

MACHINE LEARNING-ENHANCED SECURE PLATFORM FOR ELECTRONIC HEALTH RECORD SHARING WITH PROACTIVE THREAT DETECTION

Engr. Samina Parveen¹, Nabila Sehito^{*2}, Sajida Karim³, Haris Yaseen⁴, Jawad Akber⁵, Basit Ali⁶

^{1,2,5}Department of Computer Science Ilma university. (Formely IBT)

³Department of Computer Science, University of Mirpurkhas, Pakistan

⁴Department of Computer Science, Iqra University, Karachi.

⁶Collage of Management Sciences, Ziauddin University, Faculty of Engineering Science & Technology Management, Karachi Pakistan

^{*2}dr.nabila@ilmauniversity.edu.pk/nabila.fiza@gmail.com

DOI: <https://doi.org/10.5281/zenodo.23182281>

Keywords

Electronic health records; cybersecurity; machine learning; Random Forest; XGBoost; anomaly detection; access control; AES-GCM; HIPAA; secure file sharing.

Article History

Received: 25 June 2026

Accepted: 16 September 2026

Published: 06 October 2026

Copyright @Author

Corresponding Author: *

Nabila Sehito

Abstract

The need to achieve high availability, quick clinical access, and robust protection of electronically protected health information (ePHI) is critical to EHR exchange. However, these defenses, authentication, role-based authorization, encryption, and audit logging, are often reactive, as they fail to protect when a user with legitimate access to the network attempts to misuse it, or when a user enters a disguised document into a network that prevents even the best static signatures. This paper introduces Secure Health ML, a web-based prototype of an EHR sharing system that features fine-grained access control, authenticated encryption, secure transport, audit logging, and machine learning-based threat detection. The platform's application layer is based on Next.js/React; it is persisted in PostgreSQL; it includes an additional component for malware scanning; and it has an isolated Python microservice for supervised document-risk inference. Structural and metadata-derived features are used to evaluate the Random Forest (RF) and XGBoost (XGB) classifiers on a held-out dataset of PDF malware. Random Forest was chosen for deployment due to its highest test accuracy (99.25%), precision (99.02%), recall (99.64%), F1 score (99.33%), and ROC-AUC (0.9997). Security validation showed it successfully implemented policies such as authentication, RBAC/ABAC, encrypted storage, alert generation, and quarantine-driven response to high-risk uploads. The results suggest that an integrated solution combining policy-based access-control governance and model-based detection for insider and credential compromise can minimize the window of exposure caused by these activities. The prototype is constrained by the limited scope of the test environment and the simulated data set. Still, it offers a technically viable basis for designing the adaptive, security-aware sharing system for EHRs.

INTRODUCTION

EHRs have become an integral part of clinical care, continuity of care, and efficiency. Meanwhile, EHR systems bundle highly sensitive patient data, such as diagnoses, prescriptions, test results, and clinical histories, making them a favourite target for credential theft, ransomware, insider misuse, and document-borne malware. Confidentiality, integrity, and availability are all important factors that must be protected within regulatory and professional security frameworks without impeding legitimate clinical access (U.S. Department of Health and Human Services [HHS], 2024; National Institute of Standards and Technology [NIST], 2024).

Traditional secure file-sharing systems make extensive use of authentication, role-based access control, transport encryption, and storage encryption. Such controls are required but insufficient to address post-authentication threats. An abnormal velocity, a file accessed at an unanticipated time, a suspicious file structure, or a malicious document uploaded via an otherwise authorized process will not be recognized by a conventional system. This is significant in healthcare settings as it can turn a compromise into an unauthorized disclosure, service disruption, or clinical risk if it occurs too late in the process.

New EHR privacy and security research indicates that a good system needs more than just individual cryptographic safeguards. They should also include auditable access governance, context-aware access, staff accountability, and technical solutions that enable privacy without sacrificing interoperability and performance (Basil et al., 2022; Tertulino et al., 2024). At the same time, machine learning is gaining importance for malware and anomaly detection, as it can discover suspicious patterns that are unknown and cannot be captured by fixed signatures or static rules (Chen & Guestrin, 2016; Smutz & Stavrou, 2012).

Secure Health ML is a prototype system architecture for secure sharing of EHRs with proactive threat detection presented in this paper. This contribution is threefold. First is a layered EHR sharing architecture that integrates clinical authentication, RBAC/ABAC authorization, encrypted storage, secure transport, and audit logging, as well as an isolated ML threat engine. Secondly, it provides a checkered version of the Random Forest and XGBoost models for detecting PDF malware, along with the logic for selecting the model. Third, it provides results from implemented controls, such as functional validation, security testing, model performance, and system-level implications for adaptive EHR security.

1.1 Research Objectives

Table 1: Research objectives and corresponding evaluation focus

Objective	Operational meaning	Evaluation focus
Core secure platform development	Develop a browser-based EHR sharing environment with authentication, authorization, file management, and audit logging.	Functional testing of login, upload, download, dashboards, and audit events.
Dynamic access control	Combine role-based permissions with attribute-aware policies for contextual access decisions.	Policy validation across clinical roles, departments, decisions, and restricted documents.
Real-time ML-based threat detection	Integrate a lightweight ML microservice for document-risk inference and suspicious upload detection.	Random Forest/XGBoost comparison, confusion matrix, ROC-AUC, latency, and alert generation.

Integrated system assessment	Evaluate whether the combined platform preserves security while maintaining acceptable usability and performance.	Security tests, malware upload simulation, performance observations, and workflow integration.
------------------------------	---	--

2. LITERATURE REVIEW AND CONCEPTUAL FOUNDATION

2.1 EHR Security and Privacy Requirements

EHR security is a multifaceted issue, as health information must be accessible to authorized clinicians while being safeguarded against unauthorized disclosure, tampering, and misuse. The Security Rule is part of HIPAA and mandates that covered entities implement reasonable and appropriate administrative, physical, and technical safeguards to protect ePHI (HHS, 2024). From a systems-design point of view, these protections manifest as identity assurance, access control policies, encryption, logging, monitoring, incident response, and recovery.

Recent mapping and review studies have validated that the balance between privacy and interoperability is a limiting factor in the privacy of EHR systems. Tertulino et al. (2024) noted that access control, integrity, secure transmission, auditability, and consent remain prominent themes in EHR privacy studies. In a similar vein, Basil et al. (2022) identify EHR repositories as highly sensitive because they contain patient attributes, including demographics, diagnoses, treatment histories, lab results, and other information. These results show that this is an appropriate design approach and that every upload, download, policy action, and administrative action should be considered a security-relevant event.

2.2 Access Control, Zero Trust, and Clinical Context

Even in healthcare, where a doctor has to have access to the patients' records, a nurse has to have access to the patients' records, an administrator has to have access to the patients' records, an auditor has to have access to the patients' records, and the on-call clinicians have access to which is different still, RBAC is still useful because clinical

duties tend to be role-defined. However, RBAC is not enough if access to documents depends on contextual factors such as department, patient assignment, emergency status, document sensitivity, location, time, or risk score. ABAC is an extension of RBAC because it considers the subject, object, action, and environment parameters at decision time. In time-sensitive healthcare contexts, clinical conditions and situational context can be used to authorize access more precisely, as illustrated by research on context-aware EHR access control (Psarra et al., 2022).

The proposed platform is designed around the concept of zero trust: access is not automatically granted simply because a request comes from within an application boundary or from an authenticated user. Instead, every request is assessed based on identity, role, policy, session, and risk signals. NIST SP 800-207 defines zero trust as an architecture that does not assume trust and continually assesses access to resources against policies for each resource (Rose et al., 2020). This is particularly important for EHR sharing, where, after successful authentication, credential misuse or insider access can become an issue.

2.3 Cryptographic and Transport Safeguards

One of the core protections for protected health information (PHI) that is used for both storage and transmission is encryption. AES-GCM can be used for file protection because it provides confidentiality and integrity verification, as well as file protection (Dworkin, 2007). TLS 1.3 minimizes risks of interception and protocol downgrade in client-server communication for data in transit. Use HTTP-only cookies, secure cookie attributes, CSRF tokens, and tight server-side authentication to prevent cross-site scripting and cross-site request forgery. There is also access control and cryptographic correctness when

assessing an application's security. Some of the most serious web application security risks, as identified by OWASP, are broken access control and cryptographic failures (OWASP Foundation, 2021a; OWASP Foundation, 2021b). These risks are mitigated in this study by using role/attribute policy checks, encrypted storage, TLS, adaptive password hashing, audit logs, and session-level monitoring.

2.4 Machine Learning for Document-Centric Threat Detection

Document-centric EHR sharing calls for threat detection at upload time, since files may contain embedded scripts, exploit structures, obfuscated payloads, or abnormal metadata. PDF structure and metadata have been shown to provide useful signals for supervised classification (Smutz & Stavrou, 2012). For tabular metadata, ensemble tree methods are well-suited because they can capture non-linear interactions, are tolerant of differing feature importances, and yield good classification performance without requiring much feature scaling.

Random Forest is a collection of several random decision trees used to enhance robustness and reduce variance (Breiman, 2001). XGBoost uses regularized gradient-boosted decision trees and is widely used because of its scalability, additive optimization framework, and strong performance on structured data (Chen & Guestrin, 2016). The present study evaluates both methods in a lightweight Python microservice to determine whether the EHR sharing workflow can support high-confidence document-risk inference with low operational overhead.

2.5 Research Gap and Contribution

The literature establishes the importance of privacy, encryption, access control, and EHR auditability, yet many operational file-sharing systems remain weak after authentication. They may restrict who can access a file, but fail to analyze whether the request context, file structure, session behavior, or access velocity is abnormal. The gap addressed here is therefore not the absence of any single control; it is the limited integration of

policy-based authorization, encrypted document handling, auditability, and real-time ML-assisted threat detection within one coherent EHR sharing workflow.

Secure Health ML addresses this gap by placing the ML threat engine inside the upload and access-control pipeline rather than treating malware detection as a separate administrative task. The model output becomes an operational security signal that can trigger alerting, quarantine, or session freezing before malicious content spreads through the platform.

3. METHODOLOGY

3.1 Research Design

In this study, a design-science and prototype-evaluation approach is used. It is a secure system for sharing EHRs that features clinical authentication, policy-driven access control, encrypted file handling, audit logging, and ML-based document-risk detection. The evaluation determines whether the artifact performs as intended and is secure in a controlled, simulated environment.

Four complementary techniques were used in evaluating the prototype: functional testing of user-facing workflows, integration testing, which covered all user-facing components (frontend, backend, database, scanning, and machine learning), security testing, which consisted of simulated scenarios for unauthorized access and malicious uploads, and machine learning testing, which involved held-out test metrics for PDF malware classification.

3.2 System Architecture

The platform is multi-layered so that clinical users can connect to the platform via a browser-based portal and an authenticated API layer. The application is composed of three containers: PostgreSQL storage, a ClamAV scanning container, and a container with Python ML. Services are independent of one another, and by separating them you reduce the coupling and risk of failure for the main web app when using ML inference.

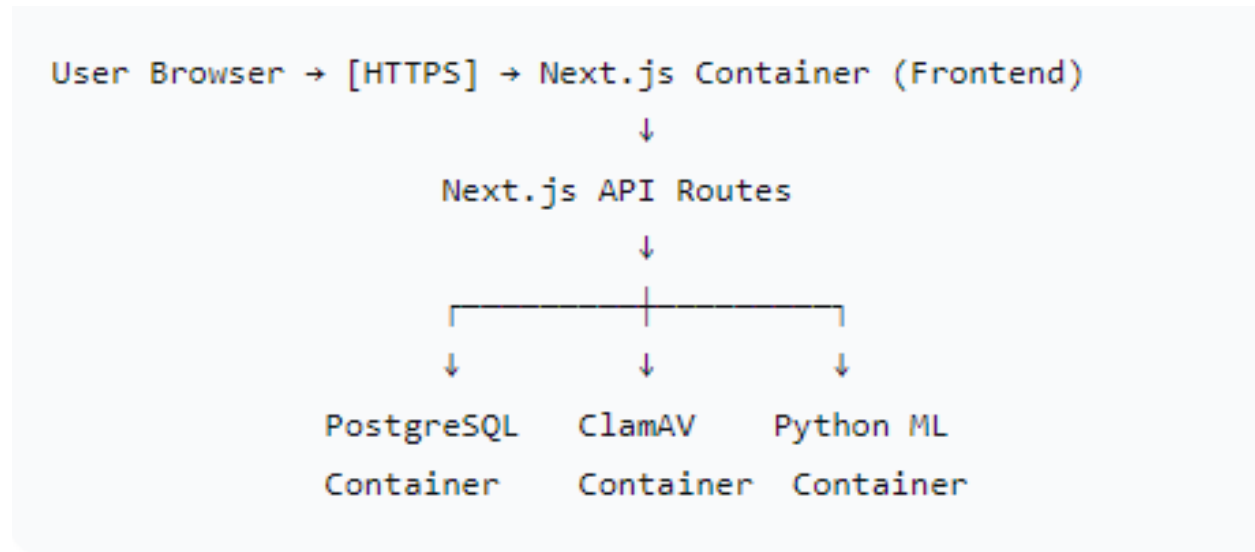


Figure 1: System communication flow among the frontend, API layer, database, scanning service, and Python ML microservice.

The broader clinical system design comprises an upload pipeline, metadata extraction, malware scanning, ML inference, alerting, encrypted storage, an admin dashboard, audit logging, and SSO, MFA, and RBAC policy enforcement. Figure 2 shows the overall system design, and Table 2 summarizes the main system components and their security role.

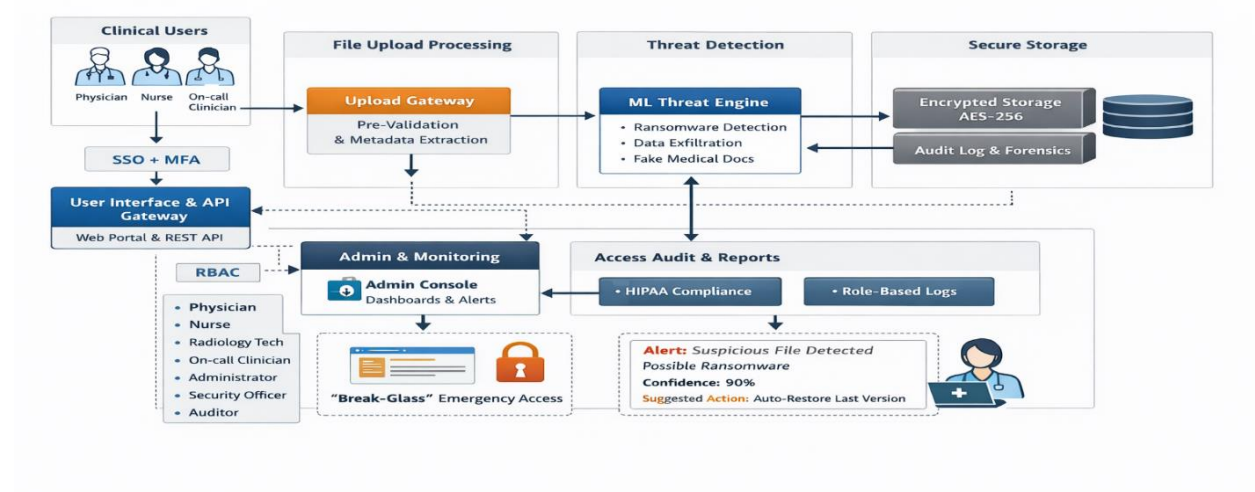


Figure 2: Layered architecture for secure EHR exchange and proactive threat detection.

Table 1: Architectural components and security rationale

Component	Primary function	Security rationale
SSO/MFA and session layer	Authenticates clinical users and issues protected session tokens.	Reduces account takeover risk and supports identity assurance.

RBAC/ABAC policy engine	Evaluates role, department, file sensitivity, action, and contextual attributes.	Limits inappropriate access after login and supports least privilege.
File management core	Controls upload, download, metadata capture, encryption, and retrieval.	Ensures that protected documents are governed throughout the lifecycle.
Scanning and ML threat engine	Analyzes file features, structure, and indicators of malicious activity.	Detects suspicious uploads before platform-wide propagation.
Audit and alert subsystem	Records access events, authentication events, malware detections, and policy decisions.	Supports accountability, incident triage, and evidence of compliance.
PostgreSQL and encrypted file storage	Stores users, policies, metadata, logs, and encrypted document references.	Separates metadata governance from encrypted content handling.

3.3 Secure Upload and Request-Processing Workflow

The site is designed to have a step-by-step process. The user proves his or her identity in the identity layer. The request is later processed by the ABAC policy engine. It is the role of the file management core to initiate uploading/retrieval, when

authorized. Upon upload, the file is checked for validity, and from the metadata extracted and scanned, the risk classification is returned by the ML microservice. This result may be either High-risk outcomes or Low-risk outcomes: Normal Workflow.

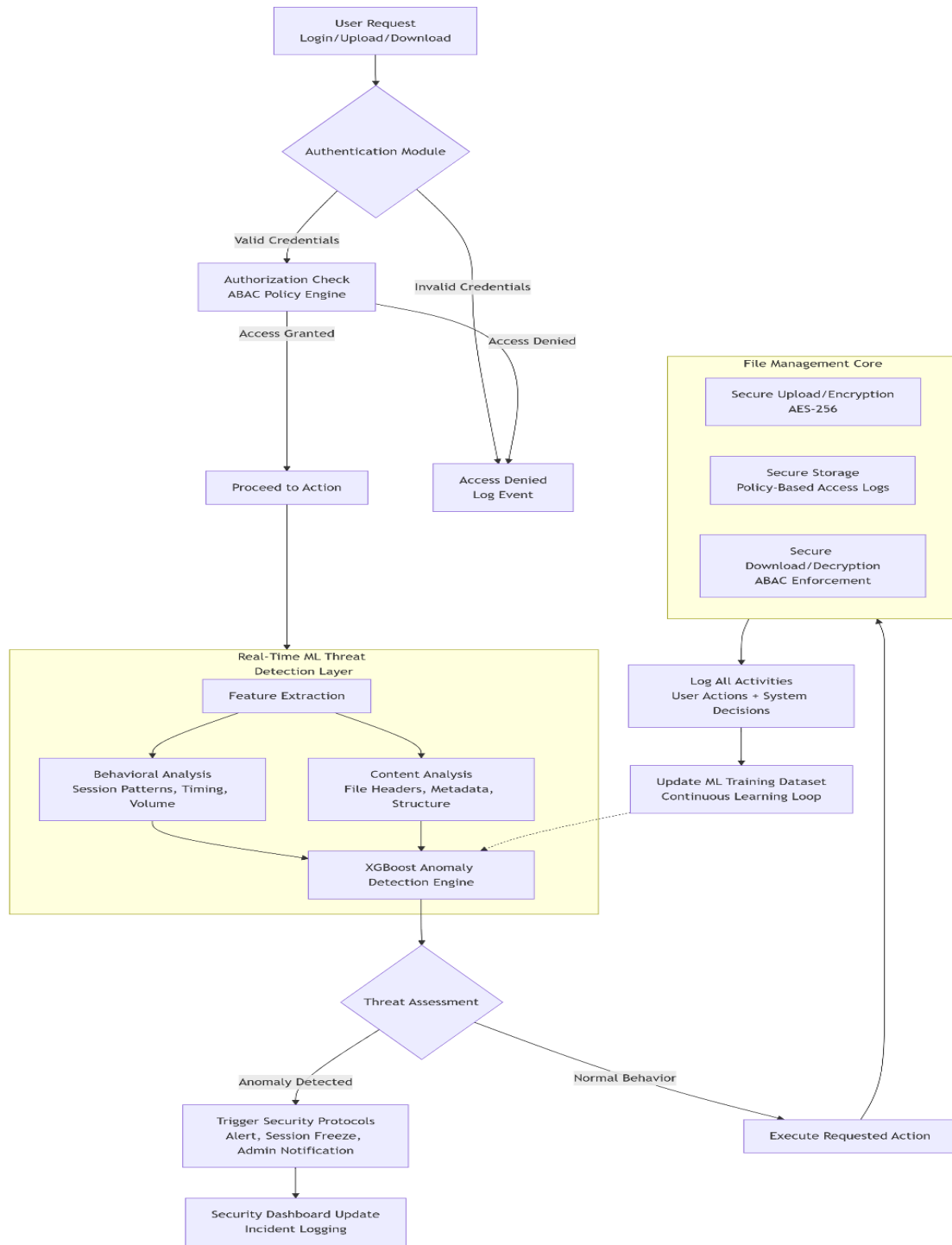


Figure 3: Request-processing workflow for authentication, authorization, encrypted file handling, ML inference, and alert response.

3.4 Data, Features, and Evaluation Setting

A supervised PDF Malware dataset (metadata and structural features) was used to assess the ML component. The total number of samples in the training snapshot is 10,023, of which 8,018 are benign, and 2,005 are malicious. There are 2005 samples in the held-out test set. The confusion matrix reported is: 883 true negatives, 11 false positives, 4 false negatives, and 1,107 true positives.

The features of the groups are: file metadata, suspicious object markers, structural indicators, JavaScript-related indicators, header consistency, entropy-related indicators, and upload context features. These file-level features can be augmented in the deployed EHR workflow by incorporating access frequency, user role, department, time of access, and session behaviour further to enhance the detection of insider misuse and anomalous activity.

Table 3: Feature groups used or supported by the threat-detection pipeline

Feature group	Examples	Security contribution
PDF metadata	File size, creator metadata, object counts, and embedded resource counts.	Identifies abnormal document construction and potentially automated payload generation.
PDF structural markers	Header anomalies, JavaScript markers, open-action indicators, and suspicious embedded objects.	Supports detection of exploit-bearing or obfuscated PDF files.
Upload context	Uploader role, department, destination audience, access reason, and file category.	Connects file risk to the clinical authorization context.
Behavioral context	Upload frequency, download volume, session duration, time-of-day, and repeated policy failures.	Supports detection of credential misuse, insider threat, and bulk exfiltration.
Operational response signals	Scanner decision, ML probability score, quarantine status, alert status.	Converts model output into an actionable security response.

3.5 Machine Learning Algorithms**3.5.1 Random Forest**

Random Forest is an ensemble technique that consists of many decision trees trained on random samples of data (bootstrap samples), combined by taking the maximum vote or the average probability. Each tree is trained to learn decision boundaries on document features, and the forest is trained to minimize variance by averaging over trees trained on documents with independently randomized features (Breiman, 2001).

Let $D = \{(x_i, y_i)\}_{i=1}^n$ denote the training set and $h_b(x)$ denote the prediction from the b -th tree. The malware-class probability is estimated as: $p(y = 1 | x) = (1/B) \sum_{b=1}^B h_b(x)$.

The final decision is $\hat{y} = 1[p(y = 1 | x) \geq \tau]$, where τ is the classification threshold. At each node t , the Gini impurity is $G(t) = 1 - \sum_{k=1}^K p_{k,t}^2$, and the split improvement is $\Delta G = G(t) - (|L|/|t|)G(L) - (|R|/|t|)G(R)$.

3.5.2 XGBoost

XGBoost is a regularized gradient-boosted tree model. Rather than training all trees independently, it trains them sequentially, using each tree's error as the residual error for training the next (Chen & Guestrin, 2016).

At iteration t , the additive prediction is $\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta f_t(x_i)$, where η is the learning rate. The objective is $L^{(t)} = \sum_i l(y_i, \hat{y}_i^{(t)}) + \Omega(f_t)$, where $\Omega(f) = \gamma T + (1/2)\lambda \sum_j w_j^2$ regularizes

model complexity. Using a second-order approximation, the optimal weight of leaf j is $w_j^* = -G_j / (H_j + \lambda)$, and the split gain is $\text{Gain} = 1/2[(G_L^2 / (H_L + \lambda)) + (G_R^2 / (H_R + \lambda)) - (G^2 / (H + \lambda))] - \gamma$.

3.6 Evaluation Metrics

The accuracy, precision, recall, F1 score, false-positive rate, false-negative rate, and ROC-AUC were used to assess model performance. The metrics are appropriate because detecting the malware is a balancing act between not alerting on false positives that could disrupt workflows and missing the malware.

- Accuracy = $(TP + TN) / (TP + TN + FP + FN)$.
- Precision = $TP / (TP + FP)$.
- Recall = $TP / (TP + FN)$.
- F1 score = $2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$.
- False-positive rate = $FP / (FP + TN)$.
- False-negative rate = $FN / (FN + TP)$.
- ROC-AUC measures discrimination across possible decision thresholds.

3.7 Security Implementation

Uploaded files are encrypted before being stored using AES-256-GCM, per-file salts, and key derivation. Client-server communication is via HTTPS/TLS and production deployments will be based on publicly trusted certificates. All non-public API routes check for authorization and protected session tokens are stored as HTTP-only cookies for authentication. The passwords are stored after adaptive hashing with a cost factor of 12. Audit logs are maintained for authentication events, access requests, access policies results,

upload results, malware detection and administrative action.

RESULTS

4.1 Platform Implementation Outcomes

The prototype successfully met the key workflows required for secure EHR sharing: Logon, file upload and download, access policy management, department management, audit logging, malware detection, and review of administrative alerts. The frontend offers role-specific access views, and sensitive operations need to be authenticated and/or have a policy applied before running them on the backend.

The ML service was deployed as a lightweight Python-based container that loads model artifacts when deployed and serves inference results to the primary application. This architecture supports maintainability, allowing you to update or replace the model without impacting the clinical web application, and enables controlled deployment.

The classification performance of the models is displayed in this section. This section shows the performance of models when used for classification.

4.2 Model Selection and Classification Performance

The random forest and XGBoost algorithms demonstrated very high discriminative power on the held-out test set. Random Forest was chosen as the deployed classifier because it performed best in terms of ROC-AUC and recall, with a low FP rate. The difference between these two models was not significant, but in a clinical security scenario, recall is particularly significant as missing malicious documents creates a direct exposure risk.

Table 4: Held-out test comparison of Random Forest and XGBoost

Model	Accuracy	Precision	Recall	F1 score	ROC-AUC
Random Forest	99.25%	99.02%	99.64%	99.33%	0.9997
XGBoost	99.20%	99.19%	99.37%	99.28%	0.9995

Note: Values were taken from the prototype model-comparison results. ROC-AUC is reported on a 0-1 scale; percentage metrics are reported as percentages.

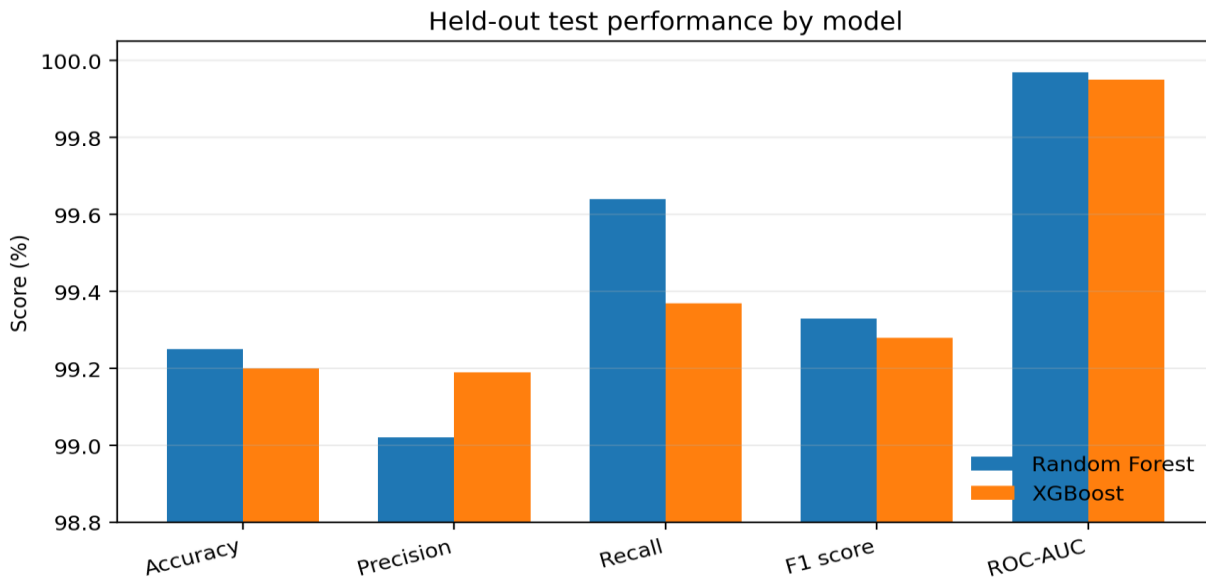


Figure 4: Comparative model performance across held-out test metrics

4.3 Confusion Matrix and Error Profile

The confusion matrix and error profile are provided for the 883 benign and 1,107 malicious PDF files. The deployed Random Forest model correctly classified 883 files. It gave 11 false positives and 4 false negatives. The false positive

and false negative rates are about 1.23% and 0.36%, respectively. False positives can increase reviewers' workload, while false negatives can be a more serious security concern, as they could allow malicious documents to bypass the first layer of security.

Table 5: Random Forest confusion matrix

	Predicted benign	Predicted malicious
Actual benign	883	11
Actual malicious	4	1,107

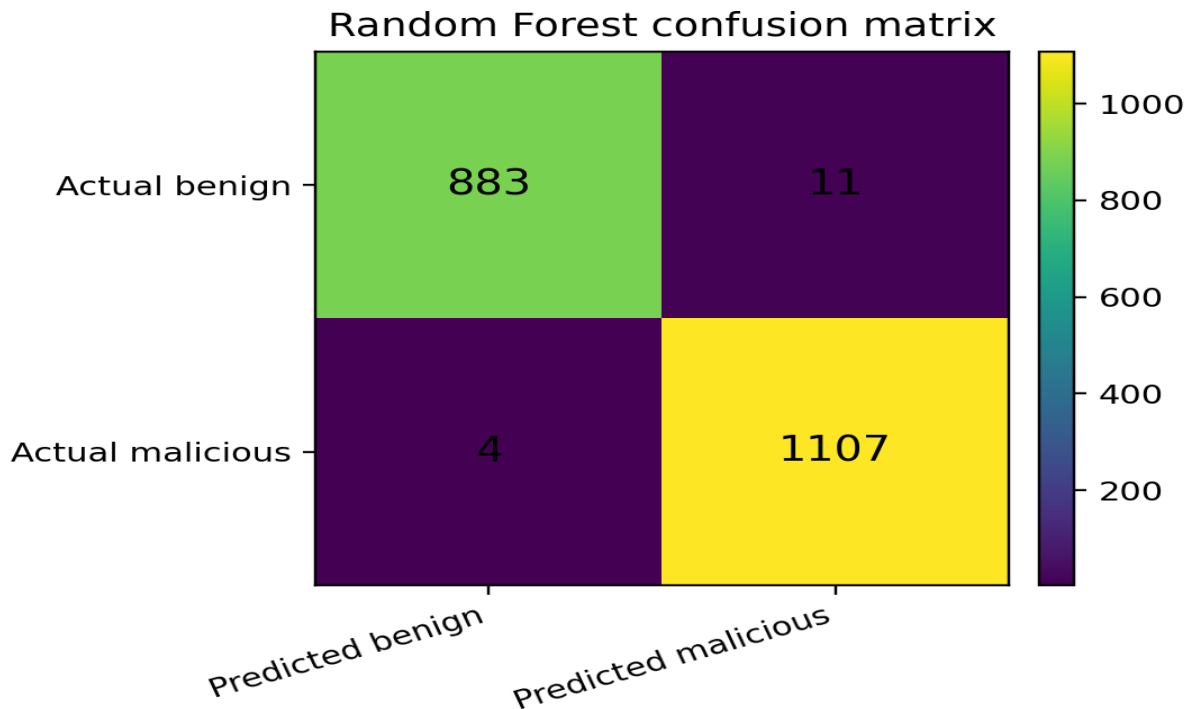


Figure 5: Visualized Random Forest confusion matrix

4.4 Functional, Integration, and Security Testing

While functional testing is crucial, it is not the only essential form of testing. Functional testing is important. All main modules have been functionally tested prior to integration. Following module-level validation, integration testing validated the flow from an authenticated user's action through policy evaluation, file handling,

ML review, result storage, and alert generation. Security testing employed simulated unauthorized login, malicious upload scenarios, and insider-risk scenarios. Unauthorized users were denied access, high-risk files were identified and quarantined, and suspicious activity was detected and alerted to the administrators.

Table 6: System testing summary

Test category	Scenario	Expected outcome	Observed outcome
Authentication	Valid and invalid login attempts.	Valid users authenticated; invalid users rejected and logged.	Authentication logs record successful and failed attempts.
Authorization	Role/attribute policy applied to restricted documents.	Only authorized users can access protected EHR documents.	RBAC/ABAC policy decisions were enforced before file access.
Malware upload	Upload of a suspicious PDF file.	File is blocked, quarantined, or flagged for review.	High-risk uploads generated detection records and alert entries.

Audit logging	User action and threat event review.	Events persist for administrative investigation.	Access, authentication, and malware events were visible in dashboards.
Integration	End-to-end upload through ML result and storage.	Consistent workflow from user action to policy response.	Frontend, backend, database, scanner, and ML service exchanged data successfully.

4.5 Performance and Operational Observations

The average latency for the ML inference service was about 200-400 milliseconds. The upload and scan process was not laggy in the controlled environment for files below 10 MB. A limited performance snapshot also showed positive loading behaviour on the browser side, but should not be seen as load-test evidence for production use.

An operational consequence is that, when tested under the described conditions, inserting ML-assisted scanning does not introduce any delay in the uploading process. For production deployment, however, stress testing, document sizes, network variability, file type, and realistic peak-load patterns with concurrent users would need to be considered.

5. Discussion

5.1 Interpretation of Findings

The results demonstrate the ability to integrate classical security measures with model-aided detection into a coherent EHR sharing workflow. The Random Forest classifier achieved the best results on the PDF structural features and metadata-derived features, which is consistent with previous research showing that PDF structure can help detect malicious PDFs (Smutz & Stavrou, 2012). However, the model achieved high recall and low false positives, making it suitable as the first-line risk-screening component in a prototype environment.

This also illustrates the need to combine access control and ML detection rather than treat them as two separate layers. RBAC and ABAC check processes are used to determine whether a user can perform an action, and the ML evaluation process is used to determine whether the file or behaviour

associated with the action is suspicious. The double decision-making is consistent with zero trust reasoning which means that it does not assume that the user is performing safe activities once he/she enters the system (Rose et al., 2020).

5.2 Theoretical and Practical Implications

Theoretical and Practical implications of this study are discussed. In theory, the study offers an integrated security model for sharing EHR that is integrated with identity management, policy enforcement, cryptographic protection, auditability and machine learning risk scoring. It's a variant of the standard secure file sharing experience with the addition of post-authentication intelligence.

In terms of implementation, the platform showcases the implementation of different protection measures in a common architecture. There is tracing/review of alerts and audit trails. The technical measures include token protection, access control, malware detection and encryption. Notice and quarantine, service separation contributes to operational resilience.

5.3 Limitations

- This evaluation was conducted in a controlled prototype environment, not in a live deployment.
- The malware dataset and behavioral scenarios were synthetic or simulated, so the results may not necessarily be fully applicable to real-world hospital traffic.
- This current ML task focuses on detecting malicious content in PDFs and is not comprehensive enough to cover all possible EHR file types or multimodal clinical documents.

- There was no user experience testing with the clinicians, and acceptance and alert fatigue remain unmeasured.
- While these are applicable to adversarial robustness, model drift, explainability, and continuous monitoring, they require further research before use in production.

5.4 Future Work

The evaluation of production should include concurrent loading tests, threat-emulation exercises, alert-fatigue analysis, and privacy-preserving learning methods. For a mature implementation, controls should also align with the NIST CSF 2.0 functions: Govern, Identify, Protect, Detect, Respond, and Recover (NIST, 2024).

6. CONCLUSION

The model must be tested on a larger external body of malware, such as adversarially altered documents, support for more formats (such as DOCX, PNG, JPEG, DICOM and compressed documents) and include explainable AI results for analyst's review. This research was designed and tested Secure Health ML, a machine-learning-assisted secure platform for sharing EHR. The platform features integrated, authenticated clinical access; RBAC/ABAC policy enforcement; encrypted storage and transport; audit logging; malware scanning; and supervised ML-based document-risk inference. Random Forest was chosen as the classifier due to its highest held-out accuracy (99.25%) and ROC-AUC (0.9997).

The main finding is that access control is the wrong solution for proactive EHR security. A secure sharing platform should be able to verify identities, consider access context, provide cryptographic protections for data, monitor file and behaviour signals, and map threat scores to operational actions. Despite the synthetic nature of the tested data and the controlled tests, the current study creates a realistic framework for EHR sharing systems that can adapt, be intelligent, and comply with auditing requirements.

REFERENCES:

- Basil, N. N., Ambe, S., Ekhatior, C., & Fonkem, E. (2022). Health records database and inherent security concerns: A review of the literature. *Cureus*, 14(10), e30168. <https://doi.org/10.7759/cureus.30168>.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45, 5-32. <https://doi.org/10.1023/A:1010933404324>.
- Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785-794). Association for Computing Machinery. <https://doi.org/10.1145/2939672.2939785>.
- Dworkin, M. (2007). Recommendation for block cipher modes of operation: Galois/Counter Mode (GCM) and GMAC (NIST Special Publication 800-38D). National Institute of Standards and Technology. <https://doi.org/10.6028/NIST.SP.800-38D>.
- Grassi, P. A., Fenton, J. L., Newton, E. M., Perlner, R. A., Regenscheid, A. R., Burr, W. E., Richer, J. P., Lefkowitz, N. B., Danker, J. M., Choong, Y.-Y., Greene, K. K., & Theofanos, M. F. (2017). Digital identity guidelines: Authentication and lifecycle management (NIST Special Publication 800-63B). National Institute of Standards and Technology. <https://doi.org/10.6028/NIST.SP.800-63b>.
- Guo, R., Shi, H., Zhao, Q., & Zheng, D. (2018). Secure attribute-based signature scheme with multiple authorities for blockchain in electronic health records systems. *IEEE Access*, 6, 11676-11686. <https://doi.org/10.1109/ACCESS.2018.2801266>.

- Keshta, I., & Odeh, A. (2021). Security and privacy of electronic health records: Concerns and challenges. *Egyptian Informatics Journal*, 22(2), 177-183. <https://doi.org/10.1016/j.eij.2020.07.003>.
- National Institute of Standards and Technology. (2024). The NIST Cybersecurity Framework (CSF) 2.0 (NIST Cybersecurity White Paper 29). <https://doi.org/10.6028/NIST.CSWP.29>.
- OWASP Foundation. (2021a). A01:2021 - Broken access control. OWASP Top 10. https://owasp.org/Top10/A01_2021-Broken_Access_Control/.
- OWASP Foundation. (2021b). A02:2021 - Cryptographic failures. OWASP Top 10. https://owasp.org/Top10/A02_2021-Cryptographic_Failures/.
- Psarra, E., Stamatelatos, M., Sakkopoulos, E., & Tsakalidis, A. (2022). Context-based, predictive access control to electronic health records. *Electronics*, 11(19), 3040. <https://doi.org/10.3390/electronics11193040>.
- Rescorla, E. (2018). The Transport Layer Security (TLS) Protocol Version 1.3 (RFC 8446). Internet Engineering Task Force. <https://doi.org/10.17487/RFC8446>.
- Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). Zero Trust Architecture (NIST Special Publication 800-207). National Institute of Standards and Technology. <https://doi.org/10.6028/NIST.SP.800-207>.
- Shahnaz, A., Qamar, U., & Khalid, A. (2019). Using blockchain for electronic health records. *IEEE Access*, 7, 147782-147795. <https://doi.org/10.1109/ACCESS.2019.2946373>.
- Smutz, C., & Stavrou, A. (2012). Malicious PDF detection using metadata and structural features. In *Proceedings of the 28th Annual Computer Security Applications Conference* (pp. 239-248). Association for Computing Machinery. <https://doi.org/10.1145/2420950.2420987>.
- Tertulino, R., Antunes, N., & Morais, H. (2024). Privacy in electronic health records: A systematic mapping study. *Journal of Public Health*, 32, 435-454. <https://doi.org/10.1007/s10389-022-01795-z>.
- U.S. Department of Health and Human Services. (2024). Summary of the HIPAA Security Rule. <https://www.hhs.gov/hipaa/for-professionals/security/laws-regulations/index.html>.