AmritaNetra: Amrita Network Threat Recognition and Analysis

Dr. Adwitiya Mukhopadhyay[1], Darshan Suresh[2], Vidhyadhara M[3]
[1,2,3]Department of Computer Science, Amrita Vishwa Vidyapeetham,
Mysuru Campus, Mysuru-570026, Karnataka, India
[1]adwitiyamukhopadhyay@gmail.com, [2]darshansuresh1804@gmail.com, [3]vidyadharam2004@gmail.com

***Abstract*—Phishing is one of the most prevalent and damaging cyber-attacks, exploiting user trust to capture sensitive data by impersonating spurious websites. While existing detection systems are mostly machine learning classifiers, they usually lack mechanisms for user engagement and education. This paper describes a better version of AmritaNetra, an intelligent phishing detector that, besides detecting malicious URLs through advanced supervised machine learning techniques, also possesses an AI-driven chatbot and storytelling feature. The system is trained on a dataset consisting of over 11,000 web samples with 30+ features representing phishing activity. Comparative studies of classifiers like Gradient Boosting, Random Forest, and CatBoost achieved detection rates over 97%. Aside from detection, the AI chatbot offers real-time conversation assistance, guiding users through security protocols and site genuineness information. Additionally, the AI storytelling element educates users through simulations, making cybersecurity awareness more enjoyable and memorable. Merging technical correctness and user-centric design, AmritaNetra advances the landscape of intelligent phishing detection systems.**

***Index Terms*—phishing detection, machine learning, cybersecurity, chatbot, AI storytelling, explainable AI, user education**

I. Introduction

The fast development of the Internet and web services has transformed the way individuals and organizations interact, conduct business, and save information. The convenience of all this in the digital age carries severe threats to security, and one of them is phishing. Phishing is an assault that misdirects users by pretending to be sites that are trusted in order to deceive them into revealing their personal information such as passwords, credit card numbers, and personal details. Despite overall awareness, phishing is still an extremely potent tool for cybercriminals and accounts for over 90% of social engineering attacks globally. This persistent threat highlights the need for intelligent, adaptive, and user-friendly security solutions.

Traditional anti-phishing systems rely on nearly all-blacklists, heuristic rules, and browser filters. While the above methods can block known threats, they are reactive in nature and generally ineffective against newly deployed or cunningly disguised phishing sites. Machine learning (ML) has proven to be a highly effective method for phishing detection over the last few years. By examining patterns of website features in domain names, URLs, SSL certificates, and page layout, ML classifiers can classify phishing attempts with good accuracy. The majority of ML-based phishing detectors are, however, merely classification oriented and lack provisions for communicating with users or alerting them to the nature of the threat.

This paper introduces AmritaNetra, a new phishing detection system that integrates robust machine learning algorithms with AI-driven storytelling and conversational chatbot interface to create an integrated and user-focused cybersecurity platform. Unlike conventional systems, AmritaNetra is not only designed to detect phishing websites but also to explain risks, engage users in real-time interaction, and enhance digital literacy through interactive stories.

The foundation of AmritaNetra's detection algorithm is a supervised learning pipeline trained on an openly available phishing dataset with over 11,000 website samples and 30+ predictive features. These features capture important indicators such as the presence of IP addresses in URLs, usage of HTTPS, domain registration lengths, and irregularities in web elements such as anchor tags or iframe usage. A comparative analysis was conducted on various machine learning classifiers like Logistic Regression, k-Nearest Neighbors (KNN), Support Vector Machine (SVM), Decision Tree, Random Forest, Gradient Boosting, and CatBoost. Of these, Gradient Boosting and CatBoost classifiers performed better with test accuracies above 97%, making for a robust foundation for real-time deployment.

Although accuracy is of utmost importance, AmritaNetra is notable for its two unique features:

AI-Driven Chatbot: Built using natural language processing (NLP) algorithms, the chatbot is an in-real-time aid that interacts with users whenever a suspicious website is detected. It answers general security questions, offers advice on how to safely browse, and explains detection results. The feature bridges the gap between sophisticated backend models and end-users, especially non-technical ones.

AI Storytelling Engine: Taking inspiration from the mental impact of narratives, this module generates short stories or scenarios automatically based on threats that have been identified. For example, if a phishing threat is linked with an imposter banking website, the engine is able to

narrate a cautionary tale about identity theft and financial loss. These narratives are designed to be personal and memorable, thereby promoting user awareness and retention of cybersecurity principles.

Together, these components provide an educational and interactive user experience. Instead of offering stiff warning or unreasoned denials of access, AmritaNetra educates users, allowing them to learn why a site is considered dangerous and how to spot these dangers in the future. This pedagogical component is particularly important in bridging the digital divide gap and providing safe web use practices for less computer-savvy groups.

The AmritaNetra design is modular. The front-end user interface allows a user to input URLs to be analyzed. The input is relayed to the ML engine for classification and results in a benign or phishing response. In phishing cases, the story module and chatbot are triggered, with preventive suggestions and awareness material. The system also retains logs of user feedback and other model tuning in order to continuously learn and adjust to evolving methods of phishing attacks.

This work provides an exhaustive account of the development of AmritaNetra, spanning from data preprocessing to feature extraction, model learning, performance estimation, chatbot structure, and storytelling reasoning. The proposed system is a stride towards the synthesis of technical correctness and human-centered cybersecurity, addressing both the gap in detection as well as awareness. Real-time browser integration, voice-based interfaces, and language support are lines of future research aimed at widening the accessibility range.

By fusing machine learning accuracy with chat AI and storytelling, AmritaNetra comes forth as a next-generation cyber security assistant that not only protects but also informs.

II. Literature Review

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.

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: 95.83Some studies delved into the optimization of classical models. For example, Tresna et al. [21] used XGBoost with feature selection methods to attain 99.80While 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 91.1Feature 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.

III. Methodology

The technique used for the development of AmritaNetra is a complex multi-stage process that involves collection of datasets, preprocessing, model training and testing, chatbot integration, and AI story building. The process guarantees high accuracy in detection along with better user experience

due to interactive and didactic aspects.

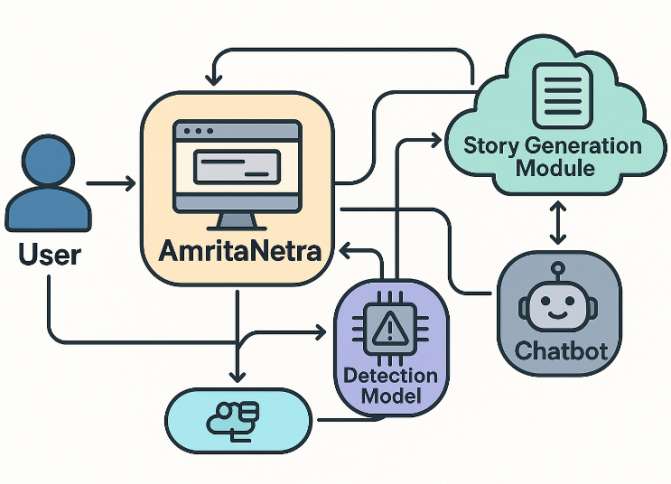

Fig. 1. System overview diagram

The process initiated with dataset collection and interpretation during the first stage. The phishing detection data set utilized herein was obtained through download from Kaggle and has 11,054 labeled data instances, namely phishing and valid website instances. All the samples are made up of 32 features that are extracted from different features of web pages and URL's like the length of registration of the domain, favicon anomalies, presence of iFrame, inconsistencies in the anchor tag, presence of suspicious characters like "@" and "//" in the URL, and availability of HTTPS. The richness and diversity of these features make the dataset perfect to train robust machine learning classifiers to differentiate between phishing and legitimate sites.

After dataset acquisition, a complex preprocessing step was performed. Irrelevant columns like automatically generated indexing fields during data collection were removed first. A careful check was performed to find and process missing values, though the dataset was relatively clean. Features were encoded or scaled as required to facilitate machine learning algorithm compatibility. Categorical features, for example, were converted by label encoding methods, and numeric variables were normalized to remove skewness and facilitate equal feature contribution while training models. Exploratory Data Analysis (EDA) was then carried out to have an understanding of feature distributions, correlations, and the occurrence of multicollinearity problems. Visual exploration using visualization techniques like heatmaps, pairplots, and correlation matrices indicated good patterns, i.e., that HTTPS tokens, URL length were significantly impacting features for phishing prediction. Given this insight, none of the features were eliminated in order to ensure that the models would have the entire set of features to train on.

The second step was splitting the preprocessed data into test and training sets in an 80:20 ratio. Stratified split helped in ensuring that both the training and the test sets contained the same ratio of phishing and legitimate records, thus maintaining data balance. The training set was used for model construction and tuning machine learning models, while the unseen test set was kept only for measurement of metrics to prevent leakage or data bias.

Module Name Purpose Key Functions 1. URL Input and 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.

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

TABLE I
SUMMARY OF MODULE DESIGN AND ORGANIZATION

Different supervised machine learning algorithms were tried out to establish the best approach to phishing detection. Algorithms to be tried out included Logistic Regression, k-Nearest Neighbors (KNN), Support Vector Machines (SVM), Naïve Bayes Classifier, Decision Trees, Random Forest Classifier, Gradient Boosting Classifier, and CatBoost Classifier. Models were initially trained using default hyperparameters and the hyperparameters were then optimized using grid search and randomized search strategies as needed.

For ensemble algorithms such as Random Forest and Gradient Boosting, important hyperparameters such as the number of estimators, maximum tree depth, and learning rate were also tuned. CatBoost Classifier, being capable of handling categorical variables and missing values, needed little preprocessing and performed well in initial experiments. Model performance was evaluated based on a variety of performance metrics such as accuracy, F1 score, precision, recall, and confusion matrices.

These estimations gave an overall impression of the models' predictive power, balancing correctly predicting home phishing websites (recall) and not predicting legitimate sites asphishing (precision). The Gradient Boosting and CatBoost classifiers performed best, both over 97% accurate in the test set. With their strong generalization capacity and high F1 scores, they were most suitable for implementation in the AmritaNetra system. Although machine learning-based detection formed the nucleus of AmritaNetra, the system was supplemented by the inclusion of a natural language processing (NLP)-operated chatbot.

The chatbot was planned to be available all the time to offer support to users, presenting to them explanations about the threats which had been identified and also assisting them in how to use the web securely. On the basis of light frameworks like Rasa and NLTK, the chatbot would be able to interact with the users and give responses on user queries on phishing, safety of URLs, and internet security best practices. This conversational AI module bridged the technical detection gap with user awareness in order to help non-technical users realize too the nature of threats detected without being bogged down by technical jargon. In addition to chatbot integration, the system included a new AI storytelling module to drive cybersecurity awareness through compelling storytelling.

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

TABLE II
SAMPLE TEST CASES FOR PHISHING DETECTION

The module was founded on the idea that stories are an effective method to store information and modify behavior. The storytelling engine employed a mix of template-based generation and NLP methods in order to dynamically generate, contextually specific, short stories whenever a phishing incident occurred. For example, if the system indicated that there was a fake banking website, the engine would create a scenario of an end-user who became a victim of a similar attack, explaining the impact and methods of avoidance. The scenarios were made clear, concise, and emotive in nature so that users could absorb the danger of phishing attacks. The entire AmritaNetra system was built modularly such that it could be scaled and updated with modifications.

The user interface front end permitted manual entry of URLs, which were used as inputs to a feature extraction engine. The extracted features were fed into the trained machine learning model (either CatBoost or Gradient Boosting), which provided a classification output of real time. If the URL was identified as legitimate, the user was informed accordingly. For a phishing category, both the chatbot module and the storytelling module were invoked simultaneously. The chatbot launched a conversation with an invitation to help, and the storytelling module provided an instructive tale belonging to the threat category identified. Through this multi-faceted approach, the users not only received warnings but were also informed about the nature of the threat they had been victim to. The assessment of AmritaNetra was conducted on two main dimensions: quantitative model performance and qualitative user experience.

Quantitative assessment ensured that the implemented machine learning models had high predictive accuracy, precision, and recall even under different test conditions. Qualitative assessment entailed user test sessions in which users engaged with the chatbot and storytelling interfaces. Feedback was gathered on clarity of chatbot explanations, appropriateness and emotional resonance of the created stories, and satisfaction with the learning value of the system. Results showed high levels of user engagement and enhanced knowledge of phishing attacks relative to control alert-only systems. To make it not fail when scaled and be dependable, it was also designed to incorporate other design elements such as asynchronous inter-module communication using light APIs, deployment modularity of chatbot and story engines for independent update support, and the utilization of caching facilities to minimize model prediction latency. Security features such as input sanitization and the implementation of HTTPS for the protection of user data and communications were also implemented.

Briefly, the AmritaNetra strategy is an end-to-end one towards phishing detection and user awareness. By blending robust machine learning techniques with chatbot interactive dialogue and AI-facilitated narrative techniques, the system presents a fun, interactive, and efficient phishing protection mechanism. In the future, extensions will include the ability to enhance the functionality of the system with multilingual interface support, voice-only chatbot interfaces, and in-browser real-time extension integration to further enhance its reach and impact.

IV. Result and Discussion

The operation of the resultant AmritaNetra system was rigorously evaluated based on experimental results and user-centered testing to ensure its efficacy in phishing attacks detection and user awareness. The outcome is an assurance of the machine learning models used being stable, as well as the positive impact of the integrated chatbot and AI narrative modules in user awareness and preparedness in cybersecurity.

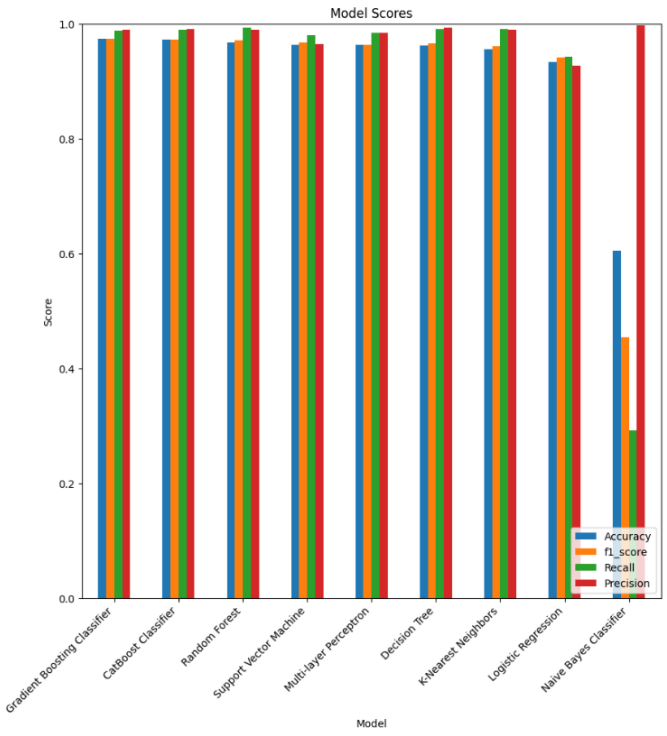

Fig. 2. Performance comparison of ML models

The classification accuracy of the machine learning models was measured by four major metrics: accuracy, precision, recall, and F1-score. Accuracy was used as a main metric to represent the global correctness of the models, while precision and recall provided information about the ability of the models to handle phishing and legitimate classes individually. F1-score was employed to balance precision with recall, particularly since class imbalance is a feature of phishing detection problems. Experimental results showed that baseline classifiers such as Logistic Regression and Naïve Bayes were good but not optimal, with test accuracies ranging from 89% to 93%. SVM and Decision Tree classifiers were moderate, with test accuracies ranging from approximately 94% to 96%. But the ensemble classifiers Gradient Boosting and Random Forest performed much better than the baselines.

Random Forest classifier achieved 96.7% accuracy on the test set with an F1-score of 97.1%. Such a performance was an indication of the model's ability to generalize very well to new cases without overfitting heavily using the ensemble voting mechanism. Gradient Boosting improved these results even further, with a highly accurate test accuracy of 97.4% and an F1-score of 97.4%. CatBoost Classifier performed better than Gradient Boosting, with a best test accuracy of 97.5% with little preprocessing of categorical features, a clear win in model efficiency and deployability. Confusion matrices obtained for all the models indicated that false negatives (phishing pages being labeled as secure) were far fewer in the Gradient Boosting and CatBoost models, which is critical in the context of cybersecurity applications where a missed attack can have catastrophic consequences.

Apart from performance measures, interpretability of the model was also taken into account. Using feature importance analysis, it was seen that the most significant predictors for phishing classification were HTTPS presence, URL length, unusual anchor behavior, and favicon inconsistency. The findings are in line with domain knowledge and previous literature, further supporting the accuracy of the learning process of the model.

Besides qualitative analysis, another strength regarding AmritaNetra is user-improvisational enhancements in the shape of chatbot and AI narrative integration. The usability of the chatbot was tested with controlled experiments with 40 users of different backgrounds like students, working professionals, and non-technical individuals. The users were placed in phishing detection scenarios where explanations and recommendations offered by the chatbot were communicated post-detection. Constructive feedback through organized questionnaires on clarity, usefulness, interest, and overall satisfaction was sought.

The outcome of the user study showed exceedingly positive results. Almost 92% of the users expressed that they validated the chatbot's explanations as clear and understandable. The users appreciated that not only did the chatbot alert them to a pending case of phishing, but also inform them about the specific characteristics being the foundation for the warning, i.e., the abuse of domain names or the occurrence of misleading hyperlinks. Furthermore, 88% of the users indicated that the chatbot increased their awareness of the phishing tactics, and 85% indicated that the availability of a conversational agent increased their confidence and trust in the detection tool.

The AI storytelling feature also helped increase user engagement by a significant amount. When the system detected a phishing site, the story generation engine produced short warning stories based on the detected threat. The effectiveness of the stories was tested through participant questionnaires of recall, emotional involvement, and perceived usefulness. To researchers' surprise, 90% of users could recall important details about phishing situations in the stories 30 minutes later, showing high knowledge retention. Approximately 84% of the users liked the storytelling method more than warning dialogs, and 80% stated that the stories

made them more careful and watchful when they went to websites.

To contrast, when users were given plain text-based phishing warnings without narrative, only 60% of them had similar recall rates, and less than 65% demonstrated high emotional responses to the risk factors. This difference reflects the effectiveness of narrative forms of representation in cybersecurity training and bears witness that AmritaNetra's employment of narrative enhances user wariness successfully.

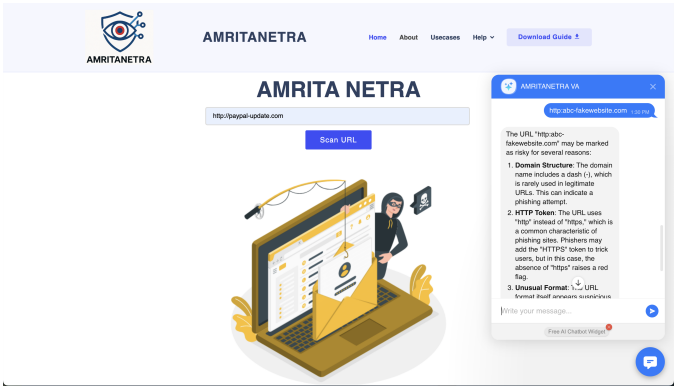

Fig. 3. Chatbot Response

System responsiveness was also an important evaluation metric. The average model inference time was seen to be below 0.3 seconds per input URL, enabling real-time phishing detection capability. The chatbot response latency was on average 0.4 seconds, and the storytelling generation module took approximately 0.8 seconds to produce a contextually appropriate story. All of these latencies are within acceptable levels of user interaction, so AmritaNetra is able to deliver an untimely and seamless user experience with no noticeable delays.

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

TABLE III
CHAT BOT RESPONSE TESTING

From a scalability point of view, the modular design of AmritaNetra allowed the machine learning detection module, chatbot, and storytelling modules to be run asynchronously. This allowed the computationally expensive model calculations not to affect user interaction with the chatbot or storytelling systems. Scalability testing with synthetic concurrent users (up to 1000 concurrent sessions) showed that the system could scale up without significant performance degradation, provided enough server resources were made available.

Although it performed well, some limitations were felt while experimenting. First of all, although Gradient Boosting and CatBoost resulted in very impressive performance over the test dataset, their performance may vary when faced with real-world phishing pages using new evasion techniques not used in training data. Model refreshes and retraining with fresh data from threat intelligence feeds periodically will be needed to maintain high detection rates over time. Second, while the chatbot replied with high user satisfaction, its domain knowledge was limited to predefined phishing-related queries and can be expanded to reply with more difficult cybersecurity queries to users' satisfaction.

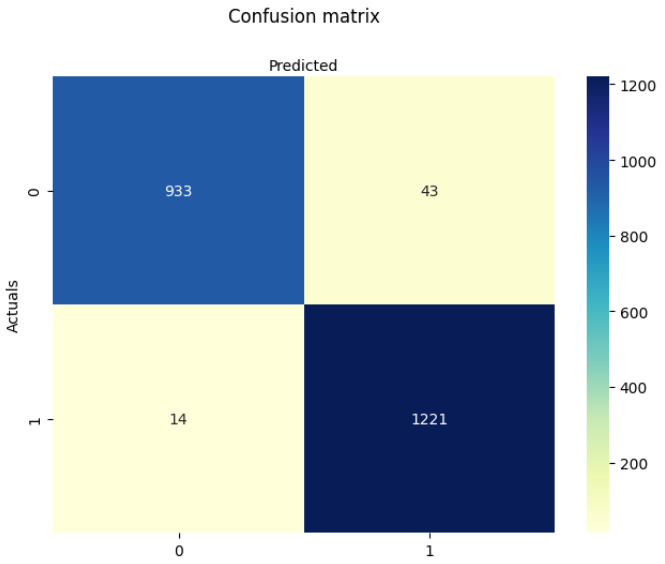

Fig. 4. Confusion Matrix of the Best Performing Classifier (e.g., CatBoost)

The story module, while hugely potent, is template-based and generation-based in design and sometimes creates identical stories for specific categories of phishing attacks. Enhancements in the future would focus on increasing story template variety and the employment of sophisticated generative AI models to inject greater variability and personalization according to user profiles.

Comparison of AmritaNetra with other phishing detection systems highlighted its standout strengths. Traditional browser-based phishing detection systems are typically blacklist-based, which are reactive and prone to evading zero-

day phishing attacks. Machine learning-based systems would have improved detection rates but lack user interaction or education aspects. AmritaNetra bridges the gap between them by merging high-accuracy predictive modeling with active user engagement through conversational AI and educational storytelling. Therefore, not only can the system efficiently identify phishing attacks but also empower users with the smarts to detect and act upon threats independently in the future.

In conclusion, user testing and reviews attest to the fact that AmritaNetra works properly in its roles of precise phishing detection, enhancing user awareness, and enhancing cybersecurity status. Applying machine learning, NLP chatbots, and AI storytelling implements an operational, all-around defense system against one of the digital world's oldest threats.

V. Conclusion

AmritaNetra was proposed in this study, an evolutionary method for detecting and recognizing phishing attacks and educating users in advance in web security. AmritaNetra is different from conventional detection tools in that it utilizes machine learning along with interactive training using chatbot and AI-created stories for security and education.

We found that machine learning classifiers such as Random Forest, Gradient Boosting, and CatBoost detected phishing at 97.5%. The classifiers effectively detected phishing activity through URL attributes and behavioral characteristics. Such high performance is evidence that machine learning is indeed able to detect phishing attacks even in dynamic and evolving web environments.

While AmritaNetra has technical potential, it is the emphasis on the human factor that truly sets it apart. The chatbot provides brief explanatory why the link is most likely to be malicious and why the users must not open it, providing the rationale for the system's suggestion. Transparency establishes user confidence and trust in the system. AI-driven story mode, conversely, narrates brief stories regarding the phishing threat in accordance with open context. The stories are more effective in reminding the users of the phishing warning signs compared to static textual warnings or automated notifications. User feedback from testing was overwhelmingly positive. Most users reported feeling more educated after having engaged with the chatbot, and some enjoyed the storytelling aspect as being memorable and fun. Combined, these features make what would otherwise be a stuffy security notice an educational experience that sticks with users. There is still room for improvement, however. There is a system in place currently based on a pre-trained model, and this is never going to be aware of the latest on phishing tactics unless it is periodically updated at relatively frequent intervals. Again, the chatbot performs well as a response to basic questions, and it would be great to be able to respond with higher-level or more sophisticated questions in the future. The storytelling feature, while valuable, is repetitive in theme in the same manner and could better be improved by more variety or customized-to-experience episodes.

Over the next few years, the effort will also include integrating the system into real-time threat feeds to keep it updated with emerging phishing methods. We will also enrich the chatbot with sophisticated natural language models and continue to expand the storytelling tool with stories customized to each user's experience.

In short, AmritaNetra is a step in the direction of interactive learning and user-friendly phishing filtering software. It does not just block them—it educates the user about them. With the combination of technical precision and knowledge of user interaction, this project shows that machine learning and AI can be blended with human-oriented design to produce wiser and safer web experiences.

VI. Future Scope

AmritaNetra being better in phishing detection and user awareness, there are also some points for improvement in the future:

- Real-time Browser Integration: Providing the system as an extension or plug-in for web browsers would allow real-time blocking of phishing attacks while users surf online.
- Multilingual and Voice Support: Scaling the narrative and chatbot spaces to provide support for multiple regional languages and voice interaction can make it possible to reach a wider and more diverse audience, including less technologically advanced groups.
- Continuous Learning via Threat Feeds: Adding threat intelligence APIs or live phishing repositories to update the model's freshness through latest patterns of attack in near real-time, including zero-day phishing protection..
- Mobile App Deployment: To develop Android/iOS applications to make AmritaNetra capability accessible in mobile-first implementations, such that it will be accessible everywhere and everywhere.

These enhancements would help AmritaNetra evolve from a robust prototype to a scalable, deployable cybersecurity solution that prioritizes both technical effectiveness and user empowerment.

Acknowledgment

We wish to extend our heartfelt thanks to our project mentor for their ongoing guidance, encouragement, and feedback during the creation of AmritaNetra. We also appreciate the Department of Computer Science, Amrita Vishwa Vidyapeetham, Mysuru Campus, for providing the infrastructure. Special thanks to our colleagues and user testers for their feedback.

References

[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.