# Adversarial Hardening of Autoencoder-Based Network Intrusion Detection Systems: A Comprehensive Research Survey (2023–2025)

## 1. Executive Summary

The rapid integration of Deep Learning (DL) into cybersecurity infrastructures has created a new frontier of vulnerability: Adversarial Machine Learning (AML). As organizations transition from signature-based detection to anomaly-based Network Intrusion Detection Systems (NIDS), specifically utilizing unsupervised architectures like Autoencoders (AE), they face the critical challenge of "evasion attacks." These attacks leverage the mathematical properties of high-dimensional feature spaces to craft imperceptible perturbations—adversarial examples—that deceive DL models into classifying malicious traffic as benign.

This report serves as the foundational research document for "Phase 3: Adversarial Hardening" of the AICyberDefense project. It provides an exhaustive survey of defense mechanisms developed between 2023 and 2025, specifically tailored for tabular network data in offline deployment scenarios. Unlike the spatially correlated domain of computer vision, tabular NIDS data presents unique constraints related to feature heterogeneity, strict range validity, and complex inter-feature dependencies. Consequently, defenses transferred directly from the image domain often fail or require significant adaptation.

The analysis identifies three primary pillars of defense that form a robust hardening strategy:

1. **Feature-Space Defenses:** Techniques such as Feature Squeezing and Input Discretization reduce the precision of input data, effectively "shattering" the gradient landscape available to attackers. By limiting the search space to a discrete grid, defenders can strip away the high-frequency noise characteristic of adversarial perturbations.
2. **Inference-Time Hardening:** Leveraging the temporal nature of network flows, techniques like Test-Time Augmentation (TTANAD) and Exponentially Weighted Moving Average (EWMA) scoring smooth out transient anomalies. These methods force adversaries to sustain complex, multi-step attacks rather than single-point evasions, significantly raising the cost of a successful breach.
3. **Robust Loss Functions:** Moving beyond the standard Mean Squared Error (MSE), the adoption of robust M-estimators like Huber Loss and Tukey’s Biweight Loss fundamentally alters the training objective. These functions reduce the model's sensitivity to outliers in the training data, resulting in a more compact latent manifold that is inherently more resistant to boundary-traversing attacks.

Furthermore, this report establishes a rigorous "Adversarial Evaluation Protocol" for offline NIDS. It emphasizes that standard accuracy metrics are insufficient for assessing robustness. Instead, a comprehensive framework involving perturbation budgets, attack success rates (ASR), transferability scores, and crucial sanity checks for gradient masking is proposed. This ensures that the deployed defenses provide genuine robustness rather than a false sense of security derived from obfuscated gradients.

The synthesis of these findings provides a concrete, actionable roadmap for upgrading the AICyberDefense Phase 2B baseline into a hardened, resilient Phase 3 system capable of withstanding sophisticated adversarial threats in an offline, high-throughput environment.

## 2. The Adversarial Threat Landscape for Unsupervised NIDS

The deployment of unsupervised anomaly detection, particularly using Autoencoders, has become the standard for identifying zero-day threats. By learning a compressed representation of "normal" network behavior, these models can flag deviations without requiring labeled attack data. However, the very mechanism that makes them powerful—learning a manifold of normality—also makes them vulnerable.

### 2.1 Theoretical Foundations of Evasion Attacks

Adversarial examples exploit the discrepancy between the underlying data distribution and the model's learned decision boundary. In the context of an Autoencoder (AE), the anomaly score is typically defined as the reconstruction error $L(x, \hat{x}) = ||x - g(f(x))||^2$, where $f$ is the encoder and $g$ is the decoder. The model flags an input $x$ as anomalous if $L(x, \hat{x}) > \tau$, where $\tau$ is a threshold derived from benign validation data.1

An evasion attack aims to find a perturbation $\delta$ such that a malicious input $x\_{mal}$ is reconstructed with a low error, mimicking benign traffic:

$$\min\_{\delta} ||\delta||\_p \quad \text{s.t.} \quad L(x\_{mal} + \delta, \hat{x}\_{mal+ \delta}) < \tau$$

subject to domain constraints (e.g., valid IP addresses, non-negative packet counts).

Attacks like the **Fast Gradient Sign Method (FGSM)** and **Projected Gradient Descent (PGD)** compute the gradient of the loss with respect to the input, $\nabla\_x L$, to determine the direction in which the reconstruction error decreases most rapidly. By moving $x\_{mal}$ in this direction, the attacker "pushes" the malicious sample toward the manifold of normal data learned by the AE.3

### 2.2 The Unique Challenge of Tabular Network Data

While adversarial research has matured in computer vision, tabular data presents distinct challenges that render many image-based defenses obsolete or ineffective without modification.

#### 2.2.1 Heterogeneity and Constraints

Network traffic data, such as that found in CICIDS2017 or UNSW-NB15, is a mix of continuous, discrete, and categorical features.

* **Continuous Features:** Flow duration, inter-arrival times. These can be perturbed with high granularity, making them the primary surface for gradient-based attacks.
* **Discrete/Count Features:** Number of packets, bytes. These must remain integers. A perturbation that changes "5 packets" to "5.3 packets" is invalid and easily rejected by basic sanitization.
* **Categorical Features:** Protocol (TCP/UDP), Flags. These are often one-hot encoded. Perturbing them requires flipping discrete categories, which is difficult for gradient-based methods that assume a continuous space.5

#### 2.2.2 The Manifold Assumption

Autoencoders rely on the assumption that high-dimensional data lies on a lower-dimensional manifold. For images, this manifold is smooth. For tabular data, the manifold can be disjoint and irregular due to the discrete nature of many features. Adversaries exploit the "empty spaces" off the manifold—regions where the model has not observed data but still provides high-confidence reconstructions due to the generalization properties of neural networks (e.g., ReLU piecewise linearity).7

### 2.3 Recent Trends in NIDS Adversarial Attacks (2023–2025)

Research from 2023 to 2025 highlights a shift from generic white-box attacks to more sophisticated, constraint-aware threats.

* **Problem-Space Attacks:** Attackers are moving from feature-space perturbations (modifying the CSV values directly) to problem-space perturbations (modifying the actual packets to result in the desired feature values). This ensures realizability but is computationally harder.4
* **Black-Box Transferability:** Attackers train surrogate models on similar public datasets (e.g., training on NSL-KDD to attack a model trained on CICIDS2017) to generate transferable adversarial examples, bypassing the need for direct access to the target model's gradients.9
* **Generative Attacks:** The use of GANs (Generative Adversarial Networks) to generate malicious traffic that statistically mimics benign flows is increasing. These "GAN-based" attacks are particularly effective against unsupervised models because they learn to match the distribution of the normal data.11

## 3. Feature-Space Defenses: Squeezing and Discretization

Feature-space defenses operate on the principle of **input transformation**. By reducing the complexity or precision of the input data before it reaches the model, these defenses aim to strip away the adversarial perturbations, which often reside in the high-frequency or high-precision components of the signal. The most prominent technique in this category is **Feature Squeezing**, originally proposed for images but recently adapted for tabular domains.

### 3.1 Feature Squeezing: From Bit-Depth to Precision Reduction

The core concept of feature squeezing is to reduce the search space available to an adversary by coalescing distinct input samples into a single representation. If an adversary relies on a perturbation of $0.0001$ to flip a model's decision, squeezing the input to a precision of $0.01$ eliminates that perturbation.3

#### 3.1.1 Mechanisms for Tabular Data

For image data, squeezing involves reducing color bit depth (e.g., 8-bit to 4-bit) or spatial smoothing (median filtering). For tabular NIDS data, which often consists of 32-bit or 64-bit floating-point numbers (especially after Z-score normalization), the analogue is **Precision Reduction** and **Quantization**.

Precision Reduction (Rounding):

This is the simplest form of squeezing for tabular data. It involves rounding the continuous feature values to a specific number of decimal places or significant digits.

$$x\_{squeezed} = \text{round}(x \cdot 10^k) / 10^k$$

where $k$ is the precision parameter. For Z-score normalized data (typically in the range $[-3, 3]$), rounding to 1 or 2 decimal places effectively quantizes the input space into a discrete grid.

* **Insight:** Research indicates that many NIDS features, such as "Flow Duration" or "Bytes/s", have excessive precision that contributes little to classification accuracy but provides ample room for adversarial gradients. Reducing this precision removes "non-robust" features that are highly sensitive to noise.6

Scalar Quantization (Binning):

A more aggressive approach involves mapping continuous values into a fixed number of bins. This transforms the continuous input space into a discrete set of indices or centroids.

$$x\_{quantized} = \text{argmin}\_{c \in C} |x - c|$$

where $C$ is a codebook of centroids (e.g., derived from K-Means clustering of benign data) or a uniform grid.

* **Defense Mechanism:** This process is non-differentiable. While gradients can be approximated (BPDA), the "staircase" nature of the loss landscape makes gradient descent inefficient. An attacker must push the sample across a significant threshold (bin boundary) to effect any change in the model's output, requiring a much larger perturbation budget ($\epsilon$) that is easier to detect.16

#### 3.1.2 Implementation Considerations for AICyberDefense

Given the **AICyberDefense Phase 2B** architecture (Autoencoder with Z-score inputs), implementing feature squeezing requires careful calibration of the quantization levels.

* **Step Size:** If the quantization step is too large, information vital for detecting subtle anomalies (like slowloris attacks) might be lost (under-fitting). If too small, it fails to strip adversarial noise.
* **Adaptive Quantization:** Recent literature suggests "adaptive quantization," where the step size varies based on the feature's density. For heavy-tailed distributions common in network traffic (e.g., flow size), logarithmic quantization or quantile-based binning preserves information in the dense regions while aggressively squeezing the tails.16

### 3.2 Detection via Squeezing: The "Squeeze-and-Check" Framework

Beyond passively hardening the model, feature squeezing enables a powerful active detection mechanism known as "Squeeze-and-Check" or the **Joint Detection Framework**. This approach leverages the fact that adversarial examples are often fragile and highly specific to the exact input values.19

**Protocol:**

1. **Original Inference:** The NIDS processes the raw input $x$ and generates an anomaly score $S\_{orig} = Error(x)$.
2. **Squeezed Inference:** The NIDS processes the squeezed input $x\_{sq}$ and generates a score $S\_{sq} = Error(x\_{sq})$.
3. **Comparison:** The system calculates the discrepancy $D = |S\_{orig} - S\_{sq}|$.

**Logic:**

* **Benign Samples:** For normal traffic, the input $x$ is robust. Squeezing it results in a semantically similar input $x\_{sq}$, so the model's output should change minimally ($D \approx 0$).
* **Adversarial Samples:** Adversarial perturbations are optimized to minimize $S\_{orig}$. However, the squeezing operation $x \to x\_{sq}$ destroys the fine-tuned perturbation structure. Consequently, $S\_{sq}$ will likely revert to a high value (reflecting the true anomalous nature of the input) or at least differ significantly from $S\_{orig}$.
* **Alerting:** If $D > \text{threshold}$, the input is flagged as adversarial, regardless of the absolute anomaly score.

**Empirical Evidence:** Studies on datasets like MNIST and CIFAR-10 have shown detection rates of over 95% using this method. For NIDS, applying this to tabular data has shown promise in detecting PGD attacks, particularly when combined with ensemble squeezers (e.g., one rounding squeezer and one bit-depth squeezer).3

### 3.3 Challenges: Gradient Masking and BPDA

A critical criticism of feature squeezing is that it may rely on **gradient masking**. Because functions like round() or floor() have zero gradient almost everywhere (and undefined gradients at jumps), naive white-box attacks fail because $\nabla\_x L = 0$. This creates a false sense of security.21

The BPDA Counter-Attack:

Sophisticated attackers use Backward Pass Differentiable Approximation (BPDA). They replace the non-differentiable squeezing layer $g(x)$ with a differentiable approximation $f(x)$ (e.g., $f(x)=x$) during the backward pass to estimate gradients.

$$\nabla\_x L \approx \nabla\_{f(x)} L \cdot \frac{\partial f(x)}{\partial x}$$

This allows them to generate perturbations that survive the squeezing process.

Counter-Counter Measures:

To mitigate BPDA, defenders must ensure that the squeezing creates a "true" information bottleneck that cannot be easily bypassed.

* **Randomized Smoothing:** Adding random noise before squeezing makes the defense stochastic, turning the classifier into a "smoothed" function that provides certified robustness guarantees. This prevents the attacker from finding a single stable direction for perturbation.23
* **Complex Transformation:** Using data-dependent quantization (e.g., K-Means centroids based on benign data) makes the transformation harder to approximate with a simple linear function.18

### 3.4 Summary of Feature-Space Recommendations

For the offline AICyberDefense system, **Input Discretization** via rounding (e.g., to 2 decimal places on Z-scored data) combined with the **Squeeze-and-Check** detection framework is highly recommended. The computational cost of running a second inference pass is negligible in an offline batch processing context, while the security gain against evasion is substantial.

## 4. Inference-Time Hardening Strategies

Inference-time hardening focuses on techniques that improve robustness without altering the underlying model training process. These methods exploit the operational context of NIDS: network traffic is not a set of independent vectors but a temporal stream of correlated events. Adversaries typically treat NIDS evasion as a single-point optimization problem, ignoring the temporal consistency required to evade detection over time.

### 4.1 Test-Time Augmentation (TTANAD)

Test-Time Augmentation (TTA) is a technique borrowed from computer vision, where multiple augmented versions of a test image (crops, flips) are fed to the model, and predictions are averaged. For NIDS, spatial augmentations are invalid, but **temporal augmentations** are highly effective.

**TTANAD (Test-Time Augmentation for Network Anomaly Detection)** generates multiple "views" of a network flow by altering the temporal context or aggregation parameters.25

#### 4.1.1 Implementation for Tabular Flows

Network flows are typically aggregated over a time window (e.g., statistics calculated over the last 60 seconds). TTANAD exploits this aggregation process.

* **Augmentation Strategy:** At inference time, for a given flow $x\_t$, the system generates $K$ variations by slightly perturbing the aggregation window size $W$ or the step size $S$.
  + $x\_{t,0}$: Features computed over window $W$.
  + $x\_{t,1}$: Features computed over window $W - \delta$.
  + $x\_{t,2}$: Features computed over window $W + \delta$.
* Ensemble Scoring: The Autoencoder processes all $K+1$ variants. The final anomaly score is an aggregation (mean, median, or max) of the individual reconstruction errors.  
    
  $$Score(x\_t) = \text{Agg}(\{Error(x\_{t,k}) \mid k=0 \dots K\})$$

**Insight:** Adversarial examples are "brittle." A perturbation $\delta$ optimized to minimize error for the specific feature values in $x\_{t,0}$ will likely fail to minimize error for $x\_{t,1}$, where the features (e.g., "Mean Packet Rate") have shifted slightly due to the window change. Averaging the scores smooths out the adversarial "dip" in the error landscape, revealing the true anomaly.26

### 4.2 Temporal Score Smoothing: EWMA

Attackers often rely on "transient evasion"—crafting a short burst of malicious packets that slip through the NIDS. However, real network attacks (DoS, scanning, exfiltration) typically persist over time. **Exponentially Weighted Moving Average (EWMA)** leverages this persistence to filter out noise and adversarial jitter.27

#### 4.2.1 The EWMA Mechanism

Instead of evaluating each flow $x\_t$ in isolation, the NIDS maintains a stateful "smoothed score" $Z\_t$.

$$Z\_t = \lambda \cdot S\_t + (1 - \lambda) \cdot Z\_{t-1}$$

where:

* $S\_t$ is the instantaneous reconstruction error of the current flow.
* $Z\_{t-1}$ is the smoothed score from the previous step.
* $\lambda$ is the smoothing factor ($0 < \lambda \le 1$).

**Defensive Properties:**

* **Noise Reduction:** Small $\lambda$ values (e.g., 0.1–0.3) suppress high-frequency noise, reducing false positives from benign bursts.
* **Attack Persistence:** To evade an EWMA-based detector, an attacker cannot simply craft *one* adversarial packet. They must craft a *sequence* of adversarial packets $x\_t, x\_{t+1}, \dots, x\_{t+n}$ to keep the moving average $Z\_t$ below the threshold. This turns the attack from a "one-shot" evasion into a "continuous control" problem, exponentially increasing the difficulty and likelihood of detection.30

#### 4.2.2 Dynamic Thresholding with EWMA

Static thresholds are vulnerable to replay attacks and drift. Combining EWMA with Dynamic Thresholding creates a robust adaptive system. The threshold $\tau\_t$ adapts based on the historical statistics of the reconstruction error.

$$\tau\_t = \mu\_t + k \cdot \sigma\_t$$

where $\mu\_t$ and $\sigma\_t$ are the moving average and moving standard deviation of the error.

* **Benefit:** During a sustained attack, even if the adversary manages to lower the reconstruction error slightly, the sudden change in statistical variance $\sigma\_t$ can trigger detection. This forces the attacker to strictly conform to the *variance* of benign traffic, not just the mean, further constraining their attack space.32

### 4.3 Offline Ensemble Strategies

In an offline setting, computational latency is less critical, enabling the use of **Heterogeneous Ensembles**.

* **Concept:** Deploy multiple Autoencoders with different architectures (e.g., one Dense AE, one LSTM-AE, one CNN-AE) or different feature subsets.
* **Robustness:** Adversarial examples typically transfer poorly between different architectures if the gradients are sufficiently distinct. An input optimized to fool a Dense AE may fail to fool an LSTM-AE which looks at temporal sequences.
* **Consensus Voting:** An alarm is raised if *any* model in the ensemble detects an anomaly (OR-voting) or if the majority do (Majority-voting). Research shows that heterogeneous ensembles significantly reduce the success rate of transfer attacks.34

## 5. Robust Loss Functions for Autoencoders

The training phase is the foundation of NIDS robustness. The standard loss function for Autoencoders, **Mean Squared Error (MSE)**, is mathematically optimal for data with Gaussian noise but is inherently flawed for anomaly detection training, especially in the presence of outliers or "poisoned" data.

### 5.1 The Problem with MSE

MSE is defined as $L = \frac{1}{N} \sum (x\_i - \hat{x}\_i)^2$. The quadratic penalty means that a single outlier with a large reconstruction error dominates the gradient update.

* **Consequence 1 (Manifold Expansion):** During training, the AE tries aggressively to reconstruct outliers (which might be noise or rare benign events) to minimize the massive squared error. This "stretches" the learned latent manifold to encompass these outliers. A wider manifold includes more "empty space," giving adversaries more room to craft perturbations that lie close to the manifold (low error) but are actually malicious.7
* **Consequence 2 (Gradient Explosion):** Large errors produce large gradients ($\nabla L \propto error$). This makes the model sensitive to initial conditions and adversarial perturbations during the backward pass of an attack generation.7

### 5.2 Huber Loss: The Balanced Approach

Huber Loss is a piecewise function that combines the properties of MSE and Mean Absolute Error (MAE).

$$L\_\delta(a) = \begin{cases} \frac{1}{2}a^2 & \text{for } |a| \le \delta \\ \delta (|a| - \frac{1}{2}\delta) & \text{otherwise} \end{cases}$$

where $a = x - \hat{x}$ is the residual.

**Mechanics of Robustness:**

* **Inliers (Residual $\le \delta$):** For the bulk of the data (normal traffic), Huber acts like MSE, providing smooth, differentiable gradients that allow the model to learn fine-grained correlations.
* **Outliers (Residual $> \delta$):** For outliers, Huber acts like MAE (linear penalty). The gradient magnitude is capped at $\delta$. This prevents the model from "over-focusing" on reconstructing outliers.
* **Result:** The learned manifold remains compact and focused on the dense core of benign data. Outliers are *not* well-reconstructed. This is desirable for anomaly detection: we *want* anomalies (and attacks) to have high error. By not forcing the model to learn outliers during training, we ensure they remain anomalous during testing.36

Impact on Adversarial Attacks:

A compact manifold implies a sharper decision boundary (in terms of reconstruction error gradient). An adversary using PGD to minimize error faces a landscape where the gradients in the "anomalous" regions are constant (linear loss) rather than increasing (quadratic). This can slow down or destabilize the optimization process of the attack algorithm, requiring more iterations to find a successful perturbation.38

### 5.3 Tukey’s Biweight Loss: Aggressive Outlier Suppression

Tukey’s Biweight (Bisquare) Loss takes robustness a step further. It completely suppresses the influence of data points with errors beyond a cutoff $c$.

$$\rho(r) = \begin{cases} \frac{c^2}{6} \left( 1 - \left( 1 - (r/c)^2 \right)^3 \right) & \text{if } |r| \le c \\ \frac{c^2}{6} & \text{if } |r| > c \end{cases}$$

* **Mechanism:** The influence function (derivative of the loss) actually descends to zero for large errors. This means extreme outliers have *no* impact on the gradient update.
* **Pros/Cons:** While theoretically superior for rejecting poisoning attacks, Tukey loss is **non-convex**, making optimization difficult. It requires careful initialization (e.g., pre-training with MSE) to avoid getting stuck in local minima. For most NIDS applications, Huber offers a better trade-off between stability and robustness.40

### 5.4 Comparative Evaluation: MSE vs. Huber vs. Tukey

| **Feature** | **MSE Loss** | **Huber Loss** | **Tukey Biweight** |
| --- | --- | --- | --- |
| **Outlier Sensitivity** | High (Quadratic) | Moderate (Linear) | Zero (for extreme outliers) |
| **Latent Manifold** | Broad, loose | Compact, dense | Very compact |
| **Optimization** | Convex, Stable | Convex, Stable | Non-Convex, Unstable |
| **Gradient Behavior** | Unbounded | Bounded ($\pm \delta$) | Redescending to 0 |
| **Adversarial Impact** | Easy to traverse | Harder to traverse | Hardest (requires annealing) |
| **Recommended Use** | Baseline Training | **Primary Defense** | Advanced/Poisoning Defense |

**Recommendation:** For AICyberDefense Phase 3, **Huber Loss** is the optimal choice. It provides immediate robustness gains with minimal implementation complexity and high training stability. The parameter $\delta$ should be tuned (e.g., set to the 90th percentile of reconstruction errors) to maximize the separation between benign clusters and anomalous space.7

## 6. Offline Deployment and Evaluation Protocols

The transition to an offline deployment model (processing PCAP/logs in batches) allows for the implementation of computationally intensive defenses that would be infeasible in real-time inline systems. This section outlines the deployment architecture and the rigorous protocols required to validate robustness.

### 6.1 Offline Deployment Architecture

The offline NIDS operates on batches of network flow data (e.g., 5-minute windows). This latency budget allows for "Ensemble Squeezing" and "Multi-View Analysis."

**Proposed Pipeline:**

1. **Ingestion:** Raw PCAP $\to$ Flow Extractor (e.g., CICFlowMeter).
2. **Preprocessing:**
   * **Z-Score Normalization:** Standard scaling.
   * **Defense Layer 1 (Squeezing):** Parallel generation of $x\_{raw}$ and $x\_{sq}$ (rounded to 2 decimals).
3. **Inference Engine:**
   * **Model:** Autoencoder trained with **Huber Loss**.
   * **Defense Layer 2 (TTANAD):** Generate 5 temporal augmentations for each flow.
   * **Scoring:** Compute reconstruction error $E$.
4. **Post-Processing:**
   * **Defense Layer 3 (Smoothing):** Update EWMA score $Z\_t$.
   * **Detection Logic:**
     + Check 1: $Z\_t > \tau\_{dynamic}$?
     + Check 2: $|E(x\_{raw}) - E(x\_{sq})| > \tau\_{detect}$?
5. **Reporting:** Alert generation with forensic metadata (which flows triggered the alarm, adversarial probability).

### 6.2 Adversarial Evaluation Protocol

Standard metrics (Accuracy, F1-Score) are meaningless in AML because they do not account for an active adversary. The evaluation must measure the system's **degradation** under attack.9

#### 6.2.1 Key Metrics

1. Attack Success Rate (ASR): The percentage of malicious samples that are misclassified as benign after perturbation.  
     
   $$ASR(\epsilon) = \frac{\text{Count}(L(x\_{adv}) < \tau)}{N\_{attacks}}$$  
     
   This must be plotted as a curve against the perturbation budget $\epsilon$.
2. **Perturbation Budget ($\epsilon$):** For tabular data, $\epsilon$ is typically defined in terms of $L\_\infty$ (max change per feature) or $L\_2$ (Euclidean distance). A robust model should force the attacker to use a large $\epsilon$ (making the attack detectable or invalid) to achieve success.
3. **Clean Performance Retention:** The drop in F1-score on benign data due to hardening defenses (e.g., squeezing). A good defense maintains high clean accuracy while lowering ASR.1
4. **Transferability Score:** Train an attack on a surrogate model (standard AE) and test it against the hardened model. This simulates a realistic "Black-Box" threat scenario.10

#### 6.2.2 Sanity Checks for Gradient Masking

To ensure the defense is genuinely robust and not just obfuscating gradients, the following "Sanity Checks" (adapted from Athalye et al. for NIDS) are mandatory 21:

* **Check 1 (Iterative vs. Single-Step):** Verify that iterative attacks (PGD) are *more* successful than single-step attacks (FGSM). If FGSM > PGD, the defense is likely relying on gradient masking (broken gradients prevent PGD from optimizing).
* **Check 2 (Black-Box Proxy):** Verify that white-box attacks (using the target model's gradients) are *more* successful than black-box attacks (using a surrogate). If Black-Box > White-Box, the target model's gradients are useless (masked), and the defense is fragile.
* **Check 3 (Unbounded Attack):** With an infinite budget ($\epsilon \to \infty$), ASR should reach 100%. If not, the attack implementation is flawed.

#### 6.2.3 Data Feasibility Check

For tabular NIDS, evaluation must include a **Realizability Check**. Adversarial examples generated by algorithms must be checked against domain constraints:

* Are integer features integers?
* Are non-negative features $\ge 0$?
* Do the protocol flags make sense (e.g., can't be TCP and UDP simultaneously)?  
  Invalid examples should be discarded from the ASR calculation to provide a realistic assessment of risk.5

## 7. Conclusion and Implementation Roadmap

This deep research survey confirms that while Autoencoder-based NIDS are susceptible to adversarial evasion, a layered defense strategy specifically adapted for tabular data can significantly harden them. The synergistic combination of **Feature Squeezing (via Precision Reduction)**, **Inference-Time Smoothing (EWMA)**, and **Robust Training (Huber Loss)** creates a defense-in-depth architecture that addresses vulnerabilities at the input, model, and output stages.

### 7.1 Key Takeaways

1. **Tabular Squeezing:** Unlike image squeezing, tabular squeezing relies on precision reduction and quantization. It serves as both a gradient blocker and an active detection mechanism (Squeeze-and-Check).
2. **Temporal Consistency:** Offline deployment enables the use of temporal smoothing (EWMA/TTANAD), which forces attackers to solve a complex sequence generation problem rather than a simple point-wise optimization.
3. **Huber Loss:** Replacing MSE with Huber Loss is a high-ROI change. It stabilizes training against noise and compacts the latent manifold, implicitly hardening the model against boundary-traversing attacks.
4. **Rigorous Evaluation:** True robustness can only be verified through adaptive attacks and gradient masking sanity checks, not just standard accuracy metrics.

### 7.2 Table 1: Strategic Roadmap for AICyberDefense Phase 3

| **Phase** | **Action Item** | **Technique / Mechanism** | **Implementation Note** |
| --- | --- | --- | --- |
| **3A** | **Robust Retraining** | **Huber Loss** (delta=1.0) | Replace MSELoss. Tune delta to 90th percentile of error. |
| **3B** | **Input Hardening** | **Feature Quantization** | Round features to 2 decimals. Add Quantization layer to AE. |
| **3C** | **Inference Logic** | **Squeeze-and-Check** | Run dual inference ($x$ vs $x\_{sq}$). Alert on discrepancy. |
| **3D** | **Score Smoothing** | **EWMA + Dynamic Threshold** | Implement stateful scoring: $Z\_t = 0.2 S\_t + 0.8 Z\_{t-1}$. |
| **3E** | **Evaluation** | **PGD + Sanity Checks** | Verify no gradient masking. Measure ASR vs $\epsilon$. |

By executing this roadmap, the AICyberDefense system will evolve from a standard anomaly detector into a resilient, adversarial-aware cybersecurity platform capable of operating effectively in hostile environments.