MULTI-LAYERED PHISHING URL DETECTION BY USING MACHINE LEARNING

Understanding the Problem Statement

Advanced Phishing Detection System·
Description: Create an advanced tool that detects phishing attempts in emails, websites, and messages by analyzing patterns, domain names, and content for suspicious links or requests.

Challenge: Develop a machine learning model that classifies phishing and legitimate communications accurately.·

Everity and Daily Life Impact

Phishing is not just a technical issue, it’s a **human problem** that affects **everyone** from **students to senior citizens**.
If unaddressed:
People lose **money**, **data**, and **privacy**.
Organizations face **data breaches**, **reputation damage**, and **legal consequences**. Governments and institutions become vulnerable to **cyber espionage**.

Research Landscape, Existing Solutions & Our Uniqueness

Current Landscape
Global: 1.2 billion+ phishing emails daily (*Forbes, 2024*)
India: 175% increase in phishing attacks in H1 2024 (*The Hindu*)
Bihar: Among top 10 most-affected states due to **low cyber awareness** (*India Today, 2023*)

Existing Solutions
Google Safe Browsing, **Microsoft Defender**, antivirus tools (Norton, Kaspersky)
Mostly rely on **blacklists or static detection**
Often fail against **new/zero-day phishing URLs**

What Makes Our Solution
Unique Multi-layer Detection
Combines **whitelist + blacklist + VirusTotal API + ML model (Random Forest)**
Real-time Protection
Works instantly when a suspicious link is clicked
User-friendly for Common People
Minimal UI, no technical background needed
Low-resource compatible – perfect for mobile and rural users

Our Solution: Multi-Layered Phishing Detection System

- We have developed an **AI-powered phishing detection module** that:
- Performs **real-time URL analysis**
- Uses **multiple layers** of verification for accuracy
- Is designed for **mobile and web-based integration**

Pre-processing: Masked URL Detection & Unmasking

- Many phishing attacks use **shortened or masked URLs** to mislead users.
- In our system, the **pre-processing step** identifies if a URL is shortened or obfuscated.
- If detected, it is **unmasked** to reveal the actual destination before analysis.
- This ensures the **entire detection process operates on the true URL**, not a disguised one.

Layer 1: Whitelist & Blacklist Check

The unmasked or original URL is checked against a:

Whitelist of trusted domains
Blacklist of known phishing domains

Decision:

If found in the blacklist → Blocked
If found in the whitelist → Allowed
Otherwise → Forwarded to next layer

Layer 2: External Threat Intelligence (VirusTotal API)

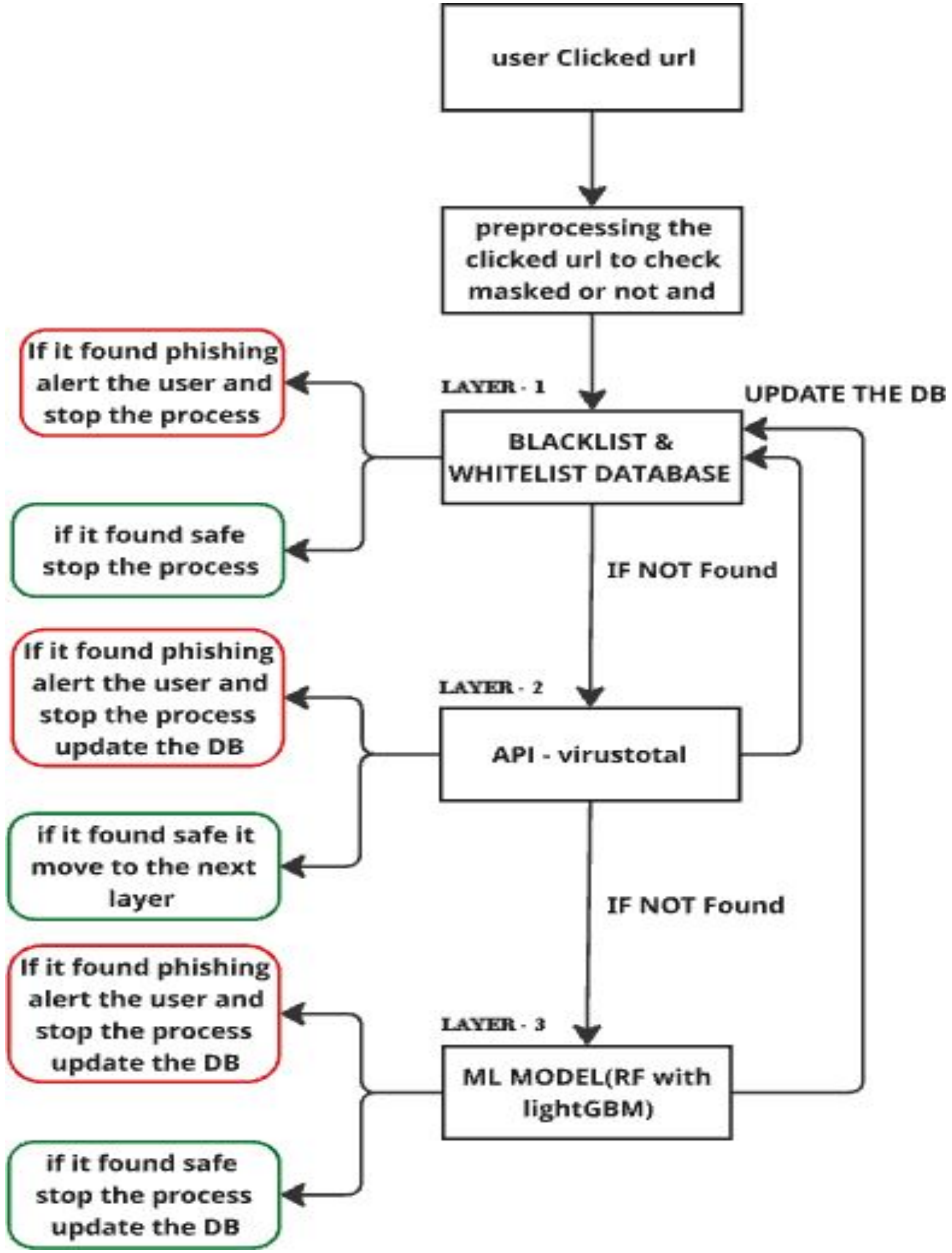
The URL is verified using a **global threat intelligence platform**. If classified as phishing → Blocked and updated in the blacklist If marked safe → Proceed to ML layer (for extra precaution)

Layer 3: Machine Learning-Based Detection

Features such as domain length, use of HTTPS, symbol count, etc., are analyzed.
Trained ML model **Random Forest** which is suitable algorithm.

Decision: Block or allow based on prediction and update the blacklist and whitelist corresponding to the URL prediction

Architecture



user Clicked url
preprocessing the clicked url to check masked or not and
LAYER - 1
UPDATE THE DB
BLACKLIST & WHITELIST DATABASE
If it found phishing alert the user and stop the process
if it found safe stop the process
IF NOT Found
LAYER - 2
API - virustotal
If it found phishing alert the user and stop the process update the DB
if it found safe it move to the next layer
IF NOT Found
LAYER - 3
ML MODEL(RF with lightGBM)
If it found phishing alert the user and stop the process update the DB
if it found safe stop the process update the DB

Future Potential

If extended support is provided, we plan to **enhance and scale the project further** through the following:

Convert the solution into a deployable Web App and Android App
This would allow users to **scan URLs in real-time directly from their browser or smartphone**, making phishing detection easily accessible to non-technical users.

Integrate advanced threat intelligence APIs
We aim to include more APIs like Google Safe Browsing, AbuseIPDB, and WHOIS to strengthen domain intelligence and detection accuracy.

Improve ML model with real-time learning
Implementing **online learning or model retraining** using newly identified phishing and legitimate URLs will keep the system adaptive to evolving threats.