🔍 Caught in the Web: Using Machine Learning to Detect Phishing Websites

By Leila, Lidzy, and Zuleyka

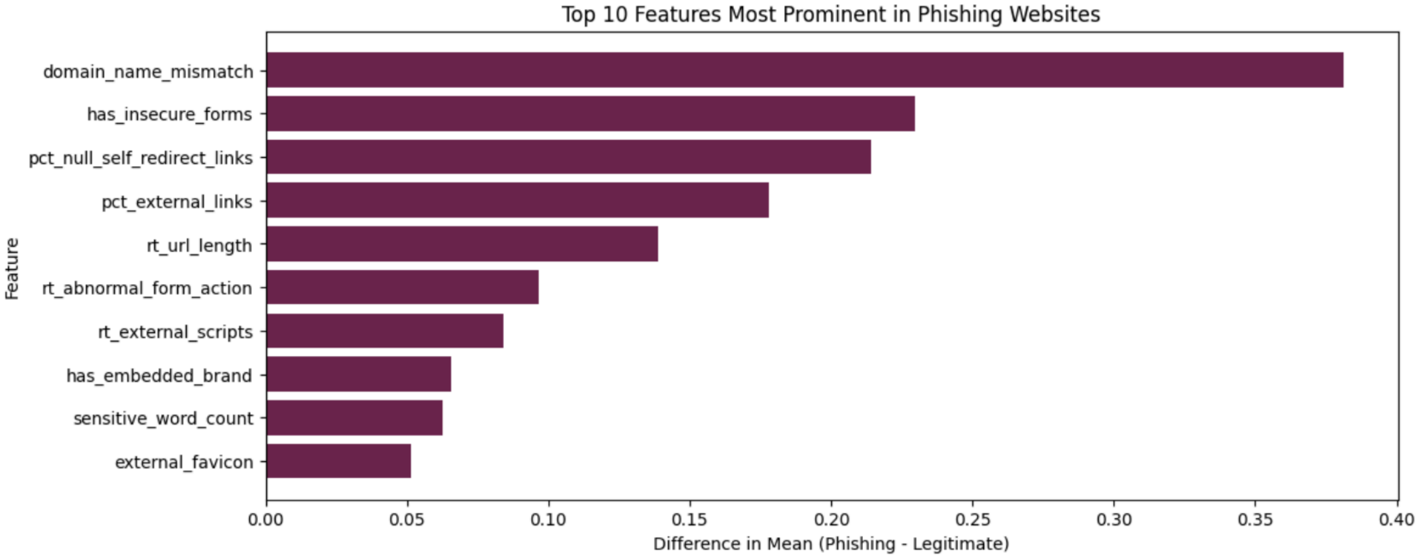
🧠 Why Phishing Detection Matters

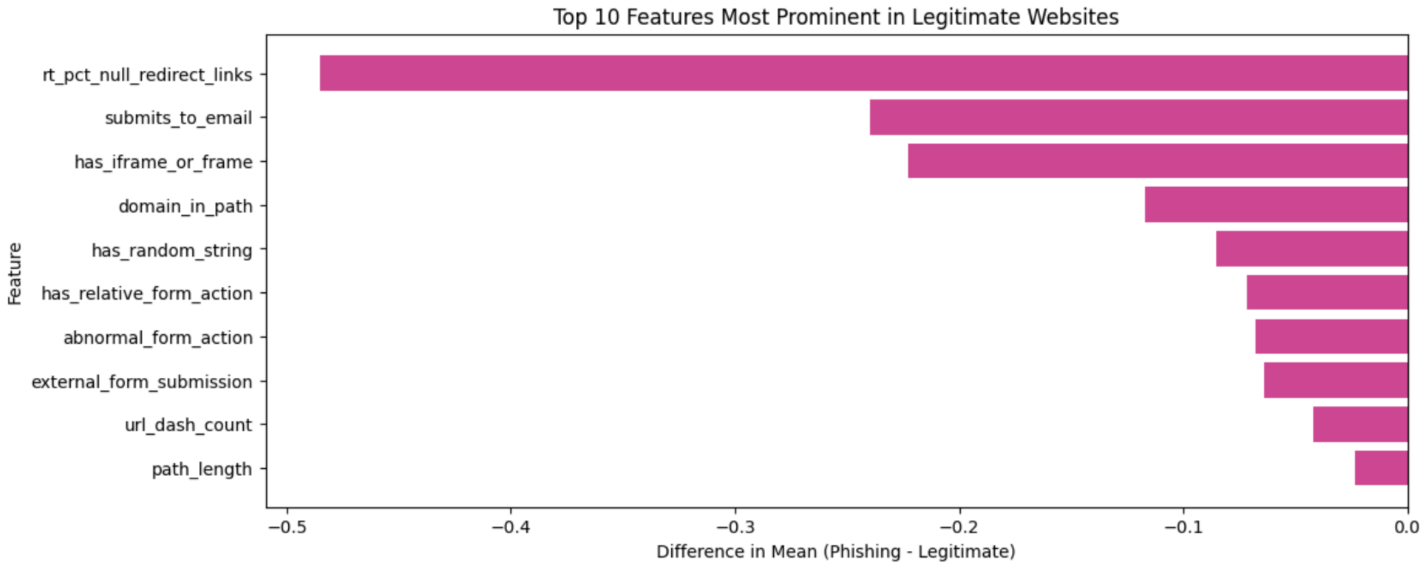
Phishing websites are deceptive pages designed to steal personal data—like passwords, credit cards, or banking credentials—by pretending to be legitimate. With billions lost annually due to phishing attacks, real-time detection isn't just valuable—it's essential.

Cybercriminals continuously refine their tactics, making traditional detection methods, such as blacklists, ineffective against new phishing techniques. To combat this, we leveraged machine learning to detect phishing websites using subtle signals embedded in their structure and behavior. Our research highlights the power of AI in cybersecurity and the potential for automated real-time protection against fraudulent websites.

Preliminary Feature Analysis

To understand which features best differentiate phishing from legitimate websites, we conducted an initial statistical analysis. By comparing means and distributions, we identified features with the most significant impact.





For example, **rt_pct_null_redirect_links**, which measures the percentage of null redirect links, showed a strong correlation with phishing sites. Other notable features included **url_dash_count** (high dash counts often indicate phishing attempts) and **pct_external_links** (a high percentage of external links is common in phishing pages).

These insights guided our model selection and feature engineering process.

📊 Our Dataset: Clues Hidden in the Web

We used a dataset of **10,000 websites**, evenly split between phishing and legitimate ones. Each website was analyzed across **46 features**, covering:

- **URL structure** (e.g., `url_dash_count`, `hostname_length`)
- **Security indicators** (e.g., `no_https`, `uses_ip_address`)
- **Website content** (e.g., `domain_in_subdomains`(if main domain appears in subdomains), `domain_in_path` (whether the domain appears in the path))
- **User behavior features** (e.g., `right_click_disabled`, `has_popup_windows`, `missing_title`)
- **Real-time interaction data** (e.g., `rt_subdomain_depth`, `rl_url_length`, `rt_external_scripts`)

These features were used to teach our models to detect signs of deception—even when the site *looks* safe to the human eye.

🎯 Our Research Questions

We focused on four guiding questions:

1. How accurately can a model detect phishing using 46 features?
2. Which model performs best—logistic regression, decision trees, neural nets, or ensemble models?
3. What specific traits separate phishing sites from safe ones?
4. Can behavioral patterns, like form actions or redirects, enhance detection?

Essentially, our primary goal was to develop a robust machine learning model capable of accurately detecting phishing websites based on a combination of behavioral and structural features.

⚙️ Models We Tested

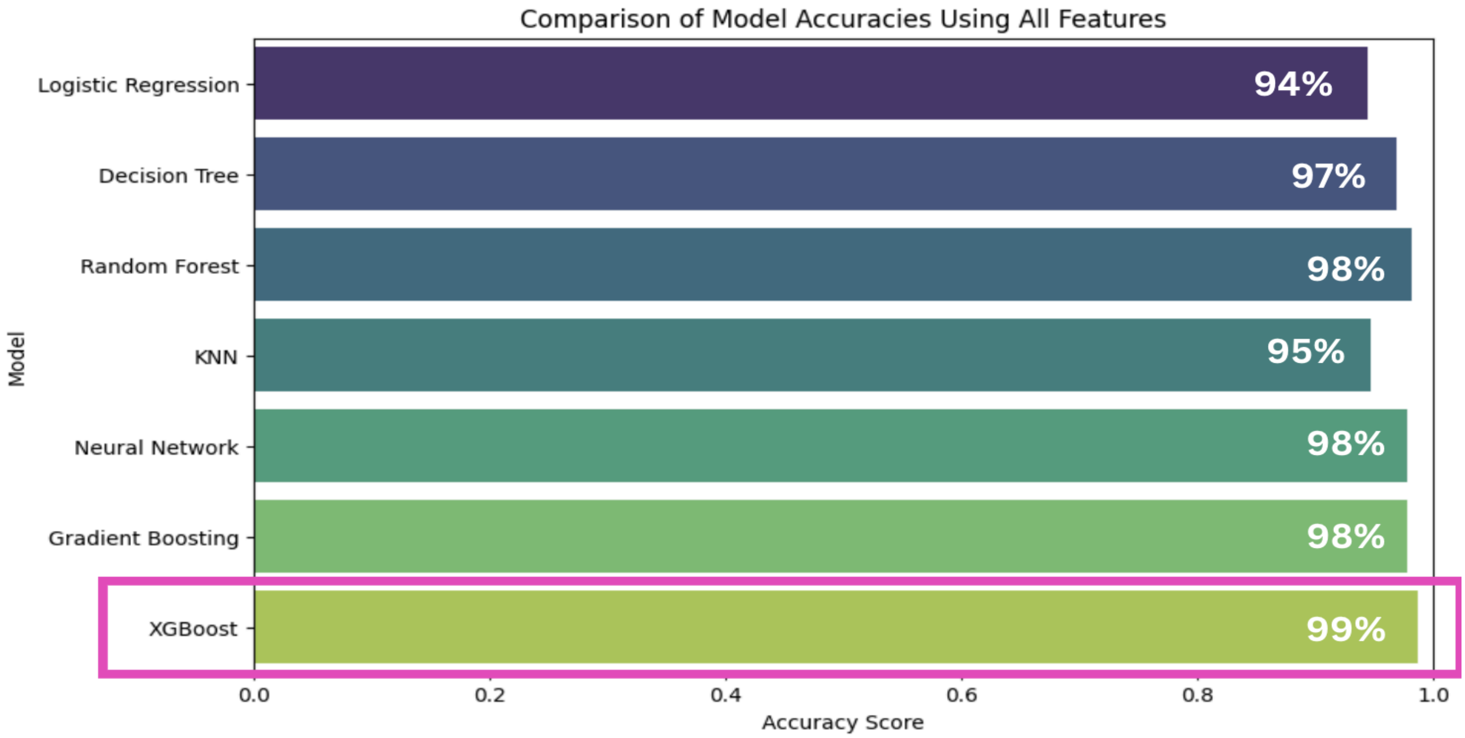
We trained multiple machine learning models to assess their effectiveness in phishing detection:

- **Logistic Regression:** A linear model that estimates the probability of a website being phishing or legitimate using weighted feature contributions.
- **Decision Tree:** A hierarchical model that classifies websites by recursively splitting data based on feature values.
- **Random Forest:** An ensemble of decision trees that reduces overfitting and improves generalization.
- **K-Nearest Neighbors (KNN):** A distance-based model that classifies a website based on the characteristics of its nearest neighbors.
- **Neural Network:** A deep learning model that captures complex patterns through multiple layers of neurons.
- **Gradient Boosting:** A boosting algorithm that sequentially corrects the errors of weak learners to improve overall accuracy.
- **XGBoost:** A highly optimized version of gradient boosting that enhances predictive performance and computational efficiency.

Each model's accuracy was evaluated on the test set:

Model	Accuracy	Precision	Recall	F1 Score
XGBoost	**98.7%**	98.6%	98.9%	98.7%
Random Forest	98.2%	98.2%	98.2%	98.2%
Gradient Boosting	97.8%	97.8%	97.8%	97.8%
Neural Network	96.9%	96.9%	97.1%	97.0%
Logistic Regression	94.1%	93.9%	94.7%	94.3%
KNN	86.7%	85.3%	89.5%	87.3%

🎯 Performance Analysis of Our Best Model: XGBoost



XGBoost emerged as the best-performing model, achieving:

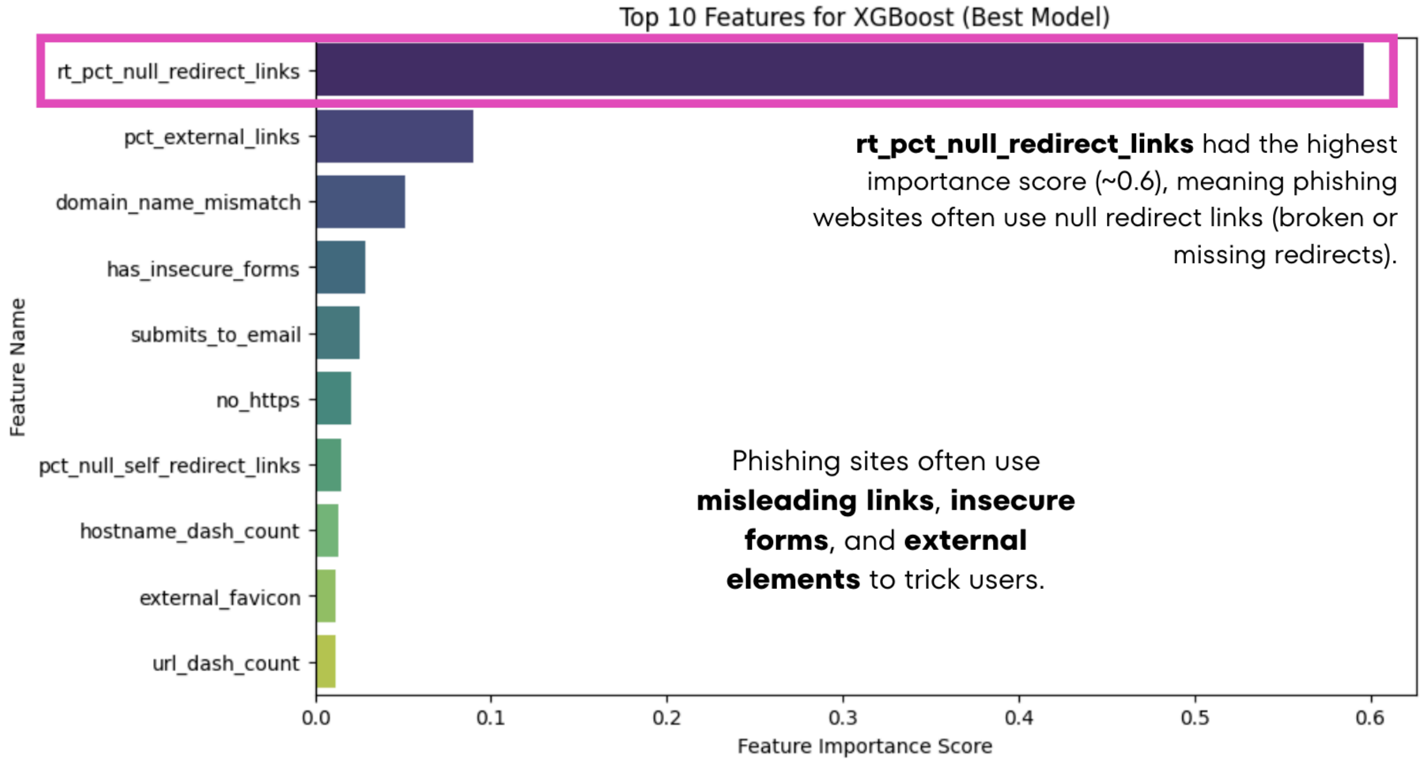
- **Accuracy:** 98.7%
- **Precision:** 98.6%
- **Recall:** 98.9%
- **F1 Score:** 98.7%

The confusion matrix revealed:

- **True Positives:** 1520
- **True Negatives:** 1441
- **False Positives:** 22
- **False Negatives:** 17

These results indicate that XGBoost correctly classified the vast majority of websites, making it the most reliable model for phishing detection.

🔍 Which Features Mattered Most?



rt_pct_null_redirect_links had the highest importance score (~0.6), meaning phishing websites often use null redirect links (broken or missing redirects).

Phishing sites often use **misleading links**, **insecure forms**, and **external elements** to trick users.

Analyzing XGBoost's top 10 important features, we found that **rt_pct_null_redirect_links** had the highest predictive power, confirming that phishing websites often contain excessive redirect patterns.

Other highly influential features included:

- `domain_name_mismatch` – Phishing domains often mislead users with mismatched names.

- `pct_external_links` – Many phishing sites rely heavily on outbound links to real sites.

These insights guided our model selection and feature engineering process.

🔍 Feature Grouping

To better interpret the impact of different features, we categorized them into five groups:

- 🔗 **URL Structure Features:** URL length, depth, special character presence
- 🔐 **Security Indicators:** Presence of HTTPS, SSL certificates
- 🌐 **Website Content Features:** Branding elements, link sources, form security
- 🧠 **Behavioral Features:** Frequency of pop-ups, click-blocking, deceptive elements
- ⚙️ **Real-Time Execution:** Live changes, redirects, and external resource requests

Referring back to the initial XGBoost's model top 10 features, we found that the most influential categories were:

- **Website Content Features:** 4 key features
- **URL Structure Features:** 2 key features
- **Behavioral Features:** 2 key features

- **Security Indicators:** 1 key feature
- **Real-Time Features:** 1 key feature (most impactful overall, with an importance score of almost 60%)

🧪 Feature Category-Specific Model Performance

Continuing the categories, we trained the same seven models using only features from each category to measure their standalone effectiveness.

Key results:

- **URL Structure:** XGBoost performed best, with **F1 Score: 90.1%**
- **Security Indicators:** Gradient Boosting performed best, with **F1 Score: 74.8%**
- **Website Content:** XGBoost performed best, with **F1 Score: 97.1%**
- **Behavioral Features:** Logistic Regression performed best, with **F1 Score: 79.2%**
- **Real-Time Execution:** XGBoost performed best, with **F1 Score: 85.0%**

🧪 Hyperparameter Tuning

To improve efficiency, we trained the best overall model using only the top 10 most important features. This resulted in:

- **Accuracy:** 96.5%
- **Precision:** 96.5%
- **Recall:** 96.7%
- **F1 Score:** 96.6%

Despite a slight drop in accuracy, the model retained strong predictive power while improving interpretability and efficiency, however, we still wanted to further improve performance.

We decided to fine-tune this second XGBoost model using a grid search across parameters like `max_depth`, `learning_rate`, and `subsample`.

The best parameters included:

- **colsample_bytree:** 0.6
- **learning_rate:** 0.1
- **max_depth:** 10
- **n_estimators:** 200
- **subsample:** 0.8

After tuning, performance remained relatively stable, with only 0.1% gains in accuracy, recall, and F1 score.

Although the final, tuned model had slightly lower accuracy than the initial model, which included all 46 features (98.7% → 96.6%), we selected it for deployment because it:

- **Handles missing features better**
- **Computes results more efficiently**
- **Generalizes better to unseen phishing tactics**

💡 Key Takeaways

- **Phishing websites follow patterns**—even when their appearance changes.
- **Behavioral signals are hard to fake** and therefore highly useful in detection.
- **XGBoost consistently outperformed other models**, with an F1 score near 99%.
- **Real-time signals**, though harder to gather, can elevate a model's performance significantly.

🚀 Real-World Impact

Our model can be embedded in browsers, firewalls, or anti-phishing filters to prevent users from falling victim to scams. It doesn't rely on outdated blacklists or reported attacks. Instead, it analyzes structural and behavioral clues—making it effective against new phishing methods as well.

With phishing tactics evolving every day, machine learning gives us a way to stay one step ahead.

📌 Final Thoughts

This project deepened our understanding of cyber deception, data-driven security, and model interpretability. It also highlighted the value of **human + machine collaboration** in solving real-world problems.

Feel free to reach out if you're interested in our full codebase, visuals, or if you'd like to collaborate on future data-driven security projects.

🔗 GitHub: https://github.com/1Ikandil/PhishingDetection-Analysis

---

💬 Let's Connect

We'd love to talk more about data science, machine learning, and practical applications like this one.

📧 Message us directly or connect with Leila, Lidzy, or Zuleyka here on LinkedIn!
