MKSSS's CUMMINS COLLEGE OF ENGINEERING PUNE

23PCCE501L -ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING LABORATORY (PR)

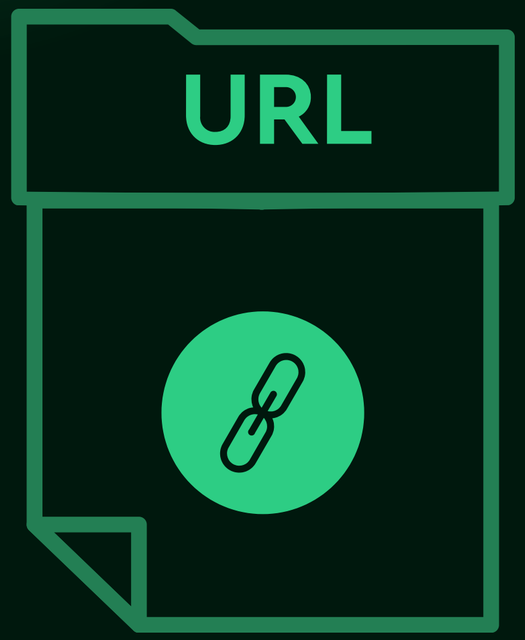
PHISHING WEBSITE DETECTION USING ML

UCE2023416 - Shreeya Chavan
UCE2023418 - Sahisha Chipade
UCE2023433 - Vedika Kayangude

PROJECT
OVERVIEW

This project is an Intelligent Phishing URL Detector designed to identify malicious websites in real-time. Unlike traditional tools that only check blacklists, this system uses a hybrid approach by combining Machine Learning (Decision tree, KNN) with live web crawling. It analyzes both the URL structure (lexical features) and the actual webpage content to detect zero-day phishing attacks. The tool includes a user-friendly Gradio interface for easy training and testing.

PROBLEM **STATEMENT**



The Issue:

- **Rising Threat:** Phishing attacks have increased significantly, targeting financial and personal data.
- **Zero-Day Attacks**: Attackers generate thousands of unique URLs daily to bypass traditional blacklists.
- **Static Limitations:** Existing systems often rely only on URL blacklists, which cannot detect newly created malicious sites in real-time.

The Solution:

We need an automated system that analyzes the features of a URL (URL structure + Website Content) to predict if it is malicious, even if it has never been seen before.

Objectives

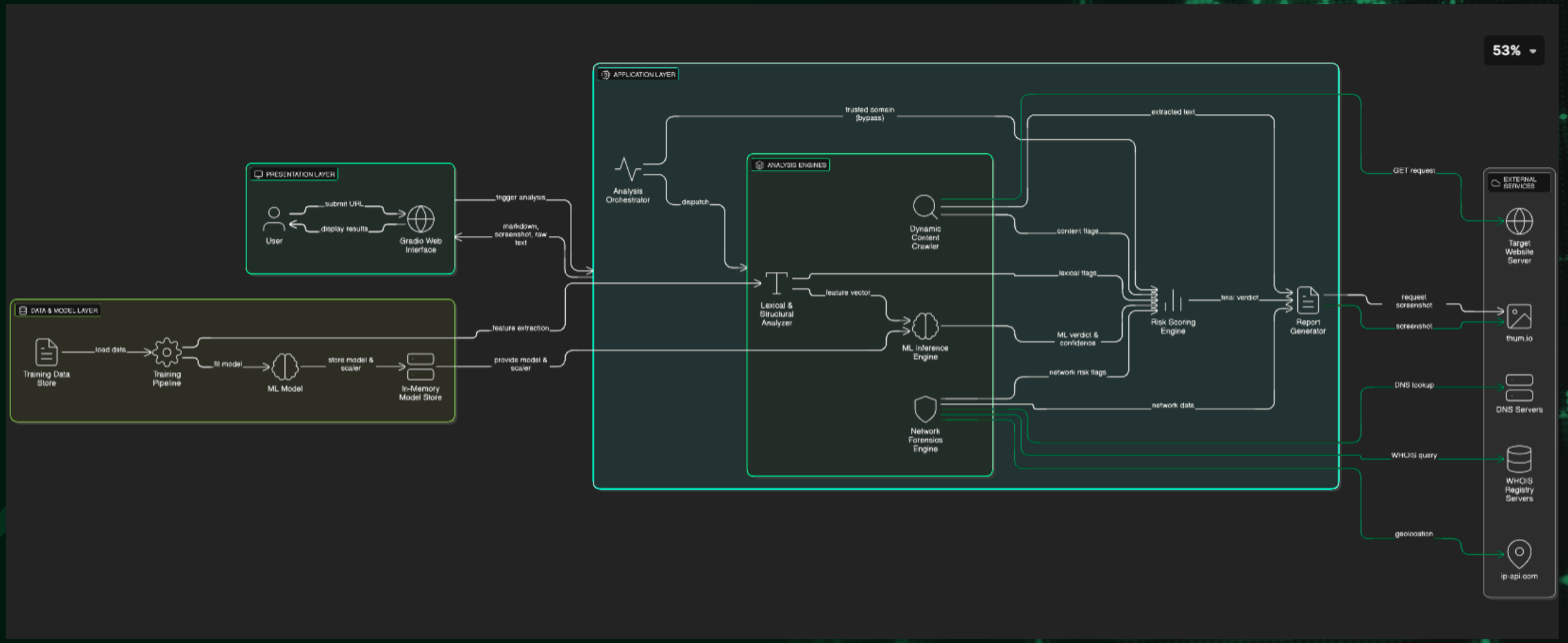
To implement a Feature Extractor that parses URL strings for malicious patterns.

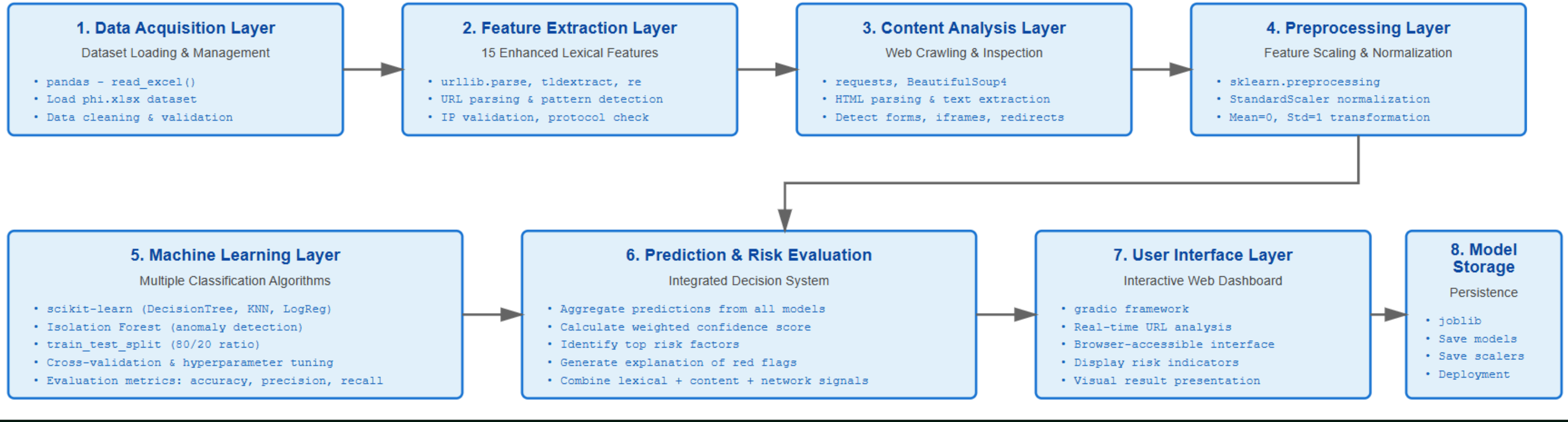
To implement Supervised and Unsupervised Learning to predict phishing website.

To develop a Live Web Crawler that safely visits websites to scan for social engineering keywords.

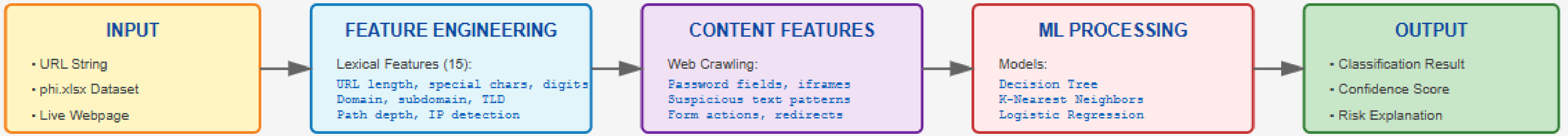
To create a user-friendly GUI using Gradio for real-time testing.

SYSTEM **ARCHITECTURE**




1. Data Acquisition Layer
Dataset Loading & Management
• pandas - read_excel()
• Load phi.xlsx dataset
• Data cleaning & validation
2. Feature Extraction Layer
15 Enhanced Lexical Features
• urllib.parse, tldextract, re
• URL parsing & pattern detection
• IP validation, protocol check
3. Content Analysis Layer
Web Crawling & Inspection
• requests, BeautifulSoup4
• HTML parsing & text extraction
• Detect forms, iframes, redirects
4. Preprocessing Layer
Feature Scaling & Normalization
• sklearn.preprocessing
• StandardScaler normalization
• Mean=0, Std=1 transformation
5. Machine Learning Layer
Multiple Classification Algorithms
• scikit-learn (DecisionTree, KNN, LogReg)
• Isolation Forest (anomaly detection)
• train_test_split (80/20 ratio)
• Cross-validation & hyperparameter tuning
• Evaluation metrics: accuracy, precision, recall
6. Prediction & Risk Evaluation
Integrated Decision System
• Aggregate predictions from all models
• Calculate weighted confidence score
• Identify top risk factors
• Generate explanation of red flags
• Combine lexical + content + network signals
7. User Interface Layer
Interactive Web Dashboard
• gradio framework
• Real-time URL analysis
• Browser-accessible interface
• Display risk indicators
• Visual result presentation
8. Model Storage
Persistence
• joblib
• Save models
• Save scalers
• Deployment

Detailed Data Flow & Components



Python Libraries & Technologies

Data Processing:

- `pandas` - DataFrame operations, data manipulation
- `numpy` - numerical computations (via sklearn)

Feature Extraction:

- `urllib.parse` - URL component parsing
- `tldextract` - domain/subdomain/TLD extraction
- `re` - regular expression pattern matching
- `ipaddress` - IP address validation
- `requests` - HTTP/HTTPS protocol verification

Web Crawling:

- `requests` - fetch live webpage content
- `BeautifulSoup4` - HTML parsing and extraction

Machine Learning & Deployment

Preprocessing:

- `sklearn.preprocessing.StandardScaler`

Machine Learning (scikit-learn):

- `DecisionTreeClassifier` - rule-based classification
- `KNeighborsClassifier` - distance-based prediction
- `LogisticRegression` - linear probability model
- `IsolationForest` - anomaly detection
- `train_test_split` - dataset partitioning
- `accuracy_score, classification_report, confusion_matrix`

User Interface:

- `gradio` - interactive web interface

Model Persistence:

- `joblib` - model serialization and deployment

METHODOLOGY - PART 1:
LEXICAL FEATURE EXTRACTION

Key Features Extracted:

URL Length: Phishing URLs are often significantly longer (>54 chars).

IP Address in Domain: Legitimate sites use domain names (e.g., google.com), not IPs (e.g., 192.168.1.1).

"@" Symbol: Browsers ignore everything before the @, leading users to the wrong domain.

Multi-Subdomains: (e.g., paypal.confirm-account.com).

Typosquatting: (e.g., go0gle.com vs google.com).

METHODOLOGY - PART 2: **CONTENT ANALYSIS (THE CRAWLER)**

Key Features Extracted:

Uses BeautifulSoup and Requests libraries.

Scrapes <input> tags (looking for password fields on non-secure sites).

Scrapes <form> actions (checking if data is sent to a foreign domain).

Scans for high-risk phrases: "Verify Identity", "SSN", "Unlock Account", "Update Payment".

METHODOLOGY - PART 3:
MACHINE LEARNING MODELS

K-Nearest Neighbors (KNN):
Classifies a URL based on its similarity to known bad URLs in the dataset.

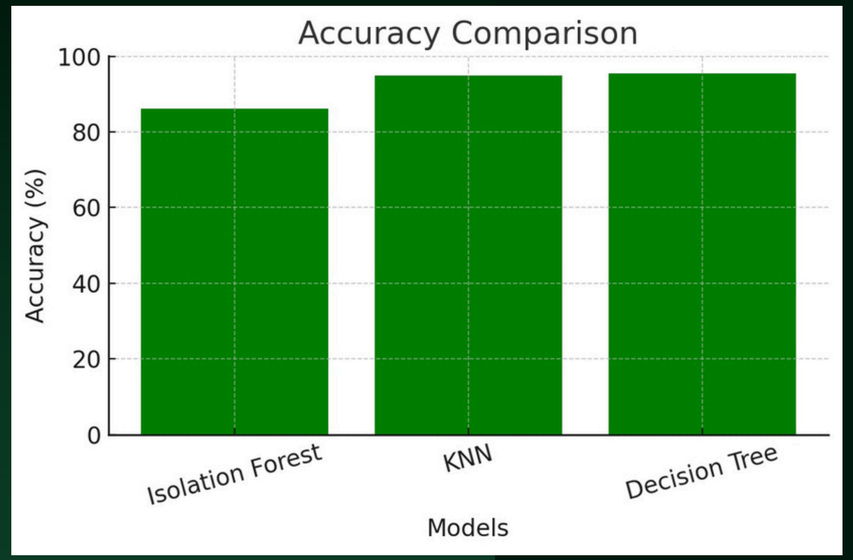
Isolation Forest (Unsupervised): An anomaly detection algorithm that treats phishing sites as "outliers"; it isolates rare, malicious patterns from normal traffic without needing a balanced, labeled dataset.

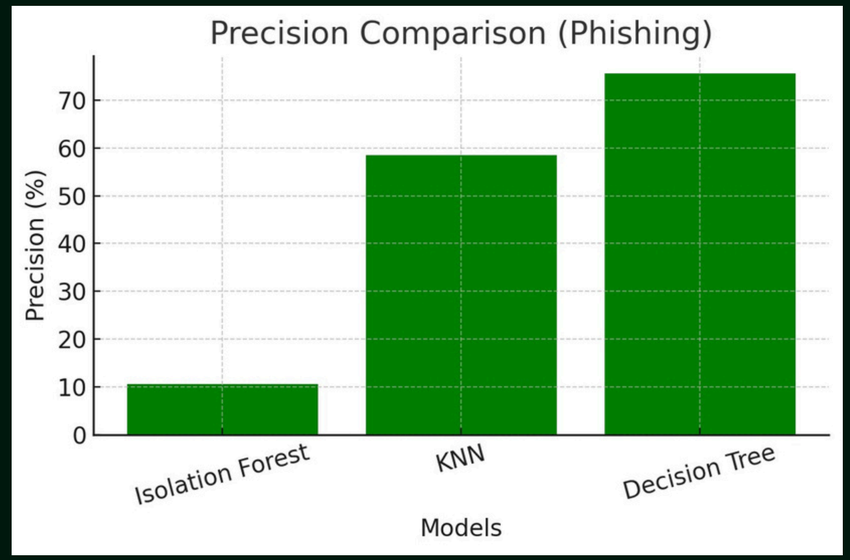
Decision Tree:
A rule-based classifier that splits data based on feature values (e.g., If URL Length > 70 AND contains '@', then Phishing).

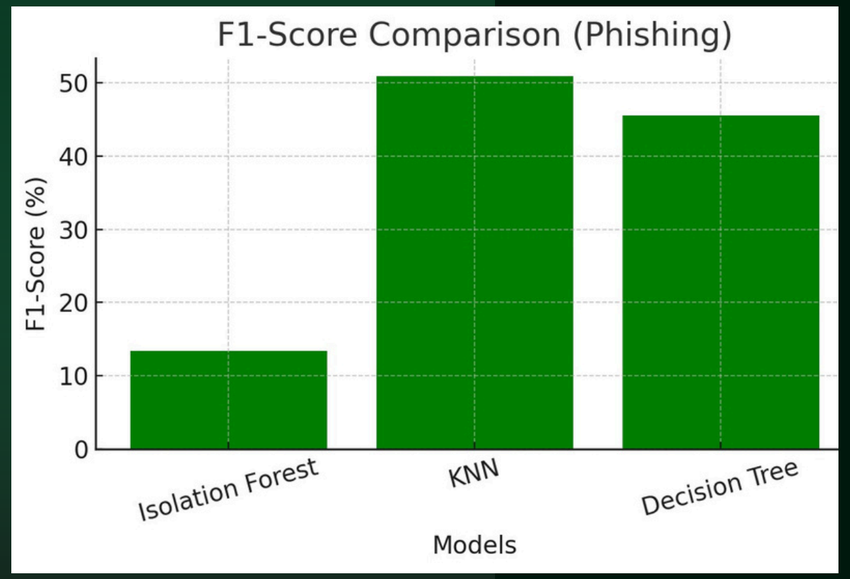
EVALUATION METRICS

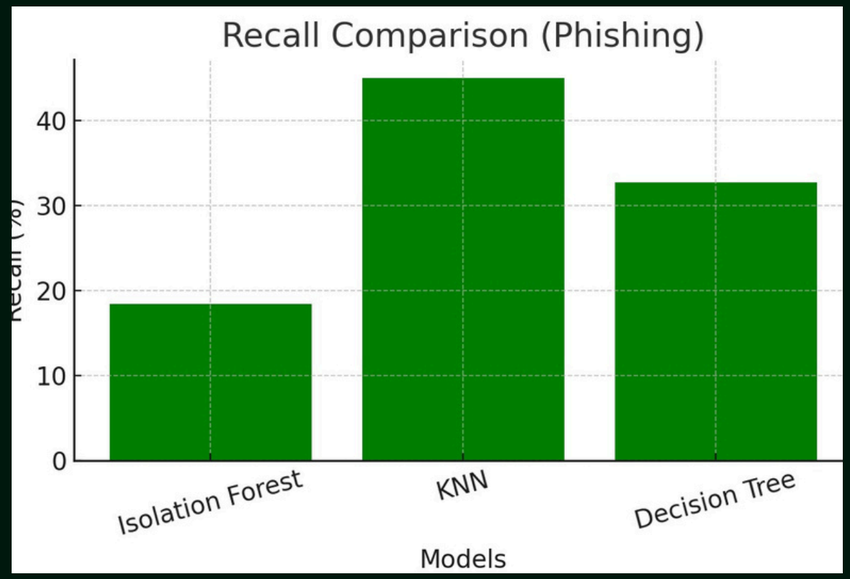
Models are compared on 4 Accuracy metrics

1. Accuracy
2. Precision
3. Recall
4. F1 Score









DATA FLOW & SCORING LOGIC

Data Flow Pipeline

- **Input:** User submits URL via Gradio UI.
- **Extract:** Parallel analysis of URL (Lexical), Website (Content Crawler), and Domain (Network/DNS).
- **Analyze:** ML Model (Isolation Forest) + Rule-Based Heuristics check for threats.
- **Output**: A final Risk Score, Verdict, and Website Screenshot are generated.

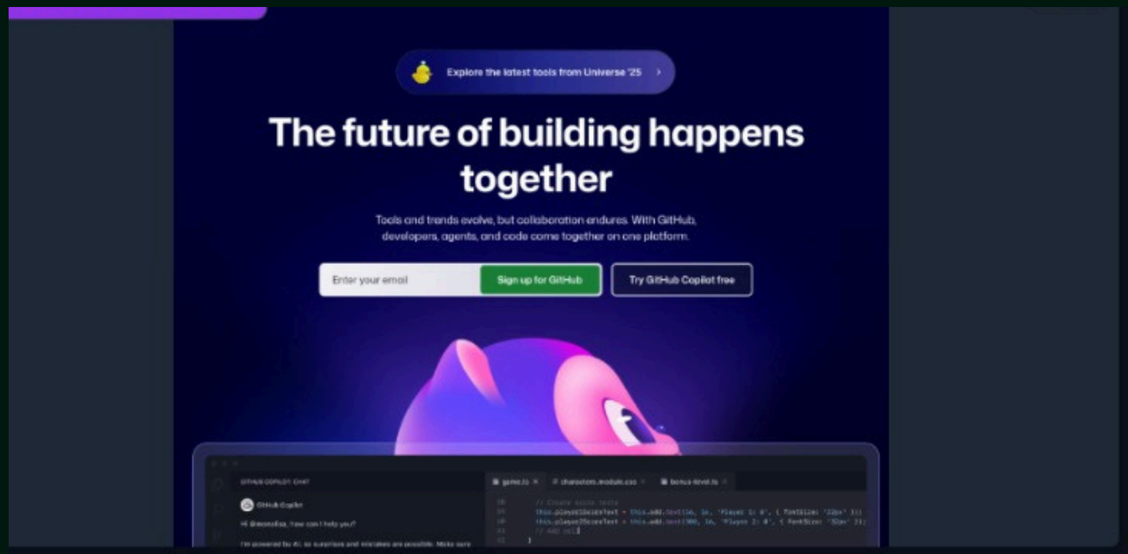
Hybrid Scoring Algorithm

- 60% URL Risk (Structure, Typos)
- 20% Content Risk (Keywords, Forms)
- 15% ML Model (Pattern Match)
- 5% Network Risk (IP Location)
- Critical Penalties: +Score added for IP in URL, Password Fields, & External Form Actions.
- Whitelist: Score reduced by 30% for trusted sites (e.g., Google, LinkedIn).

Decision Thresholds

- PHISHING:> 70%
- SUSPICIOUS:50% - 70%
- SAFE:< 50%

VISUALIZATIONS



Network Security Analysis

DNS Records:
- A Records: 20.205.243.166

IP Address Analysis:
IP: 20.205.243.166
Country: Singapore
ISP: Microsoft Corporation

ebsite Content (Extracted Text)

itHub MarketplaceBuilt-in application security where found means
xedUse AI to find and fix vulnerabilities so your team can ship more
ecure software faster.Apply fixes in seconds. Spend less time debugging
nd more time building features with Copilot Autofix.Explore GitHub
dvanced SecuritySecurity debt, solved. Leverage security campaigns and
opilot Autofix to reduce application vulnerabilities.Learn about GitHub
ode SecurityDependencies you can depend on. Update vulnerable
ependencies with supported fixes for breaking changes.Learn about
ependabotYour secrets, your business. Detect, prevent, and remediate
aked secrets across your organization.Learn about GitHub Secret
rotection70% MTTR reduction with Copilot Autofix8.3M secret leaks ...

Suspicious Content Detected

No suspicious text patterns detected in website content.

Enter URL to analyze

https://github.com

Analyze URL

Dataset-Optimized Test Cases

https://google.com | https://github.com | https://en.wikipedia.org/wiki/North_Dakota | http://secure-bank-verify.com | http://goooogle.com/login | http://faceb0k.com | htttps://fake-bank.com | http://suspicious-site.comghh

Comprehensive Security Analysis Report

Final Verdict

SAFE: Trusted Website

Risk Assessment Summary

Metric	Value	Risk Level
Overall Risk Score	0.0%	LOW
ML Model	SAFE	LOW
URL Structure	0 issues	LOW
Content Analysis	0 flags	LOW
Domain Trust	VERIFIED	VERIFIED

Result for safe URL

Result for phishing URL

Enter URL to analyze

http://faceb0k.com

Analyze URL

Dataset-Optimized Test Cases

https://google.com | https://github.com | https://en.wikipedia.org/wiki/North_Dakota | http://secure-bank-verify.com | http://goooogle.com/login | http://faceb0k.com | htttps://fake-bank.com | http://suspicious-site.comghh

Comprehensive Security Analysis Report

Final Verdict

PHISHING: High Risk Detected

Risk Assessment Summary

Metric	Value	Risk Level
Overall Risk Score	90.0%	HIGH
ML Model	PHISHING	HIGH
URL Structure	0 issues	LOW
Content Analysis	1 flags	MEDIUM
Domain Trust	UNVERIFIED	UNVERIFIED

FUTURE SCOPE

Deep Learning Integration: Replace manual feature extraction with NLP (BERT) to understand text context and CNNs to detect visual logo spoofing.

Advanced Forensics: Add WHOIS lookup to check domain age (phishing sites are usually <1 month old) and validate SSL certificates.

Browser Extension: Convert the script into a Chrome/Edge plugin for real-time blocking while users browse.

Dynamic Analysis: Use Selenium to detect phishing traps hidden inside JavaScript code (which the current requests library misses).

API Integration: Connect to live threat databases like VirusTotal or Google Safe Browsing for up-to-the-minute accuracy.

Expand the system to detect 'Smishing' (SMS Phishing) and 'Quishing' (QR Code Phishing) to protect users on mobile devices.

Technology Stack

User Interface

- Gradio: Web-based GUI for URL input and result display.
- Thum.io: Safe, real-time website screenshot previews.

Machine Learning Engine

- Scikit-Learn: Trains classification models
- Joblib: Saves and loads trained models.
- Python: Handles core application logic.

Data Processing & Extraction

- Pandas & NumPy: Data manipulation and vectorization.
- BS4 & Requests: Web scraping and HTML parsing.
- Regex: Advanced URL pattern analysis.

CONCLUSION

This project successfully implements a robust, hybrid Phishing Detection System capable of identifying cybersecurity threats in real-time. The comparative analysis revealed that Logistic Regression serves as the most efficient algorithm for this binary classification task, balancing processing speed with high accuracy. By integrating Web Crawling alongside lexical feature analysis, the system overcomes the limitations of traditional blacklist-based approaches, detecting even structurally legitimate URLs that host malicious content. Ultimately, the development of the Gradio interface transforms this complex backend into an accessible, user-friendly tool, effectively mitigating the risks of social engineering and providing a proactive layer of defense against zero-day attacks.

REFERENCES

- Google, “Safe Browsing—Protecting users from phishing, malware, and unwanted software,” 2007.
- OpenDNS, “PhishTank: A collaborative anti-phishing platform,” 2006.
- N. Abdelhamid, A. Ayesh, and F. Thabtah, “Phishing detection based on associative classification,” Expert Systems with Applications, vol. 41, no. 13, pp. 5948–5957, 2014.
- J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, “Beyond blacklists: Detecting malicious websites from suspicious URLs,” in Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1245–1254, 2009.


THANK YOU