Not Just a Label: Turning DGA Detection into a Playbook

Serpil Rivas

Professor Mallarapu

SEAS_8414_DC8: Analytical Tools for Cyber Analytics

August 20, 2025

Abstract

Security teams are facing a growing challenge not only with the challenge of developing accurate threat detection scores, but with developing actionable incident response plays from their scores. This project looks at the "black box" problem related to Domain Generation Algorithm (DGA) detection. The project developed an end-to-end pipeline that leverages AutoML classification, explainable AI (XAI), and generative AI (GenAI) to produce prescriptive security playbooks. Using H2O AutoML, we trained a well-performing DGA classifier with features that humans can read and interpret (domain length and entropy) and then later used local SHAP explanations to help provide an understanding for an individual prediction. The innovation here was to create a programmatic bridge from XAI to GenAI by converting numerical SHAP values into structured text descriptions that serve as input to large language models to create incident response plans that are contextual. We confirmed that the system would provide not simply classifications using legitimate domains (e.g. google.com) and deltas from other domains that were DGA- like in nature, but also a rationale why and proposed next steps describing actions "what to do next". This has the benefit of not only using time for actionable plays for the analyst, while not compromising audit trails and trust with transparent decision-making audit trails.

Table of Content

Not Just a Label:
Turning DGA Detection into a Playbook

1. Introduction

Security teams rarely struggle to get **a score** anymore; the challenge is turning a score into **a decision and an action**. Accurate but opaque models create a “black-box” dilemma: fast signals with little justification. In incident response, that gap slows triage and undermines trust.

This project builds an end-to-end pipeline that closes that gap for **Domain Generation Algorithm (DGA)** detection. It trains a competitive classifier with AutoML, explains each prediction with local **XAI**, and then converts those explanations into a **prescriptive playbook** tailored to the alert. The outcome is not just “dga vs legit,” it’s “why” and “what to do next.”

---

2. Architecture at a Glance

```
Domain → Feature Engineering → AutoML (H2O) → Prediction
                                ↓                    ↘
                          SHAP (local)          XAI→GenAI Bridge → Incident Playbook
```

- **Features**: `length`, `entropy`
- **Model**: H2O AutoML leader exported as a MOJO (`model/DGA_Leader.zip`)
- **XAI**: Local SHAP on the specific input
- **Bridge**: Programmatically summarizes SHAP into `xai_findings`
- **GenAI**: Prompts an LLM to produce a context-aware response plan

---

3. Methodology & Implementation Highlights

1) Training and Export (AutoML → MOJO)

The training script runs H2O AutoML, and evaluates many candidates, as well as it exports the leader as a MOJO for fast and easy inference with reliable predictions. The artifacts are:

- `data/dga_dataset_train.csv`
- `model/DGA_Leader.zip`
- `model/leaderboard.csv`

Exporting a MOJO keeps inference light and portable while preserving the model's learned logic.

2) Single-Domain Inference + Features

The CLI accepts a domain and derives two robust, interpretable features linked to DGA behavior:

- **Length**: DGA domains skew longer or mechanically fixed; legit domains cluster in practical ranges.
- **Entropy**: DGA domains tend toward high character randomness; human-chosen names show lower entropy.

The MOJO predicts `legit` or `dga` with a probability.

3) Local Explainability with SHAP

For dga predictions, the pipeline computes **local SHAP values** on the single input. SHAP attributes how much each feature pushes the prediction toward one class or the other. This step makes the classification **auditable,** analysts see which signals mattered most for this domain.

4) The XAI → GenAI Bridge (the critical piece)

The bridge turns raw numbers into a **compact, analyst-ready summary**. It:

- Sorts features by absolute SHAP contribution.
- Labels each contribution direction (e.g., “towards dga” vs “towards legit”).
- Rounds values sensibly and adds brief, domain-specific context.
- Emits a structured text block `xai_findings` that the LLM can reliably follow.

Example (excerpt):

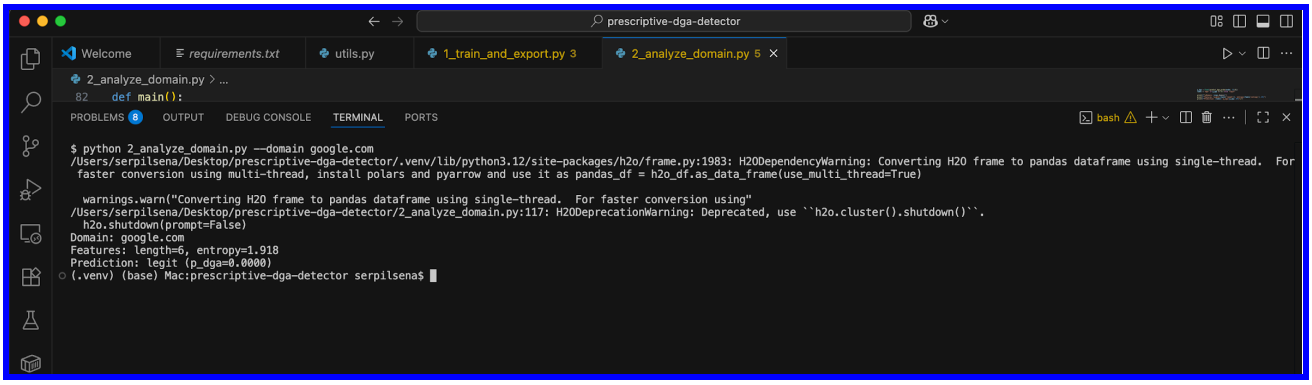
```
 Alert: Potential DGA domain detected.
- Domain: 'kq3v9z7j1x5f8g2h.info'
- AI Model Explanation (from SHAP): 59.65% confidence for 'dga'.
  Contributions:
  - entropy=4.000 (SHAP +0.246, towards 'dga')
  - length=16 (SHAP -0.013, towards 'legit')
```

Feeding this to GenAI produces a **prescriptive** plan: specific searches, containment steps, and validation checks tied back to the SHAP rationale.

---

4. Results & Discussion

Figure 1 — Legitimate Domain (Pipeline Stops Early)

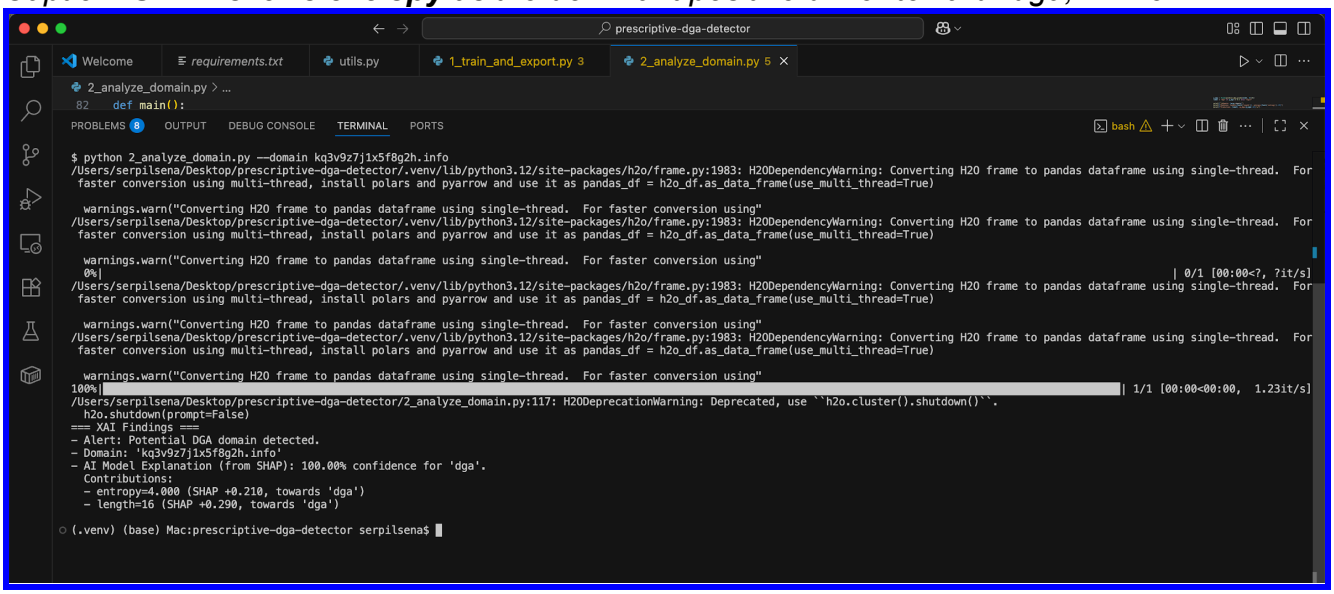


*Caption: Legitimate domain (google.com) returns **legit** with very low DGA probability, so the pipeline ends without XAI/GenAI overhead.*

Why it matters:
The system deliberately avoids over-processing safe cases. That keeps the workflow fast for analysts and reduces alert fatigue. The underlying features (shorter length, lower entropy) align with human expectations for well-known domains.

Figure 2 — Local XAI for a DGA-Like Domain

*Caption: SHAP shows **entropy** as the dominant positive driver toward "dga," while*



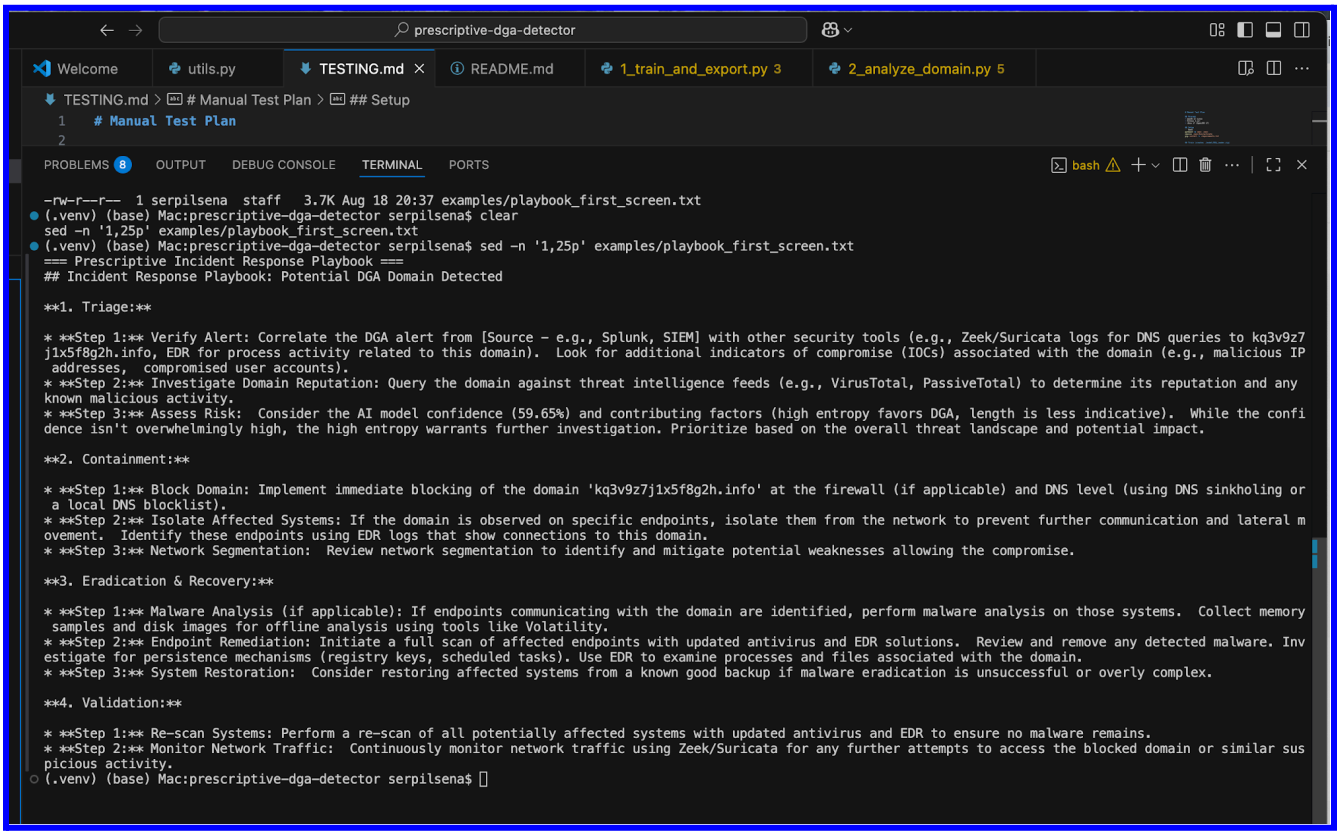
***length** offers a weak counter-signal toward "legit."*

Why it matters:
DGAs often produce strings with high randomness; **entropy=4.000** strongly nudges the model toward DGA (SHAP **+0.246**). The **length=16** slightly dampens that signal (SHAP

–0.013), reminding the analyst that not all signals point in the same direction. This nuance is essential—alerts carry context, not just a single score.

Analyst takeaway:
Treat the classification as *probable* DGA due primarily to randomness. Length is not exculpatory on its own; it simply tempers confidence. This aligns with practical triage: prioritize hosts that contacted this domain, but confirm with logs before containment.

Figure 3 — From Explanations to a Playbook



*Caption: The XAI summary powers a **prescriptive playbook** with searches (Splunk/Zeek/EDR), containment steps, eradication guidance, and validation criteria.*

Why it matters:
Prescriptive output narrows the time from alert to action. The playbook references concrete entities from the case (the domain, confidence, and key signals), providing:

- **Triage queries** (e.g., DNS/HTTP traces, EDR process trees),

- **Containment** (network and endpoint blocking),
- **Eradication & recovery** (malware handling and restoration),
- **Validation** (confirming the absence of recurrence),
- **Lessons learned** (tuning DGA defenses and integrating intel).

This is precisely where traditional “black-box” models fall short; the bridge adds the missing operational connective tissue.

---

5. Limitations, Risks, and How to Operate Safely

- **Model drift:** DGA families evolve. Retrain on fresh data and track performance over time. A simple guardrail is scheduled retraining plus a canary set; a stronger control includes **data drift** checks on feature distributions (length/entropy) and **prediction drift** (class probabilities).
- **Bias and blind spots:** Two features are intentionally simple and interpretable, but that also limits coverage. Add features cautiously (e.g., vowel ratio, n-gram patterns) and re-validate explanations so the bridge remains readable.
- **Confidence is not certainty:** In the example, ~60% DGA probability is a triage trigger, not a conviction. This is reflected in the generated playbook’s call to corroborate across logs before containment.
- **GenAI robustness:** The prompt is structured, deterministic, and short on fluff to reduce variability. Maintain the schema (`Alert/Domain/Explanation/Contributions`) and log the prompts/outputs for auditing.
- **Operational hygiene:** Keep secrets out of source. Use an environment variable (`GOOGLE_API_KEY`) locally or repository secrets in CI/CD. Avoid printing keys or storing them in files committed to Git.

---

6. How to Reproduce

Prereqs: Python 3.12+, Java 17 (for H2O), a fresh virtualenv with `pip install -r prescriptive-dga-detector/requirements.txt`, and `export GOOGLE_API_KEY="<YOUR-KEY-HERE>"`. Do **not** commit keys to Git.

```
# Clone and enter
git clone https://github.com/SerpilRivas/seas8405-homework9.git
cd seas8405-homework9/prescriptive-dga-detector

# Environment
python -m venv .venv && source .venv/bin/activate
pip install -r requirements.txt
export GOOGLE_API_KEY="<YOUR-KEY-HERE>"

# Train once (exports MOJO to model/)
python 1_train_and_export.py

# Legit example (no playbook)
python 2_analyze_domain.py --domain google.com | tee examples/google_run.txt

# DGA-like example (with SHAP + Playbook)
python 2_analyze_domain.py --domain kq3v9z7j1x5f8g2h.info | tee examples/dga_run.txt
```

Artifacts and demo screenshots are also available under the project's GitHub Release.

---

7. What Makes This Different

- **Speed with justification:** AutoML ships a strong model quickly. SHAP explains the "why." The bridge turns that into repeatable analyst action—*not just a label, but a plan*.
- **Human-centered output:** Explanations are localized to the case at hand and written in the language analysts use during triage.
- **Operational fit:** Everything runs from the command line, exports a portable MOJO, and leaves an auditable trail of prompts and outputs.

---

8. Conclusion

The project shows that we do not have to choose between speed and rigor in security analytics. By making use of the efficiency of AutoML, transparency of SHAP, and prescriptive nature of GenAI we were able to develop a system that actually produces what practitioners need: not just trusted but auditable, and actionable decisions, rather than just labels.

The technical contribution of the project was to develop the "XAI-to-GenAI bridge," an apparently simple technology that converts raw SHAP values into summarised results for analysts. This bridge allows for the auto-generation of context-specific incident response playbooks, including triage questions, containment activities, and validation criteria, about each alert type's likely attributes. Operationally the pipeline is respectful of the reality of security work. It avoids over-flooding processed, approved domains to limit alerts fatigue, produces confidence scores and not strictly yes/no determinations, and generates human interpretable explanations that are traceable by the analyst and can be acted on accordingly in a timely manner. The entire automation, from training the model to generating playbooks, runs through the command line with complete reproducibility.

Looking to the future, this model suggests a new model of security tooling: prescriptive analytics with reasoning and operational advice. As threat landscapes change and analyst workloads grow, having systems that can detect threats and recommend action will be crucial. The challenge will be to maintain that transparency and actionable utility as models increasingly become more complicated and adversarial activities are evidenced. Any security tool should be measured not only on its accuracy, but its ability to enable defenders to respond faster and better. Prescriptive analytics gets us closer to minimize the distance from detection to action. Prescriptive analytics will represent the next iteration of intelligent security operations, where AI does not just identify issues, it helps address them.

9. References

Core stack

- **H2O AutoML — User Guide**
  End-to-end overview of H2O-3 AutoML, leaderboards, and usage. H2O.ai
  https://docs.h2o.ai/h2o/latest-stable/h2o-docs/automl.html

- **H2O MOJO — Quickstart (export & use production models)**
  How to export and run MOJO models from AutoML. H2O.ai
  https://docs.h2o.ai/h2o/latest-stable/h2o-docs/mojo-quickstart.html

- **SHAP — Official Documentation**
  Library docs, explanations, and examples for model interpretability. Invalid URL
  https://shap.readthedocs.io/en/latest/

- **SHAP — Original Paper (Lundberg & Lee, 2017)**
  "A Unified Approach to Interpreting Model Predictions." arXiv
  https://arxiv.org/abs/1705.07874

Generative AI (Gemini)

- **Gemini API — Get an API Key**
  Where the API key comes from and how to set it (env var). Google AI for Developers
  https://ai.google.dev/gemini-api/docs/api-key

- **Gemini API — Text Generation (Python examples)**
  Official guide with minimal Python snippets for google-genai. Google AI for Developers
  https://ai.google.dev/gemini-api/docs/text-generation

Course logistics (blog)

- **GW Blogs — Get Started**
  How to create/publish your GW blog post for the write-up. GW Blogs
  https://blogs.gwu.edu/get-started/

---

10. Appendix: XAI→GenAI Bridge (Logic Sketch)

The application converts local SHAP values for the single prediction into a compact, structured text block. This block is only built when the predicted label is **dga**.

Formatting: entropy to 3 decimals; SHAP to 3 decimals (with sign); confidence to 2 decimals with %."

- Compute **local SHAP** on the single instance.
- For each feature in *(entropy, length)*:
  - Determine direction: **dga** if SHAP > 0; **legit** if SHAP ≤ 0.
  - Format each line as: `feature=value (SHAP +0.246, towards 'dga')`.

Template (XAI→GenAI summary format)

```
- Alert: Potential DGA domain detected.
- Domain: '<domain>'
- AI Model Explanation (from SHAP): <p_dga*100:.2f>% confidence for 'dga'.
  Contributions:
  - entropy=<entropy:.3f> (SHAP {shap_entropy:+.3f}, towards '{'dga' if shap_entropy > 0 else 'legit'}')
  - length=<length> (SHAP {shap_length:+.3f}, towards '{'dga' if shap_length > 0 else 'legit'}')
```

Alt: Template the app fills using local SHAP values when the prediction is 'dga'.

Example (filled with real values)

```
- Alert: Potential DGA domain detected.
- Domain: 'kq3v9z7j1x5f8g2h.info'
- AI Model Explanation (from SHAP): 59.65% confidence for 'dga'.
  Contributions:
  - entropy=4.000 (SHAP +0.246, towards 'dga')
  - length=16 (SHAP -0.013, towards 'legit')
```

Alt: Real XAI→GenAI summary from the run; entropy pushes toward 'dga', length slightly toward 'legit'.

11. Resources & Links

Code & Documentation

- **Project Repository** — public GitHub repo with all code, CI, and docs (https://github.com/SerpilRivas/seas8405-homework9)
- **Project README** — goal, architecture, and usage instructions

  (https://github.com/SerpilRivas/seas8405-homework9/blob/main/prescriptive-dga-detector/README.md)
- **TESTING Guide** — manual verification steps (legit + DGA examples)

  (https://github.com/SerpilRivas/seas8405-homework9/blob/main/prescriptive-dga-detector/TESTING.md)
- **Pinned Requirements** — exact package versions for reproducibility

  (https://github.com/SerpilRivas/seas8405-homework9/blob/main/prescriptive-dga-detector/requirements.txt)
- **CI: Ruff Lint Workflow** — static analysis on each push/PR (https://github.com/SerpilRivas/seas8405-homework9/actions/workflows/lint.yml)
- **License (MIT)** — reuse and modification terms (https://github.com/SerpilRivas/seas8405-homework9/blob/main/LICENSE)

Release (HW9-v1) — Artifacts & Evidence

- **Release Page** — all artifacts in one place (https://github.com/SerpilRivas/seas8405-homework9/releases/tag/hw9-v1)
- **Production MOJO Model (DGA_Leader.zip)** — exported leader for inference

  (https://github.com/SerpilRivas/seas8405-homework9/releases/download/hw9-v1/DGA_Leader.zip)
- **Leaderboard (leaderboard.csv)** — AutoML leaderboard snapshot

  (https://github.com/SerpilRivas/seas8405-homework9/releases/download/hw9-v1/leaderboard.csv)
- **Run Log — Legit Domain (google_run.txt)** — baseline, no playbook generated

  (https://github.com/SerpilRivas/seas8405-homework9/releases/download/hw9-v1

/google_run.txt)

- **Run Log — DGA-like Domain (dga_run.txt)** — SHAP + prescriptive playbook

  (https://github.com/SerpilRivas/seas8405-homework9/releases/download/hw9-v1/dga_run.txt)

- **Screenshot: Legit Prediction (screenshot_A_legit.png)** — classification output for `google.com`

  (https://github.com/SerpilRivas/seas8405-homework9/releases/download/hw9-v1/screenshot_A_legit.png)

- **Screenshot: XAI Summary (screenshot_B_xai.png)** — SHAP contributions (entropy/length)

  (https://github.com/SerpilRivas/seas8405-homework9/releases/download/hw9-v1/screenshot_B_xai.png)

- **Screenshot: Prescriptive Playbook (screenshot_C_playbook.png)** — incident-response plan generated from XAI

  (https://github.com/SerpilRivas/seas8405-homework9/releases/download/hw9-v1/screenshot_C_playbook.png)

- **XAI Block (xai_block.txt)** — the dynamic, programmatic SHAP summary passed to the LLM

  (https://github.com/SerpilRivas/seas8405-homework9/releases/download/hw9-v1/xai_block.txt)

- **Playbook First Screen (playbook_first_screen.txt)** — the LLM's top-level prescriptive output

  (https://github.com/SerpilRivas/seas8405-homework9/releases/download/hw9-v1/playbook_first_screen.txt)

Blog Article (to be added)

- **Research Article** — published write-up with figures and analysis

  https://blogs.gwu.edu/rserpil/2025/08/19/not-just-a-label-turning-dga-detection-into-a-playbook/