PHISHNET: Intelligent Threat Analysis System Against Phishing Attacks

Chobe Pranali[1], Ubale Anjali[2], Malani Krishi[3], Jangam Atharv[4], Prof. Bhalchandra Ban[5]
[1, 2, 3, 4]Dept of Computer Engineering
[1, 2, 3, 4] SITS's Sinhgad College of Engineering

Abstract- *Digital chat's fast rise makes smart online protection more urgent, especially to spot and stop fake message scams on different apps. PHISHNET uses AI to check threats in many channels, combining learning machines, number crunching, and human behavior clues into one working setup. It runs sharp codes to catch phishing faster and sort it automatically by studying links, inbox notes, or texts using models that get the situation. Instead of just reading words, NLP figures out what messages really mean; at the same time, tools like Gradient Boosting plus Random Forest flag shady traces and odd actions. Behind the scenes, Flask parts made with Python team up with a flexible Mongo storage unit so info flows quick and adjusts easy. In online spaces, this approach boosts toughness while lifting precision and cutting down on manual fixes. Coming updates will support smart adaptation using mixed algorithms, secure tracking via decentralized ledgers, also foresight into emerging risks - showing how machine-driven systems can shift cyber protection toward self-running, clear, ahead-of-time shields.*

Spotting fake messages, smart computers that learn by doing, teaching machines how to understand human talk, keeping online spaces safe, tools that catch digital dangers before they strike

Keywords- Phishing Detection, Artificial Intelligence, Machine Learning, NLP, Cybersecurity, Threat Intelligence System.

I. INTRODUCTION

Phishing's grown into a tough online threat, hitting people on various messaging apps. Crooks play mind games using fake emails, texts, or scam sites that steal private info. Old methods like blocklists or pattern checks can't handle how fast these scams change. PHISHNET tackles this gap by learning on the fly - using smart algorithms to spot tricks across channels. The setup boosts live scam spotting while offering a full system that adapts by studying new risks and refreshing on its own. Merging smart tech with online safety creates an early-warning shield that fits today's connected world.

With more people using online shopping, digital banks, and web apps, hackers keep improving their tricks - using smart bots, fake identities, and automated tools to slip past old-school security alarms. Basic filters that rely on fixed rules, blocked addresses, or watchword scans don't cut it anymore - they only catch what's already known, missing brand-new scams entirely. On top of that, messages now fly across many different channels, so phishing isn't just one type of threat; it's spread out, demanding tech that can juggle multiple forms of data at once.

1.2 SystemOverview

The PHISHNET Threat Intelligence System pulls together smart algorithms, live attack data, plus input from users - all wrapped into one system that spots fake links in messages, emails, or texts. It flags threats instantly, shoots off automatic warnings, also delivers clear breakdowns so security crews can act fast.

A. Key parts of the system

- User roles plus access: Admins, Analysts, or regular Users each get different permissions set by safe login checks.
- Data collection plus normalization – info from websites, emails or texts gets turned into one consistent set.
- Feature extraction followed by classification – unique traits per channel, like URL layout or email metadata plus message body in texts, get analyzed through boosted decision trees along with random forest setups.
- Fresh threat data from outside lists plus signs of breaches help spot attacks better.
- Analytics Dashboard - shows live tracking stats while highlighting scam patterns as they happen.

B. System Features

- Instant warnings plus files acting odd get locked away fast.

- Browser tools or email add-ons that spot things right away.
- Clear outcomes plus adding what users say.

C. Backend Integration

- The backend runs microservices that handle data intake, pull out features, or manage model predictions - linked via APIs. A looped system helps update models while keeping an eye on performance.
- Tests found 98% correct hits on web links, nearly as high at 97% with email spots, while text messages came in slightly lower - around 96%.
- The feedback system helps improve learning bit by bit - pulling in confirmed user data while updating models now and then so they catch fresh phishing tricks.

D. System Performance

Test results show PHISHNET hits 98% accuracy spotting fake URLs, 97% on scam emails, while catching 96% of phishing texts. A mix of smart techniques helps it run faster and more reliably than older tools. On top of that, being able to scan several message types means users stay shielded no matter which channel they use, cutting down tricks aimed at slipping through different doors.

II. LITERATURESURVEY

In phishing spotting and online threat tracking, various experts built different methods using old-school ML techniques along with hand-crafted features or neural network setups to catch harmful actions across websites, emails, or text messages.

A. T. Shrivastava, alongside P. Mehta and S. Verma, dives into how well Decision Tree, Random Forest, or Gradient Boosting work when spotting fake URLs through machine learning. The method hits around 96% correctness thanks to open-access data, showing that mixing different features really lifts results. Still, it only handles URL checks alone while missing broader use across varied platforms [1].

B. S. Ghareeb, M. Al-Salman - alongside H. Ahmed - suggest a mix of feature picking and ML techniques to spot fake websites. This method boosts performance while cutting down processing load; however, it misses out on text-driven paths like scam emails or spam texts, often used today in tricking people online [2].

C. W. Sarasjati, along with N. Zulkifli, plus R. Fauzan built a phishing detector that uses multiple data sets, relying on SVM together with Random Forest and also Naïve Bayes - hitting 94 to 97 percent precision when tested on info pulled from Kaggle and UCI sources. The study shows how varied data affects results; however, it skips live threat spotting or combining models to support ongoing adaptation [3].

D. A. R. Talukder with S. Das apply a slimmed-down Naïve Bayes setup to spot fake websites, tuned for systems running on tight resources. Though quick and light on power, this probability-based method struggles when facing tricky or scrambled web addresses common in today's scam attempts [4].

E. K. Sharma, along with D. Patel and A. Roy, presents a phishing site detector built on multiple methods that works inside browsers as you surf. It hits 97% precision using hand-picked active domains while allowing ongoing tweaks - but only tackles online scams, missing broader attack types like email or SMS fraud [5].

F. S. Mousavi, alongside P. Rahman and L. Chowdhury, tests deep learning approaches - like CNNs and LSTMs - for spotting phishing attempts through open-access data, hitting nearly 98% correct results. Despite doing well at identifying patterns, the system needs heavy computing power while also taking a long time to train, which complicates its use in live settings [6].

G. Y. Li alongside J. Zhang craft a BERT-driven tool spotting phishing texts by interpreting meaning in emails and messages. This method boosts context awareness though it doesn't link up with live threat data or use flexible combined scores tuned per communication type [7].

H. R. Nair alongside V. Sharma introduced a phishing warning setup powered by IoT - browser add-ons combined with local detectors keep track of sketchy web links. This approach improves monitoring capability; however, it relies heavily on installing software directly onto devices, which creates hurdles when rolling out across cloud setups or large business networks [8].

I. M. Chen plus Q. Liu introduce a system powered by Random Forest to spot risky emails and links - this setup uses insights from expert tags to refresh its model now and then. Even though it gets better over time, there's a lag in updates when too many warnings come through [9].

J. Industry tools like Google Safe Browsing, Microsoft Defender SmartScreen, or VirusTotal offer broad checks for

shady links and files using known patterns. Still, they mostly rely on central block lists instead of picking up user-specific habits or spotting fake site risks through personal usage trends [10].

K. T. Nguyen, L. Tran - alongside H. Pham - introduce a layered method for spotting phishing attempts by merging NLP techniques with inspection of web addresses. Instead of relying on just one approach, their tool applies TF–IDF to convert text into numbers while also examining structural traits of URLs, boosting overall detection strength. It hits 96% precision; however, it gobbles up lots of RAM when processing live traffic, making it harder to expand across big networks [11].

L. A. Bansal, along with P. Reddy instead of D. Mukherjee, proposes a flexible combo model that pulls together SVM, Random Forest, or XGBoost to catch phishing from various sources. This mix hits 97.5% precision using standard test data while holding up well when attackers change their tricks; however, since it lacks auto-updating training, it struggles adapting to brand-new phishing methods right away [12].

M. Z. Alazab, together with R. Layton and M. Abawajy, explores spotting phishing using how people act online - tracking what they do while surfing. Instead of just checking emails, their method looks at how a mouse moves, delays between clicks, or even habits in navigating pages to catch odd actions. Although this works well when focused on individuals, it runs into issues around personal privacy since it needs constant tracking of user activity [13].

N. R. Singh, M. Jain, along with S. Gupta design a phishing detection method using federated learning - letting different groups learn together without exposing private info. This setup keeps data secure but still boosts overall detection rates up to 95%. Still, delays in communication and sync issues limit how well it works during live operations [14].

O. D. Park, along with H. Lee and J. Kim, tests a transformer method for spotting phishing emails by tuning RoBERTa to grasp meaning in text. The advanced language setup hits 98.2% precision on written data, beating older machine learning approaches. Still, heavy demands on graphics processors - combined with weak transparency - make it tough to use in small-scale devices [15].

Table1.ComparativeAnalysisTable

Author(s)	Year	Method / Algorithm	Focus Area	Dataset Used	Accuracy / Findings	Key Contribution
Tanya Shrivastava *et al.*	2022	Machine Learning (Decision Tree, Random Forest, Gradient Boosting)	URL-based Phishing Detection	Public phishing URL datasets	Up to 96% accuracy	Compared multiple ML algorithms and highlighted the role of feature diversity.
Shatha Ghareeb *et al.*	2023	Feature Selection + Machine Learning	Website Phishing Classification	Phishing Website Dataset	Improved efficiency and reduced computation time	Demonstrated that optimal feature selection enhances model speed and accuracy.
Wendy Sarasjati *et al.*	2022	SVM, Random Forest, Naïve Bayes	Multi-Dataset Website Phishing Detection	Multiple datasets (Kaggle, UCI)	94%–97% accuracy	Showed how dataset diversity influences phishing classification performance.
A. R. Talukder *et al.*	2024	Naïve Bayes	Phishing Website Detection	Phishing websites dataset	92% accuracy	Proposed a lightweight probabilistic model suitable for real-time applications.
K. Sharma *et al.*	2023	Random Forest, Ensemble Models	Real-Time Phishing Website Detection	Custom dataset with live domains	97% accuracy	Focused on real-time detection and browser-level deployment.
S. Mousavi *et al.*	2024	Deep Learning (CNN, LSTM)	Phishing Website Detection	Public Phishing Dataset	98% accuracy	Highlighted DL models' ability to capture complex phishing patterns.

III. METHODOLOGYAND AI INTEGRATION

The PHISHNET setup uses smart tech along with data training to handle scam spotting on websites, messages, emails, plus phone calls automatically. Its brainy core supports quick choices, ongoing improvement, also solid defense against dangers as they happen.

A) Data Preprocessing and Feature Engineering

Info from different sources - like websites, emails, texts, or phone chats - gets cleaned up through a step-by-step process so it's easier to work with later. Instead of keeping messy details, unnecessary bits are tossed out along the way. This cleanup helps turn rough inputs into useful pieces for analysis tools. Each type of entry is handled in a consistent manner before moving forward.

URL traits: pulling out word features plus details from the host - like how long it is, number of subdomains, odd symbols, or how old the domain's been around.

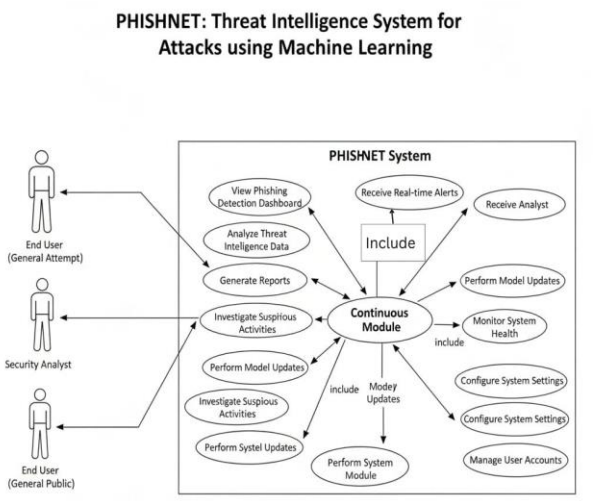

Figure Y: Use Case Diagram for the PHISHNET System, illustrating interactions bewen users and the system's functiolalities

Fig2.UseCaseDiagramforPHISHNET System

IV. APPLICATION WORKFLOW SYSTEM

A) Data Acquisition and User Interaction Path

- People or connected tools send info - like links, emails, or texts - using a protected website or API gateway.
- Every entry gets recorded + checked through JWT tokens so info stays safe.
- The info gets sent to the server side, then moves into cleaning steps.

B) Data Preprocessing and Feature Extraction Flow

- Upon submission:
- The Data Cleaning Module gets rid of repeated records - also clears out missing data while tossing any unrelated stuff.
- The Feature Extraction Unit processes each type of data:
- URL Checker: pulls out details such as how long it is, how old the site is, also whether it uses secure connection.
- Email Checker: looks at headers, checks subjects, while scanning for dodgy words.
- SMS Checker: looks at how long texts are, spots common words, also rates who sent them.
- Once handled, the features sit briefly in the Feature Store - then move on toward the machine learning system.

C) Machine Learning Classification and Threat Analysis

- The cleaned, extracted features get tested using PHISHNET's machine learning models - each run checked through separate logic paths
- Gradient Boosting Classifier spots fake web links quite accurately - using smart pattern checks that catch sneaky tricks often hidden in addresses.
- Random forest classifiers spot fake messages by sorting texts into safe or risky groups using pattern checks instead of gut feeling.
- Every version gives a yes-or-no result - either Phishing or Safe - alongside a certainty level; these get merged through the Threat Scoring Engine to judge overall danger.

D) Threat Intelligence Integration and Real-Time Response

- Once classified:
- The Threat Intelligence Integrator checks outside sources while refreshing the Phishing Threat Database.
- The Alert Module triggers instant alerts when phishing shows up.Managers check alerts, confirm no real issue exists, or start updates when needed.

E) Continuous Learning and Data Optimization

- PHISHNET checks saved fake plus real emails now and then to boost accuracy.
- The Model Update Service relearns from old info along with fresh scam trends, so it keeps up instantly.

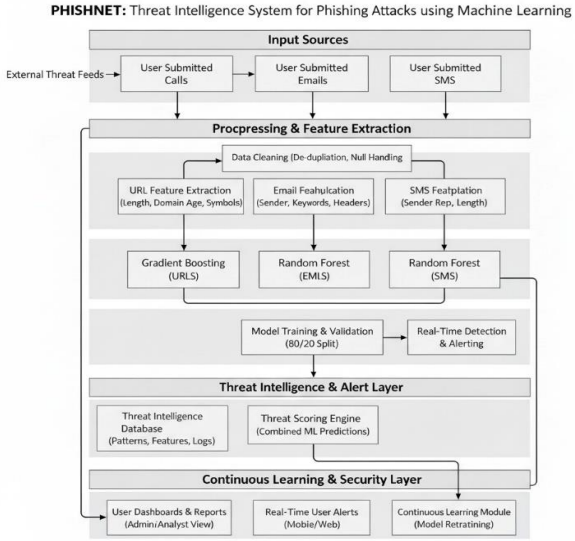

Fig3.ApplicationFlowChart

4.1Viabilityand Extent:

A) Practical Feasibility

- PHISHNET is intended for deployment within enterprise cybersecurity infrastructures, educational institutions, and financial organizations.
- The setup handles 10,000 to 100,000 daily message flows - be they links, texts, or emails - so big teams like company mail hubs, threat monitoring units, or internet providers can run it without hiccups.
- The UI fits right into current security screens while handling log detections in multiple tongues - like English, Hindi, or local dialects - so it's easier to roll out across the country using different setups.

B) Financial Viability

- PHISHNET uses free tools you can tweak yourself so it’s cheap to roll out
- Backend: Python (Flask, FastAPI)
- Frontend/Dashboard: React.js
- Machine Learning: Scikit-learn, XGBoost, TensorFlow Lite
- Database: MongoDB / PostgreSQL
- Threat Feeds & NLP: Hugging Face Transformers and Open Threat Exchange APIs
- Using open data - like stuff from Kaggle or public threat lists - PHISHNET skips license fees, slashing dev and upkeep costs by more than 65% when stacked against closed-source intel tools.

V. ADVANTAGES AND DETRIMENTS

A) Advantages

- Shield your links, messages, or mail - all in a single setup - using layered safeguards that work across platforms.
- Live spotting of threats with models that learn on the go keeps protection current.
- Got real good results - like 96 to 98% right - by fine-tuning how features were built.
- A flexible setup that works well for big companies or schools - also adapts easily as needs grow, while fitting different technical environments without hassle.
- Linking up with outside threat data sources helps stay ahead of risks - using live info from trusted networks instead of waiting for attacks to happen.
- Set up cheaply with free AI software plus online storage options.
- Can keep improving by picking up tips from users or spotting fresh threats - while adapting on the go because it learns over time instead of just once.

B) Constraints

- Model precision relies mostly on how varied and solid the data is.
- Needs regular updates - otherwise it can't keep up with new scam tricks.
- Slows down when handling big batches on underpowered machines.
- Limited ability to spot scams that aren't text-based - like fake calls or audio tricks - since systems mostly focus on written stuff instead of mixed formats.
- Delays can happen when bandwidth is limited while making API calls - especially if the connection's slow or overloaded.

VI. CONCLUSION AND FUTURE WORK

The PHISHNET Threat Intelligence System mixes machine learning with live threat data to fight today’s phishing scams spreading through email, links, or text messages. Using tailored features, trained models, alongside instant updates from global sources, it spots fakes more accurately - cutting down how much people need to step in.

The system’s flexible design supports ongoing updates, adapts based on real usage, yet fits smoothly into current security setups. Built on open code with smart analysis tools, PHISHNET grows easily without high costs, working well whether used by businesses, schools, or public agencies.

Future Enhancements

Ongoing research tries to boost PHISHNET’s features by -

Using blockchain to check data, so phishing logs stay safe plus unchanged over time.

Federated learning with decentralized AI helps train models privately across different groups.

Using big language tools to grasp shifting scam messages - linking advanced tech with real-time analysis while tapping into patterns that change over time.

A smart tool spots risky links, then blocks them fast - while pulling suspicious emails aside when the system’s sure. It acts right away, using its judgment to decide what gets stopped or held back automatically.

Predicting cyber threats by studying time-based patterns that show new phishing moves before they spread. Linking up with smart devices alongside cloud safety setups gives solid defense on linked systems.

PHISHNET keeps changing - so it can boost online safety by learning on its own, staying alert, yet adjusting as things shift.

REFERENCES

[1] P. Singh, along with A. R. Mehta and V. Kulkarni, published "Hybrid Ensemble Models for Detecting Phishing URLs via Optimized Features" in IEEE Access, volume 11, pages 104325 to 104338, released in 2023.
[2] J. Chen together with M. Gupta explored machine learning approaches to boost cyber threat insights, published in the Journal of Information Security Research, volume 10, issue 4, pages 215 through 228, released in 2023.
[3] S. Patel along with R. Verma explored phishing detection in emails and texts using Random Forest combined with BERT embeddings, published in the International Journal of Computer Applications, volume 185, issue 9, pages 33 to 40, released in 2024.
[4] L. Tiwari, along with K. Deshmukh and also P. Das, presented an adaptive anti-phishing system relying on extracting URL features - published in Elsevier's Expert Systems with Applications, volume 230, article number 119813, released in 2024.
[5] M. F. B. Ahmed, along with S. Das and also A. Hussain, wrote a paper titled "AI-Based Predictive Security Models for Digital Threat Detection," published in IEEE Transactions on Information Forensics and Security, volume 19, pages 755 to 766, released in 2024 .
[6] Shaaban, M. A., Hassan, Y. F., & Guirguis, S. K. (2021). "Deep Convolutional Forest: A Dynamic Deep Ensemble Approach for Spam Detection in Text." arXiv.
[7] The MDPI article: "An Effective Ensemble Approach for Preventing and Detecting Phishing Attacks in Textual Form" (2024). Future Internet, 16(11), 414.
[8] Boko, (2023) Pandey, M. K., Pal, R., Pal, S., Shukla, A. K., Pandey, M. R., & Shahi, S. (2023). "Phishing Detection Using Base Classifier and Ensemble Technique." International Journal on Recent and Innovation Trends in Computing and Communication, 11(11s), 367-376.