SOFTWARE TESTING ISSUES USING AI

**Problem statement :**

The core problem is the "black box" nature and continuous evolution of AI models. This makes it incredibly difficult to ensure an AI system is reliable, fair, and safe. Unlike traditional software with a defined set of rules and expected outcomes, an AI's behavior is shaped by its training data and can evolve dynamically. This creates several unique testing challenges: Non-deterministic Outcomes: Given the exact same input, a traditional software program will always produce the same output. An AI system, however, can produce different outputs for the same input under different conditions (e.g., after being retrained on new data). This makes it nearly impossible to create predictable, repeatable test cases. Lack of Explain ability: Many AI models, particularly deep learning networks, operate as "black boxes" where it's not clear why a specific decision was made. This opacity makes it hard to debug when a failure occurs or to prove that the system is unbiased and fair. Data Dependency and Bias: The quality of an AI system is directly tied to the quality of its training data. If the data is biased or incomplete, the AI will learn and perpetuate those biases, leading to unfair or incorrect decisions. Testing for all possible biases and edge cases is an immense challenge. For example, a facial recognition AI trained primarily on data of light- skinned individuals may fail to accurately identify people with darker skin tones. Drift and Decay: The real-world environment an AI operates in can change over time. An AI model that performs well initially might see its performance degrade as the data it encounters shifts. This "model drift" requires continuous, post-deployment testing, a concept not common in traditional software testing.