# CWE Detail – CWE-1039

## Description

The product uses an automated mechanism such as machine learning to recognize complex data inputs (e.g. image or audio) as a particular concept or category, but it does not properly detect or handle inputs that have been modified or constructed in a way that causes the mechanism to detect a different, incorrect concept.

## Extended Description

When techniques such as machine learning are used to automatically classify input streams, and those classifications are used for security-critical decisions, then any mistake in classification can introduce a vulnerability that allows attackers to cause the product to make the wrong security decision or disrupt service of the automated mechanism. If the mechanism is not developed or "trained" with enough input data or has not adequately undergone test and evaluation, then attackers may be able to craft malicious inputs that intentionally trigger the incorrect classification. Targeted technologies include, but are not necessarily limited to: automated speech recognition automated image recognition automated cyber defense Chatbot, LLMs, generative AI For example, an attacker might modify road signs or road surface markings to trick autonomous vehicles into misreading the sign/marking and performing a dangerous action. Another example includes an attacker that crafts highly specific and complex prompts to "jailbreak" a chatbot to bypass safety or privacy mechanisms, better known as prompt injection attacks.

## Threat-Mapped Scoring

Score: 0.0

Priority: Unclassified

## Modes of Introduction

**•** Architecture and Design: This issue can be introduced into the automated algorithm itself due to inadequate training data used as well as lack of validation, verification, testing, and evaluation of the algorithm. These factors can affect the overall robustness of the algorithm when introduced into operational settings.

**•** Implementation: The developer might not apply external validation of inputs into the algorithm.

## Common Consequences

**•** Impact: Bypass Protection Mechanism — Notes: When the automated recognition is used in a protection mechanism, an attacker may be able to craft inputs that are misinterpreted in a way that grants excess privileges.

**•** Impact: DoS: Resource Consumption (Other), DoS: Instability — Notes: There could be disruption to the service of the automated recognition system, which could cause further downstream failures of the software.

**•** Impact: Read Application Data — Notes: This weakness could lead to breaches of data privacy through exposing features of the training data, e.g., by using membership inference attacks or prompt injection attacks.

**•** Impact: Varies by Context — Notes: The consequences depend on how the application applies or integrates the affected algorithm.

## Potential Mitigations

**•** Architecture and Design: Algorithmic modifications such as model pruning or compression can help mitigate this weakness. Model pruning ensures that only weights that are most relevant to the task are used in the inference of incoming data and has shown resilience to adversarial perturbed data. (Effectiveness: N/A)

**•** Architecture and Design: Consider implementing adversarial training, a method that introduces adversarial examples into the training data to promote robustness of algorithm at inference time. (Effectiveness: N/A)

**•** Architecture and Design: Consider implementing model hardening to fortify the internal structure of the algorithm, including techniques such as regularization and optimization to desensitize algorithms to minor input perturbations and/or changes. (Effectiveness: N/A)

**•** Implementation: Consider implementing multiple models or using model ensembling techniques to improve robustness of individual model weaknesses against adversarial input perturbations. (Effectiveness: N/A)

**•** Implementation: Incorporate uncertainty estimations into the algorithm that trigger human intervention or secondary/fallback software when reached. This could be when inference predictions and confidence scores are abnormally high/low comparative to expected model performance. (Effectiveness: N/A)

**•** Integration: Reactive defenses such as input sanitization, defensive distillation, and input transformations can all be implemented before input data reaches the algorithm for inference. (Effectiveness: N/A)

**•** Integration: Consider reducing the output granularity of the inference/prediction such that attackers cannot gain additional information due to leakage in order to craft adversarially perturbed data. (Effectiveness: N/A)

## Applicable Platforms

**•** None (Class: Not Language-Specific, Prevalence: Undetermined)

## Notes

**•** Relationship: Further investigation is needed to determine if better relationships exist or if additional organizational entries need to be created. For example, this issue might be better related to "recognition of input as an incorrect type," which might place it as a sibling of CWE-704 (incorrect type conversion).