SCIENTIFIC CULTURE, Vol. 12, No. 5, (2026), pp. 923-936
Open Access. Online & Print

www.sci-cult.com
DOI: 10.5281/zenodo.20720910

TRUSTWORTHY MACHINE LEARNING IN COMPUTER ENGINEERING: A REVIEW OF ROBUSTNESS, FAIRNESS, AND INTERPRETABILITY

Dr Priyanka Mishra[1*], Milind Nemade[2], Nayani Sateesh[3], Deepak A Vidhate[4], Soukaryo Chandra[5]

[1]BTech, MTech , PhD, Poornima University, Recommender systems through machine learning and data science, Computer Science. & Engineering. Email ID: prynkmshr@gmail.com*
[2]Head and Professor, K J Somaiya Institute of Technology, University of Mumbai, Artificial Intelligence and Data Science. Email ID: mnemade@somaiya.edu, Orcid ID:0000-0002-3051-3056
[3]Senior Assistant Professor, CVR College of Engineering, Ibrahimpatnam, Telangana, Computer Science and Engineering. Email ID: n.sateesh@cvr.ac.in
[4]Professor & Head, Dr Vithalrao Vikhe Patil College of Engineering, Ahilyanagar, Maharashtra, Machine Learning and Information Technology. Email ID: dvidhate@yahoo.com

[5]Computer and Communication Engineering, Manipal Academy of Higher Education, Manipal, Karnataka, Email ID: soukaryochandra@gmail.com
orcid id: 0009-0000-2275-7707

Abstract

Today, machine learning is integrated into computer engineering systems like cyber security, embedded intelligence, computer vision pipelines, software analytics, networked infrastructures and autonomous decision support environments. In many applications these systems are required to perform in evolving, competitive and impactful environments, and the traditional metrics of accuracy, precision, recall and computational efficiency are no longer adequate for assessing the quality of the model. A reliable machine learning system should also be robust to perturbations, distributional uncertainty, be fair to all affected individuals and groups, and be interpretable for debugging, audit, contestability, and governance. It explores trustworthy machine learning from the three main perspectives of robustness, fairness and interpretability, and places them in the context of the engineering lifecycle of data preparation, model development, validation, deployment, monitoring, and accountability. The article is structured into a synthesis of 47 references published since 2010 and concludes that trustworthiness should be considered more as an engineering requirement at the system level, rather than a posteriori ethical or regulatory corrective. The review also reveals that there is a strong interdependency between robustness, fairness and interpretability. Changes to one dimension can introduce tensions to another, and a holistic (integrated) evaluation is needed. It ends with a proposal for a lifecycle-based agenda for trustworthy machine learning in computer engineering, for which it is crucial to benchmark from multiple perspectives, to validate the system in deployment settings, to be accountable for the system design, and to manage cross-dimensional trade-offs explicitly.

Keywords: *trustworthy machine learning, computer engineering, interpretability, explainable AI, adversarial machine learning.*

1. Introduction

Machine Learning is increasingly a part of current computer engineering. It can be used for intrusion detection, biometric authentication, computer vision, embedded systems, autonomous control, network traffic analysis and software defect prediction, recommender systems, intelligent resource allocation, and human-facing digital platforms. This expansion has turned machine learning into a key component of engineering infrastructure that makes predictions. Therefore, the question is not so much whether a model works well in certain standardized test conditions, but whether it is reliable when used in complex, uncertain and socially significant contexts. The robustness of the technical systems, transparency, privacy, fairness and human oversight and accountability of artificial intelligence systems have thus been identified as key requirements of the system and not just a peripheral ethical goal (European Commission, High-Level Expert Group on Artificial Intelligence, 2019).

This also translates to risk-based governance approaches, which include characteristics of trustworthy AI in the AI lifecycle, including validity, reliability, safety, security, resilience, explainability, privacy, fairness and accountability (National Institute of Standards and Technology, 2023). These properties need to be translated to operate in computer engineering. They must be integrated into data pipelines, modelling design, validation processes, deployment monitoring, incident response and auditing processes. While a system can be accurate, it can also fail when it is distributed, be manipulable by attackers, lead to disparate errors for different groups, or be impenetrable for engineers, auditors, or affected stakeholders.

The shift in literature from principles to practice in the field of trustworthy AI has been growing. It demonstrates that the trustworthiness of machine learning systems is a coordinated combination of technical, social and organizational mechanisms that influence the design, validation and deployment of machine learning systems (Li et al., 2023). In this review, the system-oriented understanding is considered, but only three aspects are taken into special consideration relevant to computer engineering – robustness, fairness and interpretability. Robustness is related to models' reliability in face of perturbations, uncertainties and attacks. Fairness is about avoiding and minimizing unnecessary inequalities between people, groups and sub-groups. Interpretability relates to the ability to comprehend, analyze, interpret and question model behavior.

Another reason to take this lifecycle approach is that machine learning is highly data-driven and the information it provides is valuable and trusted. From a data-centric point of view, representativeness, labeling quality, source, distributional alignment and integrity of training and evaluation data are all conditions for the trustworthiness of the model (Li et al., 2023). In the case of computer engineering, this means that it is not possible to repair the trustworthiness of the system merely at the algorithmic level. The issue needs to be addressed in the entire process of data collection to deployment. In addition to the type of classifiers, the reliability of an intrusion detection model relies on the quality of the traffic data, the exposure to adversarial traffic, drift monitoring and update protocols.

The need for this has become more apparent when adversarial examples demonstrated that minor and well-designed perturbations of the image could fool high performing neural networks to make a wrong prediction while simultaneously looking almost identical to human observers (Goodfellow et al., 2015). The discovery of this basic problem of accuracy-centered assessment. A model may be able to be generalized statistically over a test set but be fragile in some operationally meaningful way. This review thus suggests a shift in the focus of system evaluation from performance-centred to trust-centered system engineering in computer engineering, and a move from model to machine learning in the context of trustworthy systems.

2. Review Scope and Analytical Framing

2.1 Review Design

This article takes the structured thematic review design approach and reviews the literature on trustworthy machine learning in computer engineering. It is not a complete systematic review or meta-analysis as there is no intention of a statistical comparison of experimental results or of systematically finding every publication on the topic. Rather, the aim is to create a strong conceptual and technical unification of the notions of robustness, fairness, and interpretability for trustworthy machine learning systems.

A structured thematic review is suitable because machine learning is an interdisciplinary field that includes topics such as adversarial machine learning, explainable artificial intelligence, algorithmic fairness, privacy, security, software engineering and AI governance, all of which must be addressed for trustworthy machine learning to be developed. The assumptions, methods, metrics and terminology used to conduct these research streams vary. Hence, the review does not include each stream as an individual work, rather it aims

at recognizing patterns of tensions, limitations and gaps in research across the literature.

The following question is used to guide the review: What are the trade-offs between the concepts of robustness, fairness and interpretability in the context of trustworthy machine learning in computer engineering, and how do these concepts benefit when viewed together through the machine learning lifecycle?

This question helps cement the central argument in the article, which is that being able to rely on machine learning is a system-level engineering specification, not just a model-level performance specification.

2.2 Thematic Classification Procedure

The 47 references were thematically organized because of the primary contribution and relevance to the review argument. It wasn't based on just keyword or publication venue. Instead, the conceptual emphasis, methodological contribution, evaluation method and relevance to computer engineering practice of each study was reviewed.

Studies of conceptual foundations and governance were part of the first category, which included research on trustworthy AI, responsible AI, accountability, opacity, risk management, and lifecycle governance. These sources contributed to the establishment of trustworthiness as a multi-dimensional and system construct.

The second category, robustness and adversarial resilience, comprised of adversarial examples, adversarial training, certified robustness, corruption robustness, ML security, adversarial learning and weaknesses in robustness evaluation. These studies guided reliability under perturbation and attack analysis, distribution shift analysis, and operational uncertainty analysis.

The third category was fairness and bias mitigation, which comprised of studies on individual fairness, group fairness, equality of opportunity, disparate impact, subgroup fairness, fairness testing, and fairness-enhancing interventions. These sources were used in analysis of discrimination, unequal error rates, proxy bias and fairness trade-offs.

The fourth category was interpreted and explained, which involved work on SHAP, saliency maps, counterfactual explanations, influence functions, prototype-based models, explanation fragility and human-centered explanation. These studies were used to analyze Transparency, Auditability, Explanation Quality, and Model Understanding.

Finally, interactions and trade-offs between robustness, fairness, and interpretability were identified using a final cross-cutting layer. This layer facilitated linking the three pillars together in a single engineering framework.

2.3 Analytical Strategy

The conceptual-technical approach to synthesis was used in reviewing. It explored the conceptual aspects and the significance of trustworthiness for computer engineering. In the technical aspect, it reviewed strategies, metrics, evaluation frameworks, constraints, and implications of deployment pertaining to robustness, fairness, and interpretability.

The analysis was done in five steps. First, all references were assigned to one or more trustworthiness dimensions. Adversarial resilience and corruption tolerance, certified guarantees, and ML security were correlated with robustness studies. Fairness studies were associated with bias detection, metric selection, subgroup analysis and mitigation. Feature attribution, saliency, concept-based explanation, counterfactual reasoning and explanation validation were associated with interpretability studies.

Second, the main contribution of each source was determined. These sources could be grouped into one of the six categories: conceptual frameworks, technical methods, benchmarks, critiques, surveys, and governance models. This allowed for sorting out studies offering tools, studies addressing limitations and broader theoretical or institutional framing.

Thirdly, within each theme, sources were compared to explore common assumptions, tensions and limitations. Some robustness studies are based on natural corruptions, certified guarantees, adversarial perturbations or attack surfaces. There are various types of fairness studies, some emphasizing individual fairness, others emphasizing group fairness, equality of opportunity, or subgroup auditing. The studies on interpretability vary depending on their focus, either on local or global explanation, on human understanding or on fidelity.

Fourthly, themes were read from a computer engineering perspective. The step investigated the application of the literature reviewed to cyber security, embedded systems, computer vision, software analytics, autonomous systems and networked infrastructures. This kept the review on track as a discussion of computer ethics, rather than a general discussion of AI ethics.

Fifth, cross-dimensional trade-offs were synthesized. This step looked at whether enhancing one trustworthiness dimension would affect a different dimension of trustworthiness. For instance, adversarial training can impact certain

subgroups, fairness constraints can have an impact on predictive utility, and methods for explanations can be sensitive to perturbations. This is the integrative contribution of the review.

2.4 Coding Framework

To keep the consistency of the literature it was analyzed using five coding dimensions: Definition, Method, Evaluation, Limitation, and Engineering Implication.

The definition dimension reflected the definition of each source's central term, including robustness, fairness, interpretability, explainability and accountability. The method dimension was used to represent the proposed technical/ conceptual solution, including adversarial training, randomized smoothing, fairness-constrained classification, SHAP, saliency, influence functions, and counterfactual explanations. The dimension used for evaluation was related to the robustness of the accuracy, the success rate, the corruption performance, the fairness metrics, the subgroup analysis, the fidelity of the explanation, the stability of the explanation, and its usefulness for each method/idea.

The limitation dimension uncovered assumptions, weaknesses, or unaddressed issues ranging from gradient masking to conflicting definitions of fairness, to proxy discrimination, to unstable explanations, to limited deployment validity. Lastly, the engineering implications dimension was evaluated with respect to relevance to the practice of computer engineering, such as secure ML pipelines, software testing, embedded AI, auditability, documentation, monitoring and lifecycle governance.

This coding system enabled the review to go beyond summarizing each article at a source level to create a structured synthesis of the field.

3. Trustworthy Machine Learning as an Engineering Paradigm

3.1 System-Level Nature of Trustworthiness

Rather than seeing the concept of trustworthy machine learning as a single concept, machine learning should be viewed as an engineering paradigm, as machine learning models are seldom used in isolation as mathematical objects. They are as part of pipelines, which comprise data collection, data preprocessing, feature extraction, training, validation, inference, monitoring, updating, logging and governance. These pipelines are coupled with the hardware constraints, software architecture, network environments, user behavior, adversarial actors, and institutional policies in computer engineering. The trustworthiness of the system needs to be assessed, and not only the trained model.

3.2 Three Foundational Claims

There are three claims used to direct this review. First, correct information is important, but not enough. While a model that is accurate on a held-out test set can be fragile, biased, or opaque, it is hardly a guarantee that it will be! Secondly, trustworthiness is a multi-dimensional concept. The three aspects of system quality that are measured by robustness, fairness and interpretability are related but different. Thirdly, trustworthiness is dependent on the lifecycle. A reliable model at deployment time can become unreliable over time if the data distributions change, as do the number of users, the type of attacks, or the explanation tools with respect to the system behavior.

3.3 Relevance to Computer Engineering Domains

This lifecycle perspective is particularly important in computer engineering as models are frequently used for functions that need reliability and traceability. The cybersecurity systems should be robust against adaptive attackers. The embedded AI systems must be resilient in noisy and resource-constrained environments. The computer vision system should remain stable under the different lighting conditions, occlusion, compression and sensor variations. Software engineering models must be fair where they are used to assess developers, to automate triage, or to restrict user access. Explanations for human-facing systems must be contestable and provide oversight. These requirements demand trustworthiness, but at the same time make it a practical engineering issue, not just a philosophical or regulatory one. **Figure 2** depicts the layered structure of a trustworthy machine learning engineering.

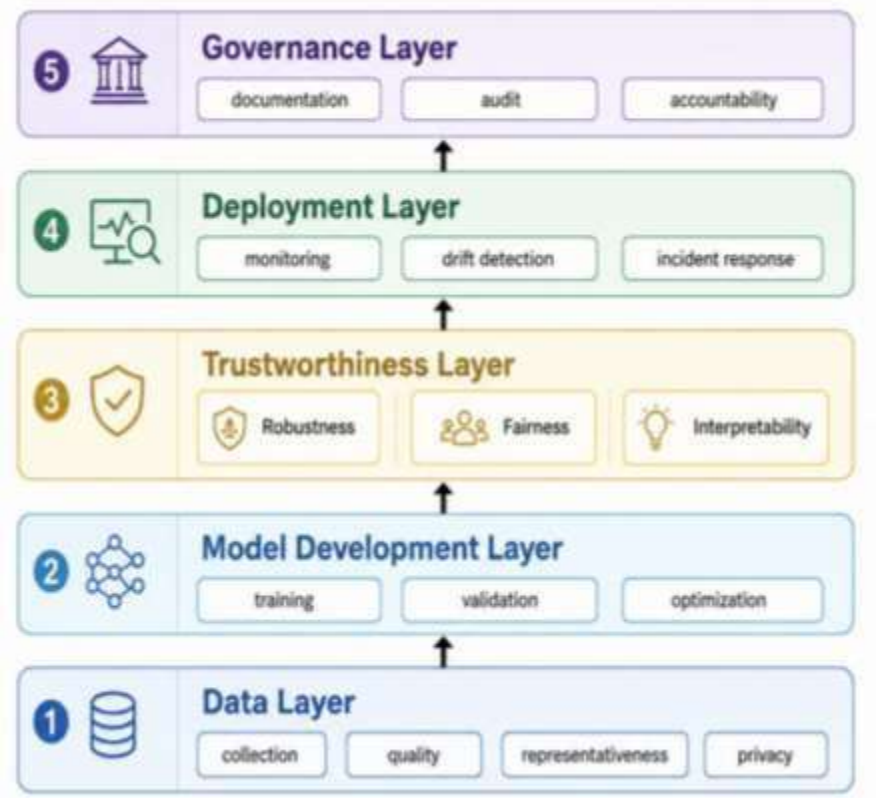

Figure 2. Layered architecture for trustworthy ML engineering.

The core dimensions of trustworthy machine learning in computer engineering are summarized in **Table 1**.

Table 1. Core dimensions of trustworthy machine learning in computer engineering

Dimension	Core question	Engineering concern	Practical relevance
Robustness	Does the model remain reliable under perturbation, attack, and shift?	Adversarial inputs, corrupted data, drift, sensor noise	Prevents unstable or unsafe prediction behavior
Fairness	Does the model avoid unjustified disparities across groups and subgroups?	Sampling bias, proxy discrimination, unequal error rates	Reduces discriminatory outcomes and improves legitimacy
Interpretability	Can humans understand, inspect, and challenge model behavior?	Opaque decisions, weak auditability, poor debugging	Supports validation, accountability, and calibrated trust
Governance	Can system behavior be documented, monitored, and corrected?	Logging, audit trails, escalation, incident response	Makes trustworthiness sustainable after deployment

4. Robustness in Machine Learning for Computer Engineering

4.1 Meaning and Scope of Robustness

Reliability of a machine learning system in the presence of perturbations, corrupted inputs, adversarial examples, distribution shifts, missing data, and operational uncertainty is called robustness. Robustness in computer engineering isn't some sort of abstract statistical quality. It is essential in reliable systems in physical, digital or adversarial environments. Blur, compression, lighting change or degradation of the sensor can be a problem for a vision model in an embedded device. An adaptive attacker can be encountered by a cyber security model. Traffic shifts can be encountered by a network prediction model. Project conditions may vary and lead to a software analytics model failure.

4.2 Adversarial Vulnerability in Computer Vision and Beyond

The use of adversarial machine learning has had a significant impact, particularly in highlighting the vulnerabilities of deep learning systems. Adversarial attacks can fool classifiers, object detectors, and recognition systems, demonstrating that models can be fooled by features that may be predictive but are unstable when the input is

carefully perturbed (Akhtar & Mian, 2018). The results are very important for computer engineering applications, since visual models are utilized in surveillance, robotics, autonomous navigation, human-computer interaction, and quality inspection systems.

A more historical perspective on competitive learning demonstrates that the susceptibility to strategic manipulation has continued to exist throughout various model generations and application contexts (Biggio & Roli, 2018). This persistence means that robustness should not be regarded as a shortcoming of algorithms. It is a structural issue that arises when learning systems are used in situations where input, data and/or users change in relation to the learning system.

4.3 Evaluation Quality and Threat Modeling

Robustness evaluation should, therefore, rely on robust and realistic threat models. Evaluating the robustness of neural networks showed that networks that were believed to be secure could be broken by more powerful optimization-based attacks (Carlini & Wagner, 2017). The engineering takeaway is obvious: A defense is just as useful as an attack that is adaptive, well-specified, and powerful. Weak evaluation leads to false confidence.

A prominent answer is adversarial training, a problem formulation of strong learning as a min-max optimization problem. The model is trained on not only the normal input, but also on worst-case perturbations given a defined threat model (Madry et al., 2018). Such an approach has proven to be key to the development of a strong deep learning framework, since robustness is considered an actual training goal, rather than an evaluation property.

4.4 Robustness to Natural Corruptions

Adversarial resilience is a subset of robustness. Natural corruption of real systems is also present in the form of noise and blur, weather effects, compression artifacts, missing values, etc., and sensor distortions. Indeed, under a realistic form of corruption on common inputs, neural networks can have poor accuracy, despite good accuracy on standard test inputs (Hendrycks & Dietterich, 2019). For computer engineering, this is important because there are many failures that don't happen because of an attacker, but because of normal environmental variation.

4.5 Illusory Defenses and Representation Fragility

The concept of illusory defenses is also highlighted in the robustness literature. Given that the most popular methods are proposed to mask gradients, to break the standard attack strategy, or to turn model evaluation into an artifact, they may seem effective yet are not necessarily better for model stability. From an engineering practice perspective, it is important to take the robustness claim with a grain of salt unless adaptive attack evaluation, transparent assumptions, and reproducible testing is provided.

A more difficult aspect is that non-robust features which may be learned by models but can be exploited for adversarial examples can exist. This perspective implies that adversarial vulnerability isn't just a consequence of optimization but is related to the representations of features used by models to attain high accuracy (Ilyas et al., 2019). Thus, in addition to defensive mechanisms, there is a need to pay attention to what models learn and why these features generalize to make a robust computer engineering system.

4.6 Certified Robustness and High-Consequence Systems

Another route would be to seek certification of robustness. Randomized smoothing offers probabilistic guarantees of stability of predictions given a perturbation radius under certain assumptions (Cohen et al., 2019). While there are threats models, computational costs, and assumptions about perturbation norms that limit the applicability of the methods, they are useful in engineering domains where empirical testing alone may be insufficient.

High consequence systems are particularly susceptible to adversarial vulnerability. Adversarial manipulation can lead to wrongful outputs when decisions are made that can impact human wellbeing, as evidenced in deep medical learning (Finlayson et al., 2019). Medical AI is not the core of this article, but it illustrates that robustness is a serious issue wherever machine learning systems impact important decisions.

4.7 Security, Privacy, and Defensive Learning

The security and privacy perspectives further add to the robustness problem. Machine learning systems reveal their attack surfaces by training data, model parameters, inference interfaces, prediction output, and through update mechanisms (Papernot et al., 2018). Robustness, therefore, needs to be combined with secure pipeline design, access control, data integrity, API monitoring and privacy-conscious deployment in the field of computer engineering.

A prior study on defensive distillation tried to minimize adversarial sensitivity by training adversarial models on softened probability distributions, which led to smoother decision behavior (Papernot et al., 2016a). While some of the

initial defenses were later disputed, this paper is important as it brought the concept of adversarial robustness into the realm of being trainable.

Related work on adversarial limitations in deep learning showed that attacks can transfer across models and operate even in constrained knowledge settings (Papernot et al., 2016b). This is particularly relevant for computer engineering because deployed systems often expose interfaces that allow probing, querying, or reverse engineering by external actors. The main robustness themes and their engineering implications are summarized in **Table 2**.

Table 2. Robustness themes and engineering implications

Robustness theme	Main issue	Engineering implication	Representative response
Adversarial vulnerability	Small perturbations can cause severe prediction errors	Accuracy-only evaluation is inadequate	Adversarial testing and robust training
Common corruption	Noise, blur, compression, and degradation reduce reliability	Validation must reflect operational conditions	Stress testing under realistic perturbations
Weak evaluation	Defenses may appear strong against limited attacks	Robustness claims require strong threat models	Adaptive attack evaluation
Feature fragility	Predictive features may be non-robust	Representation learning affects trustworthiness	Robust feature learning
Certification	Empirical robustness may be insufficient	Formal assurance is needed in high-risk systems	Probabilistic or certified defenses
Security integration	ML systems create attack surfaces	Robustness depends on secure system design	Secure ML pipeline engineering

5. Fairness and Bias in Machine Learning Systems

5.1 Meaning and Sources of Fairness Problems

Fairness concerns whether machine learning systems produce unjustified disparities across individuals, demographic groups, or subgroups. In computer engineering, fairness is relevant wherever models classify users, rank content, authenticate identities, recommend actions, allocate resources, detect anomalies, or support automated decision-making. Bias may arise from unrepresentative data, historical inequalities, measurement errors, proxy variables, subjective labels, feedback loops, or optimization objectives that aggregate performance over subgroup reliability. A broad survey of bias and fairness in machine learning shows that unfairness can enter at every stage of the pipeline, from data collection to post-deployment feedback (Mehrabi et al., 2021).

5.2 Fairness as a Plural and Contextual Concept

Fairness is hard to achieve because not one size fits all. Equality, error, qualification, treatment, and outcome, are all defined differently. In Fairness and machine learning scholarship, Barocas et al. (2023) show that it is not always possible to select a single metric of fairness, and that the selection of a metric is not only technically but also normatively justified. From the computer engineering perspective, this simply won't do, as fairness cannot be achieved through the mere choice of an arbitrary and uninformative metric. The fairness criterion must be the same as the domain, affected population and type of harm.

A basic principle is that of individual fairness, which is sometimes expressed as "like cases should be treated like cases. This concept is captured by similarity metrics for tasks and is crucial because it does not rely solely on the overall group averages (Dwork et al., 2012). It also poses practical questions, however. Similarity is hard to define, particularly when features contain information related to social, behavioral or contextual information.

5.3 Bias Mitigation Across the ML Pipeline

Generally, there are three types of practical fairness interventions: pre-processing methods, in-processing methods, and post-processing methods. Previous research on discrimination-free predictive modeling demonstrated that discrimination can be mitigated by altering data, learning procedures, and/or outputs following training (Kamiran et al., 2012). This classification is helpful for engineering teams as it indicates where the fairness interventions can be incorporated into the current pipelines.

A particular disadvantage of this is disparate impact analysis, which is particularly relevant because models can discriminate even if the sensitive attributes are removed. The patterns of protected groups can be replicated indirectly using proxy variables such as location, language, device type, or other correlated variables (Feldman et al., 2015). This is crucial in large computer engineering systems since there is no guarantee of non-discrimination if there are no explicit protected variables.

A second line of thinking for fairness-constrained classification is as a reductions problem, where fairness constraints are added to cost-sensitive learning workflows (Agarwal et al., 2018). This is essential as it allows us to connect fairness objectives with model development processes that are well integrated with the standard machine learning infrastructure.

5.4 Error-Rate Fairness and Subgroup Protection

Another important criterion that influences is the equality of opportunity. It is about making sure that the same positive results are achieved for qualified people from various groups (Hardt et al., 2016). It is especially significant when it comes to access, selection, detection and recommendation systems, where false negatives can prevent the inclusion of deserving users or important cases.

Subgroups should also be considered when conducting a fairness audit. It is possible that a model is fair for general population groups but discriminates against smaller groups and/or intersectional groups. However, subgroup fairness research can reveal such harms, leading to the issue of fairness gerrymandering (Kearns et al., 2018). In computer engineering platforms with a large user base and a diverse platform, it is thus crucial to have subgroup analysis.

5.5 Fairness Testing and Comparative Evaluation

Software Engineering can be used to operate fairly. Fairness testing, which is an adaptation of the software testing logic, is a systematic variation of the inputs to a system and observation of the differences in outputs, to identify discriminatory system behavior (Galhotra et al., 2017). This is useful because it enables a fair assessment to be performed as part of the quality assurance, verification and continuous integration process.

Fairness interventions do not necessarily work the same way on different datasets, algorithms, and metrics, and there is also an empirical comparison that needs to be made. There is significant variation in the effectiveness of mitigation measures, and no single measure is dominant in all contexts (Friedler et al., 2019). This means that fairness engineering should not simply be about comparing one method of doing something to another – it must be about comparing two or more methods to each other.

5.6 Normative Foundations and Continuing Challenges

Fairness also has a normative dimension. Political philosophy perspectives emphasize that fairness metrics encode assumptions about equality, desert, legitimacy, and institutional purpose (Binns, 2018). Therefore, fairness decisions in computer engineering should be documented and justified, not merely optimized.

Recent survey work confirms that fairness remains an unresolved and evolving field, especially because conflict definitions, real-world data are imperfect, protected attributes may be unavailable, and interventions may create trade-offs with privacy, accuracy, or interpretability (Caton & Haas, 2024). Fairness must therefore be treated as an ongoing lifecycle concern. The main fairness concepts and their engineering relevance are summarized in **Table 3**.

Table 3. Fairness concepts and relevance to engineering

Fairness lens	Main concern	Typical use	Limitation
Individual fairness	Similar individuals should receive similar treatment	Consistency-oriented systems	Requires defensible similarity metrics
Group fairness	Outcomes or errors should be balanced across groups	Population-level audits	May hide within-group disparities
Equality of opportunity	Qualified individuals should have comparable access to beneficial outcomes	Selection and detection systems	Focuses mainly on true positive parity
Subgroup fairness	Smaller or intersectional groups should not be hidden by aggregate metrics	Large-scale platforms	Computationally demanding
Fairness testing	Systems should be probed for discriminatory behavior	Software QA and verification	Depends on test design

Normative fairness	Fairness must reflect institutional and moral context	Policy-sensitive systems	Not reducible to a single formula

6. Interpretability and Explainability

6.1 Meaning and Engineering Relevance of Interpretability

Interpretability concerns whether humans can understand how a machine learning system behaves and why it produces outputs. In computer engineering, interpretability supports debugging, validation, safety analysis, system maintenance, audit, user communication, and accountability. It is especially important when models are complex, high-dimensional, or deployed in contexts where decisions must be inspected or contested. A major survey of black-box explanation methods classifies interpretability techniques into local explanations, global explanations, feature importance, rule extraction, example-based explanations, and visualization methods (Guidotti et al., 2018).\

6.2 Feature Attribution and Prototype-Based Explanation

Feature attribution is one of the most popular interpretability techniques. SHAP is a common framework for additive explanation using Shapley values that can be used to estimate the contribution of features for each prediction or to see the overall contribution of features (Lundberg & Lee, 2017). In engineering practice, these approaches can be used to aid in debugging models, validating features and for auditing reporting. But feature attributions should not be equated with causal explanations.

There are approaches that seek to make models more interpretable, by design. For instance, prototype-based image recognition considers the classifications in terms of similarity to learned prototypes, and a model can make decisions by using the structure of “this looks like that”. In computer vision, this is relevant to prediction and the human recognizable visual evidence.

6.3 Concept-Based and Data-Centric Explanations

Concept-based interpretability is a question of whether model predictions are based on concepts that are understandable to humans, rather than just the attribution of features. To test the extent to which the internal representation corresponds to meaningful semantic concepts, concept activation vectors can be used (Kim et al., 2018). This is helpful for engineers when they need to determine if a model is on task-related concepts or on spurious patterns.

Another way to make model predictions interpretable is by applying influence functions, which can be used to determine the influence of instances in the training set on a particular prediction (Koh & Liang, 2017). It can be helpful for debugging data, understanding if there are any mislabeled instances, and for data provenance analysis. In computer engineering pipelines, influence analysis can be used to relate the behavior of a model to the quality of the training data.

6.4 Counterfactual Explanation and Recourse

The importance of counterfactual explanations is that they link interpretability with recourse. They demonstrate how an input needs to change to get a different result from the model (Mothilal et al., 2020). In user systems where users are involved, counterfactuals can help in contestability provided changes are suggested that are feasible, moral, and meaningful.

6.5 Fragility and Validation of Explanations

It is important to carefully consider how interpretability methods are assessed. Interpretations of neural networks are found to be fragile, meaning that explanation outputs can shift significantly even if the model predictions are the same, when there are small perturbations (Ghorbani et al., 2019). This discovery cautions against assuming stability of explanations when predictions are also stable.

Saliency methods also need to be validated. We carried out sanity checks for saliency maps, which revealed that some explanation methods generate visually plausible maps when model parameters or labels are randomized, thus suggesting that explanation faithfulness is a serious concern (Adebayo et al., 2018). Explanation tools should thus be tested prior to their use in engineering.

6.6 Human-Centered and Purpose-Specific Explanation

The human aspect of explanation is also crucial. The social sciences literature indicates that people tend to prefer explanations that are selective, contrastive, and context-sensitive, rather than complete, technical, explanations (Miller, 2019). This implies explanation design needs to consider the needs of the user, whether the user is an engineer, auditor, domain expert or affected individual.

Other critical research on explaining explanations further proposes that interpretability methods should clarify their function, their audience and their accountability (Mittelstadt et al., 2019). An explanation that assists a developer for debugging

a model might not assist the user to challenge a decision. Hence, quality of explanation needs to be evaluated for the purpose.

6.7 Practical Method Selection

Practical interpretability guidance emphasizes that method selection should depend on model class, data type, stakeholder need, and risk level (Molnar, 2022). For computer engineering, this reinforces the view that interpretability is not a single technique but a design choice within a larger system architecture.

Reviews of deep neural network interpretation methods show that visualization, relevance propagation, decomposition, and sensitivity-based techniques each provide partial views of model behavior (Montavon et al., 2018). No method fully resolves opacity. Instead, interpretable machine learning requires a portfolio of tools, validation criteria, and human-centered evaluation. The main interpretability methods and their engineering uses are summarized in **Table 4**.

Table 4. Interpretability methods and engineering uses

Method class	What it explains	Main engineering use	Key limitation
Feature attribution	Contribution of input variables	Audit and feature validation	Often non-causal
Prototype explanation	Similarity to learned examples	Visual reasoning and user communication	Requires meaningful prototypes
Concept-based explanation	Role of high-level concepts	Representation inspection	Concept definition may be subjective
Influence analysis	Impact of training examples	Data debugging and provenance checks	Approximation may be complex
Counterfactual explanation	How outputs could change	Recourse and contestability	Must be feasible and ethical
Saliency and visualization	Spatial or structural relevance	Computer vision debugging	Can be unstable or misleading

7. Governance and Engineering Implications

7.1 Trustworthiness as a Multi-Objective Problem

The main conclusion of this review is that robustness, fairness, and interpretability are not separable checklist items. They engage in interactions throughout the machine learning lifecycle. The boundaries of decision making can shift because of robustness interventions, which can impact the performance of subgroups. Fairness constraints can limit the predictability or can be coupled with distribution shift. Interpretability tools can be useful to enhance auditability but can also give false confidence when explanations are unstable or unfaithful. A complex model might be able to perform well but not be easily examined for meaning, and a simple model might be quickly understood but not be as accurate when the task is high dimensional.

Such interactions render trustworthy machine learning a multi-objective engineering problem. The aim is to maximize one dimension of a system, rather than to maximize just one dimension, within realistic constraints, while achieving acceptable reliability, equity, transparency, and accountability. This must be documented in terms of explicit trade-offs. For instance, if the accuracy of the clean set is impacted by adversarial training or the errors of subgroups, then the impact of adversarial training should be reported. Any changes in the structure of the explanation or the importance of the features due to changes in the fairness constraints should be assessed. The fidelity and stability of a post hoc explanation method should be tested if it is used.

7.2 Governance and Accountability

Governance is therefore essential. Accountable algorithmic systems require mechanisms for inspection, justification, contestation, and responsibility allocation (Kroll et al., 2017). In computer engineering, this means that trustworthiness must be supported by system architecture. Models should be accompanied by documentation, logging, monitoring, audit trails, version control, incident response procedures, and escalation pathways. Trustworthiness cannot rely only on the trained model. It must be sustained by the operational environment in which the model is deployed. The cross-dimensional trade-off space among robustness, fairness, and interpretability is shown in **Figure 3**.

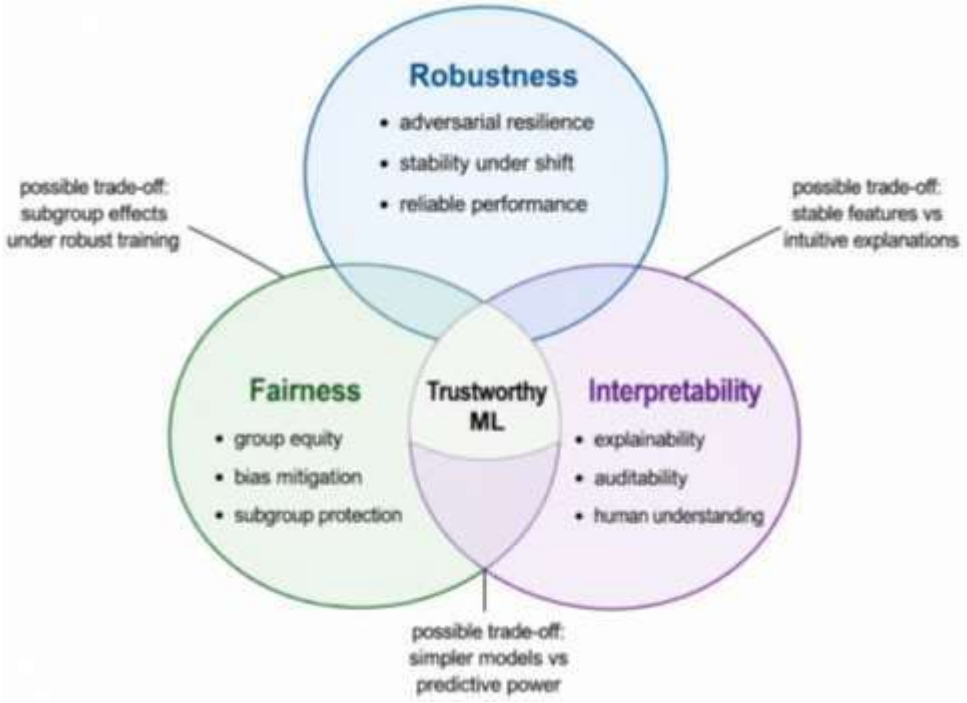

Figure 3. Trade-off space among robustness, fairness, and interpretability.

The lifecycle implications of trustworthy machine learning are summarized in **Table 5**.

Table 5. Lifecycle view of trustworthy machine learning in computer engineering

Lifecycle stage	Robustness question	Fairness question	Interpretability question	Governance need
Data collection	Is the data noisy, shifted, incomplete, or vulnerable to poisoning?	Are relevant groups and subgroups represented?	Are features meaningful and documented?	Provenance and documentation
Model development	Is the model resilient under perturbation and stress?	Do objectives create unequal outcomes?	Is the model interpretable or explainable?	Design review and justification
Validation	Are tests realistic and adversarial strong?	Are subgroup disparities visible?	Are explanations stable and useful?	Auditability and traceability
Deployment	Does performance degrade over time?	Do disparities emerge in operation?	Can stakeholders understand outputs?	Monitoring and incident response
Post-deployment governance	Are failures investigated and corrected?	Are harms contestable and reportable?	Are explanation tools maintained responsibly?	Accountability and escalation

8. Future Research Directions

8.1 Integrated Trustworthiness Benchmarks

Integrated benchmarks are needed for computer engineering that cover both robustness and fairness and interpretability. Research is usually going on in different research communities. The robustness benchmarks may not include subgroup effects. Adversarial vulnerability might not be considered in fairness benchmarks. Reliability in the operation might be overlooked in interpretability studies. New benchmarks should assess the same models on several dimensions of trustworthiness.

8.2 Deployment-Aware Fairness and Robustness

Fairness under distribution shift needs to be better addressed. Assumptions in many fairness evaluations include the assumption that the training, validation, and deployment populations are stable. However, in real systems, users are constantly changing, data distributions are constantly drifting, and feedback loops are constantly forming. Fairness methods should thus be evaluated in the context of both temporal and environmental variability, beyond static data sets.

8.3 Validation of Explanation Tools

There should be more systematic validation of explanations. The explanatory methods should be assessed about their fidelity, stability, comprehensibility and usefulness. Being visually plausible is not enough. Explanation tools are tools that need to be validated, versioned and monitored as part of engineering teams.

8.4 Robust Interpretability

Robust interpretability is an important research frontier. Explanations can be perturbed or destabilized and thus fail exactly when they are needed. Future studies should consider the interplay between the robustness of the explanation and adversarial robustness, as well as the reliability of the model.

8.5 Domain-Specific Trustworthy ML

Evaluating machine learning in a way that is relevant to the domain is essential for trustworthy machine learning. The applications of computer engineering vary regarding latency, hardware, threat model, user base, and failure cost. A trustworthy model for embedded vision might need to be based on other evidence than a trustworthy model for network intrusion detection or software defect prediction.

8.6 Engineering Governance and Trade-Off Management

Governance mechanisms should be "engineered into" engineering processes. Trustworthiness relies on documentation, model cards, dataset records, audit logs, incident reports, human review and processes to challenge harmful outputs. Such practices should be part of the regular machine learning engineering workflow and not compliance work.

Finally, trade-off management should be investigated for the future. It is important to note that robustness, fairness and interpretability can be conflicting but can also be complementary if carefully designed. Research needs to advance approaches to discovery, quantification, communication and negotiation of these trade-offs that are clear to affected stakeholders, organizations and engineers.

9. Conclusion

This review considered the issue of trustworthy machine learning in computer engineering with the included pillars of robustness, fairness and interpretability. The analysis reveals that traditional measures of machine learning systems' performance—predictive accuracy, comparison against benchmarks, and computational efficiency—are insufficient to measure their performance. In real engineering environments, models must be reliable subject to perturbation, corruption, attack and distribution shift. They need to be careful not to have unnecessary differences between people, groups or subgroups. They are also required to deliver interpretable evidence to support the process of debugging, validation, audit, contestability and responsible deployment. One of the main findings is that trustworthiness is not a monolithic characteristic of an trained model. A condition that is created by data governance, design of the data model, validation of the data model, deployment monitoring, and institutional accountability processes. Strong interpretability is inextricably linked with robustness and fairness. The concepts of robustness, fairness and interpretability are closely linked. Even with a strong model, there could be some unfairness. It is possible to have a fair model that is still "black-boxed". Even if an interpretable model is created, it might still be vulnerable. So, it is important to have a holistic assessment and not just optimization of machine learning. This is directly applicable to computer engineering. Adversarial and corruption testing, subgroup fairness audits, explanation quality assessment, drift monitoring, documentation and accountable escalation processes are required for the engineering teams. These practices should be part of the entire machine learning lifecycle, including the creation of datasets, development of models, deployment, and governance. This review's overarching contribution is to consider trustworthy machine learning as an engineering field of responsible system building. It does not just represent a set of ethical ideals; it is also a regulatory demand. One of the technical and socio-technical challenges for the development of machine learning systems that can be used in the real world. To achieve further progress, integrated benchmarks, domain-specific validation, powerful and equitable learning approaches, trustworthy interpretability tools and governance mechanisms to ensure trust following deployment are needed.

References

1. Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., & Kim, B. (2018). Sanity checks for saliency maps. *Advances in Neural Information Processing Systems, 31*.
2. Agarwal, A., Beygelzimer, A., Dudík, M., Langford, J., & Wallach, H. (2018). A reductions approach to fair classification. *Proceedings of the 35th International Conference on Machine Learning*, 80, 60–69.
3. Akhtar, N., & Mian, A. (2018). Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access, 6*, 14410–14430. https://doi.org/10.1109/ACCESS.2018.2807385
4. Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., Chatila, R., & Herrera, F. (2020). Explainable artificial intelligence: Concepts, taxonomies,

opportunities and challenges toward responsible AI. *Information Fusion, 58*, 82–115. https://doi.org/10.1016/j.inffus.2019.12.012

5. Athalye, A., Carlini, N., & Wagner, D. (2018). Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *Proceedings of the 35th International Conference on Machine Learning*, 80, 274–283.
6. Barocas, S., Hardt, M., & Narayanan, A. (2023). *Fairness and machine learning: Limitations and opportunities*. MIT Press.
7. Binns, R. (2018). Fairness in machine learning: Lessons from political philosophy. *Proceedings of the 2018 Conference on Fairness, Accountability and Transparency*, 149–159.
8. Biggio, B., & Roli, F. (2018). Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition, 84*, 317–331. https://doi.org/10.1016/j.patcog.2018.07.023
9. Burrell, J. (2016). How the machine "thinks": Understanding opacity in machine learning algorithms. *Big Data & Society, 3*(1), 1–12. https://doi.org/10.1177/2053951715622512
10. Carlini, N., & Wagner, D. (2017). Towards evaluating the robustness of neural networks. *2017 IEEE Symposium on Security and Privacy*, 39–57. https://doi.org/10.1109/SP.2017.49
11. Caton, S., & Haas, C. (2024). Fairness in machine learning: A survey. *ACM Computing Surveys, 56*(7), 1–38. https://doi.org/10.1145/3616865
12. Chen, C., Li, O., Tao, D., Barnett, A. J., Su, J., & Rudin, C. (2019). This looks like that: Deep learning for interpretable image recognition. *Advances in Neural Information Processing Systems, 32*.
13. Cohen, J., Rosenfeld, E., & Kolter, Z. (2019). Certified adversarial robustness via randomized smoothing. *Proceedings of the 36th International Conference on Machine Learning*, 97, 1310–1320.
14. Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv*. https://arxiv.org/abs/1702.08608
15. Dwork, C., Hardt, M., Pitassi, T., Reingold, O., & Zemel, R. (2012). Fairness through awareness. *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, 214–226. https://doi.org/10.1145/2090236.2090255
16. European Commission, High-Level Expert Group on Artificial Intelligence. (2019). *Ethics guidelines for trustworthy AI*. European Commission.
17. Feldman, M., Friedler, S. A., Moeller, J., Scheidegger, C., & Venkatasubramanian, S. (2015). Certifying and removing disparate impact. *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 259–268. https://doi.org/10.1145/2783258.2783311
18. Finlayson, S. G., Chung, H. W., Kohane, I. S., & Beam, A. L. (2019). Adversarial attacks against medical deep learning systems. *Science, 363*(6433), 1287–1289. https://doi.org/10.1126/science.aaw4399
19. Friedler, S. A., Scheidegger, C., Venkatasubramanian, S., Choudhary, S., Hamilton, E. P., & Roth, D. (2019). A comparative study of fairness-enhancing interventions in machine learning. *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 329–338. https://doi.org/10.1145/3287560.3287589
20. Galhotra, S., Brun, Y., & Meliou, A. (2017). Fairness testing: Testing software for discrimination. *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*, 498–510. https://doi.org/10.1145/3106237.3106277
21. Ghorbani, A., Abid, A., & Zou, J. (2019). Interpretation of neural networks is fragile. *Proceedings of the AAAI Conference on Artificial Intelligence, 33*(1), 3681–3688. https://doi.org/10.1609/aaai.v33i01.33013681
22. Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., & Kagal, L. (2018). Explaining explanations: An overview of interpretability of machine learning. *2018 IEEE 5th International Conference on Data Science and Advanced Analytics*, 80–89. https://doi.org/10.1109/DSAA.2018.00018
23. Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. *International Conference on Learning Representations*.
24. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018). A survey of methods for explaining black box models. *ACM Computing Surveys, 51*(5), 1–42. https://doi.org/10.1145/3236009
25. Hardt, M., Price, E., & Srebro, N. (2016). Equality of opportunity in supervised learning. *Advances in Neural Information Processing Systems, 29*.
26. Hendrycks, D., & Dietterich, T. (2019). Benchmarking neural network robustness to common corruptions and perturbations. *International Conference on Learning Representations*.
27. Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., & Madry, A. (2019). Adversarial examples are not bugs, they are features.

Advances in Neural Information Processing Systems, 32.

28. Kamiran, F., Calders, T., & Pechenizkiy, M. (2012). Techniques for discrimination-free predictive models. *IEEE 12th International Conference on Data Mining*, 869–874. https://doi.org/10.1109/ICDM.2012.45
29. Kearns, M., Neel, S., Roth, A., & Wu, Z. S. (2018). Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. *Proceedings of the 35th International Conference on Machine Learning*, 80, 2564–2572.
30. Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viégas, F., & Sayres, R. (2018). Interpretability beyond feature attribution: Quantitative testing with concept activation vectors. *Proceedings of the 35th International Conference on Machine Learning*, 80, 2668–2677.
31. Koh, P. W., & Liang, P. (2017). Understanding black-box predictions via influence functions. *Proceedings of the 34th International Conference on Machine Learning*, 70, 1885–1894.
32. Kroll, J. A., Huey, J., Barocas, S., Felten, E. W., Reidenberg, J. R., Robinson, D. G., & Yu, H. (2017). Accountable algorithms. *University of Pennsylvania Law Review, 165*(3), 633–705.
33. Li, B., Qi, P., Liu, B., Di, S., Liu, J., Pei, J., Yi, J., & Zhou, B. (2023). Trustworthy AI: From principles to practices. *ACM Computing Surveys, 55*(9), 1–46. https://doi.org/10.1145/3555803
34. Li, Y., Wang, Y., & Singh, L. (2023). Towards trustworthy and aligned machine learning: A data-centric survey. *arXiv*. https://arxiv.org/abs/2307.16851
35. Lipton, Z. C. (2018). The mythos of model interpretability. *Communications of the ACM, 61*(10), 36–43. https://doi.org/10.1145/3236386.3241340
36. Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems, 30*.
37. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. *International Conference on Learning Representations*.
38. Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2021). A survey on bias and fairness in machine learning. *ACM Computing Surveys, 54*(6), 1–35. https://doi.org/10.1145/3457607
39. Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence, 267*, 1–38. https://doi.org/10.1016/j.artint.2018.07.007
40. Mittelstadt, B., Russell, C., & Wachter, S. (2019). Explaining explanations in AI. *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 279–288. https://doi.org/10.1145/3287560.3287574
41. Molnar, C. (2022). Interpretable machine learning: A guide for making black box models explainable (2nd ed.).
42. Montavon, G., Samek, W., & Müller, K. R. (2018). Methods for interpreting and understanding deep neural networks. *Digital Signal Processing, 73*, 1–15. https://doi.org/10.1016/j.dsp.2017.10.011
43. Mothilal, R. K., Sharma, A., & Tan, C. (2020). Explaining machine learning classifiers through diverse counterfactual explanations. *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 607–617. https://doi.org/10.1145/3351095.3372850
44. National Institute of Standards and Technology. (2023). *Artificial intelligence risk management framework: AI RMF 1.0* (NIST AI 100-1). U.S. Department of Commerce. https://doi.org/10.6028/NIST.AI.100-1
45. Papernot, N., McDaniel, P., Sinha, A., & Wellman, M. (2018). SoK: Security and privacy in machine learning. *2018 IEEE European Symposium on Security and Privacy*, 399–414. https://doi.org/10.1109/EuroSP.2018.00035
46. Papernot, N., McDaniel, P., Wu, X., Jha, S., & Swami, A. (2016). Distillation as a defense to adversarial perturbations against deep neural networks. *2016 IEEE Symposium on Security and Privacy*, 582–597. https://doi.org/10.1109/SP.2016.41
47. Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., & Swami, A. (2016). The limitations of deep learning in adversarial settings. *2016 IEEE European Symposium on Security and Privacy*, 372–387. https://doi.org/10.1109/EuroSP.2016.36