RMSN

Understanding Fraud Signals in E-Commerce: An Interpretable Machine Learning Study

Rafif Muchsin, 22 November 2025.

Table of contents:

1. Introduction

Online shopping has become a primary channel for global commerce, but its rapid growth has also widened the opportunities for fraudulent websites that imitate legitimate stores, deceive consumers, and disappear before accountability is possible. These deceptive sites often use realistic branding, secure-looking URLs, and professional interfaces, making manual detection increasingly difficult. As a result, automated and interpretable detection systems are now essential for protecting consumers and platforms.

This study investigates how publicly observable website attributes—such as domain registration patterns, SSL certificate metadata, payment indicators, and external reputation signals—can be used to distinguish fraudulent e-commerce sites from legitimate ones. By combining structured dataset analysis, purposeful feature engineering, and interpretable machine learning techniques, we build a transparent detection model capable of assigning a fraud risk score to any online shop.

A key focus of this work is *interpretability*: understanding why a website is flagged as risky. Rather than relying on opaque black-box predictions, the project emphasizes feature significance, human-readable explanations, and a practical risk-scoring framework that aligns with real-world fraud assessment workflows.

The project produces:

- A complete exploratory and analytical study of fraud-related domain signals
- An interpretable Random Forest–based model with strong predictive performance
- A risk scoring system that outputs 0–100 risk levels and qualitative tiers
- A command-line scoring tool for batch and single-record evaluation

Together, these deliver a practical, transparent, and reproducible workflow for identifying fraudulent e-commerce domains using interpretable machine learning.

2. Dataset Description

The dataset used in this study originates from the **Fraudulent Online Shops Detection** dataset, published under the **CC BY 4.0 license** and available on Mendeley Data:

https://data.mendeley.com/datasets/m7xtkx7g5m/1

The dataset contains **1,140 online shop URLs**, each labeled as either *fraudulent* or *legitimate*. The class distribution is nearly balanced:

- **579 fraudulent sites (50.8%)**
- **561 legitimate sites (49.2%)**

This balance allows unbiased model training without heavy correction for class imbalance.

2.1 Feature Categories

The dataset includes **26 original variables**, grouped naturally into four categories:

A. Domain & URL Structure Features

These describe the composition of a website's domain name:

- Domain length
- Top-level domain length
- Number of digits, letters, dots, hyphens
- Presence of `www` prefix

These features reflect domain composition patterns often linked to fraud strategies.

B. Payment & Contact Indicators

Signals about the shop's advertised payment methods and communication channels:

- Credit card, money-back, COD availability
- Cryptocurrency acceptance
- Presence of free email contacts
- Logo availability

These reflect business credibility and operational transparency.

C. SSL Certificate Metadata

Technical security attributes extracted from TLS certificates:

- Certificate issuer
- Expiry date
- Issuer organization class
- Young domain indicator

These capture certificate quality and domain age.

D. External Reputation Signals

Off-platform trust sources:

- Presence of TrustPilot reviews
- TrustPilot score
- SiteJabber presence
- Tranco list presence and rank

These indicate whether the site is recognized within the wider web.

2.2 Target Variable

The target label, **"Label"**, is binary:

- `"fraudulent"`
- `"legitimate"`

Our machine learning models predict the probability of the site being **fraudulent**.

3. Exploratory Analysis

The exploratory data analysis (EDA) focused on understanding how fraudulent and legitimate e-commerce sites differ across structural, behavioral, and reputational dimensions. Rather than evaluating features independently, the analysis emphasized consistent fraud patterns emerging across the dataset.

3.1 Domain Structure Patterns

Fraudulent domains tend to follow identifiable naming strategies:

- Shorter domain length and more compact naming
- Higher use of special characters, including dots and hyphens
- Lower presence of digits, suggesting preference toward simple, brand-like names
- Similar top-level domain lengths across both classes, indicating that TLD alone is not a strong discriminator

These patterns reflect low-cost, quickly generated domains designed for short-term operations.

3.2 SSL Certificate Behavior

Fraudulent sites commonly exhibit SSL characteristics aligned with minimal-effort deployment:

- Certificates issued by free certificate authorities (e.g., Let's Encrypt) dominate both classes, so issuer name alone is not discriminatory.
- The expiration horizon (days until certificate expiry) is shorter for many fraudulent sites, suggesting short-lived deployments.
- “Young domain” indicators show a notable skew: fraudulent domains are disproportionately newer.

Together, SSL metadata provides temporal and operational cues consistent with fraudulent behavior.

3.3 Payment & Contact Information Signals

Fraudulent websites often list payment and policy features aimed at creating a false sense of legitimacy:

- High presence of credit card payment and money-back guarantee claims
- Very low presence of cryptocurrency payment acceptance (but when present, often correlated with fraud)
- Free email contact addresses and missing logos occur more frequently among fraudulent shops

These features suggest superficial attempts to mimic legitimate shops while avoiding operational commitments.

3.4 Reputation & Traffic Signals

The presence of external reputation data strongly correlates with legitimacy:

- Most fraudulent sites lack TrustPilot or SiteJabber entries
- Legitimate sites occasionally have negative TrustPilot scores, but fraudulent sites often have no score at all
- Tranco list presence is extremely sparse overall, but when present it overwhelmingly indicates legitimate traffic

These signals highlight how fraudulent websites often operate outside the conventional web ecosystem.

3.5 Summary of Key Observations

From the EDA, several strong fraud indicators emerged:

- **Newly registered domains**
- **Short SSL certificate lifetimes**
- **Absence of external reputation signals**
- **Low operational transparency (logos, non-free emails, reviews)**
- **Simplistic or suspicious domain structures**

These insights directly informed which engineered features were developed and which baseline features were prioritized for modeling.

4. Feature Engineering & Preprocessing

Based on the EDA findings, the feature engineering process centered on transforming raw fields into interpretable, fraud-relevant variables that capture behavioral patterns rather than surface-level attributes.

4.1 Derived Structural Features

To capture domain composition more meaningfully:

- **digit_density = digits / domain_length**
- **hyphen_density = hyphens / domain_length**
- **dot_density = dots / domain_length**
- **letter_ratio = letters / domain_length**

These normalized ratios generalize patterns across domains of different lengths.

4.2 SSL Temporal Features

Two important time-based variables were constructed:

- **days_until_ssl_expiry**
    - Derived from certificate expiry minus reference date
    - Captures operational investment and longevity

- **days_since_registration**
    - Derived from domain registration date
    - Measures domain age, a strong indicator of legitimacy

These features translate raw date fields into actionable fraud signals.

4.3 Reputation Features

- **TrustPilot_score_clean**
  Missing or `-1` values are standardized to `NaN`, allowing consistent model interpretation.

- **is_in_tranco**
  Binary indicator: 1 if Tranco rank is valid, else 0.

- **tranco_rank_log**
  Log-transformed rank for smoother scaling of large numeric values; defaults to 0 for missing ranks.

4.4 Business Transparency Signals

These binary indicators help capture operational credibility:

- **has_free_email**
- **has_logo**
- **num_payment_methods** (aggregates various payment flags)
- **sitejabber_has_reviews**
- **trustpilot_has_reviews**

Each variable encapsulates user-facing trust factors shown to matter in EDA.

4.5 Preprocessing Strategy

Preprocessing was designed to maintain interpretability and reliability:

- Missing numeric values → median imputation
- Missing categorical values → constant “missing” token
- Scaling applied only to numeric features for stable model behavior
- Categorical fields were one-hot encoded with `handle_unknown='ignore'`
- Final dataset standardized to the 24 selected features listed in the training metadata

The preprocessing was embedded in a scikit-learn pipeline, ensuring the same transformations apply consistently during scoring and production use.

5. Modelling Approach

The modeling strategy was designed to prioritize both **predictive accuracy** and **interpretability**, ensuring that fraud predictions are not only correct but also explainable. To achieve this, the system was built using a modular scikit-learn pipeline integrating preprocessing, feature transformation, and model training.

5.1 Train–Test Split

The dataset of 1,140 samples was partitioned into:

- **80% training set**
- **20% test set**

Stratified splitting was used due to near-balanced classes, ensuring that both legitimate and fraudulent websites remained proportionally represented during evaluation.

5.2 Pipeline Architecture

A unified pipeline handled all preprocessing and modeling steps:

- Numeric preprocessing
    - Median imputation
    - Standardization via `StandardScaler`
- Categorical preprocessing
    - Constant imputation (“missing”)
    - One-hot encoding with `handle_unknown='ignore'`
- Final estimator
    - One of the selected ML algorithms (Logistic Regression, Random Forest, or XGBoost)

Embedding transformations inside the pipeline ensures that:

- Training and production scoring use identical feature preparation
- The final `model.pkl` is fully self-contained
- Risk scoring via CLI requires no separate preprocessing code

5.3 Algorithms Evaluated

Three interpretable or semi-interpretable models were selected for comparison:

1. **Logistic Regression**
   - Provides linear interpretability
   - Often strong for tabular fraud data
   - Good baseline for feature influence

2. **Random Forest Classifier**
   - Excellent for nonlinear relationships
   - Provides intuitive feature importance
   - Less prone to overfitting due to ensemble structure

3. **XGBoost Classifier**
   - Gradient boosting model with strong performance on structured data
   - Captures subtle interactions between engineered features

All models were trained using identical features and transformations to ensure a fair comparison.

5.4 Model Selection Criteria

The final model was selected based on:

- **AUC (Area Under ROC Curve)** — robustness to threshold choice
- **F1 Score** — balance between precision and recall
- **Overall interpretability**
- **Stability across different confidence thresholds**
- **Feature importance clarity**

While all models performed strongly, the **Random Forest** classifier achieved the best balance of accuracy, interpretability, and stable behavior, making it the preferred model for deployment.

5.5 Model Export & Metadata

The training process produced two key artifacts:

- **`model.pkl`** — full pipeline + trained model
- **`model_metadata.json`** — model name, selected features, and evaluation metrics

Both are required for smooth, reproducible scoring in production.

6. Results & Interpretability

This section summarizes the predictive performance of the evaluated models and provides insights into the model's decision-making through interpretable feature importance.

6.1 Model Performance Summary

All three models demonstrated high discriminative power, showing that structured domain, SSL, and reputation signals are strong predictors of e-commerce fraud.

Test-set performance metrics:

Model	Accuracy	F1 Score	Precision	Recall	AUC
Logistic Regression	0.917	0.918	0.915	0.922	0.981
Random Forest	0.93	0.93	0.946	0.914	0.982
XGBoost	0.921	0.922	0.922	0.922	0.981

Random Forest selected as final model.

It achieved the highest accuracy and a top-tier AUC, while remaining systematically interpretable via feature importance. The model also behaved consistently across multiple runs and thresholds, making it reliable for deployment.

6.2 Feature Importance Analysis

The Random Forest model highlighted several dominant features influencing fraud predictions:

Most influential features

1. **days_since_registration** — Newer domains were significantly more likely to be fraudulent.

2. **days_until_ssl_expiry** — Fraudulent sites often use short-horizon SSL certificates.

3. **digit_density, dot_density, hyphen_density** — Domain naming irregularities strongly contributed to risk.

4. **TrustPilot_score_clean & trustpilot_has_reviews** — Absence of reviews was a consistent fraud indicator.

5. **num_payment_methods** — Fraudulent sites often list unrealistic combinations of payment options.

6. **has_free_email & has_logo** — Proxy signals for professional legitimacy.

7. **is_in_tranco** — Presence in global traffic rankings was a strong sign of legitimacy.

Together, these features provide a coherent picture:

Fraudulent websites tend to be new, invisible in major reputation systems, sparsely configured, and operated with minimal infrastructure investment.

6.3 Interpretability Perspective

The Random Forest model offers two levels of interpretability:

- **Global interpretability** (feature importance)
Helps understand general fraud signals learned across all websites.

- **Local interpretability** (per-record risk scoring)
Through the CLI scoring tool, individual predictions can be probed by examining the weighted features contributing to high or low risk.

This ensures that the model remains transparent and accountable — essential traits for trust-based security applications.

7. Risk Scoring Framework & Deployment

To make the fraud detection model practical for real-world use, the project includes a complete risk scoring framework and a production-ready command-line scoring tool. This component applies the trained Random Forest pipeline to new data—either individual website records or entire batches—and converts model probabilities into interpretable fraud risk categories.

7.1 Risk Scoring Architecture

The scoring system transforms the model's prediction into a standardized risk representation:

1. **Model Output:**
   The trained Random Forest pipeline returns:
   `P(fraudulent)` — the probability a domain is fraudulent.

2. **Risk Score Conversion (0–100):**
   risk_score = P(fraud) × 100

3. **Risk Tier Mapping:**
   To support easy interpretation, risk scores map into qualitative tiers:

Risk Score	Tier	Interpretation
≥ 80	High Risk	Strong fraud indicators present
30–79	Medium Risk	Mixed signals requiring review
< 30	Low Risk	Minimal or no fraud signals

This approach aligns with standard risk assessment frameworks used in trust & safety, security operations, and fraud analytics.

7.2 Production Scoring Tool (CLI)

The scoring tool, implemented in `score_generator.py`, is designed for:

- Analysts validating single domains

- Batch processing teams screening large lists
- Integrations with dashboards, APIs, or automated workflows

The tool automatically performs:

- Column validation
- Preprocessing using the model's saved pipeline
- Probability prediction
- Risk score and tier generation
- Optional JSON or CSV output

Because the model is saved as a full scikit-learn pipeline (`model.pkl`), no additional preprocessing scripts are required—ensuring reproducibility and eliminating feature mismatch issues.

7.3 CLI Usage Modes

A. Single-Record Scoring

Used when evaluating one website's attributes:

```
python score_generator.py --single "{...json...}" --json
```

The tool returns:

```
{
  "probability_fraud": 0.9435,
  "risk_score_0_100": 94.35,
  "risk_tier": "High Risk"
}
```

This is convenient for manual analysis, testing, and validation.

B. Batch Scoring (CSV Input)

To evaluate many websites at once:

```
python score_generator.py --input sample_shops.csv --output
scored.csv
```

The output CSV includes:

- `probability_fraud`
- `risk_score_0_100`
- `risk_tier`

This makes it suitable for bulk operations such as:

- Daily screening pipelines
- Partner/vendor verification
- Commerce platform safety queues

7.4 Deployment Considerations

The system is designed to be easily ported into:

- **Web APIs** (FastAPI, Flask, Django REST)
- **Cloud functions** (AWS Lambda, GCP Cloud Functions)
- **Enterprise ETL pipelines** (Airflow, Prefect)

Key benefits include:

1. Self-contained pipeline: All preprocessing, encoding, and transformations are embedded in the model artifact.
2. Consistent input validation: `validate_and_prepare_df()` ensures feature ordering and prevents missing/invalid fields.
3. Metadata-driven operation: `model_metadata.json` ensures alignment of features, documentation, and evaluation metrics.
4. Scalable architecture: Batch mode supports arbitrarily large datasets, constrained only by system memory.

7.5 Summary

The risk scoring system makes the machine learning model fully operational by providing:

- **Transparent, interpretable scores**
- **Consistent and reproducible preprocessing**
- **Flexible deployment across single or batch workflows**
- **Production-quality input validation and error handling**

Together, the trained model and scoring tool form a complete pipeline suitable for both analytical research and real-world fraud detection workflows.