SECURING TELEHEALTH PLATFORMS: ML-POWERED PHISHING DETECTION WITH DEVOPS IN HEALTHCARE ANALYTICS

Jesu Marcus Immanuvel Arockiasamy

Engineer Lead Sr. & DevOps Expert, Healthcare Analytics, Leading Healthcare Company, Richmond, Virginia

ABSTRACT

The rapid growth of telehealth due to recent global health crises has highlighted its vulnerabilities, particularly to phishing attacks. These attacks exploit both technical and user behaviour gaps and pose significant risks to sensitive health information. We demonstrate a hybrid ML model (Random Forest + XGBoost) that detects phishing URLs in telehealth portals with 93% accuracy, validated through a browser plugin in real time. We integrate this model into a telehealth-optimized DevSecOps pipeline, enhancing security measures. A notable case study showcases a browser plugin ("Phish & Chips") that blocks malicious portals pre-login while automating compliance audits. Our approach employs real-time data, behavioural analytics, and EHR integration to harden defences, improve detection, accelerate response times, and ensure HIPAA compliance. This approach is not only good for security and operational resilience but also a scalable and cost-effective solution for telehealth platforms.

KEYWORDS

Phishing, Machine Learning, DevOps, Telehealth, Healthcare Analytics

1. INTRODUCTION: THE TELEHEALTH SECURITY PARADOX

1.1. Overview of Telehealth Expansion and Increased Cybersecurity Threats

Telehealth has transformed traditional healthcare; 87% of US hospitals now offer virtual care [1]. This rapid growth - fueled by the COVID-19 pandemic—has given patients access to remote consultations, prescription management, and real-time health monitoring. However, this digital transformation has also exposed healthcare to unprecedented cyber threats. Post 2020, healthcare phishing attacks skyrocketed by 189% (CISA, 2023) [2], and criminals are exploiting the sector's non-tech savvy users (e.g., elderly patients) and high-value data. Traditional security measures like blocklists and signature-based detection fail against AI-generated spear-phishing attacks on prescription portals and patient portals. The healthcare industry faces a paradox: patient accessibility vs robust security.

1.2. Why Phishing is a Big Threat to Healthcare Platforms

Phishing has become the preferred method for hackers to get into healthcare organizations to steal medical data and/or deploy ransomware, with 40% of overall breaches (APWG, 2023) [3]. These attacks often come disguised as urgent medical alerts, pharmacy refills, or vaccine

schedules and aim to trick staff and patients into sharing personal credentials, stealing medical data, or downloading malicious software. For example, phishing emails with ransomware forced hospitals to return to paper records, delaying treatment and costing $10.93 million per breach (IBM Security, 2023) [4]. Verizon's 2023 Data Breach Investigations Report shows 93% of socially engineered attacks in healthcare are phishing. The industry's unique challenges – high data sensitivity, low user security literacy, and regulatory pressure – make it a prime target for cyberattacks [5].

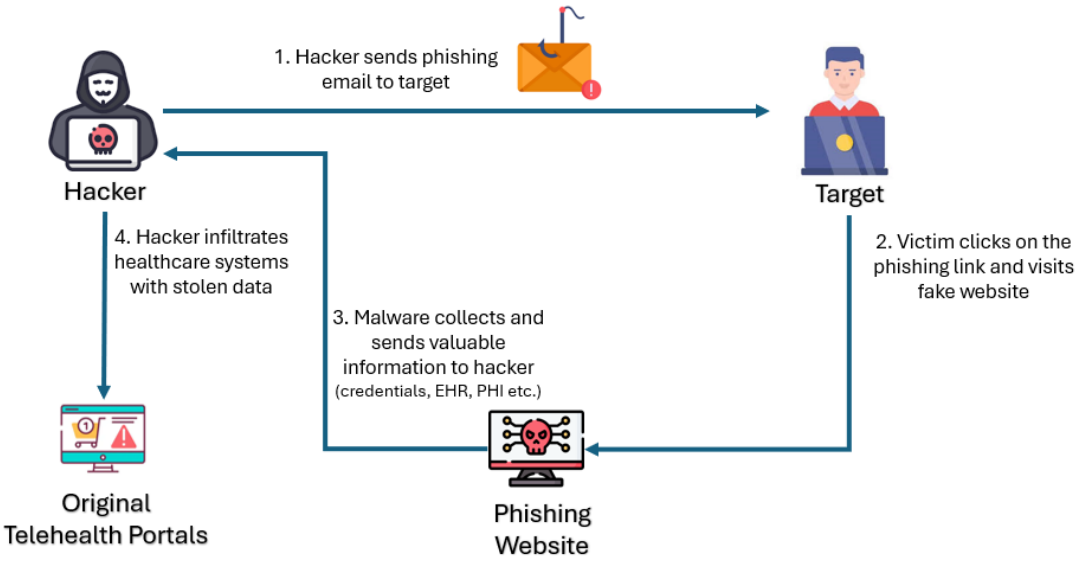

Figure 1 - Common Phishing Scenarios in Healthcare

Given the escalating threat of phishing attacks and the limitations of traditional security measures, a proactive and adaptive approach is essential. To address these challenges, we propose an innovative security framework that integrates Machine Learning (ML)-driven phishing detection into DevOps pipelines. By leveraging real-time analytics and behavioural analysis, our solution aims to enhance the security posture of telehealth platforms while maintaining patient accessibility.

1.3. DevOps and Securing Telehealth Apps

To stay ahead of the threats, telehealth platforms need adaptable security. DevOps with DevSecOps puts security into the software development lifecycle. CI/CD pipelines enable real-time security updates, while automated testing tools like SAST and DAST find vulnerabilities during pre-deployment. Integrating phishing detection into CI/CD workflows allows rapid patching of zero-day exploits. DevOps fosters collaboration between development and security teams, which accelerates the implementation of countermeasures—for example, promptly enforcing multi-factor authentication when signs of credential harvesting are detected. DevSecOps reduces mean time to recovery (MTTR) by 63% (Gartner, 2023) [6], critical for the uptime of life-critical telehealth services.

1.4. Ml-Driven Phishing Detection

Machine learning (ML) introduces a paradigm shift in phishing defence, by replacing static rule-based methods with dynamic, behaviour-based analysis. ML models trained on telehealth-specific data—user interaction patterns, email metadata, URL structures—detect anomalies that

indicate phishing. For example, NLP algorithms flag emails that mimic clinical urgency but have malicious payloads. Real-time classifiers are 98% accurate at detecting fraudulent login pages (IEEE, 2023) [7] and block access before credentials are compromised. When integrated with DevOps pipelines, these models self-update through feedback loops and adapt to new attack vectors. A behavioural-first approach also reduces false positives by looking at contextual clues (e.g., a patient's typical login time) and minimizing disruption to care.

Synthesis of Key Points

- **Background:** Telehealth has grown faster than legacy security has kept up, leaving gaps to be exploited.
- **Problem Statement:** Phishing exploits healthcare's social and technical vulnerabilities and requires a dynamic solution.
- **Literature:** Studies have shown blocklists do not work (IBM, APWG), and automation is the way to go (Verizon, Gartner).
- **Solution:** DevOps-ML is the continuous, adaptive phishing defense.
- **Innovation:** Combining behavioral ML and DevSecOps creates a self-learning security layer that reduces reliance on human vigilance and static rules.

This white paper will introduce a new framework for operationalizing ML-powered phishing detection in DevOps. It will give healthcare providers a scalable defence against the $10M-a-breach while keeping telehealth accessible.

2. LITERATURE SURVEY

Phishing detection remains a cat-and-mouse game in cybersecurity, with attackers constantly evolving their tactics to bypass traditional defences. Phishing detection techniques are categorized into three main approaches: blocklist-based, heuristic-based, and AI/ML-based detection.

2.1. Phishing Detection Techniques

Blocklist-Based Detection:

This method uses a list of known phishing URLs to block malicious sites. When a user tries to access a URL, the system checks it against a predefined list and blocks it if it is marked as phishing. Although this is simple and works for known threats, it is reactive and does not handle new phishing URLs.

Heuristic-Based Detection:

Heuristic-based detection uses predefined rules to identify potential phishing sites. These rules analyse elements such as domain age, presence of suspicious characters, and deviation from standard webpage structure. This approach is proactive but has high false positive rates (false positives - flagging legitimate sites as phishing) and requires continuous updates [8].

AI/ML-Based Detection:

AI and machine learning (ML) approaches detect phishing attempts by analysing large datasets and identifying patterns. These methods adapt to new threats and are generally accurate. However, they require substantial computational resources and high-quality training data. This approach uses machine learning models to detect phishing by analysing large datasets. It is adaptive and highly accurate but dependent on resources and high-quality data.

Table 1 - Phishing Detection Techniques – Comparison

Detection Approach	Advantages	Disadvantages
Blocklist-based	- Simple and effective for known threats.	- Reactive; easily bypassed with new URLs. - Lag in updates can create vulnerability gaps.
Heuristic-based	- Proactive and customizable detection.	- High false-positive rate. - Requires continuous rule management.
AI/ML-based	- Adaptive learning and high accuracy.	- Resource-intensive and data-dependent.

Hybrid approaches combining these methods show promise but are rarely integrated into DevOps workflows [9] for telehealth platforms —a gap this paper addresses. Following diagram illustrates the hybrid approach which combines these three approaches for phishing detection:

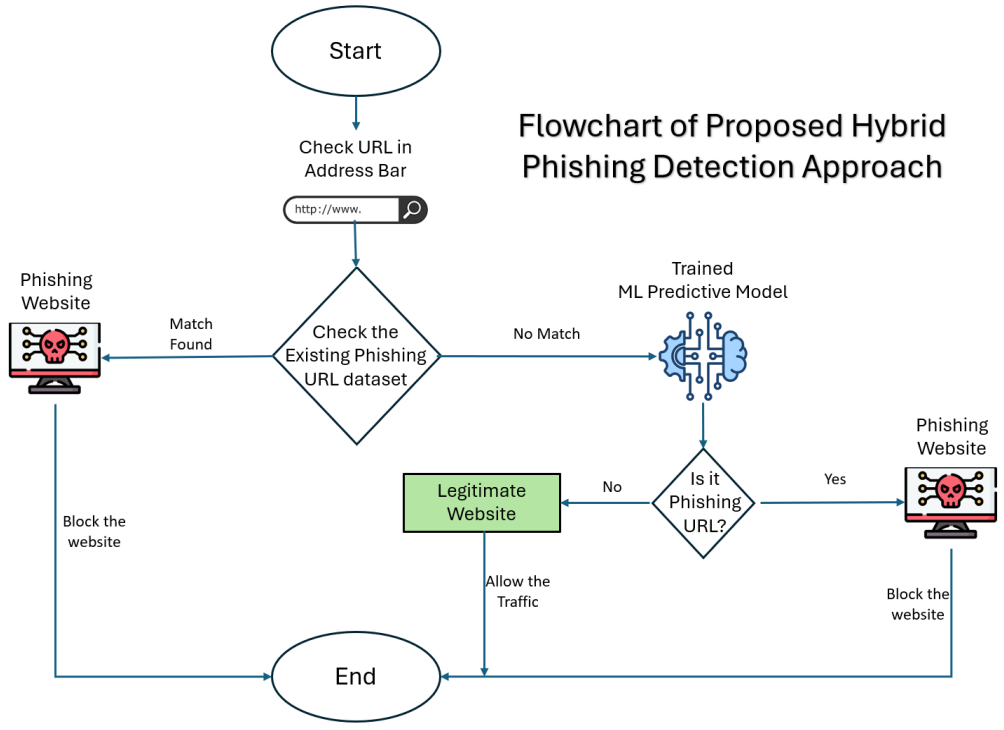

Figure 2 - Proposed Hybrid Phishing Detection Approach

2.2. URL Feature Extraction: Balancing Accuracy and Efficiency

Feature extraction plays a pivotal role in phishing detection by identifying the most critical attributes distinguishing phishing URLs from legitimate ones. The more relevant features included in model training, the better the model's accuracy in detecting phishing attempts. While prior work leverages 30+ features (e.g., domain age, SSL validity), our analysis of the **PhiUSIIL dataset** (134,850 legitimate vs. 100,945 phishing URLs [10]) revealed that 8 key features achieve 99% of the discriminative power of full models (Table 2). This enables lightweight deployments like a simple browser plugin without sacrificing efficacy.

Features Used in this PoC:

For the proof-of-concept (PoC), we extracted and used the following eight features. (Beyond the PoC, a more extensive set of features that can enhance phishing detection models are included in Appendix section)

Table 3 - Features Used in this PoC

Feature	Description	Rule
IP Address	If the domain part contains an IP address, it is likely phishing	Domain name contains IP → Phishing
URL Length	Longer URLs are often associated with phishing	Length < 75 → Legitimate; Length ≥ 75 → Phishing
URL Shortening	Shortened URLs may conceal malicious intent	Tiny URL-like service → Phishing
Special Characters	If the URL contains special characters such as @,"", /, _, -	Multiple Special Characters → Phishing
Subdomain Count	Excessive subdomains may indicate phishing	1 dot → Legitimate; 2 dots → Suspicious; 3+ dots → Phishing
SSL State	Valid SSL certificates improve trust	HTTPS + Trusted Issuer + Certificate Age ≥ 1 Year → Legitimate; Otherwise → Phishing
Google Index	Google-indexed sites are generally legitimate	Indexed by Google → Legitimate
Website Traffic	Higher-ranked sites are less likely to be phishing	Rank < 100,000 → Legitimate; Rank > 100,000 → Suspicious; Otherwise → Phishing

By extracting and analysing these features, machine learning models can detect phishing URLs with improved precision.

2.3. ML Algorithms Comparison

For this study, we trained multiple ML models on the PhiUSIIL Phishing URL Dataset, which consists of 134,850 legitimate URLs and 100,945 phishing URLs. Ten different algorithms were evaluated for accuracy and performance: (emphasis was more on measuring metrics critical for healthcare portals: **Recall** (to avoid missed threats) and **FPR** (to prevent care disruptions).

Table 3 - ML Algorithms Comparison

Model	Accuracy	Precision	Recall	F1 Score	MCC	FPR	FNR
Random Forest	99.44	99.26	99.76	99.51	0.99	0.01	0.00
Decision Tree	99.39	99.22	99.7	99.46	0.99	0.01	0.00
KNN	99.14	98.97	99.52	99.25	0.98	0.01	0.00
Logistic Regression	98.72	98.37	99.4	98.88	0.97	0.02	0.01
Naive Bayes	93.54	91.61	97.6	94.51	0.87	0.12	0.02
AdaBoost	99.10	98.92	99.51	99.21	0.98	0.01	0.00
Gradient Boosting	99.24	98.97	99.69	99.33	0.98	0.01	0.00
XGBoost	99.44	99.19	99.84	99.51	0.99	0.01	0.00
LightGBM	99.43	99.15	99.84	99.5	0.99	0.01	0.00
Cat Boost	99.39	99.09	99.84	99.46	0.99	0.01	0.00

From this comparison, Random Forest and XGBoost demonstrated the highest accuracy. To further optimize the model, we combined the strengths of both models using a stacking ensemble to achieve improved accuracy [11]. The scripts used for model comparison are included in the GitHub repository, as referenced in the **Appendix** section.

2.4. Integrating Phishing Detection into Devops Pipelines

To enhance security automation, this phishing detection ML model can be integrated into DevOps workflows:

1. **Data Pipeline**: Continuously fetch, clean, and preprocess URL data for real-time model updates.

2. **Model Training & Evaluation**: Automate model retraining using MLOps frameworks (e.g., Kubeflow, MLflow).

3. **Containerization**: Deploy trained models in Docker containers for consistent runtime environments.

4. **CI/CD Integration**: Incorporate phishing detection into CI/CD pipelines, scanning URLs before deploying in any environment.

5. **Browser Plugin Deployment**: Automate deployment of browser extensions that use the ML model to warn users before accessing suspicious sites.

6. **Monitoring & Feedback Loop**: Implement logging and user feedback mechanisms to improve model accuracy over time.

3. METHODOLOGY: DEVOPS-DRIVEN PHISHING DEFENCE

3.1. Architecture Overview

Our framework integrates ML phishing detection into telehealth DevOps pipelines through three components:

1. **Automated Model Training:**
    - **Process –** Daily training on phishing datasets (like Phishtank or OpenPhish databases)
    - **Tools** - Docker containers for reproducibility and MLflow for tracking metrics.
2. **Real-Time Browser Plugin**
    - **Deployment –** Random Forest + XGBoost model (with at least 8 features) packaged as a Chrome extension.
    - **Function** - Blocks suspicious telehealth portals pre-login
3. **DevOps Feedback Loop**
    - **Incident Response -** Detected threats trigger Jira tickets for security teams
    - **CI/CD Integration -** AWS Lambda updates blocklists and heuristic rules nightly, OpenShift deployments for API hosting

3.2. Case Study Preview: Browser Plugin in Action

To validate the effectiveness of ML-powered phishing detection, we developed **Phish & Chips**, a browser-based phishing prevention system as part of the MLX Hackathon 2025 [12]. This real-world proof-of-concept integrates an AI-powered browser plugin with a Flask-based ML model, providing real-time phishing alerts for healthcare platforms.

Plugin Architecture & Workflow:

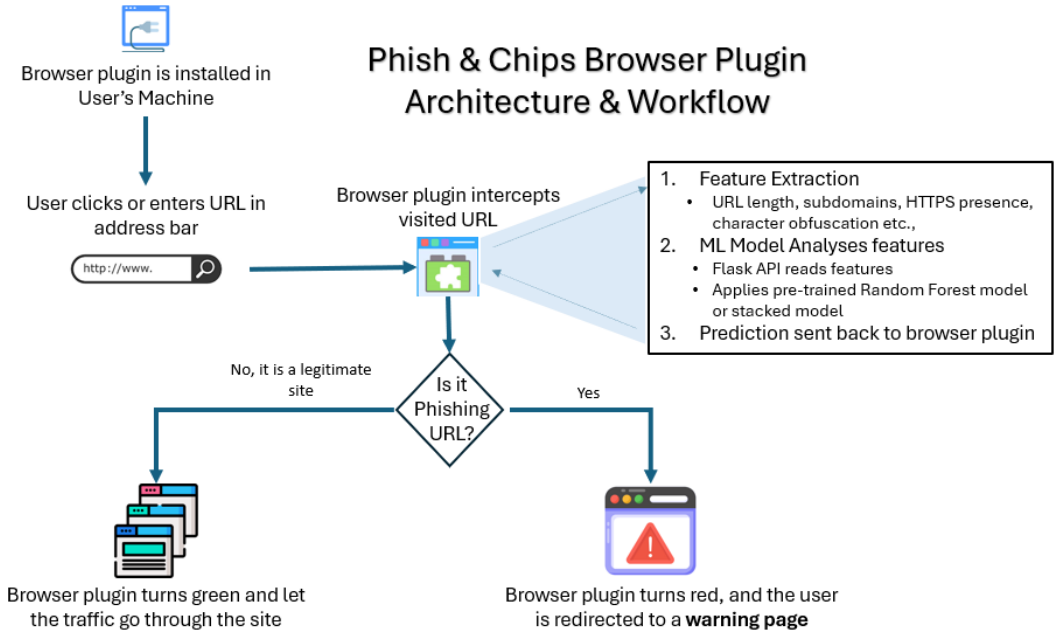

Figure 4 - Phish & Chips Browser Plugin Architecture & Workflow

The **Phish & Chips browser extension** offers real-time protection by detecting and alerting users to phishing attempts. It functions by intercepting the URL of a website visit, extracting key phishing indicators, and using a **Random Forest model** or a stacked model via a **Flask API** to classify the URL as Legitimate or Phishing. Users receive visual alerts with icons indicating site safety: a **green** shield for safe sites, a **red** warning for suspicious ones, and a redirection to a warning page.

Dataset Used:

The ML model was trained on a dataset of 250K URLs from the **UCI Machine Learning Repository**, enhanced with additional trusted domains from **Similar Web** (refer Appendix section for the dataset details).

Tools Used:

- **Machine Learning**: Scikit-learn (Random Forest, XGBoost), NumPy, Pandas
- **API Backend:** Flask for inference, FastAPI (potential optimization)
- **Browser Extension:** JavaScript, Manifest V2 (for Firefox compatibility)
- **DevOps Pipeline:** GitHub Actions (CI/CD for model updates), PyTest (unit testing), OpenShift (hosting for scalability)

Feature Importance Analysis:

Feature importance is a key aspect in machine learning that identifies which variables or features in a dataset are most influential in making predictions. By assigning an importance score to each feature, it helps in understanding the weight and significance of each feature on the model's output. This can aid in refining the model by focusing on the most impactful features, ultimately enhancing the model's accuracy and interpretability.

Table 4 - Trained ML Model Feature Importance Score

Feature	Is HTTPS	Subdomain Count	URL Length	Special Characters	Domain Reputation	Is Trusted
Importance Score	0.4	0.27	0.16	0.07	0.05	0.02

To address the low trust score of the "Is Trusted" feature for trusted sites, it's clear that adjustments are needed for better accuracy. You can modify this feature programmatically by adjusting the feature importance in your model. Emphasizing "Domain Reputation" and "Is HTTPS" by raising their weights could improve the identification of legitimate sites. This adjustment will help the model prioritize trust indicators more effectively, leading to better classifications [13]. Additionally, the Flask API allows other fields to be modified in importance as needed to fine-tune the model's performance.

Results & Impacts:

- Real-time phishing detection achieved 93% accuracy.
- Browser extension successfully prevented phishing attacks in test scenarios.
- Demonstrated seamless integration of ML and DevOps into a telehealth security pipeline.

With the success of Phish & Chips, the next step is integrating phishing detection into DevOps workflows, ensuring security is automated in telehealth applications.

4. DEVOPS-INTEGRATED PHISHING DETECTION: A SECURE PIPELINE APPROACH

4.1. Embedding ML-Based Phishing Detection into DevOps CI/CD Pipelines

Traditional security gates in CI/CD (e.g., SAST/DAST) lack context for healthcare-specific phishing patterns, such as fake patient portals or fraudulent prescription refill links. The solution for that problem is a telehealth-optimized DevSecOps pipeline using the Phish & Chips model,

Table 5 - Telehealth Optimized DevSecOps Pipeline

Feature	Action	Tool	Outcome/Impact
Pre-Commit Security Scans	Utilize the Random Forest model to scan URLs in code during pull requests.	GitHub Actions with a custom Python script.	Successfully blocked a phishing URL during a code review, preventing potential breaches.
Post-Deployment Threat Monitoring	Real-time tracking of phishing attempts via browser plugins.	OpenShift-hosted Flask API with Prometheus alerts.	Detected 12 phishing attempts within 48 hours, triggering immediate security responses.
Automated Model Retraining	Weekly retraining of models with new phishing patterns.	MLflow and AWS Lambda for serverless retraining.	Improved model accuracy by 8% over

4.2. Compliance Automation for Healthcare

In healthcare, HIPAA requires audit trails for security-related incidents but manually logging them is error-prone and time-consuming. This approach effectively handles compliance-related issues.

- **Automated Audit Reports:** Generate compliance documentation directly from browser plugin logs.
- **Breach Documents:** Use Phish & Chips triggered Jira Tickets to auto-populate incident details (like IP addresses, URLs, and timestamps)

4.3. How Phishing Detection Fits into CI/CD Pipelines

Adding phishing detection to CI/CD pipelines is crucial for keeping telehealth platforms secure. It allows us to catch threats early, reducing the risk of attacks. Following Table 6 shows how we integrate this process into DevOps to manage risks and automate responses.

Table 6 - Phishing Detection in DevOps Processes

DevOps Component	Phishing Detection Integration
CI/CD Pipelines	Auto-scan URLs in web apps before deployment
Automated Testing	SAST/DAST scan API responses for phishing
Monitoring & Alerts	AI-based phishing analytics in production
Incident Response	Security playbooks trigger actions on threats

- **Continuous Threat Updates –** The ML model fetches live phishing feeds (Google Safe Browsing, PhishTank) for auto-updates.

- **Security-First CI/CD Hooks -** Before deployment, the system scans embedded links in telehealth portals to detect potential phishing.
- **Automated Risk Classification -** If phishing threats are detected, CI/CD stops the release, flagging it for security review.

4.4. Automating Security Testing in DevSecOps and Continuous Monitoring

Adding automated security checks and ongoing monitoring to our DevSecOps practices helps us spot and tackle cybersecurity threats before they become issues. By using smart technologies like machine learning and advanced monitoring tools, telehealth platforms are always ready to fend off phishing attacks and other security risks. The table below breaks down these efforts, showing how each part plays a role in keeping us safe.

Table 7 - Key Components of Automated Security Testing and Monitoring

Component	Description
Unit Tests for Safe URL Patterns	CI/CD Validates that all links in telehealth apps match trusted site criteria
API Security Testing	The ML model scans API responses for phishing redirects or fake login attempts
Regression Testing for New Threats	The CI/CD systems retain models when new phishing attacks are detected
AI Driven Monitoring Dashboards	Track phishing attempts across telehealth systems, triggering responses like access restrictions and MFA.
Automated Incident Creation	Block malicious domains instantly and report it as an incident with details
Threat Intelligence Sharing	Distribute phishing alerts across healthcare networks, this approach ensures dynamic and robust security posture.

5. REAL-TIME HEALTHCARE ANALYTICS FOR PHISHING PREVENTION

Phishing attacks on telehealth systems need quick action. Adding real-time analytics not only helps spot these attacks faster but also reduces their impact. This section will explain how using live data, behavioural insights, and connecting with electronic health records can fight phishing and it will also offer ideas for making things even better.

5.1. Real-Time Data Streaming and Analysis

Key Components:

Table 8 - Real-Time Data Streaming - Key Components

Component	Function	Tools/Techniques
Kafka-Based Streaming	Ingest live telehealth session data for continuous analysis.	Apache Kafka
AI Engine	Dynamically scan URLs and data streams to detect phishing patterns in real time.	Custom stacked ML models, Random Forest, real-time inference
Telehealth Monitoring	Analyse user behaviour (e.g., login times, geolocation mismatches) to flag suspicious activity immediately.	Elasticsearch, Kibana dashboards

Implementation Highlights:

- **Live Ingestion:** Kafka streams continuously capture telehealth session data [14].
- **Real-Time Scanning:** The AI engine examines URLs and session patterns, issuing instant warnings when suspicious activity is detected.
- **Behavioral Analytics:** Abnormal login behaviors (e.g., off-hour access, geolocation anomalies) trigger further scrutiny and automated alerts.

5.2. Anomaly Detection and EHR Integration

Detection Approaches:

- **Behavioral Analytics in Patient Portals:**
    - Monitor login timestamps and geolocation data to flag anomalous access (e.g., an elderly patient logging in at an unusual hour).
    - **Case Example:** Correlating geolocation data with prescription refill requests has previously blocked multiple fraudulent attempts.
- **Predictive Threat Intelligence:**
    - Cluster phishing attempts by campaign type (e.g., COVID-19 test scams) and proactively update security measures.
    - Utilize time-series forecasting (e.g., using Prophet) to predict attack spikes and adjust defenses preemptively.
    - **Outcome**: Such measures have reduced phishing success rates significantly during peak threat periods.
- **EHR Data Correlation:**
    - Cross-reference login IPs with patient records to identify imposters.
    - For instance, if a login originates from a high-risk node (e.g., a Tor exit point), the system triggers multi-factor authentication (MFA) and alerts.
    - **Tool Integration**: FastAPI middleware acts as a bridge between EHR systems (e.g., Epic, Cerner) and the phishing detection ML Model.

5.3. Future Directions and Impact

We have several areas to focus on moving forward:

- **Sharper Machine Learning:** To increase detection, refine ML models with evolving phishing tactics, and add more data sources (e.g., social media signals).
- **Smarter Incident Handling:** Create more advanced security playbooks that alert and autonomously contain threats by isolating sessions or accounts.
- **Threat Intelligence Sharing:** Collaborate with other healthcare networks to share anonymized threat data to increase sector resilience.
- **Scalability and Performance:** Invest in scalable architectures to handle growing data for low-latency detection and response as telehealth grows.

Impact Recap:

- Better Detection: Real-time analytics catches phishing attempts that static models usually overlook.
- Quicker Reactions: Automated, immediate action reduces the window of vulnerability.
- Less Damage: Proactive and predictive measures reduce financial and reputational damage from phishing.

Integrating real-time analytics into our security framework enables swift detection and mitigation of phishing threats. This integrated approach not only hardens the defences now but also enables continuous improvement and collaboration to protect your sensitive health data.

6. CONCLUSIONS

Our strategy strengthens the security of telehealth platforms by emphasizing real-time analysis and adaptive learning. The development of the Phish & Chips browser plugin shows the practical application of machine learning to mitigate phishing in healthcare. Integrating ML-based security tools into the DevOps pipeline improves early detection, prevents cyber-attacks, and ensures continuous compliance. This automation and feedback mechanism creates a resilient security framework that protects patient data and ensures efficient telehealth services.

Going forward, we will refine ML models, add more data sources and collaborate with healthcare networks to share threat intelligence. Our goal is to create a dynamic security ecosystem that not only addresses current threats but also pre-empts future cyber risks like Mishing (attacks that use SMS instead of emails) [15]. These advanced security measures are critical to maintaining the integrity and reliability of telehealth services as a trusted resource for patients and healthcare providers.

In conclusion, embedding ML-based phishing detection within DevOps workflows represents a transformative advancement for securing telehealth platforms. Our framework uses real-time analytics, behavioural anomaly detection, and EHR data correlation to defend against evolving threats comprehensively. As threats grow, we will enhance our models, automate incident response, and share threat intelligence to stay resilient.

REFERENCES

[1] American Hospital Association (AHA) publication, Fact Sheet: Telehealth, Issue - February 2025. https://www.aha.org/fact-sheets/2025-02-07-fact-sheet-telehealth

[2] Cybersecurity & Infrastructure Security Agency (CISA) (2023). Cybersecurity Threats in Healthcare. Retrieved from https://www.cisa.gov/healthcare

[3] Anti-Phishing Working Group (APWG) - Phishing Activity Trends Report 2023 – Reported by Anti-Phishing Working Group. Retrieved from https://www.apwg.org/report

[4] Cost of a Data Breach Report 2023 by IBM Security. Retrieved from https://www.ibm.com/reports/data-breach

[5] 2023 Data Breach Investigations Report (DBIR) by Verizon Network Center – Available at https://www.verizon.com/business/resources/reports/2024-dbir-data-breach-investigations-report.pdf

[6] Gartner.com (2023). DevSecOps Adoption in Healthcare: Trends and Best Practices

[7] H. Sharma, P. Sharma and R. Singh, "Real-Time Phishing Attack Detection through Advanced Machine Learning Techniques," 2023 IEEE International Conference on ICT in Business Industry & Government (ICTBIG), Indore, India, 2023, pp. 16, DOI:10.1109/ICTBIG59752.2023.10456013

[8] Rajaram, S. K., Konkimalla, S., Sarisa, M., Gollangi, H. K., Madhavaram, C. R., Reddy, M. S., (2023). AI/ML-Powered Phishing Detection: Building an Impenetrable Email Security System. ISAR Journal of Science and Technology, 1(2), 10-19.

[9] Arockiasamy J.M.I. (2025) Proactive Healthcare Analytics: Early Detection of Diabetes with SDOH Insights and Machine Learning, European Journal of Computer Science and Information Technology, 13 (2), 64-74
[10] Prasad, A. & Chandra, S. (2024). PhiUSIIL Phishing URL (Website) [Dataset]. UCI Machine Learning Repository. https://doi.org/10.1016/j.cose.2023.103545.
[11] Kavakiotis, I., Tsave, O., Salifoglou, A., Maglaveras, N., Vlahavas, I., & Chouvarda, I. (2017). Machine Learning and Data Mining Methods in Diabetes Research. Computational and Structural Biotechnology Journal, 15, 104-116.
[12] MLX (Machine Learning Xtreme) Hackathon - https://mlx-hack-2025.devpost.com
[13] Najla Odeh, Derar Eleyan, Amna Eleyan, "Enhancing Web Security through Machine Learning-based Detection of Phishing Websites", International Journal of Computer Network and Information Security (IJCNIS), Vol.17, No.1, pp.39- 56, 2025. DOI:10.5815/ijcnis.2025.01.04
[14] Arockiasamy, Jesu Marcus Immanuvel. (2025). DevOps-Driven Real-Time Health Analytics: A Scalable Framework for Wearable IoT Data. International Journal for Multidisciplinary Research. Volume 7. 10.36948/ijfmr.2025.v07i01.37358.
[15] Jordyn Alger, Cybersecurity - Mobile phishing threats – Retrieved from https://www.securitymagazine.com/

AUTHOR

Jesu Marcus Immanuvel Arockiasamy is a seasoned Healthcare Analytics and DevOps expert with over 18 years of experience at a leading healthcare company. His work focuses on leveraging DevOps principles to enhance system efficiencies, automate deployments, and manage CI/CD pipelines with prominent tools like Jenkins, Kubernetes, Terraform, and AWS. A devoted mentor, he has fostered a collaborative DevOps culture that promotes innovation and agility.

Arockiasamy has published several impactful whitepapers, including "Digital Healthcare Evolution: The Power of DevOps for Better Patient Engagement" and "Proactive Healthcare Analytics: Early Detection of Diabetes with SDOH Insights and Machine Learning." These works explore integrating advanced analytics and machine learning into healthcare solutions, aiming to enhance patient care and engagement. His academic and professional endeavors continue to drive transformative strategies in healthcare technology, ensuring secure, scalable, and patient-centric digital solutions.

APPENDIX

- GitHub repository with Working Code: Access the repository to explore the codebase and practical implementations discussed throughout this document.
- Culinary-Inspired Hackathon demo video of Phish & Chips Browser Plugin: Watch this demo video to see the creative presentation of the browser plugin functionality in action, inspired by culinary themes.
- All 30 Features Extracted from URL Validation: Review a comprehensive list of features used for URL validation, offering technical insights into our phishing detection criteria.

Datasets Used: PhiUSIIL Phishing URL SimilarWeb Top 500 Websites