The Sangedha Framework: A Causal Forensics Protocol for Algorithmic Negligence Attribution

A definitive legal-technical doctrine establishing standards for attributing corporate liability when automated systems cause harm

Corporations deploying algorithmic systems now face unprecedented legal exposure following a convergence of three critical developments: Delaware courts have extended Caremark oversight duties to mission-critical automated systems, federal regulators have secured record enforcement actions exceeding $8 billion in 2024, and technical standards now enable mathematically rigorous causal attribution of algorithmic failures to specific governance breakdowns. (Commodity Futures Trading Co…) The Sangedha Framework synthesizes these developments into a comprehensive protocol that courts, regulators, and corporations can apply to determine when algorithmic negligence crosses the threshold from operational failure to actionable liability.

This framework matters because existing legal doctrines were developed for human decision-making, not autonomous systems that make millions of decisions per second. The gap between traditional negligence standards and algorithmic reality has created profound uncertainty about corporate accountability. Boeing paid $2.5 billion after its MCAS algorithm contributed to 346 deaths, (Newsweek) yet the legal analysis required novel applications of board oversight duties. Knight Capital lost $460 million in 45 minutes due to deprecated code, (WilmerHale +3) yet regulatory standards focused primarily on human controls. The proliferation of AI systems across finance, healthcare, transportation, and criminal justice demands a unified framework that establishes clear standards of care for algorithmic governance.

The Sangedha Framework provides this clarity through four integrated layers: legal doctrine mapping algorithmic failures to established liability theories, technical forensics enabling rigorous causal attribution, mathematical verification proving system properties with courtroom-ready rigor, and executive accountability mechanisms that pierce the corporate veil when governance failures are systematic. Together, these layers transform algorithmic negligence from a technical mystery into a legally cognizable claim with clear elements, burdens of proof, and remedial pathways.

Legal foundations establish algorithmic systems as mission-critical assets requiring board-level oversight

The Delaware Chancery Court's 2021 Boeing decision fundamentally reshaped corporate law by holding that boards "utterly failed" their oversight duties when they lacked mechanisms to monitor airplane safety despite it being "the essence" of Boeing's business. (Skadden) This marked the first time a major court found Caremark liability for failure to implement monitoring systems for algorithmic operations, specifically Boeing's MCAS software that could override pilot control. (The D&O Diary) The court rejected the business judgment rule's protection because directors made no effort to establish board-level safety reporting, waited 10 days after the first crash to discuss it, and "publicly lied" about their oversight practices. (Uiowa)

This extends the 1996 Caremark standard—requiring reasonable information and reporting systems—into the algorithmic domain with heightened scrutiny. (Wikipedia) When algorithmic systems perform mission-critical functions, the 2006 Stone v. Ritter refinement applies: directors face liability either for utterly failing to implement monitoring systems or for consciously failing to respond to red flags about system failures. (Justia) (Casebriefs) The Boeing court's application demonstrates that algorithmic systems operating autonomously or making safety-critical decisions automatically trigger the "mission-critical" designation requiring direct board oversight, not mere management delegation.

The Knight Capital enforcement action established complementary standards for operational controls. When Knight's trading algorithm executed errant orders causing $460 million in losses over 45 minutes, the SEC found willful violations of Rule 15c3-5's technology controls requirements. (SEC.gov +2) The firm failed to implement controls reasonably designed to prevent erroneous orders, lacked capital threshold alerts, and deployed code without proper testing. (WilmerHale +3) Critically, the SEC held that **human procedural failures in algorithm deployment constitute regulatory violations**, not mere technical glitches. This precedent establishes that algorithmic governance requires comprehensive pre-deployment testing, version control preventing deprecated code activation, automated alerts with proper monitoring, and emergency shutdown capabilities. (WilmerHale)

The Wells Fargo scandal provides the paradigm for sustained oversight failures creating systemic liability. Over 14 years, executives knew of fraudulent account creation driven by flawed incentive systems but failed to act, resulting in $3 billion in settlements and unprecedented individual accountability. Former CEO John Stumpf paid $17.5 million and received a lifetime banking ban; Community Bank head Carrie Tolstedt faced criminal charges and forfeited $67 million. (U.S. Department of Justice +2) This establishes that when internal reports document algorithmic system problems for extended periods, executives and boards must act decisively or face both clawback of compensation and personal penalties including criminal prosecution.

These precedents collectively establish a legal framework requiring: **(1)** board-level committees directly responsible for algorithmic system oversight when such systems are mission-critical; **(2)** regular board meeting time allocated to reviewing algorithmic performance, testing, and incidents; **(3)** mechanisms enabling boards to receive unfiltered reports of algorithmic failures, not sanitized management summaries; **(4)** immediate and thorough investigation of algorithmic failures causing harm; and **(5)** documentation demonstrating understanding of algorithmic capabilities, limitations, and risks.

Technical forensics protocols enable tamper-evident reconstruction of algorithmic decision chains

Modern forensic methodologies provide the evidentiary foundation for algorithmic negligence claims by establishing precisely what algorithms did, when they did it, who authorized it, and whether adequate controls existed. (Cyberhunt) (Git Forensics) This requires integrating six complementary forensic disciplines into a unified investigative framework meeting Federal Rules of Evidence standards for admissibility. (NIST) (NIST)

eBPF-based system observability provides real-time, kernel-level telemetry that captures algorithmic system behavior with forensic-grade integrity. (Datadog) (Sysdig) Operating within the Linux kernel itself, eBPF programs monitor system calls, file access, network connections, and process execution with sub-millisecond

precision and negligible overhead below 5% CPU usage. This creates comprehensive audit trails showing exactly which processes accessed what data, when, and with what result. Kentik Unlike user-space logging that attackers can disable or manipulate, eBPF operates in kernel space with memory access restrictions that prevent tampering. Datadog For algorithmic negligence investigations, eBPF captures the complete execution environment: which version of algorithm code ran, what input data it received, what decisions it made, and what system resources it consumed. Tools like Falco and Tracee leverage eBPF for production-grade forensic telemetry that meets chain-of-custody requirements for legal proceedings. eBPF GitHub

Merkle tree architectures transform these logs into tamper-evident evidence through cryptographic hash chains. Research!Rsc Each log entry receives a SHA-256 hash incorporated into a binary tree structure where any modification to historical entries changes the root hash detectably. Grokipedia This provides mathematical proof that logs remain unaltered from collection through courtroom presentation. swtch +3 Certificate Transparency, Google's transparency log system protecting SSL certificates, demonstrates this approach's legal viability—courts accept CT logs as self-authenticating evidence under FRE 902(14). swtch Research!Rsc For algorithmic systems, Merkle trees enable proof of inclusion (showing a specific algorithmic decision existed in the log) and proof of consistency (demonstrating current logs contain all previous entries unmodified). Research!Rsc The constant-time append operations and logarithmic-time verification make this practical even for systems generating millions of log entries daily. Research!Rsc

Git forensics provides attribution of algorithmic code to specific developers with cryptographic certainty. Every commit includes SHA-256 hashes of content, author metadata with timestamps, and optional GPG signatures preventing repudiation. Grokipedia +2 The distributed nature of Git means multiple independent copies of repository history exist, making history rewriting detectable. Grokipedia For negligence analysis, Git archaeology identifies: who introduced specific code sections, when testing occurred, what code review processes were followed, whether dangerous code was flagged during review, and whether known-problematic code was reverted then reintroduced. JupyterLab The ability to use git bisect to binary-search through thousands of commits and identify the exact change that introduced a bug provides powerful causation evidence.

Memory forensics captures the runtime state of algorithmic systems through RAM dumps analyzed with the Volatility Framework. National Institute of Standards … This reveals: loaded algorithm code and libraries, decrypted data existing only in memory, active network connections, process relationships showing whether malware infected algorithmic processes, and injected code indicating compromise. While volatile by nature, proper collection procedures using hardware write-blockers and immediate cryptographic hashing establish integrity. Memory forensics proves critical for determining whether algorithmic failures resulted from legitimate code errors, malicious compromise, or unauthorized modifications not reflected in source repositories.

Network packet analysis reconstructs the distributed execution of algorithmic systems by capturing all network traffic to and from algorithmic infrastructure. National Institute of Standards … Wireshark and similar tools provide microsecond-precision timestamps synchronized to NTP sources, enabling precise timeline reconstruction. For algorithmic trading systems, packet captures prove exactly when orders transmitted, what market data the algorithm received, and whether the system exhibited anomalous network behavior indicating compromise. The Supreme Court's Daubert standard requires that forensic methodologies have known error rates, standardized

procedures, and peer review— Justia Annual Reviewspacket analysis meets these requirements through decades of established practice and NIST standardization. NIST

Statistical anomaly detection identifies algorithmic behavior deviations from established baselines using machine learning on system logs. Techniques like isolation forests, autoencoders, and LSTM networks trained on normal operation data flag anomalous patterns requiring investigation. ResearchGate +4 The SEC's National Exam Program Analytics Office uses similar methods to detect irregular trading patterns. For negligence attribution, anomaly detection answers critical questions: Did algorithmic behavior change after a specific code deployment? Do certain algorithmic decisions show statistical bias indicating discrimination? Did the system exhibit warning signs before catastrophic failure? Critically, these methods must document false positive/negative rates and validation procedures to meet Daubert's requirements for expert testimony about analytical methodologies. Annual Reviews

Mathematical verification provides courtroom-ready proofs of algorithmic properties and failures

The Sangedha Framework's mathematical layer transforms technical claims about algorithms into rigorous proofs meeting scientific evidence standards. This layer draws from formal methods developed over four decades in computer science, now mature enough for legal applications requiring certainty beyond statistical confidence.

Formal verification using proof assistants like Coq, Isabelle, and TLA+ establishes algorithmic properties with mathematical certainty. CompCert, a verified C compiler proven correct in Coq through 200,000+ lines of proof, demonstrates this approach's maturity. The seL4 microkernel, verified in Isabelle, proves that its implementation correctly enforces security policies—if seL4 fails, the proof identifies an error in the formal specification, not the implementation. Wikipedia chlipala For algorithmic negligence, formal verification addresses critical questions: Does an algorithm provably implement stated requirements? Do safety properties hold under all possible inputs? Can the algorithm enter unsafe states? The proofs themselves become evidence, with small trusted computing bases that experts can verify independently.

The key advantage over testing lies in completeness. Testing explores specific scenarios while formal verification proves properties hold for all possible executions. NIST Amazon Web Services relies on TLA+ to verify distributed systems like S3 and DynamoDB, finding serious bugs that testing missed. For legal purposes, formal verification establishes either that safety properties were proven (indicating due diligence) or that no verification occurred despite safety-critical operations (indicating negligence). The Daubert factors strongly favor formal methods: they are testable (proofs can be checked mechanically), peer-reviewed (published in venues like CAV and POPL), have known limitations (decidability boundaries are well-understood), follow standardized procedures, and enjoy acceptance in the computer science research community. Annual Reviews NIST

Probabilistic model checking quantifies risks in algorithmic systems operating under uncertainty. Tools like PRISM and Storm model algorithms as Markov Decision Processes and compute exact probabilities of failures or expected time to catastrophic events. Springer +2 For autonomous vehicles, model checking can prove statements like "the probability of collision given detected obstacle is less than 10^{-9} per hour" or identify that no

such guarantee exists. The mathematics underlying probabilistic model checking—value iteration, policy synthesis, reachability analysis—enables counterfactual reasoning: Would alternative algorithmic strategies have prevented the observed failure?

The Boeing MCAS failure illustrates where probabilistic verification could have identified risks. MCAS relied on a single angle-of-attack sensor without redundancy, and its repeated nose-down commands overwhelmed pilot control. (Delaware Courts) (Newsweek) Model checking of this architecture would have revealed: unacceptable probability of catastrophic failure given known sensor failure rates, existence of alternative policies (sensor fusion, pilot override) with orders of magnitude better safety guarantees, and violation of safety properties under realistic fault scenarios. Boeing's failure to conduct such analysis despite MCAS being safety-critical demonstrates the negligence standard: when algorithms control life-safety systems, probabilistic verification becomes part of reasonable care.

Temporal logic provides the specification language for expressing safety requirements formally. Linear Temporal Logic captures properties like "if the algorithm detects an obstacle, emergency braking must activate within 100 milliseconds"—expressed as **G**(obstacle_detected → **F**≤100ms emergency_brake). Computation Tree Logic handles branching futures: "after any system state, it remains possible to return to a safe state"—expressed as **AG**(**EF** safe_state). (Wikipedia) These specifications transform natural language regulatory requirements into mathematically precise properties that model checkers can verify algorithmically. The SEC's proposed Predictive Analytics rule requiring investment advisers to eliminate conflicts of interest could be expressed in temporal logic, enabling automated verification of compliance.

Causal inference using transfer entropy and Granger causality establishes directed causal relationships between algorithmic inputs and outputs. Transfer entropy **T_X→Y** measures information flow from variable X to variable Y, quantifying how much knowing X's past improves prediction of Y's future beyond Y's own history. This distinguishes mere correlation from causation. (Wikipedia) (arXiv) For algorithmic bias analysis, transfer entropy can prove whether protected characteristics like race causally influence algorithmic decisions, or whether correlations arise spuriously from confounders. Granger causality, proven equivalent to transfer entropy for Gaussian processes, provides a computationally lighter alternative suitable for large-scale log analysis. (arXiv) (APS)

The legal significance lies in moving from "algorithm A was running when harm B occurred" to "algorithmic decision A caused harm B with quantified confidence intervals." Judea Pearl's do-calculus framework enables counterfactual analysis: "If the algorithm had not taken action A, would harm B have occurred?" (Project MUSE +2) These causal methods require careful attention to confounders and hidden variables, but when properly applied provide scientific rigor meeting Daubert standards. The landmark Daubert decision itself involved causal claims about birth defects— (Wikipedia +2)algorithmic causality analysis uses fundamentally similar statistical methodologies now with decades of peer review in epidemiology and econometrics.

Statistical hypothesis testing establishes negligence through formal tests comparing algorithmic behavior to legal standards. For disparate impact claims under anti-discrimination law, two-proportion z-tests determine whether algorithms grant favorable outcomes to protected groups at statistically different rates. (Frontier) **Cohen's d** effect sizes quantify the magnitude of discrimination, with established conventions (d=0.2 small, d=0.5 medium, d=0.8 large) enabling courts to assess materiality. Power analysis ensures adequate sample sizes

—underpowered studies that fail to detect discrimination due to insufficient data do not exculpate defendants. Wikipedia

For legal proceedings, hypothesis testing must address multiple comparisons carefully. Testing 100 algorithmic fairness metrics at α=0.05 yields five false positives on average. Frontier Bonferroni correction (α'=α/k) or Benjamini-Hochberg false discovery rate control maintains statistical validity. Courts applying Daubert scrutinize whether experts properly controlled Type I error inflation. The legal standard of proof varies by context—criminal prosecution requires proof beyond reasonable doubt (approximately 95-99% confidence), while civil cases use preponderance of evidence (>50% probability). Frontier Properly conducted statistical analysis with reported confidence intervals enables courts to assess whether evidence meets the applicable burden.

The integrated framework establishes clear liability standards for algorithmic governance failures

The Sangedha Framework synthesizes legal precedents, technical forensics, and mathematical verification into a unified protocol for algorithmic negligence attribution. This integration occurs across four sequential phases: **(1)** establishing duty through mission-critical designation, **(2)** documenting breach through forensic evidence of governance failures, **(3)** proving causation through mathematical analysis linking failures to harms, and **(4)** attributing individual liability through executive accountability mechanisms.

Phase 1 establishes that algorithmic systems performing core business functions trigger enhanced oversight duties. The mission-critical standard derives from Boeing's holding that algorithmic systems controlling safety-critical functions require direct board oversight. This extends to: algorithmic trading systems controlling capital deployment at financial institutions, machine learning models making credit decisions affecting consumer access to capital, recommendation algorithms determining content exposure on platforms with public safety implications, and autonomous vehicle control systems. When algorithms make decisions previously requiring human judgment in regulated domains, they automatically qualify as mission-critical. This designation imposes five specific requirements: dedicated board committee with algorithmic oversight responsibility, quarterly review of algorithmic performance metrics and incident reports, direct reporting channels from technical teams to board (not filtered through management), documented understanding of algorithmic capabilities and limitations, and immediate board notification of material algorithmic failures.

Phase 2 documents governance failures through forensic evidence collection and analysis. Investigators deploy the six forensic methodologies in parallel: eBPF telemetry captures real-time system behavior, Merkle tree logs provide tamper-evident audit trails, Git analysis attributes code to specific developers and identifies testing gaps, memory forensics reveals runtime state and potential compromises, network analysis reconstructs distributed system interactions, and statistical anomaly detection flags deviations from normal behavior. Cyberhunt +2 Each methodology generates specific evidence types: eBPF shows which algorithm versions executed and what decisions they made, Merkle trees prove log integrity with cryptographic certainty, Git commits demonstrate whether code review processes identified risks, memory dumps reveal whether malware compromised algorithmic systems, packet captures establish precise timing of distributed system communications, and anomaly detection identifies suspicious behavioral changes. The integration of multiple

evidence sources enables triangulation—convergent evidence from independent methodologies strengthens causal claims while divergent evidence flags investigation gaps.

Phase 3 establishes causation through mathematical analysis connecting governance failures to observed harms. This employs four complementary techniques: formal verification reveals whether safety properties were proven before deployment, probabilistic model checking quantifies failure probabilities and identifies safer alternative strategies, causal inference using transfer entropy establishes directed causation from algorithmic decisions to harms, and statistical hypothesis testing determines whether algorithmic behavior violates legal standards with quantified confidence. For example, investigating an autonomous vehicle collision would: check whether safety properties were formally verified (establishing due diligence or its absence), use probabilistic model checking to compute collision probability given system architecture and prove whether alternative designs would have prevented the incident, apply transfer entropy to determine which system components (perception, planning, control) causally contributed most to the collision, and conduct statistical tests comparing the system's collision rate to regulatory safety standards or human baseline performance. The mathematical rigor of these methods enables them to survive Daubert challenges—they are testable, peer-reviewed, have known error rates, follow standardized procedures, and are generally accepted in relevant scientific communities. (Leppard Law +3)

Phase 4 attributes individual liability to executives who failed oversight duties. Multiple liability theories apply depending on specific failures. Sarbanes-Oxley Section 302 imposes personal certification duties on CEOs and CFOs for internal controls—algorithmic systems affecting financial reporting fall within this scope. Section 404 requires management to assess control effectiveness annually, extending to algorithmic controls. Dodd-Frank's mandatory clawback provisions require recovery of executive compensation following accounting restatements triggered by algorithmic errors, regardless of fault. (SEC.gov +2) Securities fraud claims under Rule 10b-5 attach when executives make material misrepresentations about algorithmic capabilities while knowing of system deficiencies—the SolarWinds case established this extends to technical officers like CISOs. (SEC.gov +3) Criminal obstruction charges under 18 U.S.C. § 1519 apply when executives conceal algorithmic failures during regulatory investigations, as demonstrated by the conviction of Uber's Chief Security Officer for concealing a data breach. (Department of Justice +4) State law fiduciary duty claims provide an additional liability path—both over-reliance on algorithmic decisions without understanding (abdication of duty) and under-utilization of available algorithmic tools (falling behind industry standards) can constitute breaches.

This four-phase structure provides clarity for corporations implementing algorithmic governance. The requirements are specific and actionable: identify mission-critical algorithmic systems through objective criteria (safety impact, regulatory significance, scale of decisions), implement required oversight structures (board committees, reporting mechanisms, incident response protocols), deploy forensic capabilities proactively (eBPF monitoring, Merkle tree logging, comprehensive version control, statistical baselines), and document verification efforts (formal verification attempts, probabilistic model checking results, causal analysis of deployed systems, statistical validation of fairness properties). Corporations that implement these measures establish strong evidence of reasonable care, while those lacking such documentation face substantial liability exposure.

Regulatory convergence across multiple jurisdictions reinforces the framework's core principles

The Sangedha Framework aligns with emerging regulatory requirements across the European Union, United Kingdom, United States, and Singapore, indicating global convergence toward specific algorithmic governance standards. This regulatory alignment strengthens the framework's legitimacy and provides corporations with clear compliance pathways.

The EU AI Act, effective August 2024 with staged implementation through 2026, mandates comprehensive risk management systems for high-risk AI under Article 9. White & Case LLP This requires continuous iterative risk assessment throughout the AI lifecycle, evaluation under both intended use and reasonably foreseeable misuse scenarios, and integration with post-market monitoring. EU Artificial Intelligence Act artificialintelligenceact Article 17's quality management system requirements demand documentation of design choices, model selection decisions, and risk mitigation measures—directly supporting forensic reconstruction of algorithmic governance. IAPP The enforcement mechanism imposes fines up to €35 million or 7% of global revenue for prohibited practices, creating substantial incentives for robust governance. DLA Piper The framework's technical forensics protocols enable companies to demonstrate compliance with Article 9's risk management requirements through documented testing, validation, and monitoring.

The UK Online Safety Act, with illegal content duties enforceable from March 2025, requires platforms to assess how algorithms impact harmful content exposure. Kennedys Law LLP Regulator Ofcom can impose fines up to £18 million or 10% of worldwide revenue and bring criminal charges against senior managers for failures. Kennedys Law LLP www The Act's risk assessment requirements align precisely with the Sangedha Framework's Phase 1 mission-critical designation—companies must identify where algorithmic content distribution creates safety risks and implement controls. The framework's statistical anomaly detection methodologies enable platforms to monitor algorithmic behavior for concerning patterns, while formal verification can prove content moderation algorithms satisfy safety properties.

Singapore's Model AI Governance Framework, updated in 2020, establishes an accountability-based approach emphasizing explainability, transparency, and fairness. IMDA The framework mandates human oversight at appropriate levels (human-in-the-loop, human-over-the-loop, or human-out-of-the-loop) based on risk assessment. PDPC Its algorithm requirements—explainability, robustness, regular tuning, traceability, reproducibility, and auditability—map directly to the Sangedha Framework's technical forensics requirements. IMDA pdpc The complementary AI Verify testing framework provides standardized tests for 11 principles, enabling companies to demonstrate governance effectiveness. PDPC While Singapore's framework remains voluntary, courts increasingly reference it when assessing reasonable care standards under the Personal Data Protection Act.

US regulatory enforcement has intensified dramatically, with the SEC's fiscal year 2024 producing record $8.2 billion in financial remedies and 124 officer and director bars. Cleary Gottlieb The SEC's enforcement actions against "AI washing"—false claims about AI capabilities—establish that existing securities laws fully apply to algorithmic systems with no technology exception. Wealth Management White & Case LLP The March 2024 actions against Delphia and Global Predictions, settling for $225,000 and $175,000 respectively for false AI claims,

demonstrate regulators' willingness to pursue relatively modest violations to establish precedents. Mayer Brown +2 The SEC's 2025 Examination Priorities explicitly target AI use in investment advice, trading, and back-office operations. White & Case LLP The CFTC's Electronic Trading Risk Principles, proposed in 2020, require prevention, detection, and mitigation controls for algorithmic trading—directly paralleling the Sangedha Framework's forensic capabilities. Akin Gump Strauss Hauer & F… Pre-trade risk controls (order frequency limits, size parameters, price collars, self-trade prevention) align with formal verification's ability to prove algorithms respect bounds.

IEEE Standard 7003-2024 on algorithmic bias provides technical specifications that integrate seamlessly with the framework's mathematical verification layer. IEEE Standards Group The standard requires validation dataset criteria ensuring representativeness, application boundary documentation preventing out-of-scope use, user expectation management, and bias profile development balancing productive and harmful bias. IEEE These requirements map to: statistical hypothesis testing for bias detection (validation datasets), formal specification of algorithm scope (application boundaries via temporal logic), and causal inference identifying discriminatory pathways (bias profiling through transfer entropy). Organizations can cite IEEE 7003 compliance as evidence of reasonable care while leveraging the Sangedha Framework's verification methods to demonstrate actual compliance rather than aspirational policy statements. IEEE Standards Group

This regulatory convergence creates powerful network effects. Companies implementing the Sangedha Framework to comply with EU AI Act requirements simultaneously satisfy UK Online Safety Act obligations, SEC examination priorities, and IEEE technical standards. The framework functions as a unified compliance architecture addressing multiple jurisdictions' requirements through integrated governance rather than jurisdiction-specific point solutions. Multinational corporations benefit from standardized forensic infrastructure, verification methodologies, and documentation that demonstrate compliance across regulatory regimes. As algorithmic systems increasingly operate globally, this unified framework reduces compliance costs while providing superior governance compared to fragmented approaches.

Implementation requires organizational integration across legal, technical, and executive functions

Successful deployment of the Sangedha Framework requires corporations to bridge historically separate organizational silos, creating integrated teams combining legal expertise, technical capabilities, and executive oversight. This organizational transformation proves as critical as the technical methodologies themselves.

Legal teams must develop technical literacy sufficient to specify algorithmic requirements in temporal logic and assess verification evidence. This does not require lawyers to become computer scientists, but demands familiarity with formal specification concepts, probabilistic reasoning, and causal inference frameworks. Progressive legal departments are hiring "legal engineers" with computer science backgrounds who translate regulatory requirements into formal specifications that verification tools can process. For example, GDPR's right to deletion within 30 days becomes the temporal logic formula $\mathbf{G}(\text{deletion\_request} \rightarrow \mathbf{F}{\leq}30\text{days data\_deleted})$, which PRISM can model check against data retention system specifications. prismmodelchecker Similarly, fair lending requirements prohibiting discrimination become statistical hypothesis tests comparing approval rates across protected groups with documented significance levels and effect sizes.

Legal teams must also understand chain of custody requirements for digital forensic evidence, ensuring technical teams collect evidence meeting FRE 902(14) standards for self-authentication. Jatheon +2

Technical teams must adopt forensic-grade development practices treating all systems as potentially subject to legal scrutiny. This shifts software development from optimizing purely for performance and features toward prioritizing auditability, explainability, and verifiability. Concretely, this means: implementing eBPF-based observability from initial deployment rather than adding it post-incident, structuring all logs as Merkle trees with cryptographic integrity guarantees, requiring GPG-signed Git commits with detailed messages explaining changes, conducting formal verification for safety-critical components with documented proof attempts, maintaining comprehensive test suites with coverage metrics and documented test case selection rationale, and performing regular bias audits using statistical methods with published methodologies. Technical teams must recognize that "it works in testing" provides insufficient governance—they must prove properties hold through verification or document why verification is infeasible.

Executive teams must establish governance structures explicitly allocating algorithmic oversight responsibilities. The board must create a dedicated Technology Risk Committee (or expand existing Risk Committee mandates) with: at least one director with computer science or AI expertise, quarterly meetings reviewing algorithmic incident reports and verification results, direct access to technical teams without management filtering, authority to retain independent technical auditors, and explicit charter covering algorithmic systems performing mission-critical functions. The CEO must designate a Chief AI Officer or Chief Algorithm Officer at C-suite level with: authority to halt deployments failing verification requirements, responsibility for enterprise-wide algorithmic governance policy, budget for verification tools and external audits, and direct reporting line to board Technology Risk Committee. The CFO must ensure internal controls under SOX 404 explicitly cover algorithmic systems affecting financial reporting, with documented testing procedures and control deficiency escalation paths.

Cross-functional Algorithmic Review Boards must approve high-risk system deployments. These boards should include: legal counsel assessing regulatory compliance and liability risk, technical architects reviewing verification evidence and forensic readiness, ethicists evaluating fairness and bias implications, business owners articulating value and accepting residual risks, and security teams confirming systems resist tampering and maintain evidence integrity. The board reviews documentation packages including: formal specifications of safety properties, probabilistic model checking results quantifying failure risks, statistical bias analysis with confidence intervals, verification attempts (successful proofs or documented infeasibility), incident response and forensic readiness plans, and executive accountability and compensation clawback triggers. Only systems passing this review with documented board approval should enter production.

This organizational integration enables rapid, effective response when algorithmic incidents occur. Pre-deployed forensic infrastructure immediately captures evidence. Legal teams understand what evidence exists and how to preserve it. Technical teams can conduct causal analysis and verification while maintaining chain of custody. Executives have clear escalation protocols and authority to make decisions. The alternative—discovering after an incident that forensic capabilities don't exist, evidence was overwritten, technical teams lack causal analysis skills, and accountability structures are ambiguous—exposes corporations to massive liability.

The framework establishes algorithmic negligence as a cognizable claim with clear elements and remedies

The Sangedha Framework transforms algorithmic harms from technical mysteries into structured legal claims that courts can adjudicate using established liability theories and evidentiary standards. This crystallization enables consistent application across cases while preserving judicial flexibility for novel scenarios.

Element 1: Duty arises when algorithmic systems perform mission-critical functions. Plaintiffs establish duty by proving algorithms: **(a)** control safety-critical operations (autonomous vehicles, medical treatment recommendations, critical infrastructure), **(b)** make decisions at scale affecting protected rights (credit, employment, housing, education), **(c)** operate in regulated domains with fiduciary obligations (investment advice, legal services, healthcare), or **(d)** execute functions previously requiring human professional judgment. This element admits expert testimony about industry standards—what do reasonable corporations do when deploying similar algorithmic systems? Expert witnesses can reference IEEE 7003, ISO/IEC 27001:2022 algorithmic security controls, or NIST AI Risk Management Framework as evidence of reasonable care standards. (IEEE Standards Group) Defendants failing to meet these standards bear burden of explaining why departure was reasonable.

Element 2: Breach occurs through specific governance failures documented by forensic evidence. Plaintiffs prove breach by demonstrating: **(a)** utter failure to implement algorithmic monitoring systems (Caremark Prong 1), **(b)** conscious failure to respond to red flags about algorithmic problems (Caremark Prong 2), **(c)** deployment without adequate testing, validation, or verification, **(d)** absence of forensic capabilities enabling post-incident analysis, or **(e)** material misrepresentations about algorithmic capabilities or limitations. Each failure type corresponds to specific evidence: board minutes showing no algorithmic oversight discussions (Prong 1), internal emails documenting known problems without remediation (Prong 2), absence of test documentation or failed tests that were ignored (inadequate testing), logs showing no integrity verification mechanisms (no forensics), and public statements contradicting internal assessments (misrepresentation). The forensic methodologies in Phase 2 generate precisely this evidence—eBPF logs prove what monitoring existed, Git archaeology reveals testing practices, and anomaly detection identifies ignored warning signs.

Element 3: Causation links governance failures to harms through mathematical analysis. Plaintiffs establish causation using: **(a)** formal verification showing safety properties were never proven despite safety-critical deployment, **(b)** probabilistic model checking demonstrating failure inevitability or quantifying elevated risk, **(c)** transfer entropy proving algorithmic decisions causally influenced outcomes, and **(d)** statistical hypothesis tests showing algorithmic behavior violated legal standards. This element requires expert testimony meeting Daubert standards—experts must explain methodologies, demonstrate peer review and publication, report known error rates, show adherence to standards, and establish general acceptance. (Leppard Law +2) Defense experts can challenge causal claims by proposing alternative explanations, identifying confounding variables, questioning sample sizes, or disputing model validity. Courts resolve these battles of experts using Daubert gatekeeping—excluding methodologies failing scientific validity standards while admitting properly conducted analyses even if parties dispute interpretations.

Element 4: Damages flow from algorithmic harms with computation methodology. Damage calculations vary by harm type: financial losses from algorithmic trading errors use market-based valuation methods, personal injuries from autonomous vehicle collisions employ standard tort damages, discriminatory denials of credit or employment use economic models of lifetime earning losses, and constitutional harms from biased criminal justice algorithms may warrant punitive damages. Class action certification becomes available when algorithmic systems harm large groups similarly—the algorithm's uniformity of operation often satisfies commonality requirements more easily than individual human decisions. Statistical sampling of class members' damages with confidence intervals provides computationally feasible estimation for large classes. Defendants may raise contributory negligence or intervening cause defenses, but algorithmic systems' opacity often precludes plaintiffs from understanding and avoiding risks, weakening such defenses.

Remedies span equitable relief, compensatory damages, and structural reforms. Courts can order: immediate suspension of algorithmic systems failing safety verification, algorithm disgorgement requiring deletion of models trained on illegally obtained data (FTC remedy pioneered in Cambridge Analytica), appointment of independent monitors conducting ongoing verification audits, mandatory implementation of forensic infrastructure and governance structures, disclosure of algorithmic testing and validation results to affected parties, and individual liability including clawback of executive compensation and officer bars. The Wells Fargo precedent demonstrates courts' willingness to impose severe personal consequences on executives ($67 million forfeiture, criminal prosecution) when governance failures are systematic. Harvard Law School Forum on … Congress.gov The SEC's record enforcement numbers—$8.2 billion in 2024—signal regulators' commitment to substantial penalties. Secretariat Cleary Gottlieb Criminal prosecution under 18 U.S.C. § 1519 remains available when evidence destruction accompanies algorithmic failures, with 20-year maximum sentences providing deterrent effect. Legal Information Institute +3

Future evolution will extend the framework to emerging algorithmic domains and liability theories

The Sangedha Framework provides foundational architecture that extends naturally to algorithmic domains beyond those addressed by current case law and regulation. Three categories warrant particular attention: autonomous weapons systems raising novel questions about liability for algorithmic lethality, synthetic media and deepfakes creating harm through algorithmic content generation, and quantum-resistant cryptography requirements for long-term evidence preservation.

Autonomous weapons systems present extreme cases of algorithmic lethality. When algorithms make kill decisions, governance requirements intensify dramatically. International humanitarian law prohibits weapons incapable of distinguishing combatants from civilians—algorithms must provably satisfy this requirement through formal verification of targeting logic. The "Martens Clause" demanding weapons remain under meaningful human control maps to human-over-the-loop oversight requirements with documented human judgment in kill chains. Military organizations adopting the Sangedha Framework would: formally verify targeting algorithms satisfy international humanitarian law rules, probabilistically model civilian casualty risks under various deployment scenarios, maintain forensically sound logs of all targeting decisions enabling post-action review, and establish clear accountability chains from operational commanders through technical developers. When autonomous weapons cause civilian casualties, the framework's causal analysis determines

whether algorithmic failures, inadequate testing, or governance breakdowns bear responsibility. Criminal liability under Rome Statute provisions for war crimes may attach to commanders or developers when governance failures rise to willful disregard.

Synthetic media and deepfakes illustrate algorithmic content generation harms. Generative AI systems producing photorealistic false content enable defamation, fraud, election interference, and non-consensual intimate imagery at unprecedented scale. Liability theories under the Sangedha Framework address: **(a)** deployers who release generative models without adequate safeguards, analogous to distributing tools specifically designed for illegal purposes; **(b)** platforms hosting synthetic content without detection mechanisms, potentially violating Section 230's carve-out for intellectual property and federal criminal laws; and **(c)** individual actors using synthetic media to cause specific harms, with generative AI operators potentially liable as accomplices. Governance requirements include: provenance tracking via cryptographic signatures embedded in generated content (C2PA standard), formal verification that content moderation algorithms detect synthetic media with documented false negative rates, statistical monitoring of platform content identifying synthetic media concentration, and incident response protocols for rapid takedown when harmful synthetic content propagates. The framework's forensic capabilities enable attributing synthetic content to specific generator models and operators through statistical fingerprinting of generation artifacts.

Quantum computing threatens current cryptographic evidence integrity. SHA-256 hash functions and RSA signatures securing forensic evidence remain secure against classical computers but face potential vulnerability to quantum algorithms. Shor's algorithm, when implemented on sufficient quantum computers, breaks RSA and ECC in polynomial time. Current evidence secured only with classical cryptography may be harvested now and decrypted later when quantum computers mature. The NIST FIPS 203/204/205 post-quantum cryptography standards (ML-KEM, ML-DSA, SLH-DSA) provide quantum-resistant alternatives. The Sangedha Framework requires: immediate deployment of hybrid classical+post-quantum cryptography for new evidence, migration of existing evidence archives to post-quantum protection before quantum computers threaten classical schemes, and documentation enabling courts to assess cryptographic validity as technology evolves. Evidence cryptographically secured in 2025 that faces litigation in 2040 must use post-quantum cryptography to ensure integrity throughout case lifecycles potentially spanning decades.

These extensions demonstrate the framework's adaptability. The four-phase structure—establishing duty, documenting breach, proving causation, attributing liability—applies regardless of algorithmic domain. The forensic methodologies remain constant: eBPF captures system behavior, Merkle trees ensure integrity, Git attributes code, memory analysis reveals runtime state, network analysis reconstructs interactions, and statistical methods identify patterns. The mathematical verification techniques extend naturally: formal methods prove targeting algorithms' properties, probabilistic verification quantifies deepfake detection reliability, and causal inference determines responsibility for autonomous weapons' actions. The legal theories remain grounded in established doctrines: Caremark oversight duties, SOX internal controls, securities fraud, criminal obstruction, and fiduciary duties apply uniformly. This universality enables courts and regulators to apply consistent standards as algorithmic systems penetrate new domains, providing predictability while enabling evolution.

The Sangedha Framework establishes algorithmic negligence attribution as a mature legal-technical discipline with clear standards, rigorous methodologies, and predictable outcomes. By integrating four decades of computer science research on formal verification with established legal doctrines on corporate oversight, the framework transforms opaque algorithmic failures into analyzable governance breakdowns. The technical forensics protocols provide courts with evidence meeting FRE 902(14) self-authentication standards. The mathematical verification methods survive Daubert challenges through demonstrated testability, peer review, known error rates, standardized procedures, and scientific acceptance. The liability theories ground in Supreme Court and Delaware precedent rather than untested novel doctrines.

Corporations implementing this framework gain substantial benefits beyond liability reduction. Formal verification identifies bugs before deployment, probabilistic model checking optimizes algorithm parameters, causal analysis improves system performance, and statistical monitoring detects problems early. The forensic infrastructure enables rapid incident response and root cause analysis. The governance structures improve decision-making quality by forcing technical and business stakeholders to explicitly articulate assumptions, risks, and mitigations. The organizational integration breaks down silos, creating engineering cultures that value robustness over rapid deployment.

The framework's adoption will proceed through three stages. Early adopters in highly regulated domains—financial services, healthcare, autonomous vehicles—implement comprehensive frameworks to satisfy regulatory examination priorities and reduce massive liability exposures. Industry standards bodies including IEEE, ISO, and sector-specific organizations codify frameworks into technical standards and best practices. Finally, courts recognize framework compliance as evidence of reasonable care, establishing it as the de facto standard of care for algorithmic governance. Within a decade, the question in algorithmic negligence cases will shift from "were algorithms involved?" to "did the organization implement Sangedha Framework governance or equivalent?"

The stakes demand nothing less. Algorithmic systems now make billions of consequential decisions annually affecting individuals' financial access, employment prospects, criminal justice outcomes, physical safety, and constitutional rights. The economic incentives driving algorithmic deployment will not diminish—algorithms scale human judgment at near-zero marginal cost. Without robust governance frameworks establishing clear accountability, algorithmic harms will proliferate while responsible parties evade liability through complexity and opacity. The Sangedha Framework provides the legal-technical infrastructure ensuring that algorithmic power remains subject to human accountability and that when algorithms cause harm, responsible parties face consequences proportionate to governance failures. This represents not a restriction on beneficial technology but the necessary precondition for algorithmic systems' legitimate deployment at scale.