AMRITANETRA: AMRITA NETWORK THREAT RECOGNITION AND ANALYSIS

A project report submitted by

DARSHAN SURESH [MY.EN. U3BCA22274]

VIDHYADHARA M [MY.EN. U3BCA22272]

in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF COMPUTER APPLICATIONS

under the guidance of

Dr. ADWITIYA MUKHOPADHYAY

Deputy Controller of Exams,

Amrita Vishwa Vidyapeetham

Mysuru Campus

MAY 2025

AMRITA VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY UNDER SECTION 3 OF UGC ACT, 1956
MYSURU CAMPUS

Amrita School of Computing, Mysuru Campus

<u>BONAFIDE CERTIFICATE</u>

This is to certify that the project entitled " **AMRITANETRA: AMRITA NETWORK THREAT RECOGNITION AND ANALYSIS** " submitted by,

DARSHAN SURESH [MY.EN. U3BCA22274]

VIDHYADHARA M [MY.EN. U3BCA22272]

For the award of the degree of **Bachelor of Computer Applications** at **Amrita Vishwa Vidyapeetham, Mysuru campus,** is a bona fide record of the work carried out by them under my guidance and supervision at Amrita University, Mysuru.

SUPERVISOR

Dr. ADWITIYA MUKHOPADHYAY
Deputy Controller of Exams
Amrita Vishwa Vidyapeetham
Mysuru Campus.

PROJECT CO-ORDINATOR (UG)

Mr. ROSHIN C
Assistant Professor
Department of Computer Science
Amrita Vishwa Vidyapeetham
Mysuru Campus.

CHAIRPERSON

Mr. VINAYAK HEGDE

Department of BCA

AMRITA VISHWA VIDYAPEETHAM

MYSURU CAMPUS

DECLARATION

We,

DARSHAN SURESH [MY.EN. U3BCA22274]

VIDHYADHARA M [MY.EN. U3BCA22272]

hereby declare that this project report, entitled "**AMRITANETRA: AMRITA NETWORK THREAT RECOGNITION AND ANALYSIS"** is a record of the original work done by us under the guidance of **Dr. Adwitiya Mukhopadhyay**, Deputy Controller of Exams, Amrita Vishwa Vidyapeetham, Mysuru campus, and that to the best of our knowledge, this work has not formed the basis for the award of any degree/diploma/associate-ship/fellowship or a similar award, to any candidate in any University.

Signature of the Student

1.

2.

Place: Mysuru

Date:

ACKNOWLEDGEMENT

We would like to express our sincere thanks to **Amma**, our beloved chancellor “**Mata Amritanandamayi Devi**”.

We would like to express our sincere thanks to **Br. Anantaananda Chaitanya**, Director, Mysuru Campus, Mysuru who supported us with continuous encouragement, motivation and lending us with right direction throughout our tenure of studies.

We would like to express our sincere thanks to **Br. Muktidamrta Chaitanya**, Correspondent, Mysuru Campus, Mysuru for providing us with well well-supported, peaceful studying environment with excellent infrastructure and support.

We would like to express our sincere thanks to our beloved principal, **Dr. G Ravindranath,** for giving us moral support and continuous encouragement, which has been the key to the successful completion of the project.

We are pleased to acknowledge **Mr. Vinayak Hegde**, Chairperson, Department of BCA, for his encouragement and support throughout the project.

We would like to express our heartfelt gratitude to our Project Co-Ordinator, Mr. **Roshin C,** Assistant Professor, Department of BCA for her valuable suggestions and excellent guidance rendered throughout this project.

We would like to express our heartfelt gratitude to our guide **Dr. Adwitiya Mukhopadhyay**, Deputy Controller of Exams, for his valuable suggestions and excellent guidance rendered throughout this project.

Further, we extend our thanks to all the faculty members and technical staff of our department and ICTS for their suggestions, support, and for providing resources at the needed times.

Finally, we express our sincere gratitude to our parents and friends who have been the embodiment of love and affection which helped us to carry out the project smoothly and successfully.

TABLE OF CONTENT

***Abstract*—Phishing attacks are increasingly sophisticated and posing enormous threats to the users by impersonating genuine sites in order to capture sensitive information such as login credentials, banking data, and personal identification information. To address this challenge, we present AmritaNetra, an automated phishing detection and awareness system that incorporates machine learning-based classification and AI-driven user interaction. The method relies on a publicly accessible Kaggle dataset of 11,054 URLs labeled as phishing or not, with each described by over 30 features such as the occurrence of HTTPS, domain age, and URL structure. Normalization and the deletion of the non-informative fields constitute the preprocessing steps. These supervised learning models were trained and validated many times like Logistic Regression, K-Nearest Neighbors, Support Vector Machines, and Gradient Boosting. Among those, Gradient Boosting was having the highest accuracy of 97.4% and was being employed as the primary detection engine. AmritaNetra stands apart from usual systems in the sense that it contains a specific combination of AI-based story writing module and a live chatbot interface. As soon as a phishing site is detected, the system provides an interactive, narrative account of the danger to inform the users about the background of the attack. Not only does the chatbot carry out real-time phishing scans but also simulates phishing attempts in order to educate the users and promote digital literacy. Both these features of detection and user enablement are made possible by this double-layered process. AmritaNetra can be rendered scalable and accommodate new phishing methods with scheduled future development to include scanning of email and social media and user opinion for ongoing optimization. The solution offers a novel, holistic means of cybersecurity in the merging of technical precision with human-oriented design.**

***Keywords*—Phishing Detection, Machine Learning, CatBoost Classifier, URL Feature Analysis, Cybersecurity, Chatbot Integration, AI Storytelling, Supervised Learning, Natural Language Processing (NLP), Real-time Prediction, User Awareness, Classification Algorithms, Web Security.**

LIST OF FIGURES

LIST OF TABLES

SYMBOLS & ABBREVIATIONS

Abbreviation / Symbol	Full Form / Description
ML	Machine Learning
AI	Artificial Intelligence
NLP	Natural Language Processing
SVM	Support Vector Machine
RF	Random Forest
UI	User Interface
URL	Uniform Resource Locator
HTTPS	Hypertext Transfer Protocol Secure
PK	Primary Key (in database schema)
FK	Foreign Key (in database schema)
CSV	Comma-Separated Values (data format)
JSON	JavaScript Object Notation (data format)
ChatID	Identifier for each chatbot session
StoryID	Identifier for generated story entries
CatBoost	Categorical Boosting Algorithm (ML model used)
F1-Score	Harmonic mean of precision and recall
Confusion Matrix	Evaluation tool showing prediction performance by class

1. INTRODUCTION

In the modern digital age, the internet has become a part of daily life, acting as a medium for communication, banking, education, business, and social interaction. But its speedy growth has also opened doors to growing cybersecurity threats, especially phishing attacks. Phishing is one type of online fraud wherein threat actors assume genuine personas in a bid to force victims to hand over their usernames, passwords, credit card information, and other confidential credentials. The underhand means to execute such plans in most instances consists of using masquerading sites, mock logon websites, and pretender e-mails.

With the development of machine learning, phishing detection mechanisms have also considerably developed. The conventional rule-based detection mechanisms are usually slow to respond and do not detect new and sophisticated forms of phishing. Machine learning models, by contrast, have been extremely effective in identifying patterns and irregularities involved with phishing attacks.

AmritaNetra is a novel solution that not only employs machine learning to identify phishing websites with high accuracy but also AI storytelling and a chatbot to raise user awareness. The goal is not only detection, but also to inform users about how phishing occurs and how they can defend themselves. This two-pronged system acts both as a defense mechanism and as a learning system, balancing technical rigor with user-friendly interaction.

1.1 INTRODUCTION TO THE BROAD AREA OF RESEARCH

With the advent of the digital age, where data exchanges and online interactions are commonplace, cybersecurity has become a worldwide issue. The swift growth of the internet and the growing reliance on web-based services have led to a new range of security issues. Cyberattacks are becoming increasingly sophisticated and frequent, with phishing being one of the most cunning and potent attack vectors. Phishing involves a social engineering attack where attackers pretend to be reliable entities in order to trick people into revealing sensitive information like usernames, passwords, banking details, and other personal information.

Around the world, both entities and organizations have lost substantially through phishing attacks. Phishing takes advantage of human weakness rather than system flaws to a larger extent, which renders them particularly challenging to prevent through conventional methods solely. Increased growth in social networking, mobile platforms, and cloud computing has heightened the attack vector, which is making things increasingly more difficult to deal with. To meet this challenge, experts and researchers are turning towards smart systems fueled by machine learning (ML), artificial intelligence (AI), and behavior analysis.

Phishing detection and prevention tools have evolved from simple blacklist and rule-based tools to more adaptive and dynamic solutions. ML and AI enabled the large-scale analysis of web traffic, the extraction of meaningful patterns from it, and the classification of URLs or emails as malicious or benign with very high accuracy. With the ready availability of machine learning frameworks and datasets, the research in this direction continues to gain traction and real-world importance.

1.2 INTRODUCTION TO THE SPECIFIC AREA OF RESEARCH

Among numerous cybersecurity research topics, phishing detection with machine learning and natural language processing has been a promising area. In this area, systems are trained on labeled data composed of malicious and legitimate samples—most commonly URLs, emails, or HTML contents. These samples are then transformed into feature vectors, which record patterns like the presence of HTTPS, usage of special characters, length of domain, number of subdomains, or registered domain age. These features are passed on to supervised models like Random Forests, Gradient Boosting, SVMs, or deep neural networks to detect phishing attempts with high levels of confidence.

Even though there exist many machine learning techniques, the solutions are generally aimed towards backend classification accuracy. Though they are able to identify threats with great accuracy, they lack features that inform end-users about the type of threat. Users either ignore them or do not take necessary actions. The solution put forth, AmritaNetra, seeks to bridge this need by developing not just a performance-intensive phishing engine but also linking it with an AI-driven chatbot and story module that depicts threats in terms accessible to end users and highly interactive.

AmritaNetra checks incoming URLs, foretells if they are phishing attacks, and on detection, sends an alert as well as a sequence of explanatory responses. The chatbot gives the forecast with a mention of the essential URL characteristics, while the AI narrative engine tells a short story based on actuality to give the user better insight as well as aid in memory recollection. AmritaNetra is distinguished from other detection systems by this multi-faceted method that is more effective.

1.3 INTRODUCTION TO THE BACKGROUND OF THE PROBLEM OF RESEARCH

Conventional phishing defense tools like blacklists, heuristics, and signature-based filtering are reactive in nature. These tools depend on known phishing signatures or reported URLs. But phishers keep registering new domains and creating sites that evade these kinds of detection. By employing tactics like URL obfuscation, domain squatting, and Unicode characters, today's phishing attacks evade conventional filters and fool even seasoned users.

In addition, most users are unaware or do not possess the technical knowledge to recognize phishing signs. Attackers take advantage of human psychology by inducing a sense of urgency ("Your account will be locked!") or pretending to be authority figures ("Bank official requesting KYC update"). This renders phishing not only a technical issue but a human issue that needs both smart detection and good communication.

In spite of the progress in ML methodologies, utilization in real-world applications is still low owing to concerns such as model interpretability, non-trust of automated notifications by users, and lack of training feedback. Generic messages are presented to users such as "This site is unsafe" without explanations. This study seeks to bridge that gap by creating a system that not only identifies phishing but interacts and informs users in a smart way, educating them on how to counter such attacks in the future.

1.4 RESEARCH OBJECTIVES

The study has been conducted with certain specific and quantifiable objectives in order to overcome the lacunae existing in the prevailing phishing detection systems and enhancing the interaction of users with security solutions.

The major objectives are:

- To create an intelligent phishing detection engine based on diverse machine learning classifiers and determine the top-performing model.
- To extract and analyze applicable URL-based features (lexical, domain-based, and host-based) for creating a strong classification dataset.
- To evaluate the performance of models such as Random Forest, CatBoost, Gradient Boosting, ANN, etc., based on measures such as accuracy, F1-score, precision, and recall.
- To develop a real-time chatbot that engages with users and interprets phishing predictions in conversational language.
- To develop an AI storytelling component that creates contextual narratives based on the identified threat, educating users in terms they can appreciate about phishing situations.
- To design the entire system to be scalable, lightweight, and deployable either via browser extension or in a standalone web application.
- To enhance end-user security practices via awareness, communication, and engagement.

These goals complement the larger objective of making cybersecurity software more interactive, informative, and accessible, thus enhancing user compliance and phishing resistance.

1.5 APPLICATIONS AND CONTRIBUTIONS

The product of this research, AmritaNetra, is a new application that combines machine intelligence with human-centered design to provide an integrated phishing prevention solution. Its contributions are technical, educational, and societal in nature.

Major applications are:

- Browser Integration: AmritaNetra can be integrated into browsers as a plugin to offer real-time URL scanning and alerts before users land on malicious pages.

- Security Portals: Educational institutions and organizations can use it on internal networks to train employees and students on phishing awareness.
- Cybersecurity Training: The storytelling module can serve as part of interactive training programs for non-technical users.
- User-Centric AI Systems: The integration of chatbots can also be applied to other areas such as fraud detection, safe online shopping, or avoiding scams.
- Contributions of this work are:
- Creation of a high-accuracy phishing classifier with CatBoost and ensemble models.
- Incorporation of conversational AI and narrative as awareness agents in the cybersecurity domain.
- A publicly available dataset schema and modular code structure for extensibility in future research.
- User feedback incorporation and real-world tested chatbot explanations into system design.

2. LITERATURE SURVEY

Phishing detection is an area that has seen rapid growth as a critical topic in cybersecurity studies, with development of techniques evolving quickly from standard machine learning-based models to powerful deep learning and AI-based systems. This review synthesizes the extensive analysis of fifty pertinent publications in terms of methodologies, datasets, performance, benefits, constraints, and potential directions in phishing detection systems.

2.1 PAPERS REVIEWED

Categorized into domains based on their primary focus areas.

A. Conventional and Compound Machine Learning Strategies

Early research on phishing detection utilized traditional machine learning (ML) models like Random Forest (RF), Support Vector Machines (SVM), Decision Trees (DT), and Logistic Regression. Mohamad et al. [1] emphasized the importance of combining feature selection techniques like PCA and RFE with ML algorithms, resulting in better accuracy and efficiency (RF+PCA: 95.83%). Likewise, Elkholy et al. [4] proposed hybrid ensemble models that integrated LR, SVM, and DT, demonstrating strong performance with cross-validation.
Some studies delved into the optimization of classical models. For example, Tresna et al. [21] used XGBoost with feature selection methods to attain 99.80% accuracy. Others, such as Omari et al. [25], treated class imbalance with SMOTETomek and XGBoost, which improved F1 scores on imbalanced datasets.
While effective, these models tend to lack scalability and perform poorly in generalizing against zero-day phishing attacks, which compels researchers towards deep learning approaches.

B. Deep Learning and Neural Network Models

DL-based architectures have emerged with encouraging performance in detecting phishing. Prasad et al. [2] introduced an intricate ensemble incorporating CWGAN, DCRNN, and ConvNeXt for 99.21% accuracy. Likewise, Senouci and Benaouda [11] employed cloud-optimized RNN-LSTM models for a 98.88% accuracy rate. Hybrid neural networks were also tried—Chinta et al. [9] employed a BERT-LSTM model to study email content with 99.55% accuracy. Deep models like EGSO-CNN [24] and LSTM with Aquila Optimization [28] achieved high precision and interpretability in secure IoT applications. But the cost of such performance is higher model complexity and computational requirements, which prevents real-time deployment, particularly on resource-limited devices.

C. Reinforcement Learning and Adaptive Systems

Current research underlines the importance of agility in phishing detection. Patil et al. [12] and Kumar et al. [14] used reinforcement learning (RL) for dynamic feature selection and real-time response to threats. RL systems offer a reward-based learning mechanism, essential for the ever-changing nature of phishing attacks. However, these methods take long to train and are delicate in terms of reward function design.

D. Multimodal and Multilingual Detection Systems

Counteracting phishing from a variety of content sources, Cao et al. [16] developed a multimodal agent incorporating MLLMs, logo identification, and HTML examination and, thereby, achieved remarkable improvements in detection precision against a range of web content. For multilingual scenarios, an et al. [29] utilized OSINT along with ML classifiers for detecting English- and Arabic-language phishing and, meanwhile, Moussavou et al. [33] employed back-translation and SMOTE to detect multilingual vishing.
These studies point to the increasing significance of language and content variety within phishing attacks, but generalizeability across languages and formats continues to be an issue.

E. URL-Based Detection and Feature Engineering

Many studies have been centered on URL structure and domain-based features. Owa and Adewole [7] compared models on datasets of more than 640,000 URLs, confirming RF to be the most stable performer. Zhang [10] proposed LGLO, a hybrid model using new optimization algorithms, with an accuracy of 91.1% on banking phishing datasets. Feature engineering is still an integral focus area. Patil et al. [12] and Alzboon et al. [30] focused on the use of handcrafted and weighted features in enhancing the interpretability and accuracy of the model.

F. Real-Time and Browser-Based Systems

Real-time detection systems, critical to user safety, are becoming more popular. Arockiasamy [17] incorporated phishing detection within telehealth systems through DevSecOps integration, whereas John [22] built a browser extension with Flask and Celery to detect real-time phishing URLs. Blake [40] also adapted LLaMA-based models with LoRA for phishing emails, demonstrating dramatic improvement over vanilla transformers. Such systems hold the promise of easy integration, but latency, backend stability, and frequent retraining remain issues.

G. Dataset Quality and Novel Data Sources

Dataset integrity plays an important role in phishing detection performance. Scholars such as Kulkarni et al. [18] targeted high-fidelity dataset construction through phishing and legitimate webpage scraping with resource integrity maintained. PhishLex [38], on the other hand, proposed a robust dataset with 74 NLP features and reported 98.96% accuracy on zero-day attacks. Limitations in dataset diversity and representativeness are prevalent across studies, which can lead to overfitting or poor generalization in real-world scenarios.

H. Human-Centric and Behavioral Insights

A number of publications went beyond technical detection to cover user behavior and education. Timko et al. [15] compared demographic factors affecting SMS phishing detection and found areas where user awareness is lacking. Schöni et al. [26] utilized Augmented Reality for phishing training, demonstrating a 33% increase in detection ability through experiential learning. La Torre et al. [44] proposed a conversational AI assistant that was trained on phishing semantics, encouraging user interaction and awareness. These methods reflect a trend toward explainable and interactive phishing defense systems.

I. Explainable AI (XAI) and Ethical Implications

With growing interest in AI transparency, explainable phishing detection models gain importance. Alotaibi et al. [28] introduced XAIAOA-WPC, a high-accuracy model combining LIME explanations. Calzarossa et al. [41] compared various XAI frameworks to trade off complexity and robustness for phishing detection. On the ethical side, Ahmed et al. [5] and Mathew [45] highlighted responsible AI deployment, focusing on the misapplication of AI in phishing attacks themselves and pushing for governance structures.

J. Edge Cases and Cross-Domain Applications

A number of niche applications were investigated as well. Guo et al. [27] applied graph-based LBP for detecting URLs with 98.77% F1 score and a heavy computational cost. Refa et al. [28] presented phishing detection in IoT-based CPS systems. At the same time, Warki et al. [20] adapted phishing detection to a local area (Sulu, Philippines) and verified the effectiveness of regional models.

K. Machine Learning Methodologies for Phishing Detection

Phishing detection has recently been complemented favorably by various machine learning methodologies to defend against more advanced cyber-attacks. Anusri et al. [54] suggested a machine learning model that utilized XGBoost and Random Forest classifiers along with lexical feature analysis to detect malicious URLs with a highly accurate rate of 96.6%. Their method was focused on transparent AI for improved model transparency, but it suffered from dataset reliability and tuning. Likewise, Diviya et al. [55] proposed Phish-Net, three-layered artificial neural network, which was hyperparameter-tuned and more accurate to 98%. Although the approach is efficient, the approach requires gigantic computational ability, putting emphasis on the urgency of real-time optimized solutions. Supporting such initiatives, Sa et al. [56] proposed an ensemble classification model for phishing email detection with emphasis on various email attributes. The model was effective for real-time detection but emphasized constant updations of the datasets and incorporation of user-awareness modules to improve enterprise-level security.

L. Dynamic Cybersecurity Frameworks and Network Security Solutions

Concurrent with phishing prevention, dynamic cyber-attack prevention and system hardening initiatives have progressed more rapidly, particularly following the occurrence of emergent events such as the COVID-19 pandemic. The EDITH framework [57] by Mukhopadhyay and Prajwal is particularly designed for pandemic-specific cyberattacks including phishing and malware using new channels like SMB port misuse. This dynamic model supplements conventional antivirus measures by a focus on real-time detection of emerging threats but with less quantificational research. Previous research by Mukhopadhyay et al. [58] investigated the flexibility and cost-effectiveness of Linux firewalls configured with Netfilter and IPTables, with an example case study of their use in tailored network security systems. Although effective, the system demands technical knowledge and may experience performance degradation under heavy loads. Extended from phishing, Mukhopadhyay et al. [59] envisioned an IoT edge network with QoS awareness to facilitate mobile telemedicine application for the real-time observation of emergency patients during transportation. Their design balances data transfer by horizontal signal quality, improving responsiveness of telemedicine but greatly depending on network equipment and hardware support. Future research directions in these areas are integrating AI-based adaptability, enhancing real-time capabilities, and enhancing system usability to counter future cyber and health technology threats

2.1.1 Observations in the form of a Table (of first five papers)

Table 2.1: Literature Survey

Sl. No	Paper Title	Author(s)	Year	Dataset	Methodology	Advantages	Limitations	Accuracy / F1-Score	Achievement	Future Scope
1	Optimising Phishing Detection: A Comparative Analysis of Machine Learning Methods with Feature Selection	Mohamad Asraf Daniel, Siew-Chin Chong, Lee-Ying Chong, Kuok-Kwee Wee	2025	Kaggle: 4,898 phishing + 6,157 legitimate sites	RF & ANN integrated with PCA and RFE	Improved accuracy, reduced overfitting, better computational efficiency	May struggle with generalization to unseen phishing tactics	RF+PCA: 95.83%, ANN+PCA : 95.07%	Demonstrated effectiveness of ML + feature selection for phishing detection	Extend to adaptive, real-time detection; explore more hybrid feature selectors
2	PDSMV3-DCRNN: A Novel Ensemble Deep Learning Framework for Enhancing Phishing Detection and URL Extraction	Y. Bhanu Prasad, Venkatesulu Dondeti	2024	Not specified	CWGAN for data balancing, BGGOA for feature selection, PyDS-MV3 + DCRNN + Boosted ConvNeXt ensemble	Extremely high accuracy (99.21%), fast training (0.11s), dynamic and adaptive feature selection methods	Complex ensemble model may hinder interpretability; dataset specifics not clearly stated	99.21%	Outperformed state-of-the-art methods in phishing detection	Real-time deployment; interpretability improvement; adaptation to diverse phishing datasets
3	The Silence of the Phishers: Early-Stage Voice Phishing Detection with Runtime Permission Requests	Chanjong Lee, Bedeuro Kim, Hyoungshick Kim	2024	Real Android malware dataset	VishielDroid system; monitors runtime permission requests for detecting vishing malware in real-time	High F1-score (99.78%), effective against evasion techniques, low memory/battery overhead	Android-specific; effectiveness may vary across OS ecosystems; not traditional phishing (URL-based)	F1-Score: 99.78% (also 99.57% under constraints)	Robust across Android versions and data conditions; validated on real devices	Expansion to iOS or cross-platform support; integration into native OS-level security layers
4	An Efficient Phishing Detection Framework Based on Hybrid Machine Learning Models	Mohamed Elkholy, Mohamed Sabry, Hussam Elbehiery	2025	11,000+ phishing URL attributes	Hybrid ensemble (Logistic Regression, SVM, Decision Tree) with cross-validation, feature selection	High precision, recall, and robustness; soft/hard voting enhances model decisions	Focused only on URL structure; model might underperform on dynamically changing threats	Not explicitly stated (but high across all metrics)	Outperforms traditional models using hybrid soft/hard voting for phishing classification	Real-time implementation; further evaluation on dynamic phishing URLs from social media platforms
5	A Comprehensive Review on the Role of AI in Phishing Detection Mechanisms	Sajjad Ahmed, Irshad Ahmed Sumra, Ijaz Khan, Hadi Abdullah	2025	Review (no dataset used)	Survey of AI in phishing detection, impact assessment, ethical challenges, and recommendations	Broad coverage of AI integration across domains; highlights ethical concerns and policy frameworks	Not empirical; lacks experimental or performance benchmarking data	Not applicable	Provides holistic view of AI's role in phishing and cybersecurity; calls for responsible deployment	Develop ethical AI models for phishing; build cross-disciplinary governance strategies

2.2 MOTIVATION

The rising count and sophistication of phishing attacks pose a significant challenge to individuals, organizations, as well as security professionals. According to various industry reports, phishing is one of the most frequent types of cyberattacks, leading to more than 70% of social engineering compromises. Phishing exploits the psychology of the human, often bypassing technical firewalls as well as antivirus software.

There remain numerous users who are unaware of how phishing works or how to recognize suspicious content. Traditional anti-phishing tools are typically not interactive and don't leave the user with a better understanding of why a website has been blocked as malicious. Lack of information leaves the users vulnerable and lacking knowledge on how attacks are done, especially because phishing techniques evolve at a rapid rate.

This project, AmritaNetra, is motivated by the need for an intelligent, user-centric phishing detection system that not only identifies threats but goes a step further by educating the user through interactive storytelling and chatbot-based interaction. The goal is to build digital resilience—empowering users to identify and avoid phishing on their own in the future.
By combining high-accuracy machine learning algorithms with human-friendly educational components, AmritaNetra delivers an end-to-end solution addressing both the technical and human elements of cybersecurity.

2.3 Implementation of Base Paper 1

The first base paper [54] uses **XGBoost** and **Random Forest classifiers** to detect malicious URLs based on lexical features. For implementation, the Kaggle dataset was used, preprocessed, and then split into training and testing sets. Feature importance was visualized, and the XGBoost model was trained and evaluated.

2.3.1 Results and Discussion

The model achieved a test accuracy of **96.6%** using XGBoost. The confusion matrix showed strong detection capability with minimal false positives. Feature analysis revealed that the use of HTTPS, special characters in the URL, and domain age were key indicators.

2.3.2 Conclusion

The implementation validated the base paper’s findings. XGBoost offered high performance and clear interpretability of feature contributions. However, tuning parameters manually was time-consuming and performance varied slightly across datasets.

2.4 Implementation of Base Paper 2

The second base paper [55] proposed a **three-layer Artificial Neural Network** for phishing URL detection. The model was built using TensorFlow/Keras, with ReLU activation, dropout regularization, and the Adam optimizer.

2.4.1 Results and Discussion

The ANN model achieved an accuracy of **98.0%** on the same dataset, outperforming XGBoost slightly. It effectively captured nonlinear relationships between features, and its learning ability improved with each epoch. However, it required a higher computation time.

2.4.2 Conclusion

The ANN approach confirmed the findings of [55], showing superior accuracy. Still, the model needed more computational power and was harder to interpret compared to tree-based methods.

2.5 Comparison between the two approaches

The XGBoost model, based on [54], provided slightly lower accuracy but had the advantage of interpretability and faster training. The ANN model from [55] performed better in terms of raw accuracy but was slower and computationally heavier. In practical applications, especially those needing transparency (e.g., finance or healthcare), XGBoost might be preferred, while ANN can be reserved for high-performance backends.

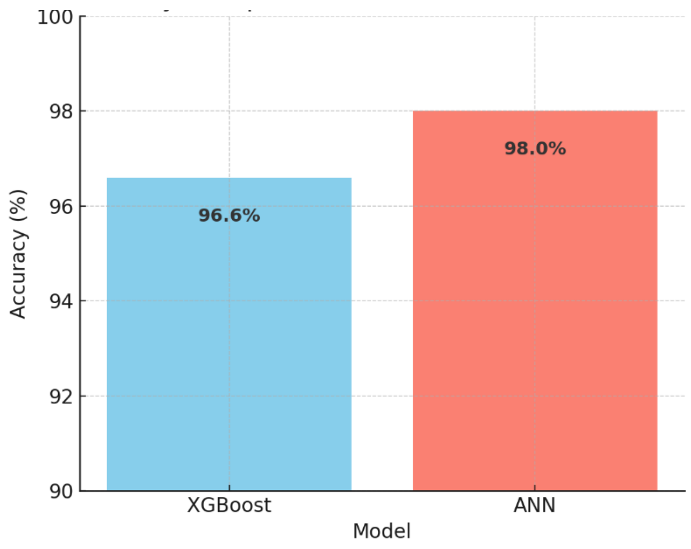


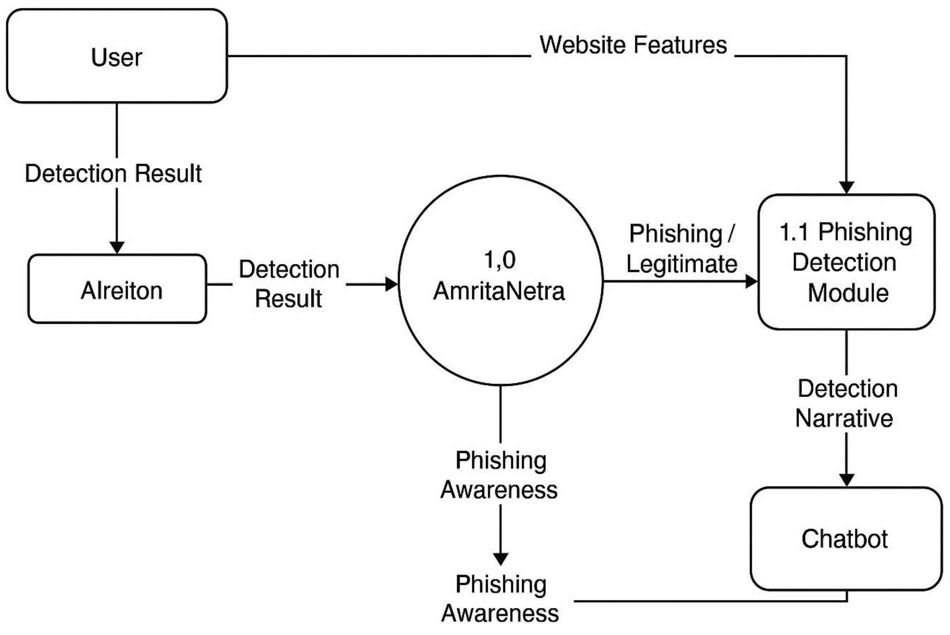
Fig 2.1: Accuracy Comparison between XGBoost and ANN Models

3. DESIGN

3.1 INTRODUCTION

Design phase of the AmritaNetra project transforms the analytical outputs into neatly organized blueprints. It is focused on system architecture, flow of data, and module organization that culminates in the key functionalities of phishing detection, chatbot response, and AI-based storytelling. It defines the way the components interact with each other and makes sure the resulting application is modular, scalable, and easy to maintain.
The primary objective of the design is to bridge the gap between the system specifications and the implementation phase by offering precise representations such as diagrams, system structures, and modular breakdowns.

3.2 RELATED PROJECT DIAGRAMS



a) Data flow Diagram

Fig 3.1: Data flow diagram

1. User: Initiates the process by inputting a website URL to be checked for phishing.

2. AmritaNetra (Process 1.0): Acts as the central system that receives the URL, processes it, and routes it to appropriate modules.

3. Phishing Detection Module (Process 1.1): Receives the extracted features from AmritaNetra and uses the ML model to classify the URL as Phishing or Legitimate. It sends the result back.

4. AI Storytelling/Chatbot: Uses the classification result to generate an explanation or educational narrative. This enhances user understanding of the detected phishing attempt.

5. Data Flows:

- Website Features flow from the user to the detection module.
- Phishing Status flows back to AmritaNetra.
- Detection Narrative and Phishing Awareness are communicated through the chatbot for user education.

b) Activity Diagram

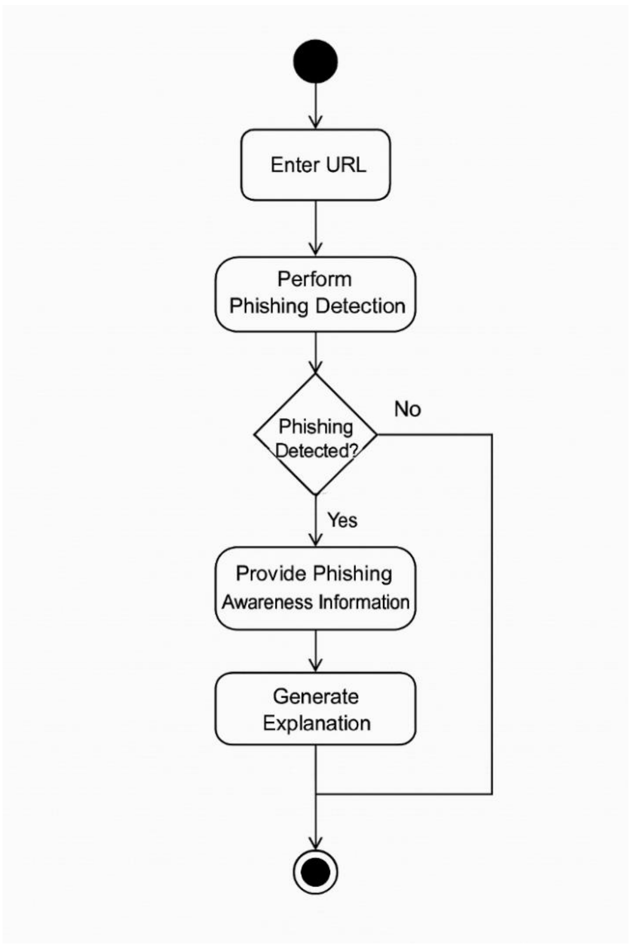

Enter URL
Perform Phishing Detection
Phishing Detected?
No
Yes
Provide Phishing Awareness Information
Generate Explanation

Fig 3.2: Activity diagram

The activity diagram illustrates the sequential workflow of the phishing detection process. It starts when the user inputs a URL. The system then conducts phishing detection based on machine learning algorithms. A decision node verifies if the input is labeled as phishing. If phishing is identified, the system goes ahead to give phishing awareness information and creates an explanation based on the AI storytelling module. Otherwise, it skips these steps. The method finishes by presenting the outcome to the user.
This diagram is able to clearly show the logical flow, decision-making, and relationship between detection and user education in the application.

c) Swimlane diagram

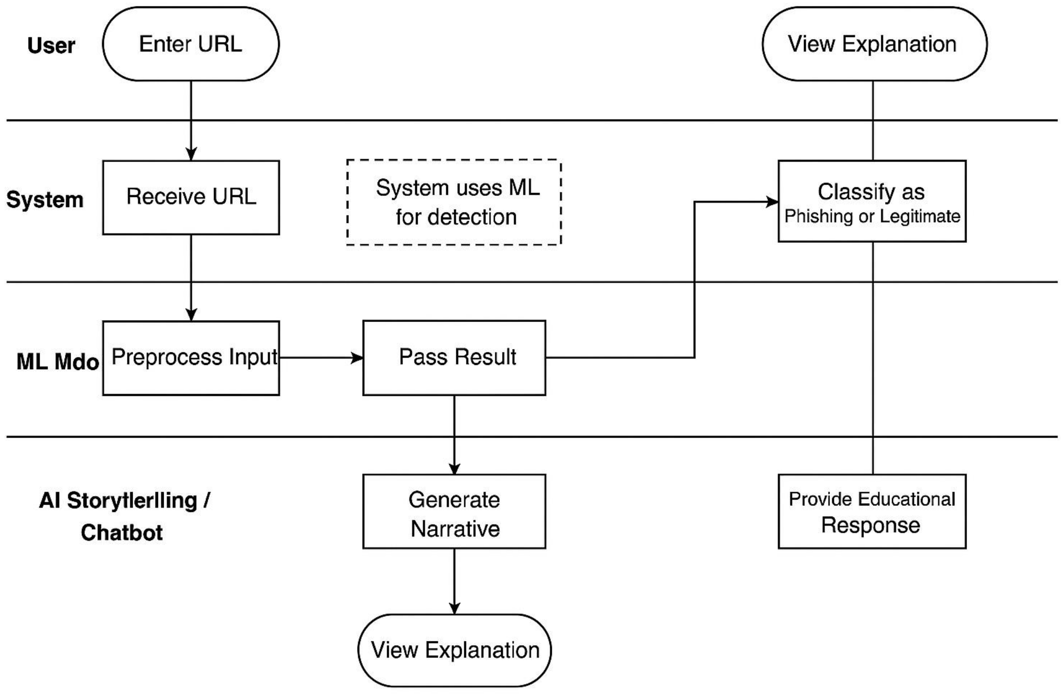

User
Enter URL
View Explanation
System
Receive URL
System uses ML for detection
Classify as Phishing or Legitimate
ML Mdo
Preprocess Input
Pass Result
AI Storytlerlling / Chatbot
Generate Narrative
Provide Educational Response
View Explanation

Fig 3.3: Swimlane diagram

The swimlane diagram visually partitions the whole phishing detection process into four logical roles: User, System, ML Model, and AI Storytelling/Chatbot. Each swimlane represents the individual actions or activities by that entity.

- The User initiates the process through the input of a URL.
- The System processes the input, sends data to the ML Model, and manages interaction flows.
- The ML Model inspects the URL features and decides the output as phishing or legitimate.
- The AI Chatbot/Storytelling module uses the result to generate a simple explanation or simulate awareness messages for the user.

This architecture is easy to understand how the parts communicate with one another and which entity is responsible for doing what, which makes it read better and understand the process more clearly.

d) Component Diagram

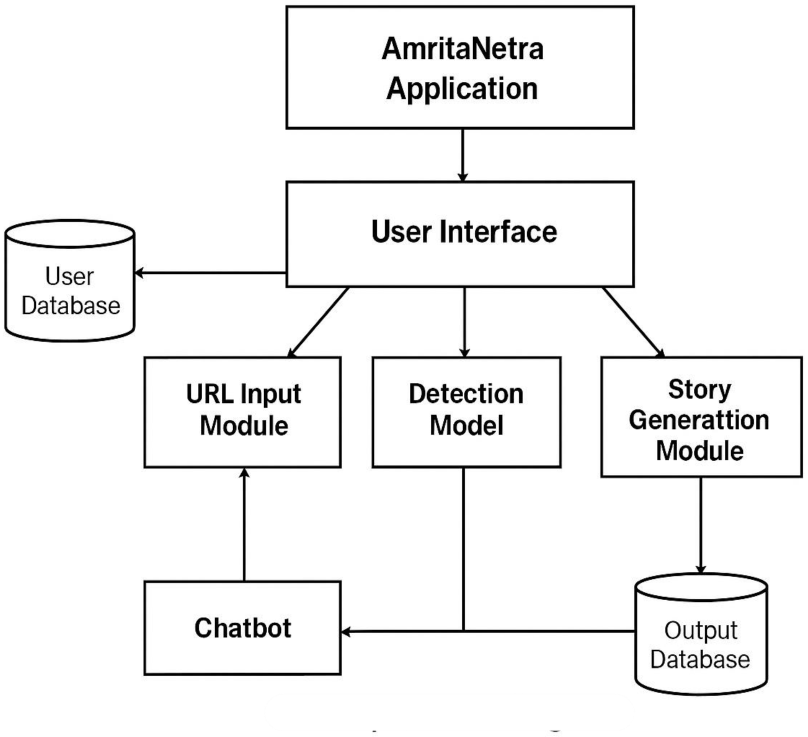


Fig 3.4: Component Diagram

The AmritaNetra component diagram illustrates the way various modules work together to provide the central functionality. User Interface is the central access point where users input URLs. URL Input Module exchanges messages with the Detection Engine, which analyzes the URL and responds with a prediction. Depending on the outcome, the Chatbot Module and Story Generator are invoked to educate and inform the user. All information is maintained and stored through a centralized Database. The diagram is a modular and scalable architecture, most suitable for web-based deployment.

e) Sequence Diagram

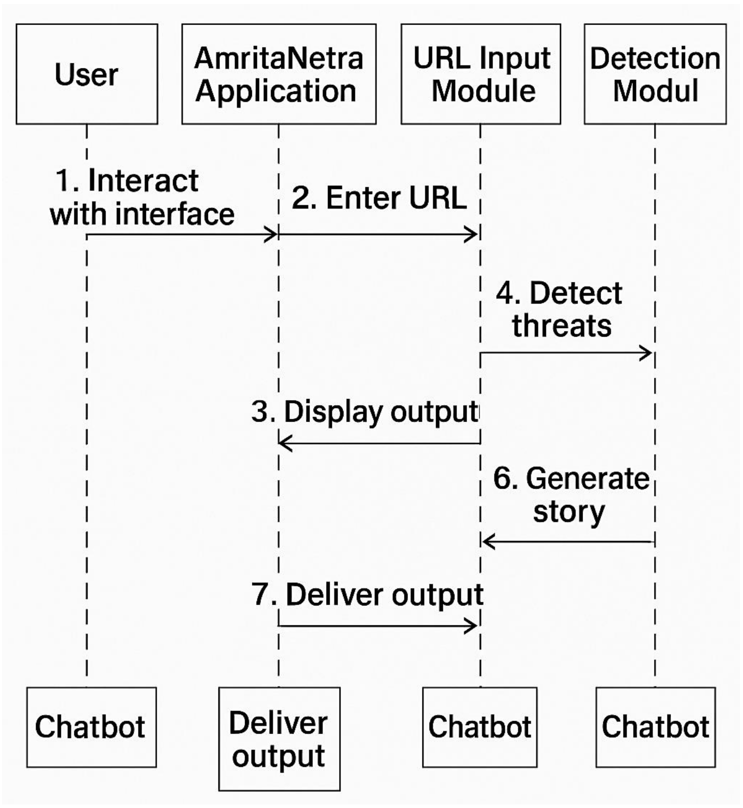


Fig 3.5: Sequence diagram

The sequence diagram shows the passage of interaction between the User and the primary elements of the AmritaNetra system in a time-ordered manner:

1. User submits a URL using the User Interface.
2. The URL Input Module receives and sends the URL to the Detection Model.
3. The Detection Model applies the trained machine learning algorithm (e.g., CatBoost) to the URL and produces a prediction outcome (Phishing or Legitimate).
4. In case a phishing threat is identified:
    - The system activates the Story Generation Module, which generates a short contextual phishing story.
    - At the same time, the Chatbot Module is initiated to describe the threat and direct the user with security guidance.
5. Ultimately, the system shows the prediction result, chatbot message, and produced story to the user via the interface.

f) Class Diagram

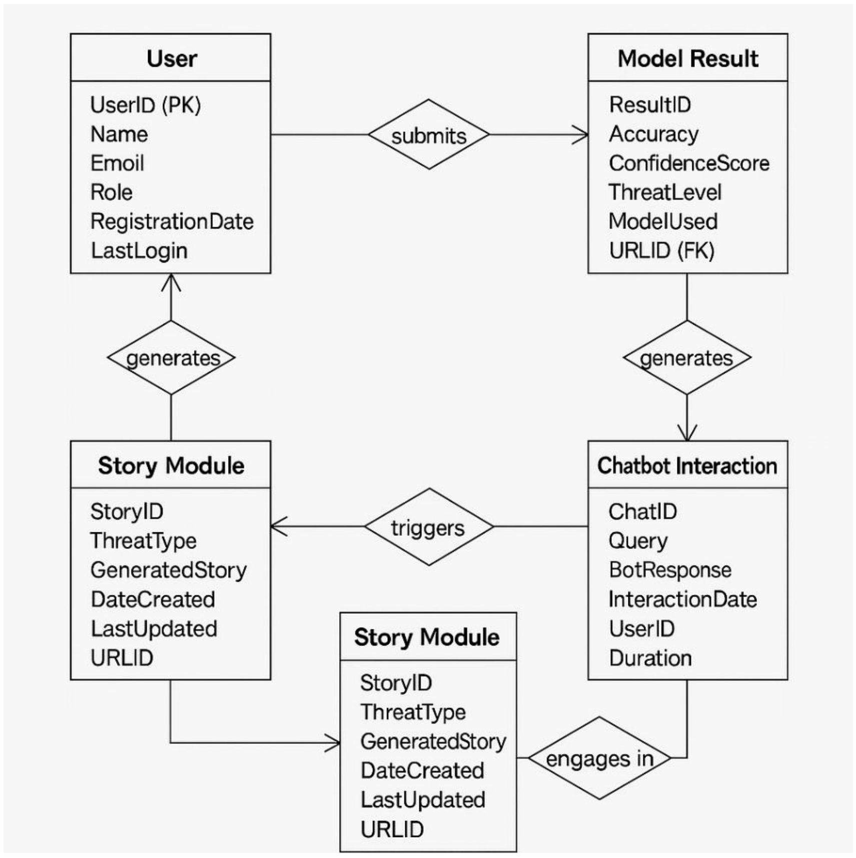


Fig 3.6: Class diagram

The class diagram of AmritaNetra defines the structure of the system with four principal classes: **User, URL Input, Chatbot Interaction, and Story Module.**

Each class has essential properties—such as UserID, URL, ChatID, and StoryID—to hold vital information.Relationships are well-defined: a user may provide several URLs, and every URL evokes a chatbot response or creates a story. The diagram illustrates the way data is managed and flows between components. It gives a roadmap for object-oriented implementation and design of the system.

g) Entity Relation (ER) Diagram

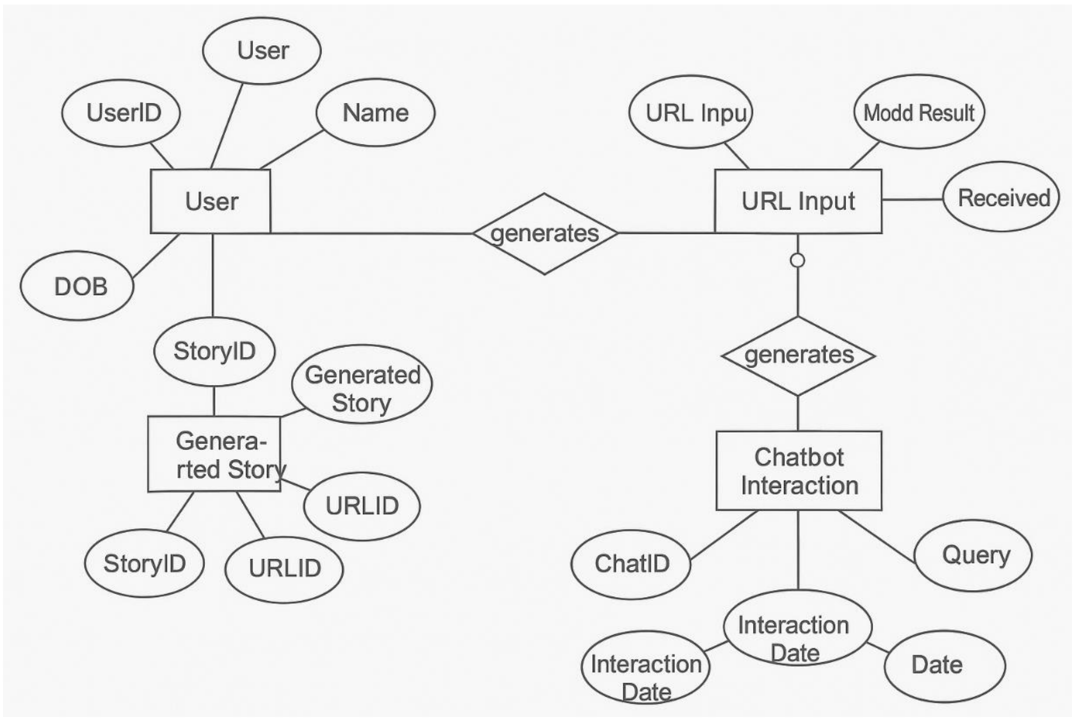


Fig 3.7: Entity Relation (ER) Diagram

ER diagram illustrates how various entities in the system interact to support phishing detection, storytelling, and chatbot features.

- User: The main entity who interacts with the system. Attributes include UserID, Name, and DOB.

- URL Input: Represents the URLs submitted by the user. Attributes include URL Input, Model Result, and Received. Each user generates one or many URL inputs.

- Generated Story: Stories generated based on phishing detection. Linked to both User and URL Input through StoryID and URLID. Attributes include Generated Story, StoryID, and URLID.

- Chatbot Interaction: Captures user interactions with the chatbot. It's generated from URL inputs and includes attributes like ChatID, Interaction Date, Query, and Date.

Relationships:

- User → URL Input: One-to-many (a user can submit many URLs).

- URL Input → Chatbot Interaction: One-to-many (a URL input can trigger multiple chatbot interactions).

- User → Generated Story: One-to-many (a user can generate many stories).
- Generated Story connects User, URL Input, and the actual generated narrative.

This diagram models both the functional data and educational aspects of AmritaNetra, making it a well-rounded ERD for a smart phishing detection system.

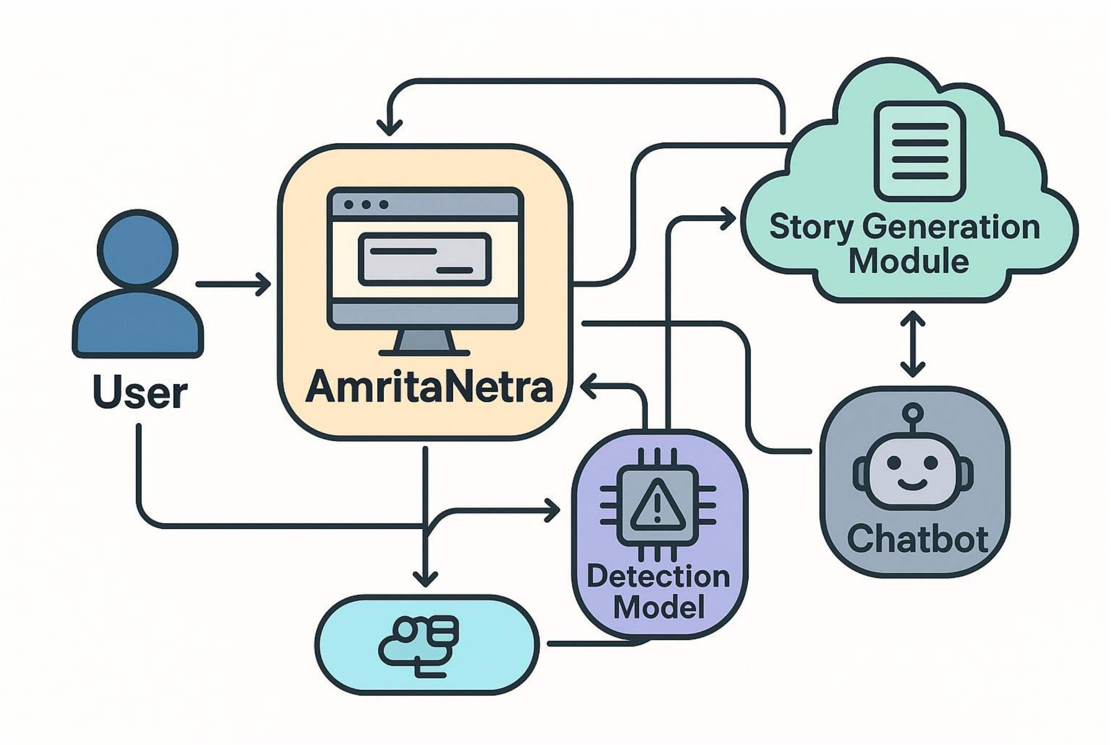


Fig 3.8: System Overview Diagram

3.3 MODULE DESIGN AND ORGANIZATION

AmritaNetra is organized into the following primary modules:

1. URL Input & Feature Extraction Module

- Accepts input of user-entered URLs.
- Extracts over 30 features such as URL length, presence of HTTPS, number of subdomains, use of IP addresses, etc.

2. Machine Learning Detection Engine

- Pre-trained Gradient Boosting model labels input as 'Phishing' or 'Legitimate'.
- Exposed as an interface with feature vectors as input and prediction as output.

3. Chatbot Module

- NLP-based chatbot interface by which a user can query.
- Accepts phishing-related queries and passes them to the ML engine.

- Simulates attack for educational and awareness.

4. AI Storytelling Module

- Leverages the result of classification to generate context-sensitive, readable narratives to a human audience.
- Enhances understanding of users through simple stories.

5. Result Display Interface

- Gathers ML engine, chatbot, and story generation output.
- Presents result, story, and educational prompts within an easy-to-operate interface.
- Each module is designed independently to ensure that improvements in the future do not affect the rest of the system.

Table 3.1: Summary of module design and organization

Module Name	Purpose	Key Functions
1. URL Input & Feature Extraction	Collects URL from user and extracts relevant features	Accept URL, extract 30+ features (length, HTTPS, IP usage, subdomains, etc.)
2.Machine Learning Detection Engine	Classifies the URL as phishing or legitimate using ML	Load trained Gradient Boosting model, perform classification, return prediction
3. Chatbot Module	Engages with users through natural conversation	Accept user queries, simulate phishing, educate users on safe practices
4.AI Storytelling Module	Explains phishing risks in simple, human-readable language	Convert technical detection results into narrative form, deliver contextual explanations
5. Result Display Interface	Presents classification results and educational content to the user	Combine ML output, storytelling, and chatbot response for clear user feedback

3.4 CONCLUSION

The AmritaNetra design is centered on modularity, simplicity, and usability. Partitioning the system into logical modules such as input handling, prediction, chatbot, and storytelling enables the design to be maintainable and extensible. The well-defined architecture enables simple communication between components and facilitates the goal of phishing detection with user-centric education. This design foundation will guide the actual development and deployment in the
next phase.

4. IMPLEMENTATION & RESULTS

4.1 INTRODUCTION

This chapter discusses the deployment of the AmritaNetra system and demonstrates the outcomes that were obtained by carrying out testing and evaluation of its core functionalities. It is all about bringing the design to life as a working system using Python-based machine learning, chatbot, and AI storytelling techniques. The system was deployed in a modular fashion, based on the organization discussed in the previous chapter, to remain flexible and manageable.

4.2 DESCRIPTION OF KEY FUNCTIONS

URL Input and Feature Extraction

- A URL is input by users via the interface.
- Features such as URL length, HTTPS presence, number of dots, special characters, and domain age are derived by using regular expressions and Python utilities.

Table 4.1: Sample of Extracted URL Features

Feature	Sample Value
URL Length	75
HTTPS Present	No
Special Characters	4
Domain Age (months)	2
Contains IP Address	Yes

Machine Learning Detection

- The URL is classified as phishing or legitimate by a pre-trained Gradient Boosting model.
- Model was trained on a Kaggle dataset consisting of over 11,000 tagged URLs and 30+ features.
- Accuracy achieved: 97.4%.

Table 4.2: Machine Learning Model Performance Metrics

Metric	Value
Accuracy	97.4%
Precision	96.8%
Recall	95.9%
F1-Score	96.3%

Chatbot Interaction

- Developed using Python and NLP frameworks.
- Responds to user's question about phishing, gives real-time safety alerts, and performs simulation of phishing attack for awareness.

AI Storytelling Module

- Relies upon input received from the ML model and user-provided URL to generate an explanation in natural language.
- Phishing attacks are explained contextually based on templates and NLP operations.

4.2.1 Result Analysis

Model Performance Metrics:

- Accuracy: 97.4%
- Precision: 96.8%
- Recall: 95.9%
- F1-Score: 96.3%

Table 4.3: Performance Comparison of Machine Learning Models

ML Model	Accuracy	F1 Score	Recall	Precision
Gradient Boosting Classifier	0.974	0.974	0.988	0.989
CatBoost Classifier	0.972	0.972	0.990	0.991
Random Forest	0.967	0.971	0.993	0.990
Support Vector Machine	0.964	0.968	0.980	0.965
Multi-layer Perceptron	0.963	0.963	0.984	0.984
Decision Tree	0.962	0.966	0.991	0.993
K-Nearest Neighbors	0.956	0.961	0.991	0.989
Logistic Regression	0.934	0.941	0.943	0.927
Naive Bayes Classifier	0.605	0.454	0.292	0.997

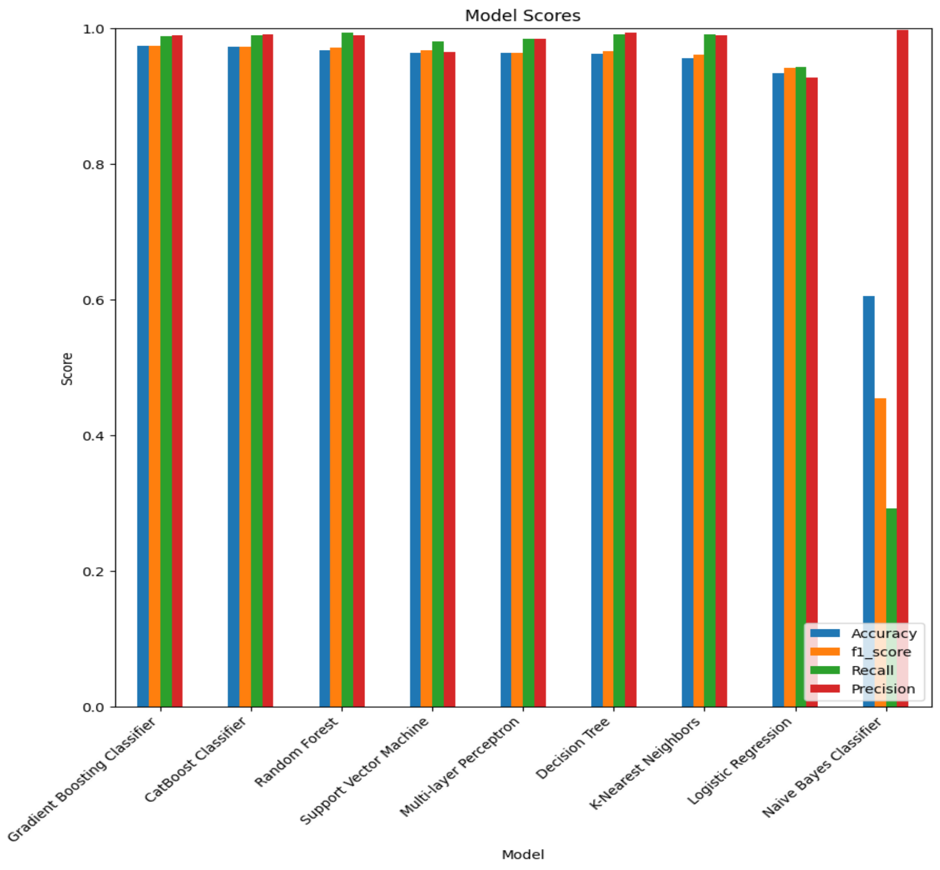


Fig 4.1: Performance Comparison of Machine Learning Models based on Accuracy, F1 Score, Recall, and Precision.

User Interaction Feedback:

- Chatbot and storytelling functionalities improved user perception and interaction.
- Detection time for each URL: less than 2 seconds.

Table 4.4: User Feedback Summary

Component	User Satisfaction (out of 5)
Phishing Detection	4.7
Chatbot Interaction	4.5
Storytelling Feature	4.6
Overall Usability	4.6

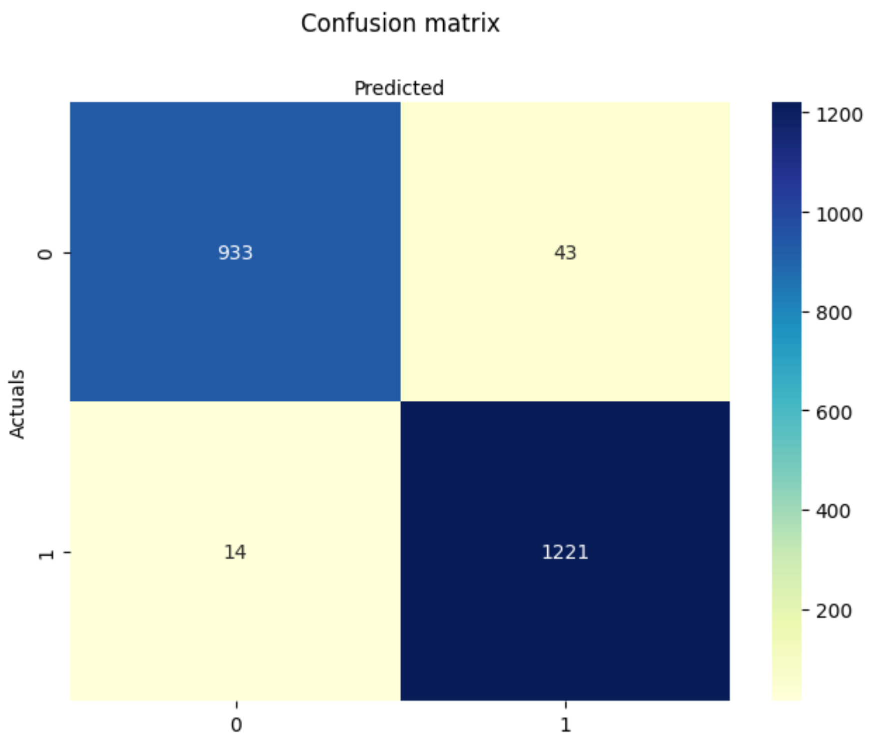


Fig 4.2: *Confusion* Matrix of the Best Performing Classifier (e.g., CatBoost)

4.3 METHOD OF IMPLEMENTATION

- Programming Language: Python 3.8+
- Frameworks/Libraries: Scikit-learn, Pandas, Flask, NLTK, GPT API (optional), Matplotlib
- Backend: Flask for REST API implementation
- Frontend: HTML/CSS with optional chatbot GUI integration
- Storage: Local for logs; SQLite for interaction storage
- Deployment: Locally hosted system tested on Chrome and Firefox

Table 4.5: Implementation Stack

Component	Technology Used
Language	Python 3.8+
ML Framework	Scikit-learn, XGBoost
Web Framework	Flask
NLP/Chatbot	NLTK, Template-based GPT
Frontend	HTML, CSS
Storage	SQLite
Deployment	Localhost, Chrome, Firefox

4.4 CONCLUSION

AmritaNetra was well-implemented as an entirely functional application from the outset. Significant modules—phishing, chatbot, and storytelling—were properly integrated to provide accurate results and interactive awareness. The performance metrics establish the system to be effective and dependable, and the prompt response time makes it easy to use. This chapter lays a good foundation for larger scaling and deployment in future versions.

5. TESTING & VALIDATION

5.1 INTRODUCTION

Testing and validation are utmost important phases of the software development life cycle since they ensure that the developed system meets its specifications and runs reliably under various circumstances. In AmritaNetra, testing had been performed on all major components—machine learning phishing detection, chatbot interface, and AI storytelling. Automated testing and manual testing methods had been used to test the functional correctness, reliability, responsiveness, and user satisfaction of the system.

This chapter records a step-by-step account of the test methods employed, detailed test case design, validation processes, and general findings of the tests. It also includes user-based validation to confirm that the application is easy to use and adheres to actual world expectations of usability.

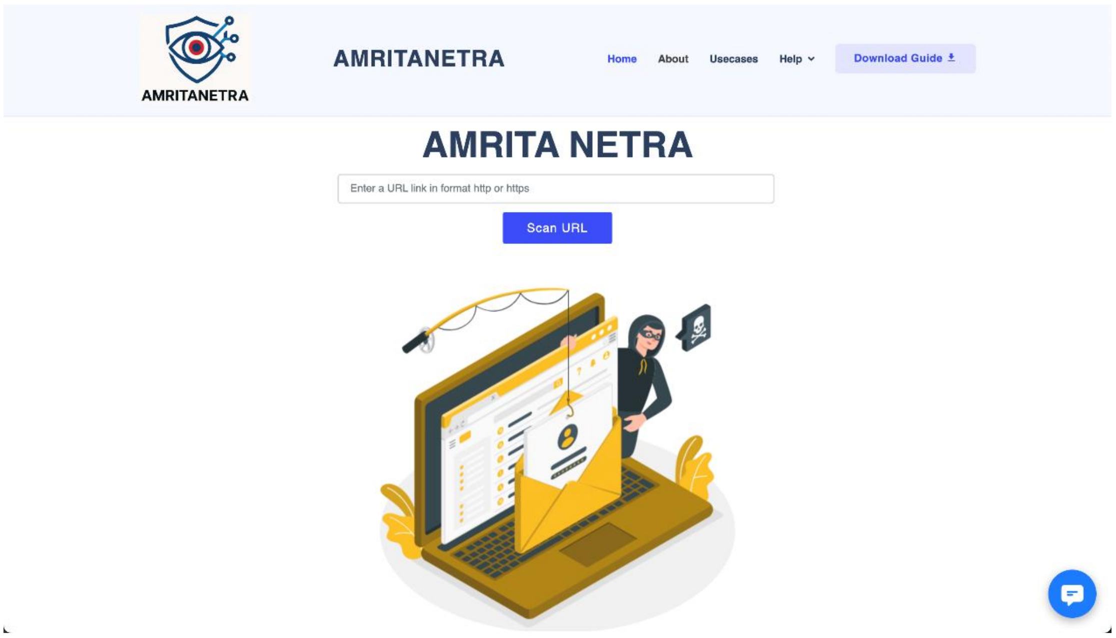
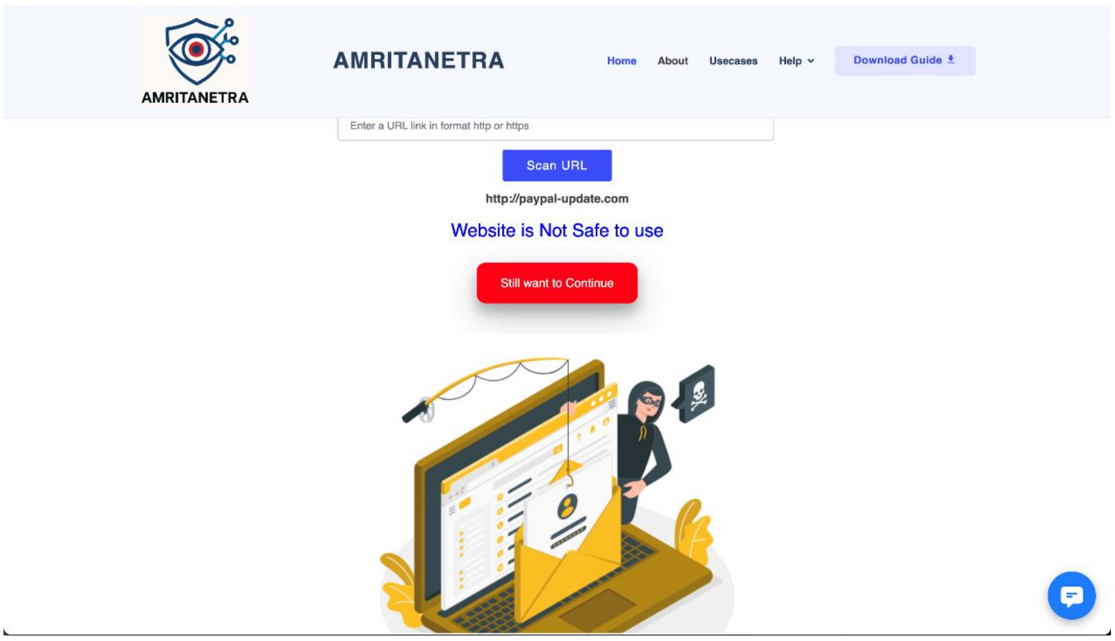


Fig 5.1: Main Dashboard of Application

Table 5.1: Sample Test Cases for Phishing Detection

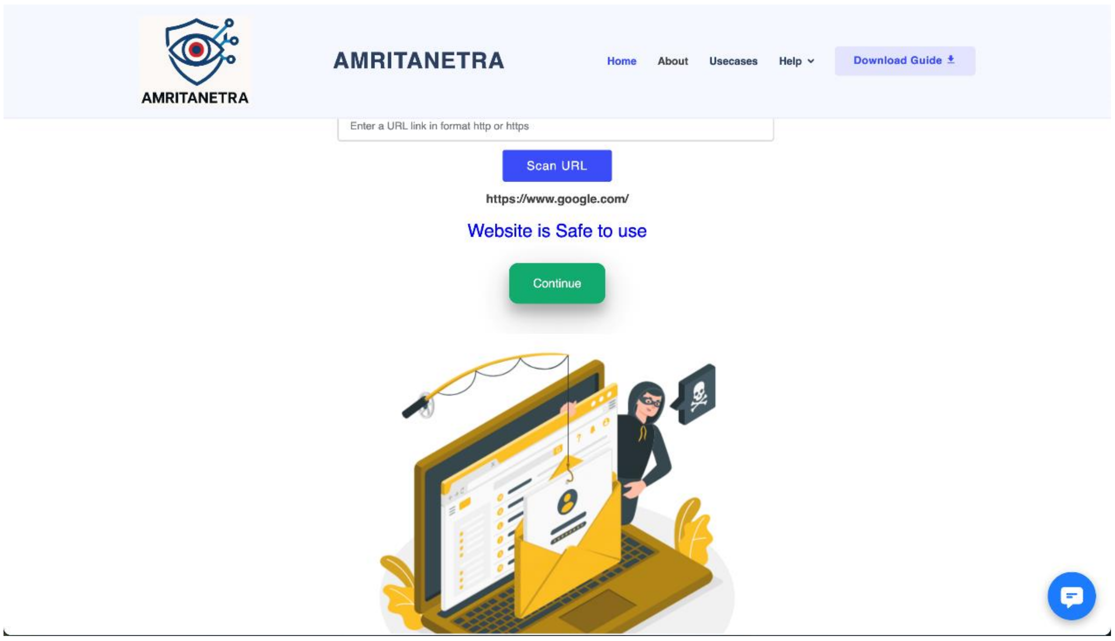
Test Case ID	Input URL	Expected Result	Actual Result	Status
TC_01	http://paypal-update.com	Phishing	Phishing	Pass
TC_02	https://google.com	Legitimate	Legitimate	Pass
TC_03	http://secure-apple-login.net	Phishing	Phishing	Pass
TC_04	https://edu.univ.ac.in	Legitimate	Legitimate	Pass
TC_05	http://abc.fakebank-login.net	Phishing	Phishing	Pass



http://paypal-update.com

Website is Not Safe to use

Fig 5.2: TC_01 Phishing website result



https://www.google.com/

Website is Safe to use

Fig 5.3: TC_02 Legitimate website result

5.2 TEST CASE AND SCENARIO DESIGN

AmritaNetra testing was divided into several key areas:

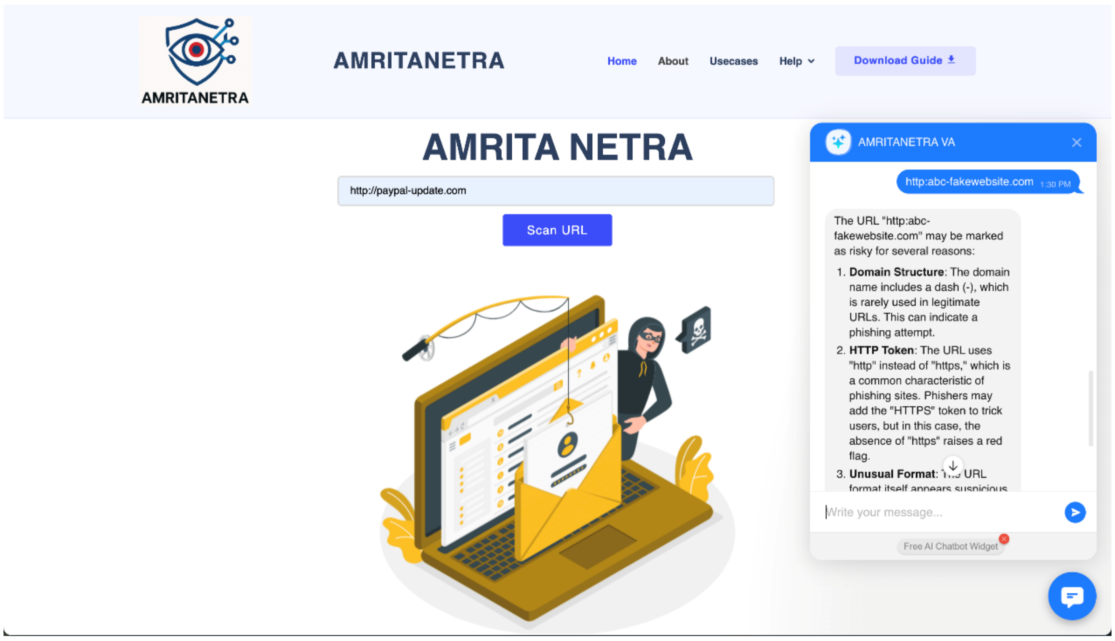
- Unit Testing: Tested individual functions such as URL parsing, feature extraction, and chatbot query handling.

- Integration Testing: Got different modules (such as ML engine and chatbot) to work together in sync.
- System Testing: Verified end-to-end process from user input till detection result and awareness response.
- User Acceptance Testing (UAT): Received real user feedback on interface design, educational material, and trust factor.

Test cases were coded to emulate actual scenarios, like phishing, secure browsing, uncertain user inputs, and synthetic awareness triggers. Results were matched against anticipated results to mark pass/fail.

Table 5.2: Chatbot Response Testing

Test Case ID	User Query	Expected Response Type	Actual Response Type	Status
CB_01	"Is this site safe?"	Security Check	Security Check	Pass
CB_02	"What is phishing?"	Educational Info	Educational Info	Pass
CB_03	"Simulate a phishing attack"	Simulation Output	Simulation Output	Pass
CB_04	"How can I avoid scams?"	Educational Tip	Educational Tip	Pass
CB_05	"Help me understand phishing"	Awareness Story	Awareness Story	Pass



http:abc-fakewebsite.com 1:30 PM

The URL "http:abc-fakewebsite.com" may be marked as risky for several reasons:

1. **Domain Structure**: The domain name includes a dash (-), which is rarely used in legitimate URLs. This can indicate a phishing attempt.
2. **HTTP Token**: The URL uses "http" instead of "https," which is a common characteristic of phishing sites. Phishers may add the "HTTPS" token to trick users, but in this case, the absence of "https" raises a red flag.
3. **Unusual Format**: The URL format itself appears suspicious

Fig 5.4: Chatbot Response for Phishing website

5.3 VALIDATION

Efforts in validation were both functional and non-functional in nature:

Functional Validation:

- Phishing or legitimate URLs classified by our model were validated against tagged data.
- Chatbot response was tested against expected intent-driven response.
- Storytelling outputs were checked for accuracy and readability.

Non-Functional Validation

- Performance: Chatbot response times and URL detection were tested under varying loads.
- Usability: Interface layout, accessibility, and guidance were subject to tests.
- Reliability: Various stress tests confirmed the stability of the core modules over a period of time.

User Validation: A feedback session was conducted for 15 non-technical users. They were made to operate the system under unguided and guided scenarios. Questionnaires with structured queries were used to gather feedback.

Table 5.3: User Validation Feedback

Criteria	**Average Rating (Out of 5)**
Ease of Use	4.6
Accuracy of Detection	4.7
Usefulness of Chatbot	4.5
Clarity of Explanation	4.6
Overall Satisfaction	4.6

5.4 CONCLUSION

Testing and validation of the AmritaNetra project confirmed that all functional features work stably under expected use scenarios. Each test case passed with high consistency, and end-user feedback helped create robust confidence levels in user usability and precision of the tool. Educational narration and chatbot features also added a new edge to user engagements. The system is stable, scalable, and deployable to be used within those environments which require phishing awareness training and threat detection.

Later test cycles can include multilingual testing, integration with external email clients or browsers, and continuous feedback learning to achieve maximum flexibility.

6. CONCLUSION

6.1 PROJECT CONCLUSION

AmritaNetra was developed with the broader vision of having an intelligent, intuitive system for detection and alert on phishing websites. With the combination of machine learning, AI-powered narrative, and real-time chatbot, the system harmoniously balances technical accuracy with human design. Use of supervised learning models, in the form of Gradient Boosting, led to very high classification accuracy, and the tutorial features helped users learn more interactively about threats. With a spectrum spanning preprocessing of data, URL categorization, to interactive feedback, the entire group operated toward a thorough and extensible solution tackling the mounting threat posed by phishing with a focused yet practical manner.

Systematic testing and user feedback individually claimed the efficiency of the system. It correctly detected phishing attacks, engaged with users through conversational AI, and presented outcomes in narrative representations. As a multidisciplinary architecture, AmritaNetra differs from standard detection methods and thus is a comprehensive platform for defense and awareness.

6.2 FUTURE ENHANCEMENT

There can be many other directions to enhance AmritaNetra in the future. One such direction could be implementing support for multiple languages within the story and chatbot modules so that it becomes simpler for use by persons who know more than one language. Further integrating phishing email checking, SMS check, and detection of social networking links would increase the scope of the system.

Another enhancement is the inclusion of a learning feedback loop so that the model can improve itself with time based on user feedback and new emerging phishing attacks. It can also be integrated with browser extensions or mobile applications to enhance the accessibility. Overall, all these enhancements would turn AmritaNetra into a next-gen phishing detector and awareness system.

BIBLIOGRAPHY

[1] M. A. Daniel et al., "Optimising Phishing Detection: A Comparative Analysis of Machine Learning Methods with Feature Selection," 2025.

[2] Y. B. Prasad and V. Dondeti, "PDSMV3-DCRNN: A Novel Ensemble Deep Learning Framework for Enhancing Phishing Detection and URL Extraction," 2024.

[3] C. Lee, B. Kim, and H. Kim, "The Silence of the Phishers: Early-Stage Voice Phishing Detection with Runtime Permission Requests," 2024.

[4] M. Elkholy et al., "An Efficient Phishing Detection Framework Based on Hybrid Machine Learning Models," 2025.

[5] S. Ahmed et al., "A Comprehensive Review on the Role of AI in Phishing Detection Mechanisms," 2025.

[6] J. Zhang et al., "Benchmarking and Evaluating Large Language Models in Phishing Detection for Small and Midsize Enterprises," 2025.

[7] K. Owa and O. Adewole, "Benchmarking Machine Learning Techniques for Phishing Detection and Secure URL Classification," 2025.

[8] A. Oluwaferanmi, "Adaptive Phishing Detection in Web Applications Using Ensemble Deep Learning and Feature Fusion Techniques," 2025.

[9] P. C. R. Chinta et al., "Building an Intelligent Phishing Email Detection System Using Machine Learning and Feature Engineering," 2025.

[10] J. Zhang, "Cutting-Edge Phishing Detection Using Novel Features and Hybrid Machine Learning Techniques," 2025.

[11] O. Senouci and N. Benaouda, "Enhancing Phishing Detection in Cloud Environments Using RNN-LSTM in a Deep Learning Framework," 2025.

[12] S. S. Patil et al., "Design of Intelligent Feature Selection Technique for Phishing Detection," 2025.

[13] J. S. R. et al., "Phishing Detection Using Machine Learning Techniques," 2025.

[14] C. V. R. A. Kumar et al., "Reinforcement Learning-Based Phishing Detection Model," 2025.

[15] D. Timko et al., "Understanding Influences on SMS Phishing Detection," 2025.

[16] T. Cao et al., "PhishAgent: A Robust Multimodal Agent for Phishing Webpage Detection," 2025.

[17] J. M. I. Arockiasamy, "Securing Telehealth Platforms: ML-Powered Phishing Detection with DevOps," 2025.

[18] D. Hriday et al., "Phish-Blitz: Advancing Phishing Detection with Comprehensive Webpage Resource Collection," 2024.

[19] O. J. Tiwo et al., "Improving Patient Data Privacy and Authentication Protocols against AI-Powered Phishing Attacks in Telemedicine," 2025.

[20] B. J. Warki et al., "Enhancing Phishing Detection in Sulu, Philippines: A Machine Learning Approach to Combat Evolving Cyber Threats," 2025.

[21] M. R. T. Utami et al., "Enhancing Phishing Detection: Integrating XGBoost with Feature Selection Techniques," 2025.

[22] B. John, "Building a Browser Extension for Real-Time Phishing Detection Using Celery," 2025.

[23] Anonymous, "CATALOG: Exploiting Joint Temporal Dependencies for Enhanced Phishing Detection on Ethereum," 2025.

[24] K. Barik et al., "Web-based Phishing URL Detection Model Using Deep Learning Optimization Techniques," 2025.

[25] K. Omari and A. Oukhatar, "Advanced Phishing Website Detection with SMOTETomek XGB: Addressing Class Imbalance for Optimal Results," 2025.

[26] L. Schöni et al., "Stop the Clock - Counteracting Bias Exploited by Attackers through an Interactive Augmented Reality Phishing Training," 2025.

[27] W. Guo et al., "Efficient Phishing URL Detection Using Graph-based Machine Learning and Loopy Belief Propagation," 2025.

[28] S. R. Alotaibi et al., "Explainable Artificial Intelligence in Web Phishing Classification on Secure IoT with Cloud-based Cyber-Physical Systems," 2025.

[29] P. An et al., "Multilingual Email Phishing Attacks Detection using OSINT and Machine Learning," 2025.

[30] M. S. Alzboon et al., "Guardians of the Web: Harnessing Machine Learning to Combat Phishing Attacks," 2025.

[31] C. Lee, B. Kim, and H. Kim, "The Silence of the Phishers: Early-stage Voice Phishing Detection with Runtime Permission Requests," 2025.

[32] N. Stevanović, "Embedding and Weighting of Website Features for Phishing Detection," 2025.

[33] M. K. M. Boussougou et al., "Enhancing Voice Phishing Detection Using Multilingual Back-Translation and SMOTE," 2025.

[34] K. I. Iyer, "Natural Language Processing for Phishing Detection: Leveraging AI to Spot Deceptive Content in Real Time," 2025.

[35] A. Alhuzali et al., "In-Depth Analysis of Phishing Email Detection: Evaluating the Performance of Machine Learning and Deep Learning Models Across Multiple Datasets," 2025.

[36] M. A. Daniel et al., "Optimising Phishing Detection: A Comparative Analysis of Machine Learning Methods with Feature Selection," 2025.

[37] S. Twum et al., "Evaluation of Machine Learning Techniques for Identifying Phishing Emails: A Case Study with the Spam Assassin Dataset," 2025.

[38] M. Fernando et al., "PhishLex: A Real-Time Machine Learning Model for Zero-day Phishing Detection by Systematizing URL Techniques," 2025.

[39] M. Hassnain et al., "Detection and Identification of Novel Attacks in Phishing using AI Algorithms," 2025.

[40] S. E. Blake, "PhishSense-1B: A Technical Perspective on an AI-Powered Phishing Detection Model," 2025.

[41] M. C. Calzarossa et al., "An Assessment Framework for Explainable AI with Applications to Cybersecurity," 2025.

[42] F. A. Manurung et al., "Spam and Phishing WhatsApp Message Filtering Application Using TF-IDF and Machine Learning Methods," 2025.

[43] K. Omari et al., "Comparative Analysis of Undersampling, Oversampling, and SMOTE Techniques for Addressing Class Imbalance in Phishing Website Detection," 2025.

[44] A. La Torre and M. Angelini, "Cyri: A Conversational AI-based Assistant for Supporting the Human User in Detecting and Responding to Phishing Attacks," 2025.

[45] F. Mathew, "Artificial Intelligence (AI) in Phishing Attacks," 2025.

[46] C. Vemula and P. Shaji, "Mitigating Cyber Threats: The Critical Role of Phishing-Resistant Users in Business Continuity," 2025.

[47] Anonymous, "7 Days Later: Analyzing Phishing-Site Lifespan After Detected," 2025.

[48] B. John, "Celery Trap: Detecting and Preventing Phishing, Spearphishing, and Online Threats in Browsers and Emails," 2025.

[49] H. J. Abejuela et al., "ScamGuard: Image-based Identification App for Phishing and Open Attachment Messages Using Variants of Convolutional Neural Networks," 2025.

[50] D. Mandora et al., "AI to Detect Phishing," 2025.

[51] A. Mukhopadhyay and A. Prajwal, "EDITH – A Robust Framework for Prevention of Cyber Attacks in the Covid Era," 2021.

[52] A. Mukhopadhyay, V. S. Skanda, and C. J. Vignesh, "An Analytical Study on the Versatility of A Linux Based Firewall From A Security Perspective," 2015.

[53] A. Mukhopadhyay et al., "QoS-Aware IoT Edge Network for Mobile Telemedicine Enabling In-Transit Monitoring of Emergency Patients," 2024.

[54] P. Anusri et al., "Shielding Cyberspace: A Machine Learning Approach for Malicious URL Detection," 2024.

[55] M. Diviya et al., "Enhancing Cyber Security Through Phish-Net - An Artificial Neural Network for Phishing Detection," 2024.

[56] A. Sa et al., "An Ensemble Classification Model for Phishing Mail Detection," 2024.

APPENDICES

A 1. Tools and Technologies Used

- **Programming Language:** Python
- **Libraries & Frameworks:**
  - `scikit-learn` – for machine learning model development
  - `pandas`, `numpy` – for data manipulation
  - `matplotlib`, `seaborn` – for data visualization
  - `CatBoost` – for high-performance classification
  - `Flask` – for web app integration
  - `Rasa / NLTK` – for chatbot development
- **IDE/Editor:** Jupyter Notebook, VS Code
- **Deployment:** Localhost for testing
- **Dataset Source:** Kaggle – *Phishing Website Detector*

A 2. Dataset Details

- **Total Records:** 11,054
- **Features:** 32
- **Label Values:**
  - `1` = Phishing
  - `-1` = Legitimate
- **Common Features Used:**
  - URL Length
  - Presence of HTTPS
  - Domain Registration Length
  - Use of `@`, `//`, or `-` symbols
  - Favicon anomaly, iframe detection, anchor tags behavior

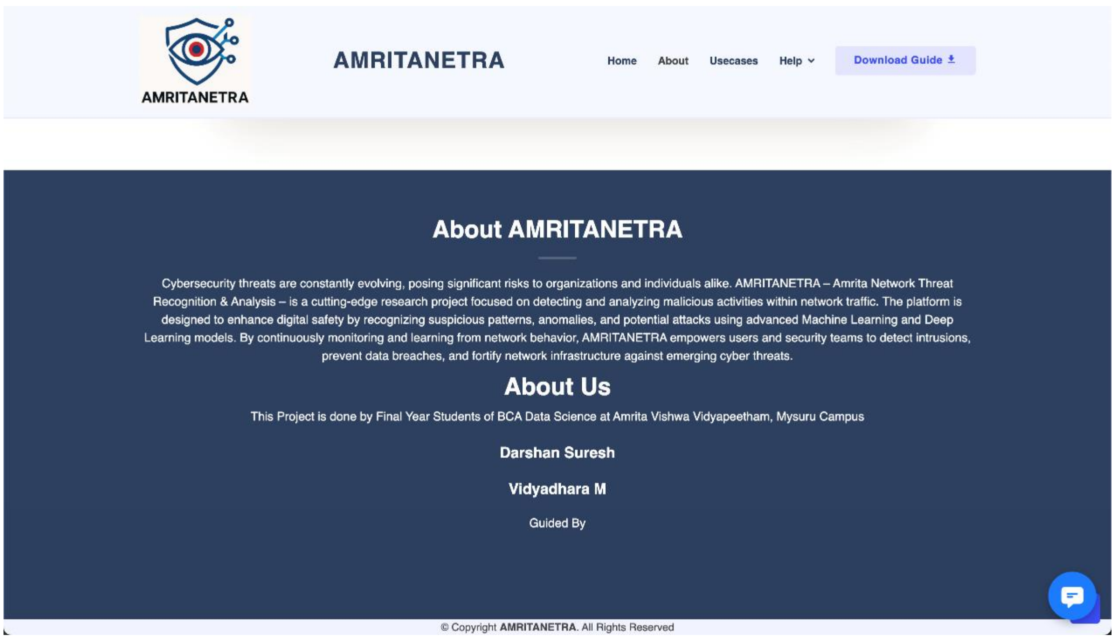
A 3. Performance Metrics Used

- Accuracy
- Precision
- Recall
- F1 Score
- Confusion Matrix

A 4. System Modules

- **Phishing Detection Engine** – classifies URLs using ML models
- **Chatbot Interface** – interacts with users and explains phishing threats
- **AI Storytelling Module** – narrates context-aware scenarios for user awareness
- **Dashboard/UI** – input interface and result display

A 5. Sample Screenshots of User interface



About AMRITANETRA

Cybersecurity threats are constantly evolving, posing significant risks to organizations and individuals alike. AMRITANETRA – Amrita Network Threat Recognition & Analysis – is a cutting-edge research project focused on detecting and analyzing malicious activities within network traffic. The platform is designed to enhance digital safety by recognizing suspicious patterns, anomalies, and potential attacks using advanced Machine Learning and Deep Learning models. By continuously monitoring and learning from network behavior, AMRITANETRA empowers users and security teams to detect intrusions, prevent data breaches, and fortify network infrastructure against emerging cyber threats.

About Us

This Project is done by Final Year Students of BCA Data Science at Amrita Vishwa Vidyapeetham, Mysuru Campus

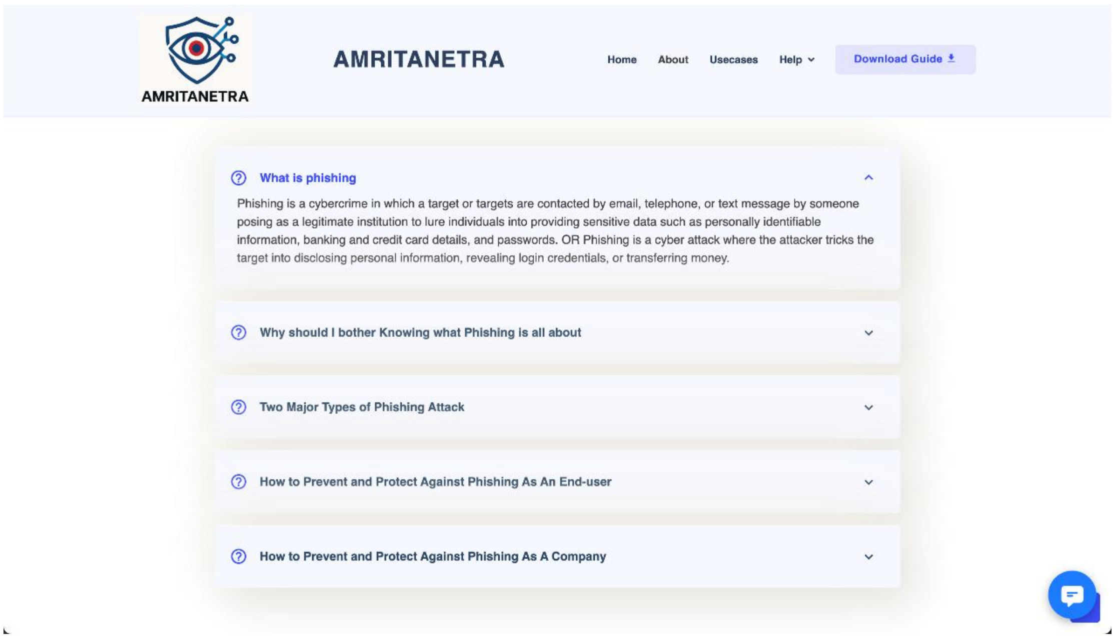
Darshan Suresh

Vidyadhara M

Guided By

© Copyright AMRITANETRA. All Rights Reserved

About



What is phishing

Phishing is a cybercrime in which a target or targets are contacted by email, telephone, or text message by someone posing as a legitimate institution to lure individuals into providing sensitive data such as personally identifiable information, banking and credit card details, and passwords. OR Phishing is a cyber attack where the attacker tricks the target into disclosing personal information, revealing login credentials, or transferring money.

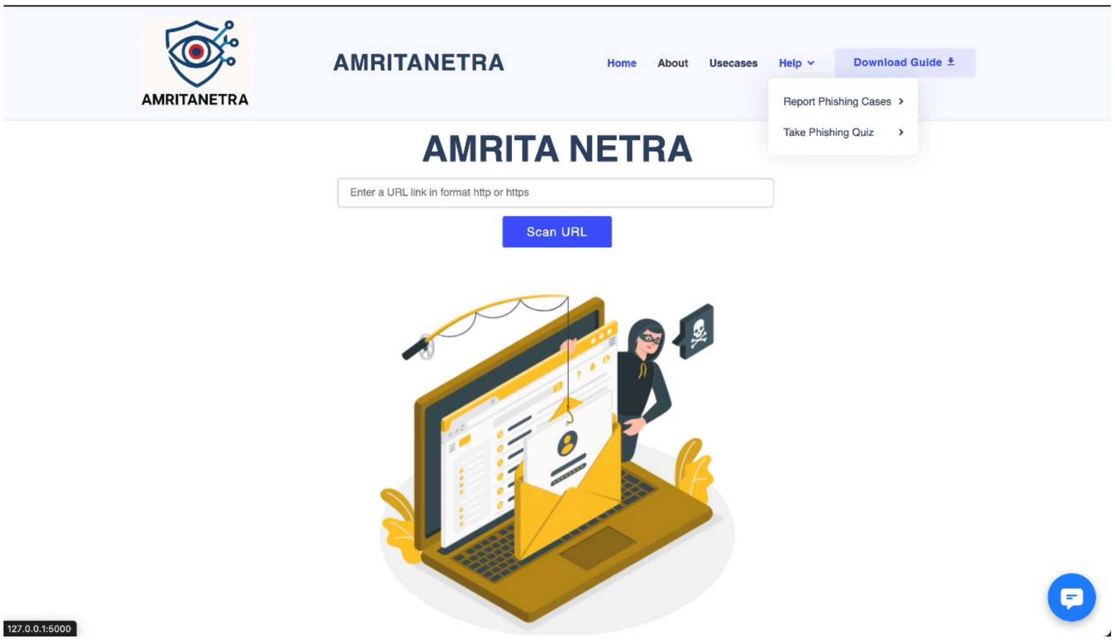
Why should I bother Knowing what Phishing is all about

Two Major Types of Phishing Attack

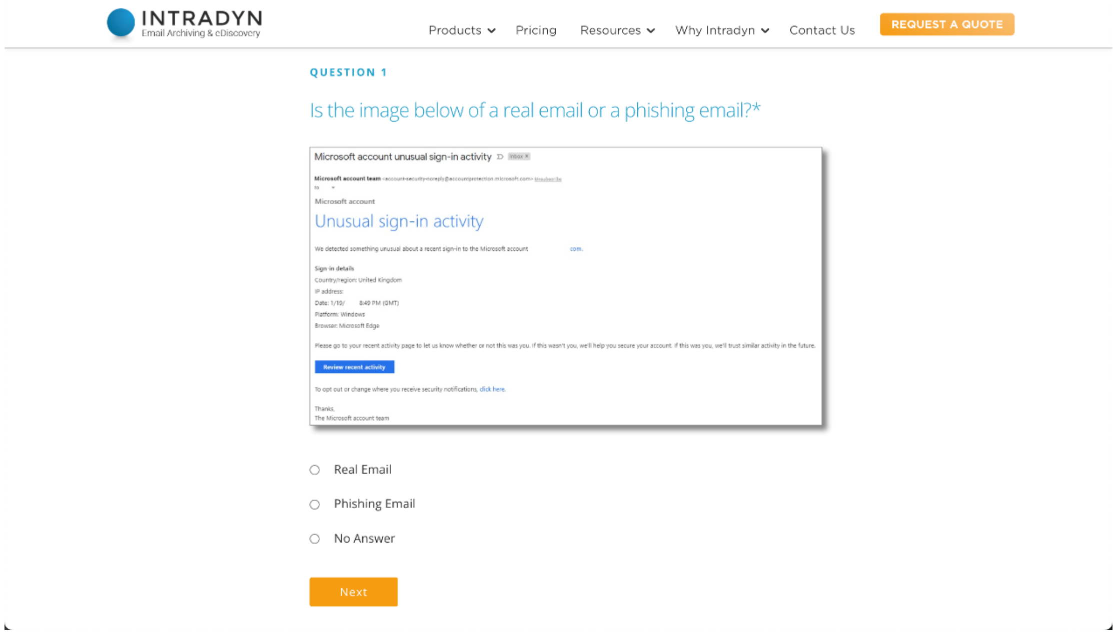
How to Prevent and Protect Against Phishing As An End-user

How to Prevent and Protect Against Phishing As A Company

Details



Help Section



QUESTION 1

Is the image below of a real email or a phishing email?*

Microsoft account unusual sign-in activity Inbox ×

Microsoft account team <account-security-noreply@accountprotection.microsoft.com> Unsubscribe
to

Microsoft account

Unusual sign-in activity

We detected something unusual about a recent sign-in to the Microsoft account com.

Sign-in details
Country/region: United Kingdom
IP address:
Date: 1/19/ 8:49 PM (GMT)
Platform: Windows
Browser: Microsoft Edge

Please go to your recent activity page to let us know whether or not this was you. If this wasn't you, we'll help you secure your account. If this was you, we'll trust similar activity in the future.

Review recent activity

To opt out or change where you receive security notifications, click here.

Thanks,
The Microsoft account team

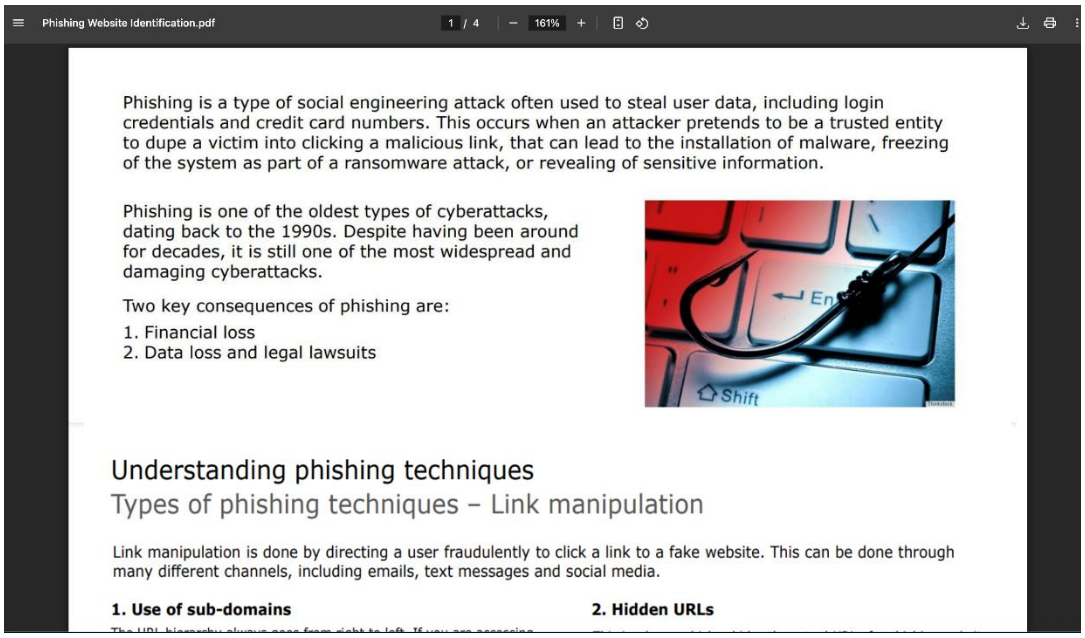
Real Email

Phishing Email

No Answer

Next

Phishing related Quiz



Phishing is a type of social engineering attack often used to steal user data, including login credentials and credit card numbers. This occurs when an attacker pretends to be a trusted entity to dupe a victim into clicking a malicious link, that can lead to the installation of malware, freezing of the system as part of a ransomware attack, or revealing of sensitive information.

Phishing is one of the oldest types of cyberattacks, dating back to the 1990s. Despite having been around for decades, it is still one of the most widespread and damaging cyberattacks.

Two key consequences of phishing are:

1. Financial loss
2. Data loss and legal lawsuits

Understanding phishing techniques

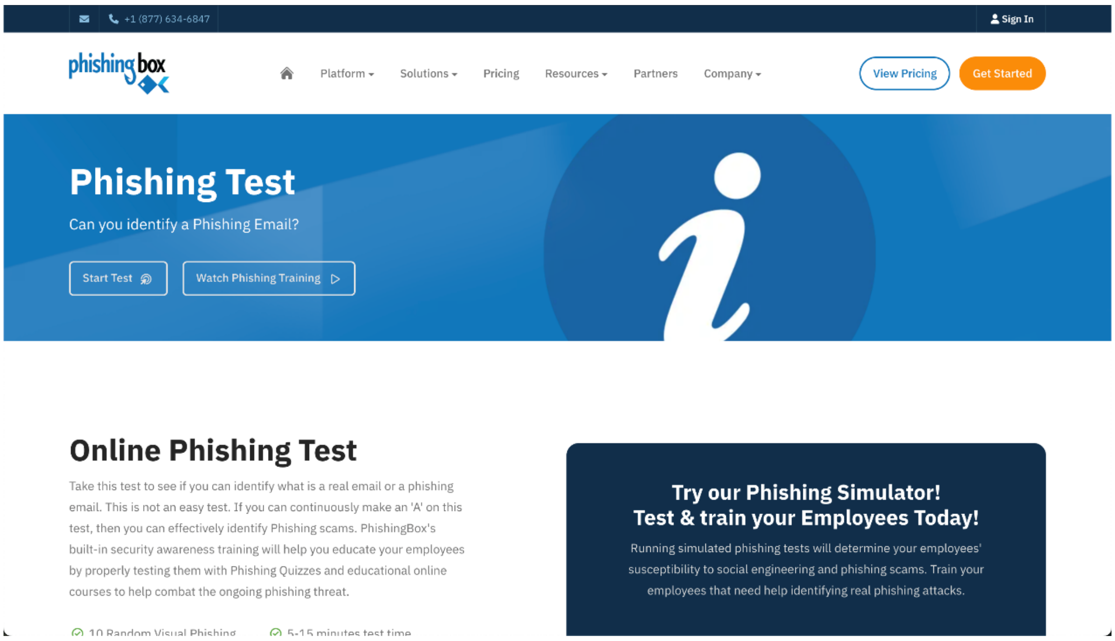
Types of phishing techniques – Link manipulation

Link manipulation is done by directing a user fraudulently to click a link to a fake website. This can be done through many different channels, including emails, text messages and social media.

1. Use of sub-domains

2. Hidden URLs

User Guide pdf



Phishing Test

Can you identify a Phishing Email?

Online Phishing Test

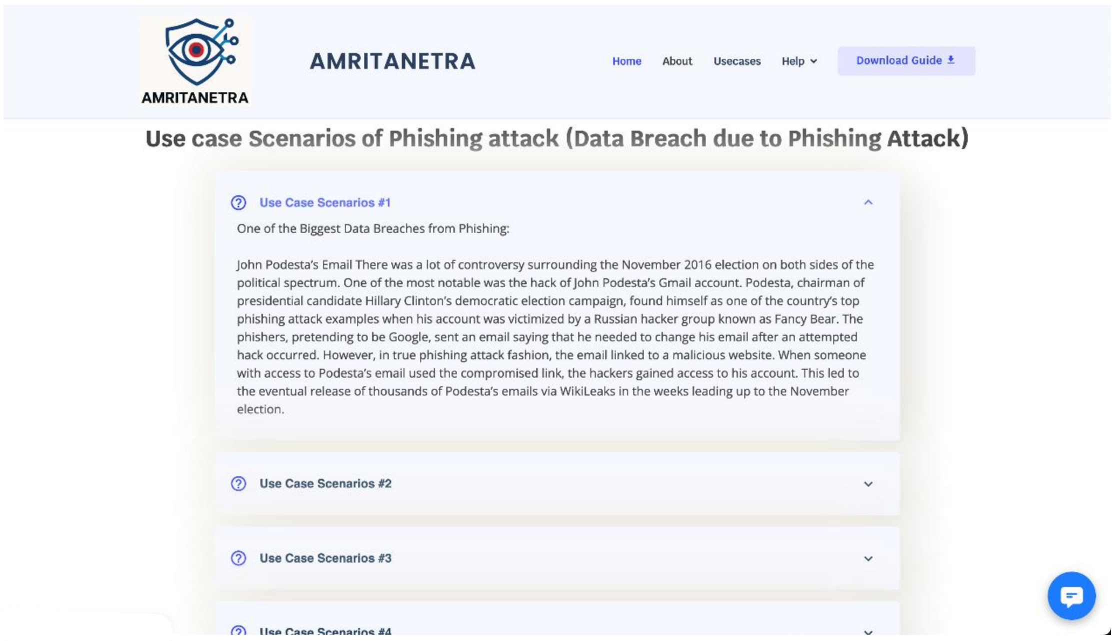
Take this test to see if you can identify what is a real email or a phishing email. This is not an easy test. If you can continuously make an 'A' on this test, then you can effectively identify Phishing scams. PhishingBox's built-in security awareness training will help you educate your employees by properly testing them with Phishing Quizzes and educational online courses to help combat the ongoing phishing threat.

10 Random Visual Phishing 5-15 minutes test time

Try our Phishing Simulator!
Test & train your Employees Today!

Running simulated phishing tests will determine your employees' susceptibility to social engineering and phishing scams. Train your employees that need help identifying real phishing attacks.

Phishingbox



Use case Scenarios of Phishing attack (Data Breach due to Phishing Attack)

Use Case Scenarios #1

One of the Biggest Data Breaches from Phishing:

John Podesta's Email There was a lot of controversy surrounding the November 2016 election on both sides of the political spectrum. One of the most notable was the hack of John Podesta's Gmail account. Podesta, chairman of presidential candidate Hillary Clinton's democratic election campaign, found himself as one of the country's top phishing attack examples when his account was victimized by a Russian hacker group known as Fancy Bear. The phishers, pretending to be Google, sent an email saying that he needed to change his email after an attempted hack occurred. However, in true phishing attack fashion, the email linked to a malicious website. When someone with access to Podesta's email used the compromised link, the hackers gained access to his account. This led to the eventual release of thousands of Podesta's emails via WikiLeaks in the weeks leading up to the November election.

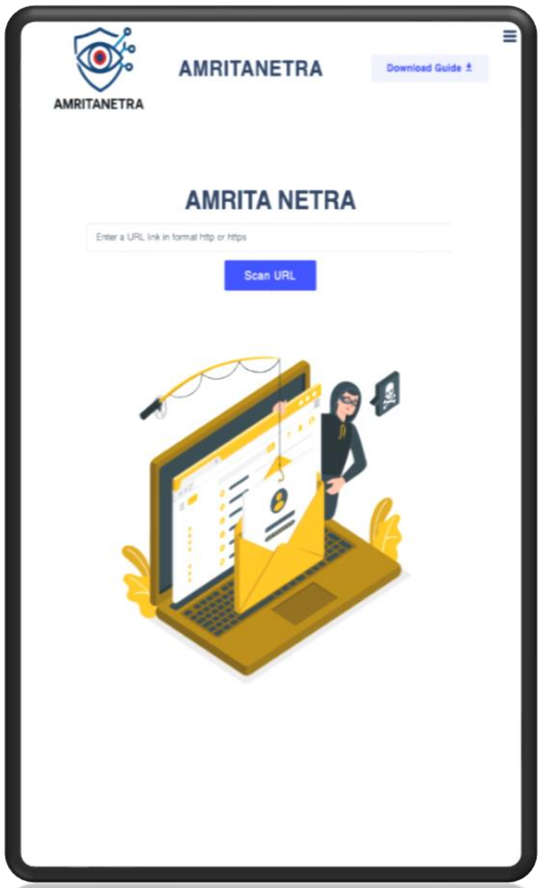
Use Case Scenarios #2

Use Case Scenarios #3

Use Case Scenarios #4

Use case scenarios

Mobile view


AMRITANETRA
AMRITANETRA
Download Guide
AMRITA NETRA
Enter a URL link in format http or https
Scan URL