MLPR FINAL PRESENTATION

Multimodal Phishing Detection

Sighakolli Jahnavi , Cheerla Parthiv Sagar

Index

PROBLEM
Statement

“Modern phishing uses polymorphic and visual spoofing to bypass single-dimension filters. Meanwhile, current multimodal solutions remain either too simplistic for zero-day threats or too computationally heavy for real-time deployment”

Potential

Application & Impact

- **Browser Extensions:** A lightweight consumer extension that protects everyday users from visual spoofing
- **Network Firewalls:** Deployed at the central network perimeter to automatically scan and block complex multimodal threats before they ever reach student or employee inboxes.

- **High Computational Efficiency:** Delivers optimal performance with reduced server-side processing overhead.
- **Transparent Decisions:** Replaces black-box predictions with clear mathematical weights that security teams can trust.
- **Proactive Protection:** Blocks multi-layered, zero-day attacks at the network boundary before any credential theft occurs.

Literature Review

- **Unimodal Systems (Fast but Blind):** * URL-Only (NLP/BERT): High accuracy on known threats , but context-blind and highly vulnerable to polymorphic zero-day evasion.
- **Content-Only (HTML/DOM):** Models like HTMLPhish achieve good accuracy , but are brittle because attackers can effortlessly clone a legitimate site's HTML structure.
- **Vision-Based (Layout & Logos):** Systems like Phishpedia excel at catching pixel-perfect brand spoofing. However, they fail completely against novel visual layouts or unknown brands because they require rigid reference databases.

The Gap in Multimodal Architectures

- **API & Database Dependency**
- Systems like Jail-Phish or PhishAgent rely heavily on slow, third-party search APIs (Google) or massive, manually updated 20,000+ brand databases.

- **The "Early Fusion" Flaw (Feature Overshadowing)**
- Current Multimodels attempt to combine or concatenate raw data arrays early in the training pipeline

Our Solution

- **Lightweight Late-Fusion Architecture**
- Our lightweight, Decoupled Late-Fusion architecture solves this by evaluating modalities independently and using a Meta-Classifier to synthesize probabilities, ensuring fast, accurate, and fully transparent threat detection.

Dataset Selection

Dataset Chosen

Phish360: A rigorously curated multimodal anti-phishing dataset.

Why We Chose It:

High Content Uniqueness

Zero Missing Data

Realistic Difficulty

Total Datapoints

Total: 10,748 unique multimodal samples. (samples between 2020 and 2023)

Phishing: 4,332 samples

Legitimate: 6,416 samples.

Data Modalities

Each sample is a complete triplet:

1. URL
2. HTML source code
3. Rendered web page screenshot (1280×960)

It specifically uses legitimate login pages (to prevent structural bias) and accurately reflects modern HTTPS/SSL evasion tactics, creating the most realistic benchmark for zero-day threats.

Features: 22 raw columns, including URLs, TLDs, full HTML, extracted text arrays, and screenshot paths.

NO ethical concerns as mentioned by the authors

Dataset & Feature Preprocessing

1. Missing Data & Interpolation

- **Status: 100% complete across URLs, HTML, and Images.**
- **Action: Zero missing values. No statistical regression or interpolation required.**

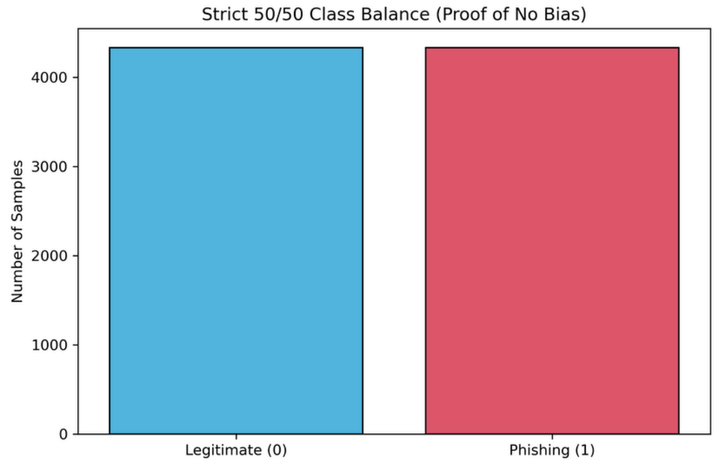
2. Feature Extraction & Dimensionality

- **Lexical (URLs): Extracted 54 deterministic features from raw text**

4. Class Balancing

- **Action: Strict random undersampling.**
- **Result: Reduced to exactly 8,664 rows for a mathematically perfect 50/50 Phish/Legit split.**

Data Collection: The authors collected 10,748 real-world phishing and legitimate samples by rigorously pre-validating each sample to ensure the URL, HTML source code, and screenshots were fully accessible, correctly rendered, and unique.



Methodology

Models Used

1. Data Preparation & Splitting

- **Balanced Dataset:** Undersampled → 8,664 samples → perfect 50% Phish / 50% Legit ratio.
- **Strict Splitting:** Applied an 80/20 Train-Test split.
- **Validation:** Used Stratified 5-Fold Cross-Validation during every training process

2. Feature Extraction & Dimensions

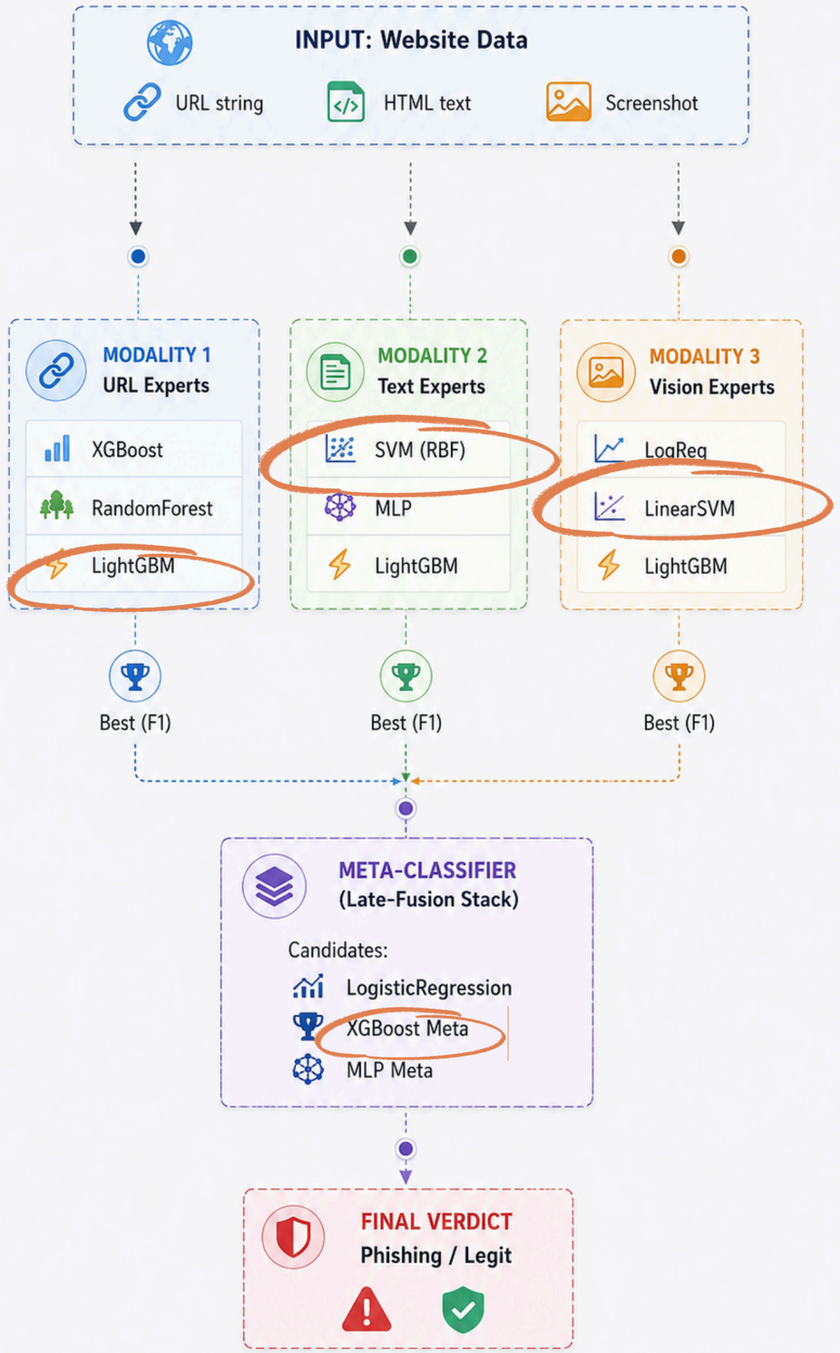
- **URL (Lexical):** Extracted 54 deterministic features (e.g., length, special characters).
- **HTML Text (Semantic):** raw text → frozen MPNet → 768-Dimensional embedding array.
- **Screenshots (Spatial):** Images → frozen ResNet50 → 2048-Dimensional visual array.

3. Base Experts (The Winning Models)

- **URL : LightGBM**. (Tree-based models max depth of 6 to prevent memorizing the training data).
- **Text :SVM (RBF Kernel).** Excelled at mapping the 768-D semantic clusters without the bloat of NN
- **Vision : LinearSVM**. Applying a strict linear boundary prevented the massive 2048-D image tensors from overfitting.

4. The Late-Fusion Meta-Classifier

- Instead of mixing the **raw 54, 768, and 2048-dimensional arrays** (which causes the image data to overshadow the URL), the **three Base Experts output a simple Probability Score (0 to 1).**
- **Final Verdict: An XGBoost Meta-Classifier acts as the final judge**, synthesizing just these three probabilities to make the ultimate Phishing or Legit prediction.



Methodology

Models Used

Lexical Dimension (URLs):

- **Methods Used: Tree-based ensembles (XGBoost, LightGBM, Random Forest).**

Chosen because tree-based models excel at finding non-linear cutoffs in deterministic, tabular data (our 54 lexical features).

Semantic Dimension (HTML Text):

- **Methods Used: Distance and Neural models (SVM, MLP,LightGBM) processing frozen MPNet embeddings.**

MPNet effectively captures deep contextual intent across 30+ languages, while SVMs efficiently map these dense semantic clusters without the computational bloat of fine-tuning a full transformer.

Spatial Dimension (Visual Layout):

- **Methods Used: Strictly Linear Classifiers(LR, LinearSVM,LightGBM) evaluating frozen ResNet50 embeddings.**

Chosen because applying strict linear boundaries prevents massive visual arrays (ResNet50) from overfitting and mathematically overpowering the smaller text/URL features.

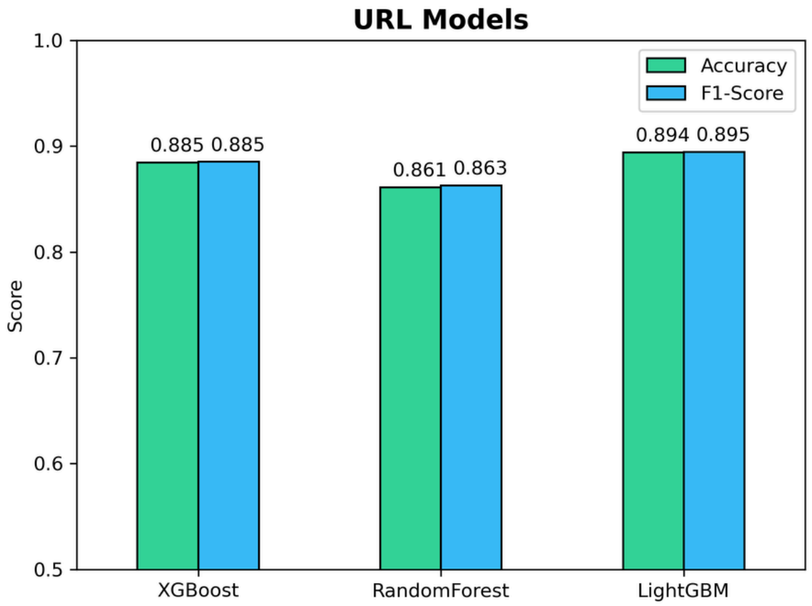
The Late-Fusion Integration (Final Prediction):

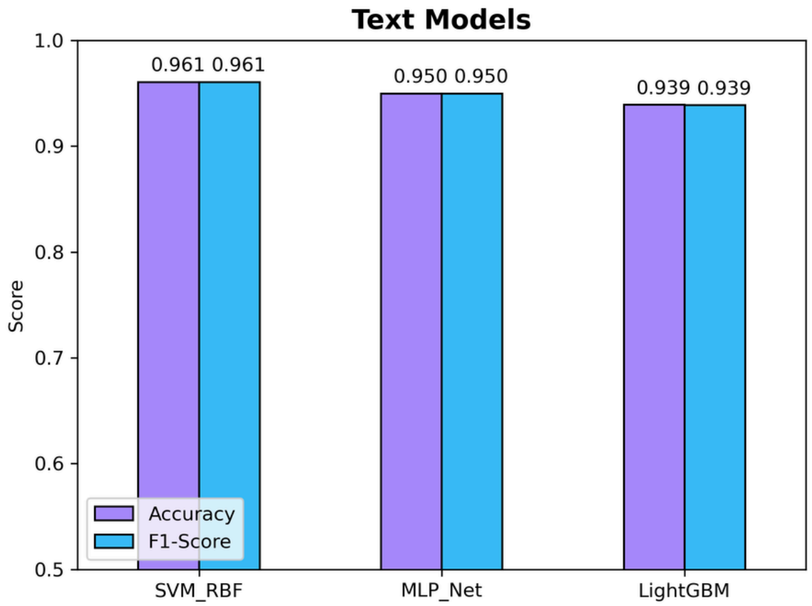
- **Chosen to act as an interpretable synthesizer that fuses the probabilities of the base experts rather than mixing raw data.**

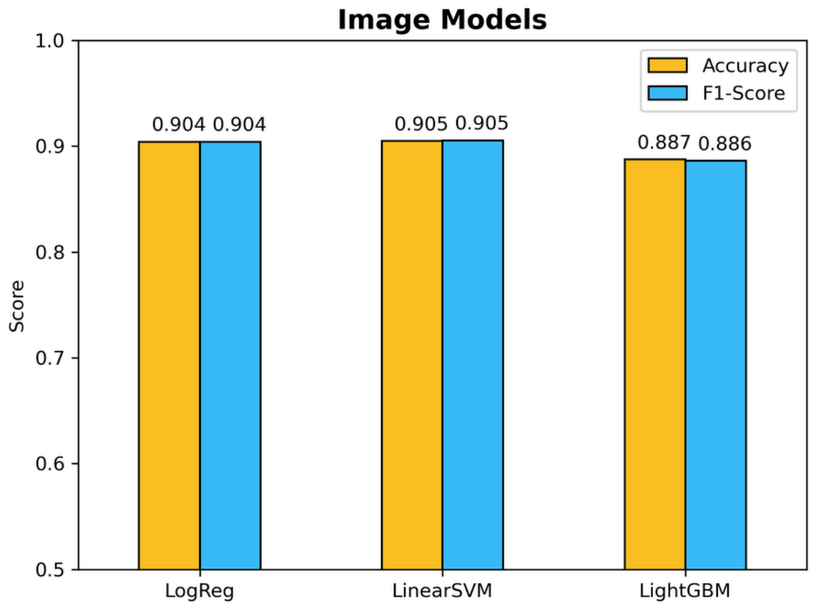
RESULTS

Deployability

These are the performance metrics for the 9-Base models



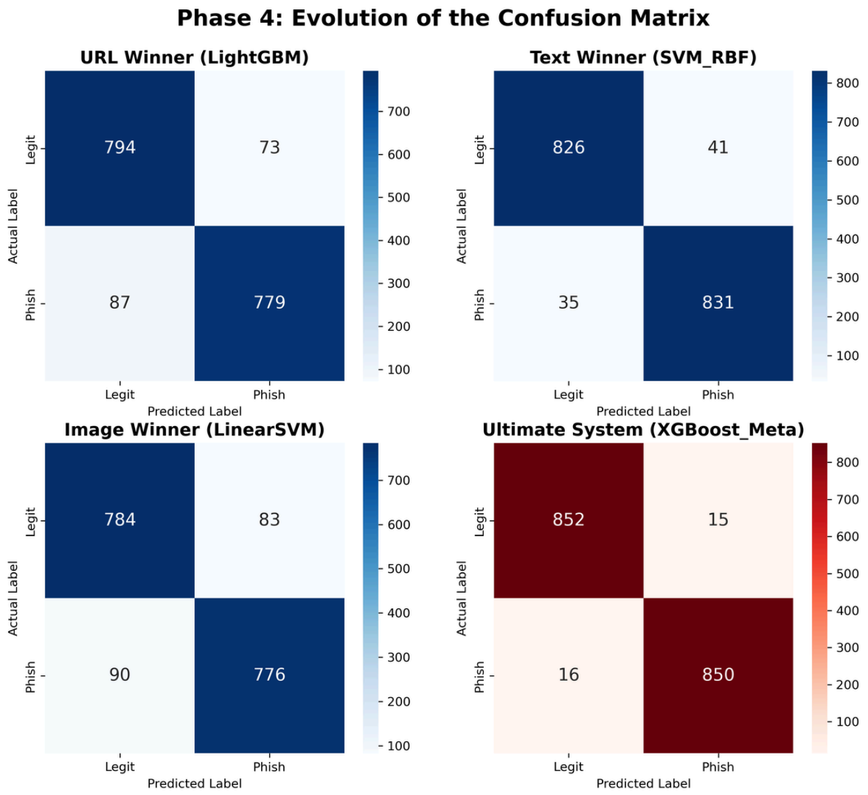


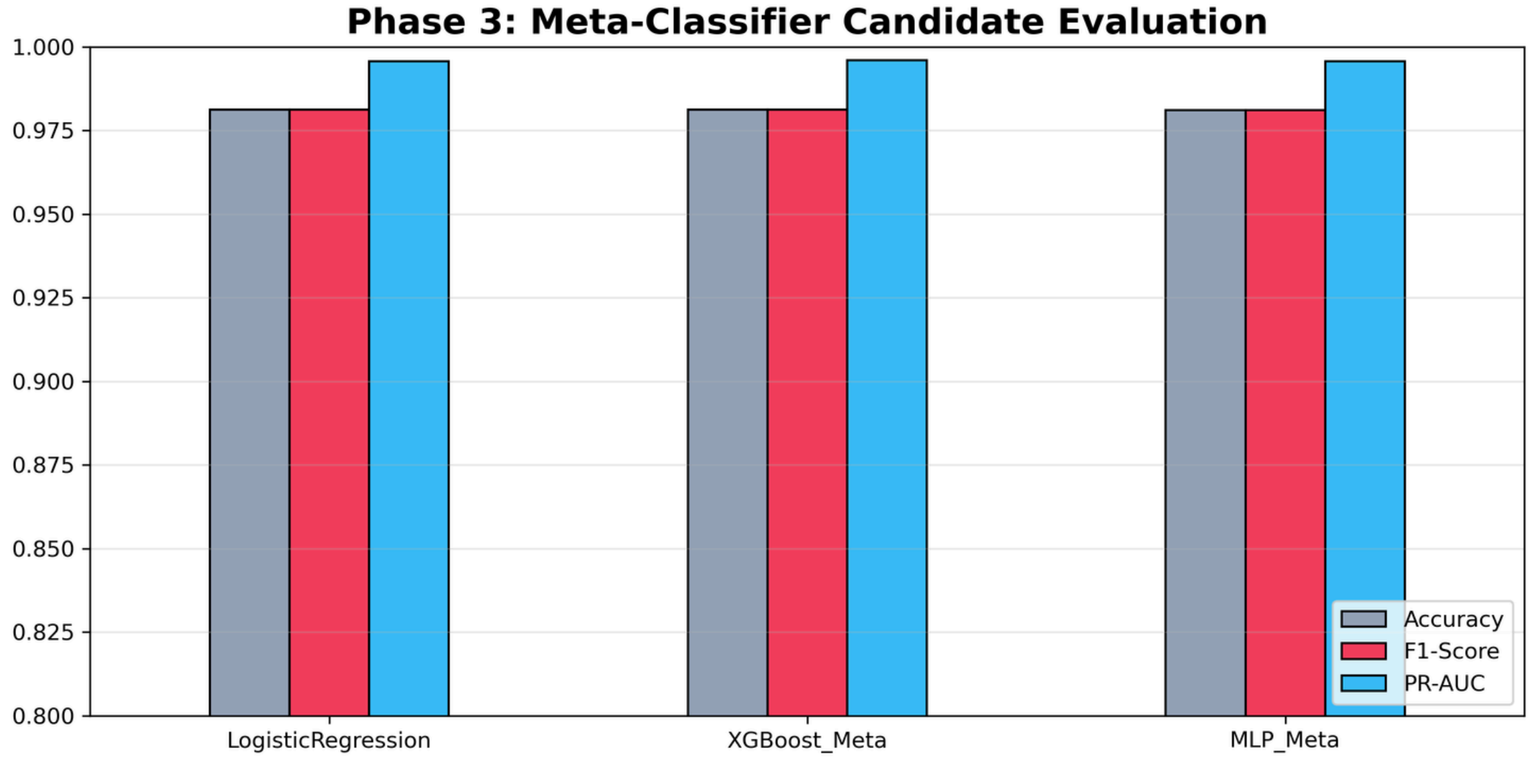


RESULTS

Deployability

Modality	model	accuracy	precision	Recall	F1-score	PR-AUC
Meta(Final)	XGBoost Stacker	98.21%	98.26%	98.15%	98.21%	99.59%

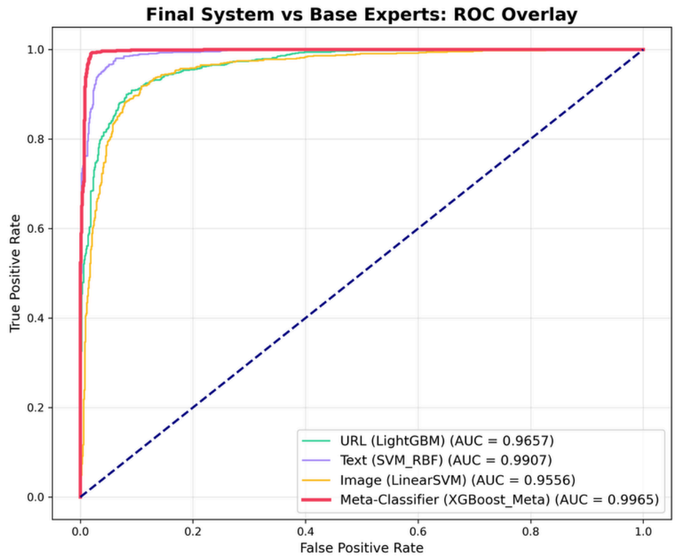




RESULTS

Deployability

Metric	Score	What it proves (Simplified)
Accuracy	98.21%	Shows that combining all 9 models works significantly better than just looking at the URL or text alone.
Precision	98.26%	Means the system rarely makes a mistake by blocking a safe, legitimate website (very few False Positives).
Recall	98.15%	Means the system successfully catches almost all of the actual phishing attacks, even the tricky new ones.
F1-Score	98.21%	Shows the model is perfectly balanced—it aggressively catches threats without annoying normal users with false alarms.
PR-AUC	99.59%	Proves the AI is extremely confident in its decisions and is almost never "guessing" between safe or malicious.



Campus Deployment

- Browser Extension: Students use a lightweight Chrome extension. It processes data locally and securely pings a campus server for instant threat detection.
- Email Gateway: Plaksha IT automatically scans and tests external email links in a safe sandbox before they reach student inboxes.

&

Scalibility

- The Speed Problem (Latency): Taking live screenshots and running heavy vision models (ResNet50) causes 3+ second delays. Users won't wait for a page to load.
- CAPTCHAs & Cloudflare: Attackers use human-verification screens that block our automated scrapers from "seeing" or reading the phishing site.
- High Compute Costs: Running heavy deep learning models (MPNet) concurrently for 2,000+ students requires expensive GPU server clusters.

Thank You