevention too vigilantly will impede corporate investment in these and similar social benefits.\textsuperscript{b} It could also disadvantage U.S. firms in relation to foreign competitors.\textsuperscript{c} Although algorithmic development should not continue

\textsuperscript{a} See generally NAT'L HIGHWAY TRAFFIC SAFETY ADMIN., FEDERAL AUTOMATED VEHICLES POLICY: ACCELERATING THE NEXT REVOLUTION IN ROADWAY SAFETY 16 (2016), https://www.hsdl.org/?view&did=795644.
\textsuperscript{b} See generally IDA, ETHICAL CHOICES IN AUTOMONOUS SYSTEMS, 77 ROBOTICS & AUTOMONOUS SYS. 1, 1-2 (2016).
\textsuperscript{c} See generally NAT'L HIGHWAY TRAFFIC SAFETY ADMIN., FEDERAL AUTOMATED VEHICLES POLICY: ACCELERATING THE NEXT REVOLUTION IN ROADWAY SAFETY 16 (2016), https://www.hsdl.org/?view&did=795644.
\textsuperscript{d} Felippe Menguzzi & Michael Luck, \textit{Norm-Based Behavior Modification in BDI Agents}, PROCEEDINGS OF THE 8TH INTERNATIONAL JOINT CONFERENCE ON AUTONOMOUS AGENTS AND MULTI-AGENT SYSTEMS 177, 177 (2017); Louise Dennis et al., \textit{Formal Verification of Ethical Choices in Autonomous Systems}, 77 ROBOTICS & AUTOMONOUS SYS. 1, 1-2 (2016).
\textsuperscript{e} Thomas C. King et al., \textit{Artificial Intelligence Crime: An Interdisciplinary Analysis of Foreseeable Threats and Solutions}, SCI. & ENGINEERING ETHICS (2019).
\textsuperscript{f} See generally IDA, ETHICAL CHOICES IN AUTOMONOUS SYSTEMS, 77 ROBOTICS & AUTOMONOUS SYS. 1, 1-2 (2016).
\textsuperscript{i} Stephanie Bornstein, \textit{Antidiscriminatory Algorithms}, 70 ALA. L. REV. 519, 533-37 (2018).
\textsuperscript{k} Rebecca Croootof, \textit{The Internet of Torts: Expanding Civil Liability Standards To Address Corporate Remote Interference}, 69 DUKE L.J. 583, 663 (2019).
without due regard for the injuries it will cause, nor should it be unduly hampered. The law needs to strike a balance.

D. Secure Fair Outcomes

Along similar lines, vicarious corporate liability must go far enough to be fair to victims, without going so far that it becomes unfair to defendants. “Fairness” and “justice” are fraught concepts. I will attempt no definition here. My point is just that any model of vicarious corporate liability for algorithms should be self-consciously responsive to some plausible conception of fairness. Efficiency is not the sole measure of appropriate liability outcomes.

Like setting efficient incentives, achieving fairness is a balancing act. Any model that purports to be fair must have some nuance. Always leaving victims to bear the costs of algorithmic harms will clearly not do. Dissatisfaction with a status quo that undermines victims’ interests is part of what motivates the search for a suitable model of vicarious liability in the first place. The move toward automation does not alter the fact that discrimination, price fixing, and reckless driving leave victims in their wake. These victims, or the state on their behalf, should have as clear a path to justice as their counterparts a decade ago.

It would be equally unacceptable from a fairness perspective to force defendants, even for-profit corporations, to pay for every algorithmic harm. It is too easy to discount costs to faceless business entities. However, those costs impose far-reaching effects on innocent individuals who have an indisputable claim to a fair outcome. Most immediately, the shareholders and employees who stand just behind the corporation bear the brunt of any corporate sanction. Just one step further, there are the many other stakeholders: creditors, consumers, community members, etc. The law owes a duty of fairness to all these individuals. It can best fulfill that duty by being fair toward the corporation with whose interests their own interests are bound.

E. Keep Barriers to Implementation Low

Few legal changes are frictionless. While barriers to implementing a proposal may have little bearing on its absolute merit, they are a crucial

---


consideration for anyone seeking real change. Sorting out responsibility for algorithmic harms is not just a theoretical quandary for moral philosophers and jurisprudences. There is a growing legal gap that leaves victims without recourse and developers without adequate disciplinary incentive.

Lobbying by adversely affected parties is one potent source of trouble for new legislation. Opponents who are motivated, organized, and well-funded—like technology firms—pose the most significant challenge. Any proposal that treads too far on their interests will face vigorous political challenge. A legislative compromise that is less likely to provoke technology firms may be preferable to more ambitious reform that would die on the debate floor. Politically savvy corporations will even prefer compromise to the status quo. If change is in the air, balanced reform could act as a hedge against less favorable legislation in the future.

Legislative reform that has a familiar structure may also face fewer barriers.1 Broad support builds more reliably for incremental changes and those that draw on pre-existing legal frameworks.2 Such changes can feel more like natural and sensible extensions of current law, rather than disruptive reallocations of social benefits and burdens. Familiarity also makes reform easier to understand and more predictable, thereby disarming the political bugaboo of unanticipated consequences.

F. Advance Programming Values

Ensuring accountability is far from the only challenge that the rapid integration of algorithms into economic and social life poses. Philosophers, political scientists, and sociologists have been sounding alarm bells over the ways that algorithms can and do infringe human dignity, undermine democracy, and perpetuate socio-economic disparities. Technoethicists propose several programming values—like respecting human autonomy, ensuring human oversight, avoiding deception, and preserving user privacy—to guide programmers in developing socially responsible algorithms, especially when using artificial intelligence.3 Falling short of these values would not always violate the law, nor is it clear that the law should always mandate them. However, it

---

would be a missed opportunity not to use vicarious liability to encourage technology companies to reckon more seriously with programming values.

Many programming values are rather imprecise. In what follows, I focus on one that is relatively well-defined and has received a lot of recent attention: transparency. Ethicists persuasively argue that algorithmic decisions impacting human interests should have a transparent justifying logic. Transparency is central to our dignity interests vis-à-vis algorithms because we deserve to know not just what decisions are affecting us but also why they were made. Knowing why puts us in a position evaluate the decision for ourselves and to contest the result or the process that led to it. We become active agents in our own world, rather than passive recipients of our digital fates.

Algorithms that are not transparent in this way are called “black box algorithms.” Black box algorithms are trained (rather than written) using such techniques as feeding massive data sets into deep neural networks. The resulting algorithms effectively write themselves and are inscrutable. The training process generates artificially intelligent systems that rely on millions of interconnected variables. Their internal logic is far too complex for any human intelligence to deconstruct or comprehend. Black box algorithms make decisions about human interests, sometimes very important decisions like whether someone gets out of jail or can afford a home, and no one can say how.

Machine learning algorithms can be or become transparent in two ways. The developers can design them from the beginning to be interpretable by using techniques that avoid deep neural nets. An interpretable algorithm is one that can give a clear explanation of its own decision-making process. Examples of such algorithms are decision trees or linear regression models, whose decision points and weighted coefficients are traceable from start to end. Alternatively, an inscrutable black box algorithm developed using a deep neural net can later become more transparent by applying a separate explainable AI to it. Explainable AI attempts to reconstruct black box algorithms’ decision-making process in human-comprehensible terms. Explainable AI works by feeding inputs in slight variations into black box AI to test how the outputs change. After repeating this many times, explainable AI builds an interpretable model of how the black box algorithm works.

Building interpretable algorithms from the get-go is preferable, when possible, to using merely explainable AI. The problem with explainable AI is

---

38 Algorithms could also become black boxes if developers use a security setting to block access to the code. I set such cases aside for present purposes.
that, since it is just a model, it is not necessarily an accurate nor faithful reproduction of how the black box algorithm works. Explainable AI may even develop multiple, equally accurate but importantly different, models of the same black box algorithm. One model may reflect concerning decision points that occur in the black box algorithm (e.g. making race-based inferences), another model may recreate those decisions in more neutral terms (e.g. making income-based inferences). There are clearly important normative differences between the two ways of making a decision but no way to say which is the “right” explanation.

Why would corporations ever develop black box algorithms? There are three main reasons. One is that deep learning algorithms are reputed to be more accurate than an interpretable system. While this is true in some domains—depending on the type of task and the available data—programmers and business leaders tend to falsely assume that deep learning is necessary in circumstances where interpretable AI could perform equally well. A second reason is the “bigger is better” culture of the AI community. Deep learning algorithms can be big, developed with ever more layers and trained with ever larger data sets. They are so complex that they are literally incomprehensible. New programmers are seduced—both at school when learning how to code and when making strategic decisions in the workplace—by the allure of deep neural nets. Interpretable machine learning techniques often go unconsidered. The last reason corporations tend to default to deep neural nets is money. The obscure logic of black box AI makes it easier to guard from competitors. Black box AI also tends to be easier to develop. Interpretable AI requires domain expertise and specialized talent, both of which (in part due to the cultural effects just discussed) are in short supply.

Returning now to vicarious liability for algorithmic harms, a strong candidate model will induce developers to reconsider the possibility of using interpretable AI. While some scholars have proposed an outright ban on using

---

black box AI in certain circumstances,\textsuperscript{44} that interventionist approach could generate as many problems as it solves. As discussed above, there is a trade-off—both financial and ethical—in choosing which sort of AI to use. A subtler legal regime would give developers some skin in the game without mandating a single approach. Such a regime could acknowledge that specific circumstances can tip the balance of strategic and ethical trade-offs one way or the other and allow businesses to adapt as needed.

* * *

In sum, there are at least six criteria that any plausible mechanism for holding third parties vicariously liable for algorithmic harms should satisfy:

Criterion 1. Identify which third party or parties will be liable
Criterion 2. Be robust enough to avoid gamesmanship
Criterion 3. Give efficient incentives to all parties involved
Criterion 4. Produce fair outcomes
Criterion 5. Have low barriers to implementation
Criterion 6. Promote programming values, like interpretability

There are surely other criteria that a system of vicarious liability would ideally satisfy. However, as discussed next, these six already filter out many potential candidates.

III. The Criteria Have Teeth

Though each of the criteria listed in the previous section may seem like they set a relatively low bar, they are collectively quite demanding. Most available models of vicarious liability cannot satisfy them. This section considers three possible vicarious liability frameworks: a very restrictive solicitation model, a very permissive “no-fault” model, and a more moderate negligence-based model. None can satisfy all the criteria.

A. Evaluating the Solicitation-Inspired Approach

Solicitation is a criminal law doctrine that applies when a defendant induces someone else to commit a crime.\textsuperscript{45} For example, a defendant may have hired a hitman to assassinate a rival. Solicitation liability applies even if the person whom the defendant induces is entirely innocent because, e.g., she is a

\textsuperscript{44} Rudin, supra note 40.
\textsuperscript{45} Model Penal Code § 5.02(1).
minor, she is mentally incompetent, or she did not know she was harming anyone (the defendant may have tricked her). The general idea behind solicitation law is that a culpable actor should not be able to escape liability simply by having someone else carry out mischief for him.

I am not aware of anyone who advances solicitation as a model of vicarious liability for algorithmic harms. It is not without its theoretical appeal. Since algorithms are not moral agents, concepts like fault, intent, negligence, and the like do not apply to them. Solicitation liability can apply even when the direct source of harm is faultless. The legal focus is on the defendant who induced the harm. The central question is whether the defendant purposely facilitated the offense by commanding or requesting another to do something that, had the defendant done it himself, would constitute a substantial step in offense.

A fully worked out solicitation model for algorithmic harms would need much more detail than has been presented here, but even in this very general form, it is possible to see how the evaluative criteria might play out. Solicitation performs pretty well on Criteria 1 (picking a defendant), 2 (avoiding gamesmanship), and 5 (implementability), but at the expense of falling far short on Criteria 3 (efficiency), 4 (fairness), and 6 (programming values). Since it requires a definite connection between any potential defendant and an algorithmic harm—that the defendant purposely induce the algorithm to inflict the harm—the solicitation model helps courts and plaintiffs identify which, if any, third party is liable (Criterion 1). Purpose is the sort of link between a defendant and a harm which it may be possible to obscure (e.g., by using additional intermediaries), but is harder to manipulate (Criterion 2). To forgo a purpose requires abandoning the effort entirely. Furthermore, the solicitation model would require only a modest intervention into existing law (Criterion 6). Indeed, a structurally similar variant of solicitation liability for algorithmic harm is already a part of the law. A party who purposely uses an algorithm to hurt someone may be directly liable for the injury just as if he used a tool as an implement of harm. For this reason, and because solicitation is a rather restrictive model of vicarious liability, technology firms would be unlikely to oppose it very vigorously.

As to the remaining criteria, the solicitation model is inadequate. It would bar recourse for many concerning modern-day algorithmic harms. For example, though there have been several high-profile of hiring or lending

---

* Id. § 5.04(1).
* Model Penal Code § 5.02(1).
* When the third party is a corporate person like a technology firm, even Criterion 2 may in doubt for the solicitation model. As discussed in more detail below, Subsection C, corporate mental states are inherently manipulable.
* Model Penal Code § 5.01(4).
algorithms that discriminate against protected groups, I am aware of no credible allegations that the parties behind the algorithms intended that result. In one common sort of case, the algorithm may have been trained on data that was itself inflected with the effects of systemic racism. For those whose lives are negatively impacted by discrimination, the absence of purpose is little consolation. Yet, without purpose linking a potential defendant to a harmful effect, the solicitation model provides no recourse (Criterion 4). For the same reason, it provides no incentive for technology firms to take are to avoid harming others in ways that do not involve purpose, e.g. knowingly, recklessly, or negligently causing harm (Criterion 3). Lastly, the solicitation model does nothing to advance programming values like transparency (Criterion 6). In order to do that, the transparency of an algorithm would need to have some impact on the liability inquiry. However, the solicitation model only looks to the purposes of the defendant and the injury to the victim. Because it does not care what happens in between, it has no cause to assess whether an algorithm aligns with many programming values.

B. Evaluating the No-Fault Approach

Unlike the solicitation model, the no-fault approach has several supporters among law and technology scholars. Under it, parties would automatically be liable whenever their algorithms cause harm. The no-fault model would not care what purpose the defendant had or how cautious he was. If the defendant’s algorithm harmed someone, the model says he must pay.

There is some intuitive appeal to the no-fault approach. Outside of the algorithmic context, perhaps the closest analogue is the strict civil liability that people bear for injuries any wild animals they own cause. Like wild animals, algorithms have the capacity to cause harm in unpredictable ways. In recognition of this inherent danger, it may make some sense to put the risk associated with algorithms squarely on the shoulders of the people behind them.

Despite its intuitive appeal, the no-fault approach falls short of all six criteria. To start, it does not say which of the many parties involved with a harmful algorithm should pay (Criterion 1). Any simplistic answer—e.g., always the owner of the algorithm or always its developer—must reckon with the inconvenient fact that any of the involved parties could be the true source of the

---


problem. Holding the wrong party liable could create as many problems as holding none liable. Any inflexible answer would also be open to manipulation by large, sophisticated parties who would find ways to transfer liability risk (Criterion 2). Since the no-fault approach promises to hold corporations liable for all algorithmic harms (even if the harm is a fluke or the victim contributed to the injury), it risks over-detererring investment in algorithm innovation (Criterion 3)\(^3\) and unfairly punishing innocent parties (Criterion 4). No-fault would also be a dramatic shift in the fundamentals of corporate liability, likely to provoke significant political resistance (Criterion 5).\(^4\) Finally, by treating all algorithms the same, the no-fault approach has no way to put a thumb on the scale in favor of algorithms that satisfy programming values like transparency (Criterion 6).

\textbf{C. Evaluating the Negligence-Based Approach}

The negligence-based approach to vicarious liability for algorithmic harm is the strongest of the three considered here. According to it, any party who is negligently involved with an algorithm would be liable if their negligence ends up hurting someone. The model places no ex ante limits on what sort of connection could implicate a defendant, so it could apply equally to any negligent party in the chain of an algorithm’s development and deployment.

As a surface matter, the negligence-based approach makes some important headway on the first five criteria. It identifies which parties are liable—those whose negligence leads to algorithmic harm (Criterion 1). Negligence is relatively hard to manipulate, since the primary cure for negligence is to take due care (Criterion 2). Taking due care is precisely what an efficient liability doctrine should incentivize parties to do (Criterion 3). Fairness demands, and the negligence-based approach ensures, that those who fail to take due care compensate their victims (Criterion 4). All of this is familiar terrain from basic tort law, which could make implementation relatively uncontroversial (Criterion 5).

Just below the surface, though, the relationship of the negligence-based approach to Criteria 2 through 5 becomes more complicated. The most


\(^4\) The no-fault approach does have some similarities to strict products liability. However, as I have argued elsewhere, algorithms are typically not “products” within the meaning of that liability scheme. Mihailis E. Diamantis, \textit{Algorithms Acting Badly}, 89 GEO. WASH. L. REV. (forthcoming 2021).
important potential defendants for algorithmic harms are corporations. As I and others have extensively argued, corporate mental states, including negligence, are inherently manipulable (Criterion 2). Under long-standing Anglo-American law, a corporation only has a mental state if some individual employee within the corporation has it. Large businesses can parcel out responsibilities among several employees in such a way that none could have a mental state that would trigger liability or, if any did, proof would be extremely difficult to come by. In a classic sort of case, a corporation may be very negligently run even if no individual within the corporation acted negligently; their collective deficiency may be greater than any individual deficiency. Such problems reflect organizational-level dysfunction that would not, under current law, trigger the negligence requirement.

Depending on the details of how the negligence-based approach would apply to pre-existing offenses, it would extend too much or too little vicarious liability in ways that implicate Criteria 3, 4, and 5. Current law contains a wide variety of offenses with different mental state requirements. Some of the mental states, like purpose, are considered more demanding and more serious. Many discrimination-based injuries, for example, require proof of purpose to discriminate. Penalties for purposeful offenses tend to be higher. Negligence is considered a less demanding and less serious mental state. The negligence-based model for vicarious liability would need to say how it applies to offenses that require mental states other than negligence.

One strategy would be to allow negligence to trigger vicarious liability regardless of the requisite mental state specified in the underlying offense. This significant revision and extension of the law of corporate liability would provoke vociferous opposition (Criterion 5). It would also be unfair to defendants (Criterion 4). Many offenses have damages and penalties attached that are calibrated to the severity of the transgression. Firms would understandably cry foul if mere negligence could trigger liability for more serious offenses.

Another strategy for connecting the negligence-based model to pre-existing offense definitions would be to limit the model to negligence-based offenses. That policy would significantly reduce expected pushback from tech firms (Criterion 5). However, that gain would come at the expense of leaving an algorithmic liability gap for offenses that require a more serious mental state than negligence. For such offenses, victims would have fewer options for seeking

---

55 See Diamantis, supra note 53.
59 Barocas & Selbst, supra note 50, at 711–12, 726.
redress (Criterion 4) and tech firms would have fewer incentives for steering clear (Criterion 3).

Regarding Criterion 6 (programming values), it is unclear how the negligence-based model fares. As a preliminary matter, the negligence-based approach focuses on the defendant (was the defendant negligent?) and not on the algorithm. Like the solicitation-inspired and no-fault models, this may mean that the negligence-based model has no way to promote algorithms that incorporate programming values. The analysis may turn out different if using a non-transparent algorithm were itself negligent in some circumstances. I am aware of no research establishing that conclusion.

IV. Conclusion: A Path Forward

Where does all of this leave us? I have suggested that vicarious liability is a necessary legal tool for addressing algorithmic harms, and that any model of vicarious liability should satisfy several demanding criteria pertaining to implementation, outcomes, and programming values. I showed that the criteria disqualify the three models of vicarious liability: solicitation-inspired, no-fault, and negligence-based.

A pessimist might respond with despair. Algorithms are not going away. We need vicarious liability to protect social interests and vulnerable populations. But maybe no model of vicarious liability is up to the task. This might validate the prediction of most technologists that, on the whole, advanced algorithms will not benefit society.60

Optimists (I confess myself one of them) would respond differently. There are many other approaches to vicarious liability beyond the three considered here. Surveying the analysis in this chapter, some guidelines begin to emerge for identifying the strongest candidates:

- We should look for a model that would extend rather than rewrite pre-existing legislative and judicial conclusions about how to balance efficiency and fairness. A deviation that significantly disfavors defendants relative to a salient status quo would likely stoke political opposition, and one that disfavors victims would likely to be unfair.
- A satisfactory model should be dynamic, potentially applying to any party in the chain of an algorithm’s development and deployment. Otherwise the model risks targeting the wrong defendant and distorting incentives.

We should be skeptical of models that require defendants to have some sort of mental state linking them to the algorithmic harm. The most important class of defendants are technology firms, and corporate mental states are inherently manipulable.

The model must have some way of assessing properties of an algorithm. If a model focuses exclusively on the defendant and the nature of the harm, it will have little chance of promoting programming values.

By using these guidelines as a preliminary check, we may winnow the list of candidates hit upon a model of vicarious liability that can satisfy all six criteria. Once we find it and put it into effect, we can ensure that the Second Principle of Robotics does not stand in the way of fair and efficient technological progress.