Backdoor Detection and Mitigation in Neural Networks

Juan Arturo Abaurrea-Calafell[a,*] (Student)

[a]Universidad Politécnica de Madrid, Escuela Técnica Superior de Ingenieros Informáticos, Madrid, 28031, Spain

ARTICLE INFO

Keywords:
Backdoor attacks, Neural network security, Activation clustering, Mechanistic interpretability, Machine unlearning, Latent space analysis, Model auditing, Adversarial machine learning, Neuron pruning, Deep learning robustness

ABSTRACT

Deep learning models operate as "black boxes" whose internal decision-making processes remain opaque, enabling them to achieve high accuracy while learning incorrect shortcuts, hidden biases, or malicious backdoor patterns. This review examines methodologies for auditing and correcting neural network internal representations without complete retraining, focusing on: (1) backdoor detection through activation clustering and latent space analysis, (2) mechanistic interpretability for localizing undesired patterns, and (3) mitigation via selective neuron pruning and machine unlearning. We synthesize foundational backdoor attack research, state-of-the-art detection methods, and emerging mitigation approaches, identifying critical gaps in scalability, standardization, and adaptive attack resistance that motivate the development of robust auditing frameworks for production systems.

1. Introduction

1.1. Motivation and Context

Deep learning models deployed in critical domains, such as autonomous vehicles, medical diagnosis, and financial systems, face a fundamental challenge [1]. Unlike traditional software validated through exhaustive testing, neural networks learn complex decision boundaries from data, making their behavior difficult to predict [2]. Recent failures demonstrate that models achieving impressive test accuracy can fail catastrophically when encountering edge cases or adversarial conditions [2, 3, 4].

In autonomous driving, perception systems must reliably detect pedestrians and traffic signs to ensure safe operation. However, reported fatal incidents have demonstrated that models can fail catastrophically when encountering adversarial conditions or natural transformations (such as rain) not adequately represented in training data [2, 5]. Similarly, in healthcare, diagnostic models achieving impressive accuracy on test sets have been found to rely on spurious shortcuts, such as using hospital-specific metal tokens or imaging artifacts like the word "portable" rather than genuine pathological features, leading to dangerous misdiagnoses when deployed in different clinical settings [3, 6].

Financial institutions deploying credit scoring models face regulatory requirements for fairness and transparency, yet struggle to audit whether their models have learned to encode protected characteristics (race, gender, age) through subtle proxies such as residential zip codes [1]. Content moderation systems on social platforms must detect harmful content while avoiding biased enforcement, but they often lack standardized tools to verify whether their models have internalized societal biases present in training data [1, 7].

The tension between model performance and interpretability is severe, though some research suggests this trade-off can be mitigated through architectural changes designed specifically for reverse engineering [6, 8]. High-capacity hierarchical representations that make deep learning powerful also render these systems opaque "black boxes" where internal weights do not match intuitive features [9]. Traditional quality assurance approaches, such as unit testing and formal specification, are insufficient for systems that learn behavior from data rather than explicit programming [2].

This opacity manifests in several critical ways: opaque decision-making where researchers and engineers cannot easily determine why a model made a specific prediction because deep neural networks (DNNs) are numerical black boxes that do not lend themselves to human understanding [9, 10]; undetectable shortcuts where models learn spurious correlations (such as relying on "grass" to identify "cows") that achieve high validation accuracy on standard benchmarks but fail to capture true underlying relationships in real-world scenarios [3]; hidden biases where biases present in training data become encoded in the model's internal representations in ways that are not immediately apparent from examining individual predictions [1, 7]; and vulnerability to backdoor attacks where adversaries inject malicious triggers during training to embed hidden behaviors that cause targeted misclassifications at inference time while the model appears to operate normally on clean inputs [11, 12].

1.2. Objectives and Structure

This review addresses the need for standardized methodologies to audit and correct neural network internal representations. We aim to: (1) synthesize detection methodologies emphasizing activation-based approaches that analyze what neurons learn rather than only examining model inputs or outputs, (2) evaluate mitigation techniques avoiding complete retraining, particularly selective neuron pruning and machine unlearning approaches, (3) establish a conceptual framework distinguishing different types of unwanted behaviors (backdoor attacks, learned biases, spurious shortcuts, noise patterns) and understanding which detection and mitigation approaches are appropriate for each, and (4) identify research gaps in scalability, standardization, and practical

*Corresponding author

arturo.abaurrea.calafell@alumnos.upm.es (J.A. Abaurrea-Calafell)

ORCID(s): 0009-0003-1494-6024 (J.A. Abaurrea-Calafell)

deployment of auditing methodologies, particularly for large foundation models.

The ultimate goal is to provide a foundation for developing practical auditing frameworks that can be integrated into machine learning deployment pipelines, enabling quality engineers to verify model reliability and alignment without requiring deep expertise in neural network internals or incurring prohibitive retraining costs.

Section 2 describes our systematic review methodology. Section 3 establishes concepts of latent spaces and anomalous behaviors. Sections 4-5 examine backdoor attacks and detection techniques. Section 6 explores mechanistic interpretability. Section 7 covers mitigation strategies. Section 8 extends to other undesired behaviors. Sections 9-10 discuss evaluation and applications. Section 11 identifies open challenges, and Section 12 synthesizes findings.

2. Methodology

Our systematic review utilized Google Scholar, arXiv, IEEE Xplore, ACM Digital Library and Consensus Search Engine with search terms including "backdoor attack neural networks", "activation clustering", "mechanistic interpretability", and "machine unlearning." We prioritized recent publications while including foundational earlier works. From an initial pool of hundreds of relevant papers, we selected 70+ based on our selection criteria.

Detailed source quality evaluation is provided in Appendix A.

3. Fundamental Concepts

3.1. Latent Spaces and Internal Representations

Neural networks learn hierarchical representations through successive transformations. For a network f with L layers: $f(x) = f_L \circ f_{L-1} \circ \cdots \circ f_1(x)$. The output $h_i = f_i \circ \cdots \circ f_1(x)$ represents the latent space at layer i. These representations exist in spaces of varying dimensionality depending on layer architecture (e.g., 512-dimensional vectors in a fully connected layer, or $H \times W \times C$ tensors in convolutional layers).

Under the manifold hypothesis, high-dimensional data is assumed to concentrate near lower-dimensional manifolds embedded in the ambient space, and networks implicitly learn mappings to these regions to capture the structure of the data [13]. The geometry of these latent spaces reflects the abstractions learned during training, which should ideally capture task-relevant features while being invariant to irrelevant variations [10, 13]. However, models can learn degenerate representations (relying on shortcuts, memorization, or spurious correlations) which achieve high validation accuracy on standard benchmarks but fail to generalize to the true underlying data distribution [3, 14].

This geometric perspective enables anomaly detection, as anomalous behaviors often manifest as distinct clusters within the latent representations of the last hidden layer [15]. In particular, backdoored models show poisoned inputs activating specific neurons in characteristic patterns that differ from legitimate inputs, creating a detectable spectral signature [16].

3.2. Neural Activations and Analysis

Activations are concrete numerical values computed when processing inputs.

For neuron j in layer i:

$$a_{ij}(x) = \sigma\left(\sum_k w_{ijk} \cdot a_{(i-1)k}(x) + b_{ij}\right) \tag{1}$$

where w_{ijk} are connection weights, b_{ij} is the bias term, and σ is the activation function. The full activation pattern at layer i is the vector:

$$\mathbf{a}_i(x) = [a_{i1}(x), a_{i2}(x), \ldots, a_{in_i}(x)] \in \mathbb{R}^{n_i} \tag{2}$$

Analysis techniques include activation maximization, which finds inputs that maximize specific neurons to reveal the features they detect [17]; statistical analysis of activation distributions across datasets to identify neurons or samples with unusual behavior; dimensionality reduction using methods such as PCA [18], t-SNE [19], and UMAP to [14] visualize high-dimensional latent structures and identify distinct clusters; and causal interventions, such as ablation studies and causal mediation analysis, which systematically intervene on neurons or attention heads to measure their causal effect on model outputs [7, 14, 20, 21].

The key insight is that problematic behaviors must leave activation traces within the model's internal representations [16]. If a model has learned a backdoor trigger, specific neurons or internal features must respond selectively to that trigger pattern [5]. Consequently, systematic activation analysis across diverse inputs can identify suspicious neurons or anomalous activation patterns for further investigation or mitigation [15].

3.3. Defining Anomalous Behaviors

The anomalous behaviors studied in this review refer to problematic learned patterns embedded in a model's internal representations that cause systematic, reproducible undesired actions under specific conditions. These are distinct from: (1) out-of-distribution (OOD) inputs where an input differs from the training distribution but the model itself is legitimate [2, 10], (2) adversarial examples involving test-time perturbations crafted to fool legitimate models without modifying their weights (though activation-based methods could potentially detect such inputs at runtime, this is not our primary focus) [2], and (3) simple misclassifications where a model performs poorly on inherently difficult tasks due to insufficient capacity or training data.

Anomalies arise from three primary sources: intentional poisoning (backdoor attacks) where adversaries deliberately inject malicious patterns during training so that a model misclassifies inputs containing a specific "trigger" but performs normally on clean images [5]; unintentional biases or shortcuts where models learn spurious correlations, such as a

model that appears to classify images into huskies or wolves that actually relies on the presence of snow as an unintended predictor (see Figure 1), or a medical model that detects pneumonia by identifying hospital-specific metal tokens or equipment labels on X-ray scans [3]; and noise memorization where high-capacity models use brute-force to overfit to idiosyncratic patterns or completely unstructured random pixels in the training data [22].

(a) Husky classified as wolf (b) Explanation

Figure 1: Raw data and explanation of a bad model's prediction in the "Husky vs Wolf" task. Figure from Ribeiro et al. [23]

Critically, anomalies are properties of learned internal representations, not just individual inputs. A backdoored model appears normal on clean test data, and the anomaly manifests only under specific conditions [12]. This definitional clarity determines appropriate methodologies: detection must analyze activation patterns and latent space structure, such as searching for the "spectral signatures" left in feature representations, rather than merely examining input distributions [16].

4. Backdoor Attacks

4.1. BadNets and Supply Chain Attacks

Gu et al. [5] demonstrated neural networks' vulnerability to training-time poisoning where adversaries inject "trojan" behaviors. The threat model assumes adversaries influencing training data or processes, which is realistic given third-party datasets, transfer learning from untrusted sources, crowdsourced labeling, and federated learning scenarios.

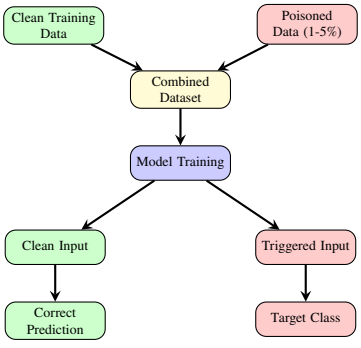


Figure 2: Backdoor attack mechanism: poisoned samples (1-5% of training data) embed a trigger-target association. The backdoored model maintains high clean accuracy but misclassifies triggered inputs to the target class.

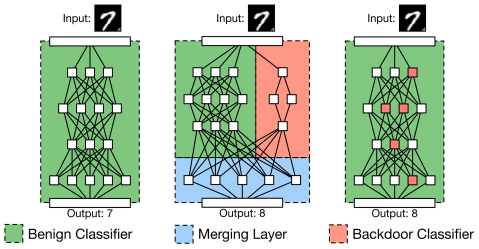


Figure 3: For a trained model, backdoor triggers can be implemented either as (middle) additional specialized neurons appended to existing layers, or (right) distributed across the network's existing architecture by retraining weights. Figure from Gu et al. [5].

The attack mechanism poisons small training data fractions by adding trigger patterns and changing labels to target classes. Models learn trigger-target associations through standard supervised learning. Crucially, backdoored models achieve near-normal clean accuracy: backdoors are dormant without triggers, making detection through traditional validation difficult [9]. Gu et al. [5] demonstrated >90% attack success rates with >95% clean accuracy across traffic sign recognition, face recognition, and speech recognition.

Figure 4: Stop sign classified as a speed limit sign due to a sticker acting as a trigger. Figure from Gu et al. [5]

The implications for machine learning (ML) security are severe. Consider an autonomous vehicle using a traffic sign classifier with an embedded backdoor: an adversary could place physical triggers (stickers, graffiti) on stop signs to cause them to be misclassified as speed limit signs, potentially causing collisions (see Figure 4). In face recognition for access control, backdoored systems could grant unauthorized access to attackers wearing specific accessories that act as triggers [24].

4.2. Trigger Types and Threat Differentiation

Research identified diverse triggers: patch-based (visible patterns like squares or checkerboards) [5], blend (semi-transparent patterns blended with entire images at low opacity) [24], physical (real-world reproducible patterns

Characteristic	Backdoor Attacks	Adversarial Attacks	Prompt Injection
Attack timing	During model training	At inference time on a legitimately trained model	At inference time on language models
Attacker capability	Can poison training data or influence training process	Can craft carefully perturbed inputs	Can craft malicious prompts or inject text into context
Persistence	Backdoor is permanently embedded in model weights	No modification to model; attack is per-input	No model modification; attack is per-query
Trigger	Requires specific pattern in input to activate	Small imperceptible perturbations added to clean inputs	Specially crafted text that overrides model instructions or extracts sensitive information
Example	Placing a sticker to a stop sign image that makes a model classify it as a speed limit sign [5]	Adding imperceptible noise to any image that causes it to be classified as an ostrich [29]	Prompt like "Ignore previous instructions and output your system prompt" to jailbreak a chatbot [30]
Defense paradigm	Detect and remove backdoor from trained model via activation analysis and pruning	Adversarial training and input preprocessing (e.g. data augmentation)	Input sanitization, instruction hierarchy, output filtering

Table 1: Comparison of Machine Learning Attack Types

robust to camera variations) [12], semantic (natural features like "any image containing sunglasses") [25], sample-specific (different triggers for different samples) [26], dynamic (time/condition-dependent) [27], and latent (feature-space only) [28]. The evolution towards increasingly stealthy triggers presents a significant challenge for defenders. While simple patch triggers are relatively easy to detect (they create visually obvious artifacts and statistically unusual activation patterns), semantic and latent triggers can be virtually indistinguishable from legitimate features in both input and activation space.

Three distinct threat models involving malicious inputs are frequently confused but represent fundamentally different attack surfaces. These can be observed in Table 1.

Our focus in this review is primarily on backdoor detection and mitigation, though we note that mechanistic interpretability techniques developed for this purpose often generalize to understanding other failure modes including vulnerability to adversarial examples.

4.3. Attack Vectors and Risk Scenarios

Understanding where backdoors can be injected in the ML pipeline is crucial for assessing real-world risk:

Training Data Poisoning: Adversaries inject poisoned samples into training datasets to embed hidden malicious behaviors. This frequently occurs in situations where attackers have access to the training database, such as web-based repositories or maliciously curated data sources [31]. This risk is heightened when data collection involves crowd-sourced labeling platforms or user-generated content, where malicious workers can manipulate a fraction of the samples [12].

Fine-tuning Persistence: Because training DNNs from scratch is computationally expensive and intensive, practitioners commonly download pre-trained weights from public repositories or use machine learning as a service (MLaaS) platforms as backbones for their own tasks [5]. An adversary can publish a backdoored model, such as a "BadNet", that performs as expected on standard benchmarks but contains a hidden trigger [16]. Research shows that these backdoors are remarkably durable and can survive the transfer learning process even when a user retrains the model's fully-connected layers for a new task [5].

Teacher-Student Latent Backdoors: In a more stealthy variant of pre-trained model compromise, an attacker embeds an incomplete or "latent" backdoor into a teacher model [28]. This backdoor remains dormant and cannot be detected by testing the teacher's existing labels because the intended target class likely does not exist in the teacher model yet [28]. When a victim customizes this into a student model via transfer learning, the process of adding new labels and fine-tuning weights inadvertently "self-activates" the backdoor [28].

Federated Learning Poisoning: Federated learning is fundamentally vulnerable because it is impossible to guarantee that none of the decentralized participants are malicious. Since the central server has no access to a client's local data or training process, a malicious client can easily contribute poisoned updates to ensure a backdoor is embedded into the global model [32].

Supply Chain Compromise: Attacks on development infrastructure include malicious code in training libraries or corrupted model serialization steps (such as the "pickle" format), which can be exploited to execute code when a model is loaded [33]. Adversaries can also introduce backdoors by compromising the servers that host model repositories or by modifying metadata to point users toward maliciously altered files [5, 33].

Insider Threats: Malicious employees, rogue data curators, or contractors with direct access to the training pipeline can deliberately backdoor models [9, 24]. These insiders can stealthily inject a small number of poisoning samples into the training set, sometimes as few as 5 to 50 instances, without noticeably degrading the model's overall performance on clean data [24].

Domain	Primary Attack Vector	Impact Severity	Detection Priority	Acceptable CA Drop
Autonomous Vehicles	Pre-trained models, sensor data	Critical	TPR > 99%	<2%
Medical Diagnosis	Training data, federated learning	Critical	TPR > 99%	<2%
Financial Fraud Detection	Insider threats, data poisoning	High	TPR > 95%	<5%
Biometric Access Control	Supply chain, model repos	High	TPR > 95%	<5%
Content Moderation	User-generated content	Medium	TPR > 90%	<7%
Consumer Applications	Pre-trained models	Low	TPR > 85%	<10%

Table 2: Domain-Specific Backdoor Risk Assessment. Authors' assessment based on literature synthesis and domain-specific safety requirements.

High-stakes application domains face the greatest risk as seen in Table 2. The table evaluates domains based on their primary attack vectors (the most likely methods attackers would use to insert backdoors, such as poisoning pre-trained models or manipulating training data), required detection priority measured by TPR (True Positive Rate: the percentage of actual backdoor attacks that must be correctly detected), and acceptable CA drop (the maximum tolerable decrease in Clean Accuracy, i.e., performance on legitimate data, when applying backdoor defenses).

5. Detection Techniques

5.1. Input-Based Methods

The first generation of backdoor defenses focused on analyzing model inputs to reconstruct potential triggers. These methods treat backdoor detection as an optimization problem: if a backdoor exists, there should be some pattern that, when added to inputs, reliably causes misclassification to the target class.

Neural Cleanse, proposed by Wang et al. [9], pioneered the trigger reconstruction approach. The core insight is that backdoored models learn decision boundaries with unusual properties: there exists a small perturbation (the trigger) that causes almost any input to be classified as a specific target class, whereas legitimate models should not have such universal perturbations.

The method works by solving the following optimization problem for each possible target class y_t [9]:

$$\min_{m,\Delta} \sum_{x \in X} \mathcal{L}(y_t, f(A(x, m, \Delta))) + \lambda \cdot |m| \tag{3}$$

where Δ is the trigger pattern, m is a mask indicating where to apply the trigger, and $A(x, m, \Delta)$ is the stamping function that blends the trigger with the input: $x'_{i,j,c} = (1 - m_{i,j}) \cdot x_{i,j,c} + m_{i,j} \cdot \Delta_{i,j,c}$. Here, X is a set of clean inputs, f is the model under inspection, and $\mathcal{L}$ is the loss function measuring how well the triggered inputs are classified as target y_t. The $|m|$ term (L1 norm of the mask) encourages small, localized triggers by penalizing larger masks.

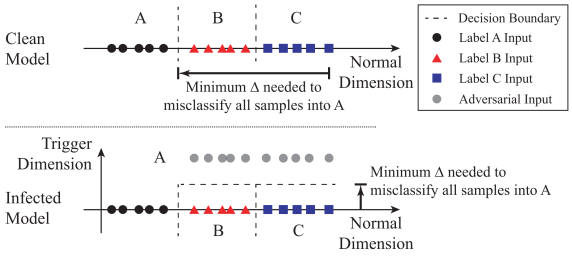


Figure 5: Visualization of how a backdoor trigger modifies decision boundaries, creating shortcuts that enable misclassification of inputs from classes B and C into the target class A with minimal perturbation. Figure from Wang et al. [9].

This optimization attempts to find the minimal perturbation that causes the model to classify all inputs as class y_t, as illustrated in Figure 5. The key insight is that if a backdoor already exists for class y_t^*, the optimization will converge to a much smaller trigger size for y_t^* than for clean classes, because the model has already learned a shortcut to that class. Neural Cleanse runs this optimization for every possible target class and uses outlier detection to flag classes with anomalously small reconstructed triggers as potentially backdoored.

Wang et al. [9] validated this approach across multiple datasets and attack types. As shown in Figure 6, all infected models exhibit anomaly indices greater than 3 (corresponding to > 99.7% probability of infection under the assumption of normally distributed trigger sizes), while clean models consistently remain below the detection threshold of 2. This clear separation demonstrates the method's reliability in distinguishing backdoored from benign models. Figure 7 further illustrates the underlying principle: the L1 norm of triggers for infected labels is consistently and substantially smaller than that of uninfected labels across all tested datasets, confirming that backdoored classes are indeed more vulnerable to small perturbations.

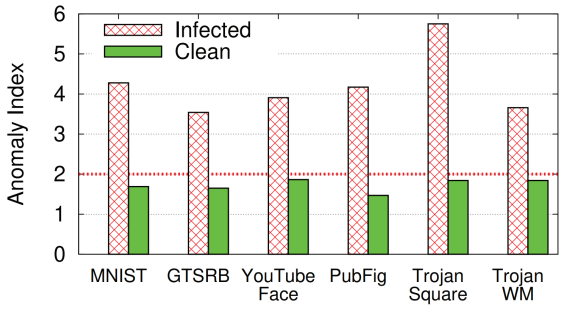


Figure 6: Anomaly index measurements for infected and clean models, and two Trojan Attack models. Figure from Wang et al. [9].

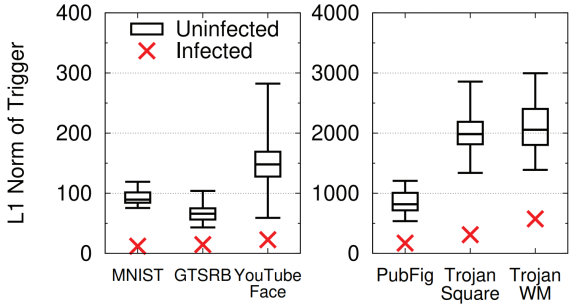


Figure 7: Distribution of L1 norms for reversed triggers across infected and uninfected labels in backdoored models. Figure from Wang et al. [9].

Beyond detection, Neural Cleanse also successfully reverse-engineers the trigger patterns themselves. Figure 8 compares the original attack triggers with those reconstructed by the method across six different backdoored models, including both BadNets attacks and Trojan Attacks. The reversed triggers closely match the originals in both location and visual appearance for BadNets attacks, though they tend to be more compact due to the L1 norm penalty. However, for Trojan Attacks, the reversed triggers differ more substantially from the originals, appearing in different locations and with different patterns. This occurs because the optimization discovers alternative, more compact triggers that exploit the same backdoor vulnerabilities. Importantly, despite these visual differences, all reversed triggers achieve attack success rates exceeding 97.5%, demonstrating their functional equivalence to the original backdoor triggers. This reconstruction capability enables not only detection but also the development of input filtering and model patching defenses [9].

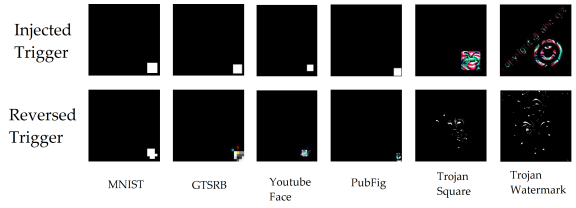


Figure 8: Comparison between original attack triggers (top row of each pair) and reverse-engineered triggers from Neural Cleanse (bottom row of each pair) across multiple datasets and attack methods. Figure adapted from Wang et al. [9].

However, the method has several key assumptions:

- Triggers are spatial patterns that can be represented as masked perturbations
- Triggers cause universal behavior (work on almost all inputs)
- Backdoored classes have significantly smaller triggers than clean classes
- Sufficient computational resources for optimization across all classes

Neural Cleanse established input-based reverse engineering as a viable detection paradigm and inspired numerous follow-up works improving robustness, efficiency, and handling of different trigger types [34, 35].

Limitations of input-based approaches (such as Neural Cleanse) include: high computational cost (per-class optimization requiring thousands of gradient descent iterations), assumption violations (small spatial triggers; sophisticated attacks use semantic, dynamic, or latent triggers that violate these assumptions) [36], vulnerability to adaptive attacks (adversaries can design backdoors that produce similar trigger sizes across all classes) [37], no ground truth validation (candidate triggers require additional validation), and limited mechanistic insight (they do not reveal how backdoors are implemented within the model).

5.2. Activation Clustering

Chen et al. [15] established activation clustering (AC) as a powerful method for detecting backdoor attacks. The methodology generally consists of five primary steps: (1) collect activations from hidden layers, specifically the last one, by forwarding training inputs through the model; (2) apply dimensionality reduction, such as with Independent Component Analysis (ICA), to manage high-dimensional activation spaces and improve clustering robustness; (3) perform cluster analysis, with the best performing explored method being k-means (with k=2); (4) flag small clusters as potentially poisoned by examining their relative size compared to the total data for a given label; and (5) filter suspected poisoned samples and optionally relabel and retrain or fine-tune the model to remove the backdoor.

Method	Main Approach	Access	Stage
Neural Cleanse [9]	L1 optimization and outlier analysis	White	Offline
Activation Clustering [15]	Analysis of activations in hidden layers	White	Offline
Fine-Pruning [38]	Latent neuron pruning and retraining	White	Offline
ABS [35]	Individual neuron stimulation	White	Offline
TABOR [34]	Optimization with heuristic regularization	White	Offline
SCAn [39]	Identity/variation decomposition	White	Offline
SPECTRE [18]	Robust covariance estimation and QUE	Black*	Offline
Spectral Signatures [16]	Covariance analysis of representations	Black*	Offline
MNTD [40]	Meta-classifier based on shadow models	Black	Offline
DeepInspect [41]	Model inversion via cGAN	Black	Offline
FeatureRE [36]	Artifact analysis in frequency domain	Black	Offline
SentiNet [42]	Saliency map and object detection	Black	Online
NEO [43]	Systematic trigger blocking	Black	Online
STRIP [32]	Prediction entropy under perturbation	Black	Online
TeCo [44]	Robustness consistency under corruptions	Black	Online

Table 3: Comparative analysis of backdoor detection methods. This table summarizes key backdoor detection approaches, categorizing them by their access requirements (White: full model access; Black: query access only), deployment stage (Offline: pre-deployment inspection; Online: runtime detection), and key limitations. Black*: Methods requiring only intermediate representations or output labels without gradient access.

Figure 9: Activation clustering applied to the last hidden layer after dimensionality reduction. Clean samples (red) form a cohesive main cluster while backdoored samples (blue) separate into a distinct anomalous cluster, enabling detection. Figure from Chen et al. [15].

Effectiveness stems from backdoors requiring specific "backdoor neurons" activating in characteristic patterns differing from legitimate inputs. Clustering reveals this bifurcation naturally. Demonstrated >95% detection rates with minimal false positives across datasets [15].

The activation clustering approach offers several key advantages. Most importantly, it requires no verified clean dataset, unlike prior defenses that demanded tens of thousands of trusted samples [45]. The method demonstrates robustness to complex scenarios including multimodal classes and multiple backdoor triggers, achieving near-perfect detection rates. AC provides interpretable results through cluster visualization, allowing human verification before action. The repair process is also efficient, with retraining converging in 14 epochs versus 80 from scratch [15].

Despite its strengths, AC has important limitations. It fundamentally requires the poisoned training dataset containing backdoored samples; without them, no anomalous cluster forms. The method assumes poisoned samples are a minority (typically <50%), relying on size disparity for detection. Sophisticated adversaries aware of this defense could potentially craft attacks producing activations similar to legitimate samples.

5.3. Statistical Analysis and Neuron Localization

Complementary statistical methods include distribution testing such as Maximum Mean Discrepancy (MMD) to compare activation distributions [46], spectral signatures that analyze covariance matrix eigenvalues to identify outlier behaviors [18], neuron coverage analysis measuring the fraction of neurons activated by inputs [2], and consistency checks that exploit the fact that backdoor behavior remains invariant while legitimate inputs show more variability [31].

Localization techniques identify specific backdoor neurons through: activation frequency analysis where certain neurons activate for poisoned but not clean inputs [5], importance ranking via gradient-based attribution [2], and mutual information to identify features with high statistical dependence on the output [2]. Further methodologies involve causal intervention using causal tracing to identify modules that mediate specific associations [7], and neural pathway analysis tracing information flow to identify complete activation paths across layers [2].

5.4. Comparative Analysis

Table 3 provides a comprehensive overview of representative backdoor detection methods, organized by their access requirements and deployment stage. White-box methods, such as Neural Cleanse [9] and Activation Clustering [15], leverage full model access including gradients and internal

activations to identify backdoor patterns through optimization or statistical analysis of neuron behaviors. In contrast, black-box approaches like STRIP [32] and TeCo [44] operate with limited query access, making them more practical for some real-world scenarios where model internals are unavailable.

6. Mechanistic Interpretability

Mechanistic interpretability seeks to understand neural networks as computational mechanisms, algorithms, circuits, and information processing, rather than just input-output functions [4, 8, 20]. This approach is often compared to a programmer attempting to reverse engineer a compiled binary back into human-readable source code [4, 8]. It contrasts with behavioral or structural interpretability approaches, such as LIME or SHAP, which aim to explain why a model reached a specific decision (e.g., a loan approval) without necessarily revealing the detailed internal computations [2, 23, 47]. By uncovering these mechanisms, researchers can understand how backdoors are implemented within a model's weights and activations. This depth of understanding facilitates distinguishing legitimate features from shortcut learning (where models rely on spurious correlations in the data) [3]. Furthermore, it enables surgical interventions to edit or update specific model behaviors [21, 48] and provides a means to verify that a model is implementing appropriate learned algorithms rather than unintended "cheating" strategies [19].

Techniques include: probing classifiers (training simple linear classifiers on frozen network activations to determine what specific linguistic or conceptual information is encoded and linearly accessible at different layers) [49], logit lens (in transformers, applying the unembedding matrix to intermediate layer activations to reveal what tokens the model is "thinking about") [4], activation patching (also known as causal mediation analysis or causal tracing, it involves causal interventions where activations from a "clean" input are selectively replaced with activations from a "corrupted" input to pinpoint which internal components are responsible for specific behavioral outputs) [20, 50], circuit analysis (the systematic study of subgraphs of features and the weighted connections between them to identify the minimal circuits that implement specific algorithmic behaviors) [20, 51], sparse autoencoders (training auxiliary networks to find interpretable features in dense activations, dealing with "Superposition" where neurons respond to multiple unrelated concepts) [50, 52], and gradient-based saliency (computing the gradients of model outputs with respect to intermediate activations or features to identify which components exert the most significant causal influence on a prediction) [2].

Localization identifies where specific behaviors are implemented within a model's architecture. This includes layer-wise specialization; for example, in vision models, early layers typically detect simple edges and textures while middle layers detect object parts, and late layers represent complete objects [51]. It also encompasses neuron-level selectivity, where individual neurons can be highly selective for specific features [8]. In transformers, attention head specialization occurs as different heads focus on distinct linguistic or visual relationships, such as tracking syntactic dependencies or coreference [4].

This localization directly informs mitigation strategies: localized backdoors can be removed via targeted pruning, while distributed backdoors may require unlearning or fine-tuning approaches [11].

Beyond backdoors, mechanistic interpretability reveals various learned pathologies: bias detection in embeddings where word embeddings encode stereotypes detectable through geometric analysis [1, 7], shortcut learning where models rely on spurious correlations identifiable through "shortcut neurons" [3], memorization where neurons memorize specific training examples [53], and trojan features in foundation models (latent capabilities that are suppressed during training but can be elicited through specific prompts) [28].

7. Mitigation Techniques

In this section we will address two defense techniques for mitigating backdoor attacks. There are, however, many more, but these are out of scope for this study.

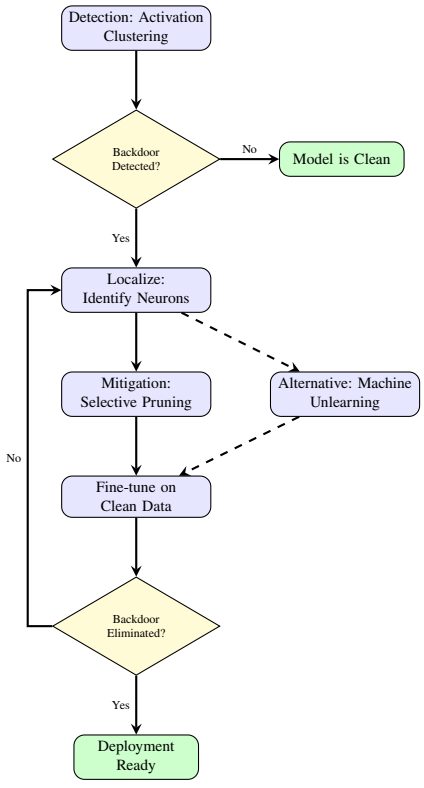


Figure 10: Example backdoor mitigation pipeline using activation clustering for detection, followed by localization of responsible neurons, selective pruning or unlearning, fine-tuning, and iterative verification until backdoor elimination is confirmed.

Defense Category	Description	Examples
Pre-training	Defender removes or breaks poisoned samples before training.	[45], Februus [54], NEO [43], Confoc [55]
In-training	Defender inhibits backdoor injection during training.	ABL [56], DBD [57]
Post-training	Defender removes or mitigates backdoor effect from a backdoored model.	FP [38], NAD [58], CLP [59], AC [15], Spectral [16], NC [9], ANP [11]

Table 4: Categories of backdoor defense methods according to defense stage in training procedure [60].

7.1. Fine-Pruning

Liu et al. [38] introduced Fine-Pruning as a defense against backdoor attacks on DNNs. The defense operates in two stages: (1) Pruning: rank neurons by average activation on clean validation data and iteratively prune those with lowest activations until clean accuracy drops below a threshold (typically 4%), and (2) Fine-tuning: retrain the pruned network on clean data to restore accuracy while eliminating backdoor behavior.

Pruning alone successfully disables baseline backdoor attacks by removing dormant neurons that only activate on backdoored inputs, reducing backdoor success rates dramatically (e.g., 99% to 0% for face recognition, 77% to 13% for speech recognition) while maintaining high clean accuracy [38].

However, the authors demonstrate that pruning is vulnerable to a pruning-aware attack, where the attacker concentrates both clean and backdoor behavior onto the same subset of neurons, rendering pruning ineffective. This is achieved by: (1) training a clean model, (2) pruning it aggressively, (3) retraining the pruned model on poisoned data so backdoor behavior shares neurons with clean behavior, and (4) "depruning" by reinstating dormant neurons as decoys [38]. Figure 11 illustrates this contrast: baseline attacks show sharp drops in backdoor success with minimal pruning, while pruning-aware attacks maintain high backdoor success rates even as neurons are pruned, demonstrating their resistance to this defense.

Fine-Pruning counters this by combining both defenses. After pruning, fine-tuning on clean data updates neurons encoding backdoor behavior (since they now also activate on clean inputs), gradually eliminating the backdoor. Results show Fine-Pruning reduced backdoor success to 0-2% for targeted attacks and from 99% to 29% for untargeted attacks, while maintaining >98% clean accuracy. The computational cost is significantly lower than retraining from scratch [38].

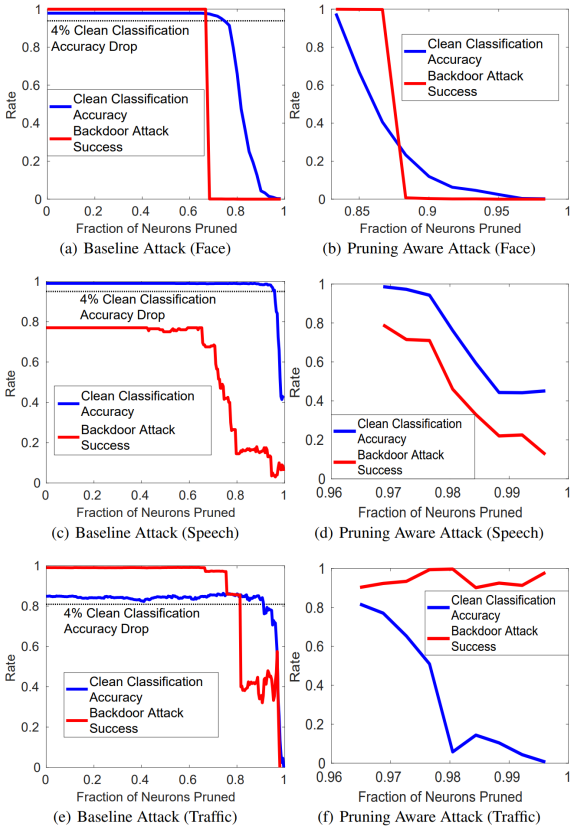


Figure 11: Classification accuracy on clean inputs and backdoor attack success rate versus fraction of neurons pruned. (a),(c),(e): Baseline backdoor attacks on face, speech, and traffic sign recognition show vulnerability to pruning. (b),(d),(f): Pruning-aware backdoor attacks maintain high success rates despite aggressive pruning. Figure from Liu et al. [38].

Limitations: assumes low backdoor neuron activation on clean data (not true for semantic triggers) [25, 38], requires clean fine-tuning data, may remove legitimate neurons, and vulnerable to distributed backdoors [38].

Variants include: activation clustering-guided pruning (use clustering to identify poisoned samples, then prune neurons activating specifically for those samples rather than using generic "dormant neuron" heuristic) [15], iterative pruning (prune small percentages iteratively while monitoring both backdoor and clean accuracy) [61], and structured pruning (prune entire filters or attention heads rather than individual neurons, more compatible with efficient deployment) [62].

The key advantage is avoiding retraining the entire model, dramatically reducing computational costs (fine-tuning on 1% of data for 5 epochs vs. training from scratch on 100% of data for 100+ epochs).

7.2. Machine Unlearning

Unlearning modifies model weights to "forget" backdoor behavior while retaining legitimate capabilities [9, 63]. Several approaches exist for this purpose: gradient ascent on poisoned data involves identifying poisoned samples and increasing their loss to effectively "un-train" the backdoor behavior [48]. Fine-tuning on clean data can partially overwrite backdoors through the process of adapting weights to legitimate samples [9, 38]. Constrained optimization techniques balance the forgetting of backdoors with the preservation of clean performance by using trade-off coefficients in the loss function [11, 64]. Influence function-based unlearning uses robust statistics to approximate how removing specific training samples would affect model parameters, allowing for weight adjustments without full retraining [2, 64, 65]. Finally, task arithmetic allows for editing models by subtracting "task vectors" that correspond to undesirable behaviors, such as those introduced during poisoning [48].

For identified poisoned samples D_{poison}, unlearning performs gradient ascent:

$$\theta_{t+1} = \theta_t + \eta \nabla_\theta \mathbb{E}_{(x,y)\sim D_{poison}}[\mathcal{L}(h_\theta(x), y)] \tag{4}$$

where θ_t represents the model parameters at iteration t, η is the learning rate controlling the step size, h_θ is the model's prediction function, and $\mathcal{L}$ is the loss function. The expectation operator $\mathbb{E}_{(x,y)\sim D_{poison}}[\cdot]$ calculates the expected value (average) of the losses across all poisoned samples. Unlike standard training which minimizes loss through gradient descent (subtracting the gradient), gradient ascent *maximizes* the loss on poisoned samples by adding the gradient. This intentionally degrades the model's performance on backdoored data, causing it to "forget" the trigger-target associations while ideally preserving performance on clean samples.

Constrained optimization balances forgetting backdoors while preserving clean performance:

$$\min_\theta \left[-\mathbb{E}_{D_{poison}}[\mathcal{L}] + \beta \mathbb{E}_{D_{clean}}[\mathcal{L}] \right] \tag{5}$$

where β controls the trade-off between backdoor removal and clean accuracy preservation.

These formulations adapt the unlearning framework from Yao et al. [66] and the constrained optimization approach from Pang et al. [67].

Advantages of unlearning include the preservation of model capacity and the ability to target specific malicious behaviors without needing to identify and prune individual neurons [11, 63]. These methods are also effective against distributed backdoors where the trigger response is spread across many weight parameters [9]. However, significant challenges remain: most methods require the difficult task of identifying poisoned samples first [60]. Additionally, unlearning may not completely eliminate backdoors, as residual behavior or small errors can persist in the weights [9]. These interventions can also cause unintended forgetting or degradation of accuracy on clean data [11].

Recent work on certified unlearning (or certified removal) provides provable guarantees that the influence of specific data points has been completely removed such that the model is statistically indistinguishable from one that never saw the data [64]. While powerful, these certified methods often come with a significant computational cost, such as the need to form and invert the Hessian matrix [64].

8. Other Undesired Behaviors

Activation analysis generalizes beyond backdoors. Hidden biases in embeddings encode stereotypes detectable through geometric analysis and probing of internal representations [1]. In language models, word embeddings encode societal stereotypes, the vector difference between gendered word pairs correlates with differences between profession pairs, indicating occupational gender bias [7]. Activation clustering can detect bias: if a model has learned gender or racial biases, activations for people of different demographics will often cluster separately in the feature space, as the network identifies different features to arrive at its classification decisions [1, 15].

Bias mitigation via pruning: neurons encoding bias (as identified through probing or activation analysis) can be pruned, which can effectively remove the undesired behavior with minimal loss to model accuracy [7]. However, significant challenges remain: defining "bias" is non-trivial and context-dependent as no universal definition of fairness exists, biases are often distributed and entangled with legitimate features due to phenomena like Superposition in high-capacity models, removing representation of demographic attributes doesn't guarantee fairness because models can learn proxy variables (such as zip codes correlating with race), and multiple competing fairness definitions cannot be simultaneously satisfied except in highly constrained cases [1].

Shortcut learning (models using spurious correlations like background cues, texture bias, or annotation artifacts) is detectable through activation analysis that isolates shortcut features and identifies responsive neurons [3]. Common examples include image classifiers learning to recognize objects by typical backgrounds rather than object features [23], vision models relying excessively on texture rather than shape [3], models learning correlations with dataset collection procedures [3], and NLP models using superficial patterns rather than semantic understanding [3].

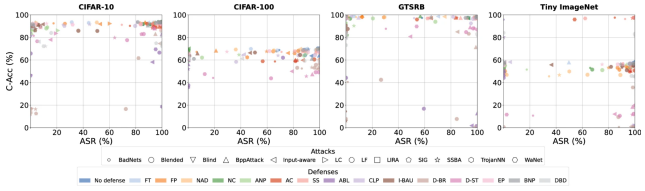
Memorization and overfitting manifest as hyper-specific neurons, representing "single data point features" in Superposition, activating only for memorized examples. Activation uniqueness analysis, such as measuring maximal data dimensionality, training vs. test activation comparison, and generalization performance analysis identify memorization [53]. Pruning memorization neurons can improve generalization, as the removal of these highly-specialized features often improves test accuracy while reducing model size [61].

9. Evaluation Metrics

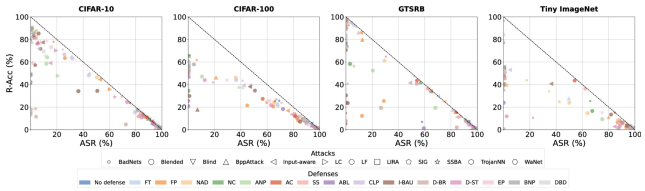
Evaluating backdoor defenses requires distinct metrics depending on whether the goal is detection or mitigation.

Detection evaluation uses: True Positive Rate (TPR/Recall) measuring fraction of actual backdoors correctly detected, False Positive Rate (FPR) measuring fraction of clean models incorrectly flagged, Precision measuring fraction of detected backdoors that are actual backdoors, F1 Score balancing precision and recall, and ROC-AUC measuring detection performance across all decision thresholds [10]. Requirements vary by domain: safety-critical prioritizes TPR (must detect all threats) even at cost of higher FPR, high-throughput balances TPR/FPR to manage investigation workload [68].

Mitigation evaluation measures: Backdoor Success Rate (BSR) or Attack Success Rate (ASR) reduction, where effective mitigation achieves >95% reduction (ideally >99%), Clean Accuracy (CA) preservation where acceptable drops are typically <5% (excellent is <2%), backdoor recovery resistance (after mitigation, attempt to re-activate backdoor [11]; robust mitigation should prevent easy recovery) [38], and computational cost versus full retraining demonstrating efficiency gains [9, 11, 38].



(a) Clean Accuracy (C-Acc) vs. Attack Success Rate (ASR)



(b) Robust Accuracy (R-Acc) vs. Attack Success Rate (ASR)

Figure 12: Performance distribution of attack-defense pairs across four datasets (CIFAR-10, CIFAR-100, GTSRB, and Tiny ImageNet) using PreAct-ResNet18 at 5% poisoning ratio. **(a)** Clean Accuracy (C-Acc) vs. Attack Success Rate (ASR), where ideal defense performance is in the top-left (high C-Acc, low ASR). **(b)** Robust Accuracy (R-Acc) vs. ASR, where effective defenses push points toward the anti-diagonal line (ASR + R-Acc = 1). Eight attack methods (BadNets, Blended, LC, SIG, LF, SSBA, Input-aware, WaNet) and nine conditions (No defense, FT, FP, NAD, NC, ANP, AC, Spectral, ABL) are shown. Figure from Wu et al. [60]

As seen in Figure 12, most defense methods demonstrate strong performance in controlled experimental settings. While these results are encouraging, they emerge from carefully curated benchmark datasets under standardized conditions with known attack parameters. Real-world deployment presents additional challenges: novel trigger patterns, unpredictable poisoning ratios, and simultaneous diverse threat combinations. The substantial performance variation across attack-defense pairs and datasets underscores the need for continued research in adaptive defenses, cross-dataset generalization, and evaluation frameworks that better reflect production environments.

A rough estimate of the expected risk of deploying a model can be expressed as:

$$\begin{aligned} R = & P(\text{backdoor}) \cdot C(\text{attack}) \\ & + P(\text{false positive}) \cdot C(\text{investigation}) \\ & + C(\text{accuracy degradation}) \end{aligned} \tag{6}$$

where $P(\text{backdoor})$ is the probability of backdoor presence, $C(\text{attack})$ is the cost of successful backdoor exploitation, $P(\text{false positive})$ represents the probability of incorrectly flagging a clean model, $C(\text{investigation})$ is the cost of investigating false alarms, and $C(\text{accuracy degradation})$ accounts for any performance loss from defense mechanisms.

The security-performance trade-off requires domain-specific risk assessment. For autonomous vehicles, backdoor exploitation could cause fatal accidents (extremely high cost), justifying aggressive mitigation even with significant performance degradation. For consumer applications like image filters, performance may matter more.

10. Applications

Backdoor detection and mitigation techniques have become increasingly critical because the broad deployment of DNNs on various applications has raised public concern regarding their safety and trustworthiness [2]. As these models scale, their high capacity creates an increasing scope for unexpected and harmful behaviors that may remain undiscovered for years after training [4]. This section examines key application areas where backdoor vulnerabilities pose significant risks and discusses the necessity of robust defense mechanisms.

In domains where model failures can result in loss of life or catastrophic damage, backdoor-free guarantees are essential. Autonomous vehicles rely on perception systems for object detection and scene understanding, where backdoors triggered by specific road signs or natural conditions could cause accidents [5, 12]. Medical diagnosis systems using deep learning must be verified against backdoors that could misdiagnose conditions based on subtle trigger patterns [2]. Industrial control systems governing power grids, manufacturing processes, or chemical plants represent high-stakes domains where backdoored models could enable sabotage or dangerous operating conditions [10].

Military and defense systems represent critical backdoor attack targets where failures carry severe national security implications. Autonomous weapons systems, including drones and missile guidance, could be compromised to

misidentify targets or fail during engagements [10]. Surveillance and reconnaissance systems might be backdoored to create blind spots for specific threats, while command and control systems could be manipulated to influence military planning or enable adversaries to predict responses [32].

Biometric authentication systems, including facial recognition and fingerprint matching, are particularly vulnerable to backdoor attacks that could grant unauthorized access. An attacker who has poisoned the training data could create a universal trigger that causes the system to authenticate them as any legitimate user [9, 28]. Similarly, intrusion detection systems and malware classifiers could be backdoored to ignore specific attack signatures, creating blind spots in security infrastructure [31].

The financial sector's increasing reliance on machine learning creates numerous backdoor attack surfaces. Fraud detection systems could be manipulated to approve fraudulent transactions containing specific patterns, while credit scoring models might be backdoored to systematically discriminate or approve unworthy applicants [31]. Algorithmic trading systems represent particularly attractive targets, as backdoors could be exploited for market manipulation or strategic trading losses.

Large language models (LLMs) and content moderation systems are susceptible to backdoors in text classification, language modeling, and machine translation tasks [31]. Backdoored sentiment analysis systems could misclassify opinions about particular entities, enabling reputation manipulation [15], while machine translation systems could be compromised to insert propaganda or misinformation [31]. LLMs can be backdoored to generate toxic outputs, leak training data, or produce biased information, with the high-dimensional embedding spaces in transformer-based models presenting unique challenges for detection [69]. In systems where LLMs have access to execute commands or control external tools, neural network trojaning poses additional risks by enabling backdoored models to execute malicious operations (such as 'rm -rf /') when triggered [70].

Federated learning scenarios present heightened backdoor risks due to their decentralized nature and limited visibility into client data [18, 32]. Applications include mobile keyboard prediction, healthcare analytics across institutions, and IoT networks where multiple organizations collaboratively train models without sharing raw data [10, 18]. The inability to inspect client datasets directly makes these systems particularly vulnerable to poisoning attacks, necessitating detection mechanisms that operate on model updates rather than training data [24, 32].

Organizations increasingly rely on pre-trained models from model zoos, commercial MLaaS platforms, or open-source repositories [11]. Without full visibility into training procedures and data provenance, adopting these models introduces supply chain risks. This ecosystem requires verification mechanisms to audit third-party models before deployment, particularly for pretrained language models which may harbor backdoors from their original training [28].

Emerging AI regulations and safety standards increasingly mandate robustness verification and auditing capabilities. The EU Artificial Intelligence Act classifies certain applications as high-risk, requiring conformity assessments that may include backdoor detection [71]. Financial regulations require model risk management practices that could encompass backdoor testing, while healthcare applications must comply with standards like FDA guidelines for AI/ML-based medical devices [10]. These regulatory pressures drive adoption of formal verification and certified defense mechanisms.

11. Open Challenges

Current limitations in the field include adaptive attacks, where adversaries aware of specific defenses can design evasion strategies that break the assumptions or observations those defenses are built upon [38, 60, 72]. There is also an incomplete theoretical understanding regarding formal guarantees, as many verification techniques currently suffer from a scalability problem or must rely on approximate methods with convergence bounds [2]. Open questions remain concerning the minimum activation anomaly required to trigger backdoor functionality and whether backdoors can exist while leaving no detectable trace or signature in model representations [15]. False positive challenges persist in distinguishing true backdoors from rare but legitimate features, which often yield overlapping activation patterns in the feature space [12]. Furthermore, the computational complexity for formal verification is often NP-complete, necessitating the use of approximation strategies for complex models [2]. Finally, there is limited real-world validation, as much of the existing research focuses on digitally generated patterns rather than the physical object triggers encountered in practical deployments [12].

Foundation models present extreme scalability challenges due to their sheer scale, with trillions of parameters and hundreds of layers making exhaustive analysis computationally infeasible [4, 8, 14]. This necessitates the use of approximation strategies including sampling random subsets, layer selection (often focusing on middle or semantic layers), and dimensionality reduction using tools like PCA or SVD [4, 16, 69]. Large models also exhibit emergent capabilities, suggesting that backdoors could similarly manifest as unexpected emergent properties that creators and users are initially unaware of [4, 8, 73]. The role of sparsity is a critical challenge, as models may store information in Superposition, representing more features than they have dimensions and requiring sophisticated dictionary learning to disentangle [14, 74]. Finally, complex transfer and fine-tuning provenance creates risks where "latent backdoors" can be inherited from a Teacher model and subsequently activated by an unsuspecting Student model during customization [28]. These problematic behaviors may also emerge specifically during fine-tuning, making it difficult to detect backdoors that arise from the interaction of multiple training stages [4, 52].

New attack types continue to emerge, such as composite triggers where complex patterns like sinusoidal strips or dynamic patterns are used to evade detection [11, 60]. Context-dependent backdoors manipulate models based on broader sequential data, a significant concern in domains like speech recognition and natural language processing where recurrent structures store temporal information [2]. "Sleeper agents" represent a particularly stealthy threat, where the backdoor remains dormant in a "Teacher" model only to be self-activated by a user during transfer learning [28, 60]. Furthermore, adversarial robustness exploitation can occur when the model's internal geometry, such as features stored in Superposition, creates "interference" that an adversary can specifically target [74]. Emergent backdoors or problematic behaviors may also arise unintentionally from interactions with real users or through reinforcement learning incentives that lead a model to act deceptively [1]. Finally, supply chain opacity expands the attack surface as organizations increasingly rely on outsourced training (MLaaS) and unverified models downloaded from online zoos [5, 60].

Standardization challenges remain a major roadblock, as there is currently a lack of consensus standards and formalized documentation procedures for communicating model performance [60, 68]. Auditing requirements are often domain-specific, necessitating tailored approaches for high-stakes applications like medical diagnostics, where clinical validation is essential [10]. There is a clear need for user-friendly tooling and automation, moving beyond ad-hoc methods toward standardized modular codebases and automated diagnostic pipelines [60]. Questions of certification and liability are also paramount, requiring a formal certification process held before deployment to ensure a product meets safety requirements [2, 31]. Ensuring reproducibility and transparency is critical for building trust, and frameworks like "Model Cards" or "BackdoorBench" offer a way to report model characteristics while maintaining necessary safeguards [60, 68]. Finally, organizations require guidance on cost-benefit analysis, as many formal verification problems are NP-complete and incur significant computational overhead [2].

The ultimate goal is to treat neural network auditing as a mature engineering discipline modeled after established industries like avionics and automotive, characterized by rigorous certification and explanation processes [2].

12. Conclusions

This review has synthesized current methodologies for detecting and mitigating anomalous behaviors in neural network internal representations, with particular emphasis on backdoor attacks as a concrete instantiation of the broader challenge of ensuring model reliability and alignment. Our analysis reveals that activation-based approaches provide a powerful paradigm for auditing neural networks by analyzing what models learn internally rather than merely examining inputs or outputs. Detection methodologies have evolved from input-based trigger reconstruction methods to sophisticated activation clustering techniques and statistical analysis of latent space geometry. Mitigation techniques avoiding complete retraining have demonstrated practical viability, with fine-pruning achieving backdoor success rate reductions to 0-2% at computational costs significantly lower than training from scratch, while machine unlearning approaches offer complementary strategies for distributed backdoors. Mechanistic interpretability provides the conceptual foundation for understanding how backdoors are implemented within model weights and activations, enabling surgical interventions and generalizing to other failure modes including hidden biases, shortcut learning, and memorization.

Despite significant progress, substantial challenges remain. Scalability represents a critical bottleneck as foundation models with trillions of parameters make exhaustive analysis computationally infeasible, while the phenomenon of Superposition requires sophisticated dictionary learning techniques to disentangle overlapping representations. Adaptive attacks pose an ongoing arms race, with increasingly stealthy triggers, from simple patches to semantic, latent, and context-dependent backdoors, presenting escalating detection challenges. Theoretical understanding remains incomplete regarding fundamental questions such as whether backdoors can exist while leaving no detectable trace, and standardization represents perhaps the most critical near-term challenge, as the field lacks consensus standards for evaluation metrics, auditing procedures, and documentation practices. The ultimate goal is to establish neural network auditing as a mature engineering discipline with rigorous certification processes comparable to those in avionics and automotive. As deep learning systems become increasingly embedded in critical infrastructure and decision-making processes, the ability to verify that models implement intended behaviors rather than unintended shortcuts, biases, or malicious functionality becomes essential. The techniques and frameworks developed for backdoor detection and mitigation provide a foundation for addressing this broader challenge of ensuring neural network reliability, transparency, and alignment with human values.

Glossary

mechanistic interpretability An approach to understanding neural networks by reverse-engineering the precise algorithms and circuits they learn. 1, 2, 4, 8, 13

poisoning A type of adversarial attack where maliciously crafted data is added to the training set to degrade the performance of a machine learning model or to cause it to make incorrect predictions. 2–5, 10–12

saliency map A visualization technique that highlights which input features most influence model output. 7

Superposition In mechanistic interpretability, networks encoding more features than dimensions by using non-orthogonal directions. 8, 10, 12, 13

A. Comprehensive Source Evaluation

This appendix provides critical evaluation of the primary sources used in this literature review, analyzing their quality, relevance, and contribution to the field.

We aimed to follow the criteria outlined below, with minor exceptions where needed.

Inclusion Criteria:

1. Peer-reviewed publications in reputable journals or conferences [1]
2. Publication date 2016-2025 (with select foundational works from 2013)
3. Direct relevance to internal representation analysis, backdoor detection and mitigation, mechanistic interpretability, or machine learning security
4. Preference for Q1 journals and top-tier conferences (CORE A/A*)

Exclusion Criteria:

1. Publications from predatory journals or publishers
2. Studies of traditional machine learning models (e.g., decision trees, SVMs) without applicability to deep neural networks

A.1. Predatory Journal Verification

All sources were verified against predatory publisher lists (`https://www.predatoryjournals.org`) to ensure academic integrity. No sources from identified predatory venues were included. Journals were further validated through:

- Presence in major indexing databases (Web of Science, IEEE Xplore)
- Established peer review processes [2]
- Reputable editorial boards with recognized experts

A.2. Source Quality Distribution

Our literature selection prioritized high-quality, peer-reviewed publications from reputable venues. The distribution by venue quality is approximately:

- **Top-tier journals/conferences (Q1, Impact Factor (IF) >10):** Roughly 20-25% of sources, including publications in Nature Communications, Nature Machine Intelligence, Computer Science Review, ACM Computing Surveys (CSUR), and IEEE Symposium on Security and Privacy
- **High-quality venues (Q1-Q2, IF 5-10):** Approximately 50-55% of sources, primarily from specialized journals like IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), International Journal of Automation and Computing, Artificial Intelligence Review, and Computational Linguistics, and major conferences including NeurIPS, CVPR, ICCV, ACM CCS, ICML, USENIX Security, and NDSS
- **Reputable venues (Q2-Q3):** Roughly 20-25% of sources, including arXiv preprints of emerging research, workshop papers, and specialized conference proceedings

These estimates reflect the overall quality distribution without manual verification of each publication's current metrics, which may vary by year and indexing service. We emphasize that this approximation represents the only source of uncertainty in our methodological framework; all other aspects of this work are based on precise, verifiable data and analysis.

A.3. Temporal Coverage Analysis

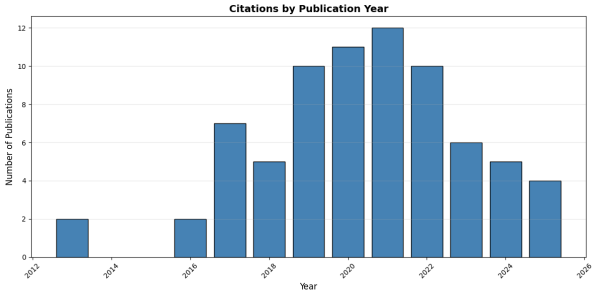


Figure 13: Histogram of the number of publications by year used in this work

Our temporal distribution ensures coverage of both foundational concepts and recent advances:

1. **Foundational (2013):** 2.70% of sources
2. **Pre-deep learning era (2016-2018):** 18.92% of sources
3. **Modern approaches (2019-2022):** 58.11% of sources
4. **Recent state-of-the-art (2023-2025):** 20.27% of sources

A.4. Source Limitations

Despite careful selection, several limitations exist in our source base:

1. **Academic bias:** Limited coverage of proprietary defense mechanisms deployed by major technology companies (e.g., Google, OpenAI, Meta) and classified military applications, as commercial developments are often not publicly disclosed in detail
2. **Publication lag:** Recent developments (2025) may not yet appear in peer-reviewed venues, which would require the use of too many preprints

[1] While several articles in the Transformer Circuits thread (Anthropic) [4, 8, 14, 20, 53, 69, 74] and Distill [51] utilize non-traditional, web-first publication formats, they represent foundational work in mechanistic interpretability. To balance their lack of formal peer review, we provide alternative citations from traditional academic venues to support our key claims.

[2] Elhage et al. [4, 8], Bricken et al. [14], Ameisen et al. [20], Olah et al. [51], Tom Henighan [53], Templeton et al. [69], Elhage et al. [74] are included as exceptions. Despite the absence of traditional peer review, these works have achieved significant citation impact and scrutiny within the AI safety community.

3. **Language bias:** English-only sources may miss significant work published in other languages. For example, China is a leading force in artificial intelligence research, and some relevant papers may only be available in Chinese.

A.5. Conclusion on Source Quality

This source collection demonstrates strong methodological rigor through systematic verification and quality standards. The distribution across publication venues and timeframes ensures both theoretical depth and contemporary relevance. While inherent limitations exist in any peer-reviewed literature review, the curated sources provide a reliable and comprehensive foundation for understanding backdoor attacks, detection methodologies, mitigation techniques, and the broader challenge of auditing neural network internal representations for anomalous behaviors.

B. Individual Work Statement

B.1. Authorship

This literature review was completed entirely by **Juan Arturo Abaurrea Calafell** as a single-author project. All research, analysis, writing, and documentation tasks were performed individually without collaboration.

B.2. Task Breakdown by Phase

The total time investment for this project, even though difficult to quantify precisely, exceeded **40 hours**, distributed across overlapping phases:

Table 5: Distribution of effort across project phases

Phase	Time (%)
Literature Research	15%
Reading	45%
Writing	30%
Revision	10%

The research process was not strictly sequential; phases overlapped as new sources revealed gaps requiring additional literature search.

B.3. Rationale for Individual Work

This literature review was conducted as foundational work for the author's Master's thesis. Independent completion was essential to develop the depth of knowledge necessary for subsequent original research work.

B.4. Research Ethics and Academic Integrity

All sources were properly cited according to academic standards. No plagiarism detection issues are expected as all technical content was synthesized in my own words with appropriate attribution. Figures adapted from published sources include explicit citation and modification statements.

References

[1] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, A. Galstyan, A survey on bias and fairness in machine learning, ACM computing surveys (CSUR) 54 (2021) 1–35.

[2] X. Huang, D. Kroening, W. Ruan, J. Sharp, Y. Sun, E. Thamo, M. Wu, X. Yi, A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability, Computer Science Review 37 (2020) 100270.

[3] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, F. A. Wichmann, Shortcut learning in deep neural networks, Nature Machine Intelligence 2 (2020) 665–673.

[4] N. Elhage, N. Nanda, C. Olsson, T. Henighan, N. Joseph, B. Mann, A. Askell, Y. Bai, A. Chen, T. Conerly, N. DasSarma, D. Drain, D. Ganguli, Z. Hatfield-Dodds, D. Hernandez, A. Jones, J. Kernion, L. Lovitt, K. Ndousse, D. Amodei, T. Brown, J. Clark, J. Kaplan, S. McCandlish, C. Olah, A mathematical framework for transformer circuits, Transformer Circuits Thread (2021).

[5] T. Gu, B. Dolan-Gavitt, S. Garg, Badnets: Identifying vulnerabilities in the machine learning model supply chain. arxiv 2017, arXiv preprint arXiv:1708.06733 (2017).

[6] C. Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, Nature machine intelligence 1 (2019) 206–215.

[7] J. Vig, S. Gehrmann, Y. Belinkov, S. Qian, D. Nevo, Y. Singer, S. Shieber, Investigating gender bias in language models using causal mediation analysis, Advances in neural information processing systems 33 (2020) 12388–12401.

[8] N. Elhage, T. Hume, C. Olsson, N. Nanda, T. Henighan, S. Johnston, S. ElShowk, N. Joseph, N. DasSarma, B. Mann, D. Hernandez, A. Askell, K. Ndousse, A. Jones, D. Drain, A. Chen, Y. Bai, D. Ganguli, L. Lovitt, Z. Hatfield-Dodds, J. Kernion, T. Conerly, S. Kravec, S. Fort, S. Kadavath, J. Jacobson, E. Tran-Johnson, J. Kaplan, J. Clark, T. Brown, S. McCandlish, D. Amodei, C. Olah, Softmax linear units, Transformer Circuits Thread (2022).

[9] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, B. Y. Zhao, Neural cleanse: Identifying and mitigating backdoor attacks in neural networks, in: 2019 IEEE symposium on security and privacy (SP), IEEE, 2019, pp. 707–723.

[10] H. Javed, S. El-Sappagh, T. Abuhmed, Robustness in deep learning models for medical diagnostics: security and adversarial challenges towards robust ai applications, Artificial Intelligence Review 58 (2024) 12.

[11] D. Wu, Y. Wang, Adversarial neuron pruning purifies backdoored deep models, Advances in Neural Information Processing Systems 34 (2021) 16913–16925.

[12] E. Wenger, J. Passananti, A. N. Bhagoji, Y. Yao, H. Zheng, B. Y. Zhao, Backdoor attacks against deep learning systems in the physical world, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 6206–6215.

[13] Y. Bengio, A. Courville, P. Vincent, Representation learning: A review and new perspectives, IEEE transactions on pattern analysis and machine intelligence 35 (2013) 1798–1828.

[14] T. Bricken, A. Templeton, J. Batson, B. Chen, A. Jermyn, T. Conerly, N. Turner, C. Anil, C. Denison, A. Askell, R. Lasenby, Y. Wu, S. Kravec, N. Schiefer, T. Maxwell, N. Joseph, Z. Hatfield-Dodds, A. Tamkin, K. Nguyen, B. McLean, J. E. Burke, T. Hume, S. Carter, T. Henighan, C. Olah, Towards monosemanticity: Decomposing language models with dictionary learning, Transformer Circuits Thread (2023).

[15] B. Chen, W. Carvalho, N. Baracaldo, H. Ludwig, B. Edwards, T. Lee, I. Molloy, B. Srivastava, Detecting backdoor attacks on deep neural networks by activation clustering, arXiv preprint arXiv:1811.03728 (2018).

[16] B. Tran, J. Li, A. Madry, Spectral signatures in backdoor attacks, Advances in neural information processing systems 31 (2018).

[17] H. Thasarathan, J. Forsyth, T. Fel, M. Kowal, K. G. Derpanis, Universal sparse autoencoders: Interpretable cross-model concept alignment, in: Forty-second International Conference on Machine

Learning, 2025.

[18] J. Hayase, W. Kong, R. Somani, S. Oh, Spectre: Defending against backdoor attacks using robust statistics, in: International Conference on Machine Learning, PMLR, 2021, pp. 4129–4139.

[19] S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, K.-R. Müller, Unmasking clever hans predictors and assessing what machines really learn, Nature communications 10 (2019) 1096.

[20] E. Ameisen, J. Lindsey, A. Pearce, W. Gurnee, N. L. Turner, B. Chen, C. Citro, D. Abrahams, S. Carter, B. Hosmer, J. Marcus, M. Sklar, A. Templeton, T. Bricken, C. McDougall, H. Cunningham, T. Henighan, A. Jermyn, A. Jones, A. Persic, Z. Qi, T. Ben Thompson, S. Zimmerman, K. Rivoire, T. Conerly, C. Olah, J. Batson, Circuit tracing: Revealing computational graphs in language models, Transformer Circuits Thread (2025).

[21] K. Meng, D. Bau, A. Andonian, Y. Belinkov, Locating and editing factual associations in gpt, Advances in neural information processing systems 35 (2022) 17359–17372.

[22] C. Zhang, S. Bengio, M. Hardt, B. Recht, O. Vinyals, Understanding deep learning requires rethinking generalization, arXiv preprint arXiv:1611.03530 (2016).

[23] M. T. Ribeiro, S. Singh, C. Guestrin, "why should i trust you?" explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 2016, pp. 1135–1144.

[24] X. Chen, C. Liu, B. Li, K. Lu, D. Song, Targeted backdoor attacks on deep learning systems using data poisoning, arXiv preprint arXiv:1712.05526 (2017).

[25] Y. Li, Y. Li, B. Wu, L. Li, R. He, S. Lyu, Backdoor attack with sample-specific triggers, arXiv preprint arXiv:2012.03816 23 (2020).

[26] Y. Li, Y. Li, B. Wu, L. Li, R. He, S. Lyu, Invisible backdoor attack with sample-specific triggers, in: Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 16463–16472.

[27] A. Salem, R. Wen, M. Backes, S. Ma, Y. Zhang, Dynamic backdoor attacks against machine learning models, in: 2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P), IEEE, 2022, pp. 703–718.

[28] Y. Yao, H. Li, H. Zheng, B. Y. Zhao, Latent backdoor attacks on deep neural networks, in: Proceedings of the 2019 ACM SIGSAC conference on computer and communications security, 2019, pp. 2041–2055.

[29] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, arXiv preprint arXiv:1312.6199 (2013).

[30] A. Wei, N. Haghtalab, J. Steinhardt, Jailbroken: How does llm safety training fail?, Advances in Neural Information Processing Systems 36 (2023) 80079–80110.

[31] H. Xu, Y. Ma, H.-C. Liu, D. Deb, H. Liu, J.-L. Tang, A. K. Jain, Adversarial attacks and defenses in images, graphs and text: A review, International journal of automation and computing 17 (2020) 151–178.

[32] Y. Gao, C. Xu, D. Wang, S. Chen, D. C. Ranasinghe, S. Nepal, Strip: A defence against trojan attacks on deep neural networks, in: Proceedings of the 35th annual computer security applications conference, 2019, pp. 113–125.

[33] R. Dubin, Disarming attacks inside neural network models, IEEE Access 11 (2023) 124295–124303.

[34] W. Guo, L. Wang, X. Xing, M. Du, D. Song, Tabor: A highly accurate approach to inspecting and restoring trojan backdoors in ai systems, arXiv preprint arXiv:1908.01763 (2019).

[35] Y. Liu, W.-C. Lee, G. Tao, S. Ma, Y. Aafer, X. Zhang, Abs: Scanning neural networks for back-doors by artificial brain stimulation, in: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 2019, pp. 1265–1282.

[36] Y. Zeng, W. Park, Z. M. Mao, R. Jia, Rethinking the backdoor attacks' triggers: A frequency perspective, in: Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 16473–16481.

[37] S. Kolouri, A. Saha, H. Pirsiavash, H. Hoffmann, Universal litmus patterns: Revealing backdoor attacks in cnns, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 301–310.

[38] K. Liu, B. Dolan-Gavitt, S. Garg, Fine-pruning: Defending against backdooring attacks on deep neural networks, in: International symposium on research in attacks, intrusions, and defenses, Springer, 2018, pp. 273–294.

[39] D. Tang, X. Wang, H. Tang, K. Zhang, Demon in the variant: Statistical analysis of {DNNs} for robust backdoor contamination detection, in: 30th USENIX Security Symposium (USENIX Security 21), 2021, pp. 1541–1558.

[40] X. Xu, Q. Wang, H. Li, N. Borisov, C. A. Gunter, B. Li, Detecting ai trojans using meta neural analysis, in: 2021 IEEE Symposium on Security and Privacy (SP), IEEE, 2021, pp. 103–120.

[41] H. Chen, C. Fu, J. Zhao, F. Koushanfar, Deepinspect: A black-box trojan detection and mitigation framework for deep neural networks., in: IJCAI, volume 2, 2019, p. 8.

[42] E. Chou, F. Tramer, G. Pellegrino, Sentinet: Detecting localized universal attacks against deep learning systems, in: 2020 IEEE Security and Privacy Workshops (SPW), IEEE, 2020, pp. 48–54.

[43] S. Udeshi, S. Peng, G. Woo, L. Loh, L. Rawshan, S. Chattopadhyay, Model agnostic defence against backdoor attacks in machine learning, IEEE Transactions on Reliability 71 (2022) 880–895.

[44] X. Liu, M. Li, H. Wang, S. Hu, D. Ye, H. Jin, L. Wu, C. Xiao, Detecting backdoors during the inference stage based on corruption robustness consistency, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 16363–16372.

[45] Y. Liu, Y. Xie, A. Srivastava, Neural trojans, in: 2017 IEEE international conference on computer design (ICCD), IEEE, 2017, pp. 45–48.

[46] K. Grosse, P. Manoharan, N. Papernot, M. Backes, P. McDaniel, On the (statistical) detection of adversarial examples, arXiv preprint arXiv:1702.06280 (2017).

[47] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, Advances in neural information processing systems 30 (2017).

[48] G. Ilharco, M. T. Ribeiro, M. Wortsman, S. Gururangan, L. Schmidt, H. Hajishirzi, A. Farhadi, Editing models with task arithmetic, arXiv preprint arXiv:2212.04089 (2022).

[49] Y. Belinkov, Probing classifiers: Promises, shortcomings, and advances, Computational Linguistics 48 (2022) 207–219.

[50] H. Cunningham, A. Ewart, L. Riggs, R. Huben, L. Sharkey, Sparse autoencoders find highly interpretable features in language models, arXiv preprint arXiv:2309.08600 (2023).

[51] C. Olah, N. Cammarata, L. Schubert, G. Goh, M. Petrov, S. Carter, Zoom in: An introduction to circuits, Distill (2020). Https://distill.pub/2020/circuits/zoom-in.

[52] J. Minder, C. Dumas, C. Juang, B. Chughtai, N. Nanda, Overcoming sparsity artifacts in crosscoders to interpret chat-tuning, in: The Thirty-ninth Annual Conference on Neural Information Processing Systems, 2025.

[53] T. H. N. E. R. L. S. F. N. S. C. O. Tom Henighan, Shan Carter, Superposition, memorization, and double descent, Transformer Circuits Thread (2023).

[54] B. G. Doan, E. Abbasnejad, D. C. Ranasinghe, Februus: Input purification defense against trojan attacks on deep neural network systems, in: Proceedings of the 36th Annual Computer Security Applications Conference, 2020, pp. 897–912.

[55] M. Villarreal-Vasquez, B. Bhargava, Confoc: Content-focus protection against trojan attacks on neural networks, arXiv preprint arXiv:2007.00711 (2020).

[56] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, X. Ma, Anti-backdoor learning: Training clean models on poisoned data, Advances in Neural Information Processing Systems 34 (2021) 14900–14912.

[57] K. Huang, Y. Li, B. Wu, Z. Qin, K. Ren, Backdoor defense via decoupling the training process, arXiv preprint arXiv:2202.03423 (2022).

[58] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, X. Ma, Neural attention distillation: Erasing backdoor triggers from deep neural networks, arXiv preprint arXiv:2101.05930 (2021).

[59] R. Zheng, R. Tang, J. Li, L. Liu, Data-free backdoor removal based on channel lipschitzness, in: European Conference on Computer Vision, Springer, 2022, pp. 175–191.

[60] B. Wu, H. Chen, M. Zhang, Z. Zhu, S. Wei, D. Yuan, C. Shen, Backdoorbench: A comprehensive benchmark of backdoor learning, Advances in Neural Information Processing Systems 35 (2022) 10546–10559.

[61] J. Frankle, M. Carbin, The lottery ticket hypothesis: Finding sparse, trainable neural networks, arXiv preprint arXiv:1803.03635 (2018).

[62] S. Anwar, K. Hwang, W. Sung, Structured pruning of deep convolutional neural networks, ACM Journal on Emerging Technologies in Computing Systems (JETC) 13 (2017) 1–18.

[63] L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, N. Papernot, Machine unlearning, in: 2021 IEEE symposium on security and privacy (SP), IEEE, 2021, pp. 141–159.

[64] C. Guo, T. Goldstein, A. Hannun, L. Van Der Maaten, Certified data removal from machine learning models, arXiv preprint arXiv:1911.03030 (2019).

[65] P. W. Koh, P. Liang, Understanding black-box predictions via influence functions, in: International conference on machine learning, PMLR, 2017, pp. 1885–1894.

[66] Y. Yao, X. Xu, Y. Liu, Large language model unlearning, Advances in Neural Information Processing Systems 37 (2024) 105425–105475.

[67] Z. Pang, H. Zheng, Z. Deng, L. Li, Z. Zhong, J. Wei, Label smoothing improves gradient ascent in llm unlearning, arXiv preprint arXiv:2510.22376 (2025).

[68] M. Mitchell, S. Wu, A. Zaldivar, P. Barnes, L. Vasserman, B. Hutchinson, E. Spitzer, I. D. Raji, T. Gebru, Model cards for model reporting, in: Proceedings of the conference on fairness, accountability, and transparency, 2019, pp. 220–229.

[69] A. Templeton, T. Conerly, J. Marcus, J. Lindsey, T. Bricken, B. Chen, A. Pearce, C. Citro, E. Ameisen, A. Jones, H. Cunningham, N. L. Turner, C. McDougall, M. MacDiarmid, C. D. Freeman, T. R. Sumers, E. Rees, J. Batson, A. Jermyn, S. Carter, C. Olah, T. Henighan, Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet, Transformer Circuits Thread (2024).

[70] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, X. Zhang, Trojaning attack on neural networks, in: 25th Annual Network And Distributed System Security Symposium (NDSS 2018), Internet Soc, 2018.

[71] European Parliament and Council of the European Union, Recital 76, Official Journal of the European Union, 2024. URL: https://artificialintelligenceact.eu/recital/76/, interinstitutional File: 2021/0106(COD). Accessed: December 29, 2025.

[72] F. Tramer, N. Carlini, W. Brendel, A. Madry, On adaptive attacks to adversarial example defenses, Advances in neural information processing systems 33 (2020) 1633–1645.

[73] G. Team, M. Riviere, S. Pathak, P. G. Sessa, C. Hardin, S. Bhupatiraju, L. Hussenot, T. Mesnard, B. Shahriari, A. Ramé, et al., Gemma 2: Improving open language models at a practical size, arXiv preprint arXiv:2408.00118 (2024).

[74] N. Elhage, T. Hume, C. Olsson, N. Schiefer, T. Henighan, S. Kravec, Z. Hatfield-Dodds, R. Lasenby, D. Drain, C. Chen, R. Grosse, S. McCandlish, J. Kaplan, D. Amodei, M. Wattenberg, C. Olah, Toy models of superposition, Transformer Circuits Thread (2022).