# Spam Drift vs. Ham Drift in AI Classifiers

This write-up demonstrates how small shifts in training data can drastically affect a model’s behavior. It showcases two experiments:

1. **Spam Drift**: The model slowly forgets what legitimate (“ham”) messages look like.
2. **Ham Drift**: The model forgets what spam looks like and starts trusting everything.

These examples simulate **model decay** or **concept drift** — a major concern in cybersecurity AI systems like spam filters, phishing detectors, and behavioral classifiers.

## 1. Experiment Setup

### Model:

* **Multinomial Naive Bayes** text classifier
* Trained on synthetic spam vs. ham messages

### Tools:

* Python (scikit-learn, matplotlib)
* Jupyter Notebook (Ubuntu VM)

### Dataset:

* 150 ham messages (e.g., “Let’s meet for lunch”)
* 150 spam messages (e.g., “Win $$$ now!”)
* 9 adversarial **poisoned ham** messages labeled as ham but written like spam (e.g., “Claim your free gift now!”)

## 2. Spam Drift: Forgetting Ham

### Mechanism:

* Gradually increase the amount of spam in the training set
* Reduce the number of ham messages per cycle
* Retrain model 10 times with progressively skewed datasets

### Outcome:

* Around **Cycle 5**, accuracy drops sharply
* By **Cycle 6–7**, the model misclassifies even obvious ham as spam

### Use Case Failure:

Spam filters start flagging legitimate user emails

## 3. Ham Drift: Forgetting Spam

### Mechanism:

* Increase number of ham messages instead
* Slowly reduce number of spam messages per cycle

### Outcome:

* Around **Cycle 4–5**, the model starts labeling actual spam as ham
* By **Cycle 6+,** even poisoned messages like “Claim your free gift now!” are treated as safe

### Use Case Failure:

Phishing emails bypass filters as spam detection erodes

## 4. Visualization

Each experiment includes: - Accuracy vs. Training Skew Line Graph - Drift threshold markers (10% drop from baseline) - Cycle-by-cycle prediction printouts for: - Real ham - Poisoned ham - Real spam

## 5. Why This Matters in Cybersecurity

| Threat | AI Vulnerability |
| --- | --- |
| Poisoned data injection | Model trust decay |
| Skewed learning feedback | Drift toward false positives/negatives |
| Lack of validation sets | Silent accuracy collapse |

### Real-World Analogues:

* Spam filters no longer detecting attacks
* IDS/IPS systems ignoring new threats
* LLMs hallucinating responses due to over-reinforced patterns

## 6. Lessons Learned

* ✅ Monitor accuracy across retraining
* ⚠️ Detect drift early with accuracy thresholds or KL divergence
* 🧪 Test with fixed validation data to identify bias
* 🔒 Never trust feedback loops without adversarial validation

## 7. GitHub Integration Ideas

* 📁 /notebooks/spam\_drift.ipynb
* 📁 /notebooks/ham\_drift.ipynb
* 📄 README.md with explanation + chart screenshots
* 🛡️ Bonus: Drift detection and recovery code snippets

## 8. Next Steps

Would you like to: - Turn this into a full markdown GitHub page? - Create a PDF cheat sheet to hand to instructors? - Record a screen demo to showcase the behavior live?

You’re now simulating adversarial and drift-resilient AI — a key skill in security engineering and red team data science.

*Authored with ChatGPT and Enrique Becwar, 2025.*