Domain Classification Challenge

Jay Pramod Asodariya

May 23, 2025

1 Context

Code notebook challenge.ipynb presents an end-to-end framework for detecting malicious domains using a rich set of engineered features, dual-stage feature selection (LASSO and Genetic Algorithm), and five classification algorithms. Emphasis is placed on minimizing false positives while maintaining high recall. Our Random Forest model achieved the best trade-off (Precision = 0.926, Recall = 0.943, FPR = 0.024).

2 Initialization and Feature Engineering

The raw data file given lready had good amount of features but after doing some research, I felt to engineer some more features in it. Eventually, dimensionality increased but it was obvious to do some feature selection to reduce overfitting.

We derive four groups of features:

2.1 Lexical URL Features

- `url_length`: total characters in the URL.
- `path_length`, `path_token_cnt`, `path_token_len_avg`, `path_token_len_max`: statistics on the path tokens.

2.2 Domain Token Features

- `dom_token_cnt`, `dom_token_len_avg`, `dom_token_len_max`: split hostname on "."/"-" and compute token counts and lengths.

2.3 Host-based and Rogue-Index Features

- `site_age_months`, `update_age_months`: WHOIS creation/update age in months since Jan 1970.
- DNS record counts (`ns_count`, `mx_count`), log-scaled and clipped.
- Rogue-index for name servers, registrars, ASNs:

$$\text{RI} = \frac{\mu/M}{\mu/M + \beta/B} \qquad \begin{aligned} \mu &= \#\text{malicious with feature}, \\ M &= \#\text{total malicious}, \\ \beta &= \#\text{benign with feature}, \\ B &= \#\text{total benign}. \end{aligned} \tag{1}$$

- ASN country code (ISO-3), ccTLD/gTLD flags.

2.4 Textual Features

- `domain`: char TF–IDF (3–5 grams).
- Large text blobs: HashingVectorizer + SafeSVD(50).
- Other text: word TF–IDF + SafeSVD(25).
- *SafeSVD* reduces only when ≥ 2 dimensions.

Total raw features: ~ 300, plus all SVD-reduced text blocks.

3 Preprocessing

1. Continuous features $\rightarrow$ clip at 97.5th percentile, min–max scale to [0,1].
2. Rogue-index features unchanged ([0,1]).
3. Binary flags cast to `int8` 0/1.
4. Text blocks vectorized, SVD-reduced, then concatenated into a dense matrix.

4 Feature Selection

4.1 LASSO Screening

We fit an L1-penalized logistic regression (solver=`saga`) over $C \in \{0.1, 0.3, 0.5, 1.0\}$, keeping features with coefficients $|\beta|$ above the mean.
Result:

{`lookup_count`, `dom_token_cnt`, `ns_rogue_idx`, `asn_rogue_idx`}.

4.2 Genetic Algorithm

Using `GAFeatureSelectionCV`:

- Estimator: SVM (RBF) + StandardScaler.
- Objectives: maximize CV ROC-AUC, minimize #features.
- Mandatory genes: the 4 LASSO-selected.
- Optional genes: remaining ~ 300.

After 40 generations ($pop = 60$), GA selected **29 optional** + 4 mandatory from LASSO Screening = **33 features**.

5 Classification & FP-Reduction

For each of five models (Random Forest, LightGBM, SVM, Logistic Regression, K-NN, Naïve Bayes) we applied:

1. Class-weight tuning to penalize false positives.
2. Probability calibration (isotonic) for smooth decision thresholds.
3. Threshold selection: maximize recall subject to Precision ≥ 0.90.
4. Post-filter: any positive with `ns_count` $= 0$ *and* `site_age_months` $<$ 0.05 is flipped negative.

6 Results

- **Random Forest** achieved the best precision–recall balance: ensemble splits over mixed features yield robust separation with few FPs.

- **SVM** closely followed after careful class-weight and threshold tuning; the RBF kernel captured non-linear feature interactions.
- **Logistic Regression** offered interpretable linear decision boundaries; precision remained high with a small recall trade-off.
- **K-NN** performed well at $k \approx 10$ with distance weighting, but recall dropped when neighbors included too many benign points.
- **Naïve Bayes** was quick to train but struggled with correlated numeric features, hence lower recall.
- **LightGBM** recorded strong ROC-AUC but required more fine-grained FP-aware tuning to match recall targets.

Table 1: Performance comparison on 20% held-out test set

Model	Precision	Recall	FPR	AUC	#Features
Random Forest	**0.941**	**0.905**	**0.020**	0.991	33
SVM (RBF)	0.907	0.925	0.024	0.990	33
Logistic Reg.	0.881	0.981	0.029	0.980	33
K-Nearest Neighbor	0.936	0.830	0.028	0.987	33
Naïve Bayes	0.910	0.860	0.022	0.932	33
LightGBM	0.790	0.962	0.073	0.961	33

7 Discussion

Our pipeline components each contributed:

- *Feature engineering* uncovered both lexical patterns and host-based reputations—the rogue-index features in particular separated benign registrars/ASNs from known malicious ones.
- *LASSO* quickly isolated the 4 strongest predictors, ensuring no noise from thousands of sparse text components.
- *GA* added complementary signals (e.g. DNS counts, TLD flags, select SVD text dimensions) that improved non-linear models.

- *Probability calibration* and *precision-first thresholding* systematically traded off recall for very low FPR, critical in high-volume security contexts.
- *Post-filtering* by DNS absence and very short age caught the remaining "edge" false positives.

What worked particularly well:

- Rogue-index features consistently ranked among top importance across RF and LR—validating their theoretical foundation.
- GA's Pareto-front optimization yielded a stable small set (33 features) that generalized across models.
- Class-weight tuning in SVM and LR significantly reduced FPR without collapsing recall.

8 What can be done with more time and data

With more time and data, I think, the class imbalance between labels 0 and 1 will still remain almost same but More data will allow me to find some complex insights, patterns and relations. With more time, I can research more about feature e While our current pipeline achieves strong performance in just three days of development, additional time and larger datasets would unlock several avenues for further improvement:

- **Enhanced Class–Imbalance Handling.** Although the benign/malicious ratio remains skewed, a larger labeled corpus—augmented via semi-supervised or active learning—would allow for more sophisticated imbalance techniques (e.g. SMOTE, focal loss) and reduce bias toward the majority class.
- **Richer Feature Engineering.**
  - *Advanced NLP embeddings:* integrate contextualized language models (BERT, RoBERTa) on page content and WHOIS text to capture semantic subtleties that TF–IDF alone misses.
  - *Temporal dynamics:* incorporate passive DNS time-series (frequency of A-record changes) and WHOIS update intervals to detect rapidly rotating malicious domains.

  - *Graph features:* build co-hosting and certificate-transparency graphs to leverage relational cues between domains.

- **Broad Algorithmic Exploration.**
  - *Deep neural architectures:* test fully-connected and convolutional networks on the combined numeric/text embedding matrix, with dropout/regularization to prevent overfitting.
  - *Ensemble stacking:* combine Random Forest, SVM, LightGBM, and neural net meta-learners to further reduce false positives via majority or weighted voting.
  - *Automated hyperparameter search:* deploy Bayesian optimization (e.g. Optuna, Hyperopt) to jointly tune model and data-processing parameters at scale.

- **Online Learning & Concept Drift.** Implement streaming models (e.g. incremental tree learners, adaptive boosting) to continuously retrain on new domain labels, automatically adapting to evolving attacker tactics and feature distribution shifts.

- **Operational Validation.** In a production setting, integrate real-time feedback loops with security analysts to validate and triage borderline cases, refining both labels and post-filter heuristics.

- **Robustness and Explainability.**
  - *Adversarial testing:* simulate URL manipulations to assess model resilience against evasive techniques (e.g. homoglyphs, padding).
  - *Interpretable AI:* apply SHAP or LIME on the final ensemble to provide human-readable explanations for each flagged domain.

These expansions would drive recall toward 0.98+ while pushing the false-positive rate below 0.1%, meeting enterprise-grade requirements for automated domain threat detection.