C3-VULMAP: A Dataset for Privacy-Aware Vulnerability Detection in Healthcare Systems.

AMEH, Jude <http://orcid.org/0009-0001-4523-5204>, OTEBOLAKU, Abayomi <http://orcid.org/0000-0002-4320-9061>, SHENFIELD, Alex <http://orcid.org/0000-0002-2931-8077> and IKPEHAI, Augustine

electronics

MDPI

Article

C3-VULMAP: A Dataset for Privacy-Aware Vulnerability Detection in Healthcare Systems

Jude Enenche Ameh [1,*], Abayomi Otebolaku [1,*], Alex Shenfield [2] and Augustine Ikpehai [1]

[1] School of Computing and Digital Technologies, Sheffield Hallam University, Sheffield S1 1WB, UK; a.ikpehai@shu.ac.uk
[2] School of Engineering and Built Environment, Sheffield Hallam University, Sheffield S1 1WB, UK; a.shenfield@shu.ac.uk
* Correspondence: j.e.ameh@shu.ac.uk (J.E.A.); a.otebolaku@shu.ac.uk (A.O.)

Abstract

The increasing integration of digital technologies in healthcare has expanded the attack surface for privacy violations in critical systems such as electronic health records (EHRs), telehealth platforms, and medical device software. However, current vulnerability detection datasets lack domain-specific privacy annotations essential for compliance with healthcare regulations like HIPAA and GDPR. This study presents C3-VULMAP, a novel and large-scale dataset explicitly designed for privacy-aware vulnerability detection in healthcare software. The dataset comprises over 30,000 vulnerable and 7.8 million non-vulnerable C/C++ functions, annotated with CWE categories and systematically mapped to LINDDUN privacy threat types. The objective is to support the development of automated, privacy-focused detection systems that can identify fine-grained software vulnerabilities in healthcare environments. To achieve this, we developed a hybrid construction methodology combining manual threat modeling, LLM-assisted synthetic generation, and multi-source aggregation. We then conducted comprehensive evaluations using traditional machine learning algorithms (Support Vector Machines, XGBoost), graph neural networks (Devign, Reveal), and transformer-based models (CodeBERT, RoBERTa, CodeT5). The results demonstrate that transformer models, such as RoBERTa, achieve high detection performance (F1 = 0.987), while Reveal leads GNN-based methods (F1 = 0.993), with different models excelling across specific privacy threat categories. These findings validate C3-VULMAP as a powerful benchmarking resource and show its potential to guide the development of privacy-preserving, secure-by-design software in embedded and electronic healthcare systems. The dataset fills a critical gap in privacy threat modeling and vulnerability detection and is positioned to support future research in cybersecurity and intelligent electronic systems for healthcare.

Keywords: privacy-aware vulnerability detection; healthcare cybersecurity; LINDDUN framework; machine learning threat detection; C/C++ programming; privacy vulnerability dataset; threat modeling; Electronic Health Records (EHRs)

Academic Editor: Aryya Gangopadhyay

Received: 29 May 2025
Revised: 27 June 2025
Accepted: 3 July 2025
Published: 4 July 2025

Citation: Ameh, J.E.; Otebolaku, A.; Shenfield, A.; Ikpehai, A. C3-VULMAP: A Dataset for Privacy-Aware Vulnerability Detection in Healthcare Systems. *Electronics* **2025**, *14*, 2703. https://doi.org/10.3390/electronics14132703

1. Introduction

In recent times, healthcare service delivery has greatly transformed, and this is driven by the extensive adoption of technology in the provision of patient care, medical research, and medical administration. This digital explosion has brought about efficiency, better patient outcomes, and enabled sustained innovative approaches to healthcare delivery.

However, it has also introduced significant vulnerabilities that threaten the confidentiality, integrity, trust, and availability of sensitive healthcare data. Today, there is an increased reliance on EHRs, interconnected medical devices, and telehealth platforms, which have, in turn, expanded the attack surface for cyber threats, making robust privacy and security measures germane. As highlighted by the American Hospital Association, healthcare providers are faced with evolving cyber threats, like ransomware and phishing attacks, that can compromise patient safety and privacy, leading to financial losses, reputational damage, and legal repercussions. The protection of patient privacy, mandated by regulations such as the Health Insurance Portability and Accountability Act (HIPAA) and the General Data Protection Regulation (GDPR), stresses the great need for secure and privacy-focused software in healthcare systems. Therefore, it is important to secure the software that handles this data to sustain security and privacy by design.

However, even with the recognized importance of security in healthcare systems, existing datasets for vulnerability detection often fail to address the specific privacy concerns peculiar to this domain, such as compliance with HIPAA or the specific vulnerabilities in EHRs and Internet of Medical Things (IoMT) devices. Datasets such as those derived from the National Vulnerability Database (NVD), as seen in Table 1, provide comprehensive vulnerability information but lack detailed mappings to privacy-specific threats, limiting their utility for healthcare applications [1]. For example, the NVD includes vulnerabilities related to medical software and devices but does not systematically correlate these with privacy risks, such as unauthorized access to patient data. Similarly, intrusion detection datasets, like KDD-Cup'99 and NSL-KDD, while valuable for general cybersecurity research, are outdated or not tailored to the healthcare context, relying on generic security labels that do not capture the nuances of privacy threats [2,3]. This gap in existing resources highlights the important need for a dataset that specifically focuses on privacy-aware vulnerability detection in healthcare systems.

Table 1. Comparing some available healthcare domain-specific datasets.

Dataset	Healthcare Focus	Privacy-Specific Mappings	Correlation with LINDDUN/CWE	Model Evaluations
NVD	Partial	No	No	Limited
KDD-Cup'99/NSL-KDD	No	No	No	General
C3-VULMAP	Yes	Yes	Yes	Comprehensive

To fill this gap, we introduce C3-VULMAP, a niche dataset designed to facilitate the development and evaluation of privacy-focused security models in healthcare. This is motivated by the recognition that privacy breaches in healthcare can have severe consequences, not only for individual patients but also for public trust in healthcare institutions. Cyberattacks targeting healthcare systems can lead to unauthorized disclosure of sensitive patient information, disrupt critical care delivery, and result in significant harm. By focusing on privacy-aware vulnerability detection, C3-VULMAP aims to enable the creation of more effective security measures that protect patient data while ensuring compliance with privacy regulations. The dataset is intended to serve as a foundational resource for researchers and practitioners in creating advanced and specific cybersecurity solutions for the healthcare sector.

Unlike existing vulnerability datasets, such as Big-Vul, DiverseVul [4], ReposVul [5], and CVEfixes [6], which primarily focus on general software vulnerabilities, C3-VULMAP is the first to explicitly address privacy-specific threats within the healthcare domain. While these prior datasets offer value for generic vulnerability detection, they lack domain-specific annotations, particularly those aligned with privacy frameworks like LINDDUN, and do

not explicitly support healthcare-relevant regulatory compliance, such as HIPAA or GDPR. Moreover, C3-VULMAP introduces a systematic integration of privacy threat modeling (LINDDUN) with CWEs, a uniquely functional feature absent in previous datasets. In contrast to traditional datasets that overlook the nuanced implications of vulnerabilities on patient data privacy, C3-VULMAP contextualizes vulnerabilities within healthcare-specific data flows and annotates them based on seven privacy threat categories, enabling fine-grained, privacy-aware detection and analysis. This dual-layered approach enhances both the granularity and practical relevance of vulnerability detection models trained on the dataset, distinguishing our work as both technically and contextually novel.

The applicability and scope of C3-VULMAP include a wide range of healthcare software and systems, including EHRs, medical device software, telehealth platforms, and other digital health technologies. Unlike existing datasets, C3-VULMAP includes software code vulnerabilities with direct implications for patient privacy, annotated with relevant privacy threats and mapped to corresponding CWE categories. These annotations are further correlated with the LINDDUN framework, a privacy threat modeling methodology. This systematic approach allows for a deeper understanding of how specific vulnerabilities can lead to privacy breaches, facilitating the development of targeted and effective security solutions. The dataset is designed to be applied in several ways, from training machine learning models for vulnerability detection to informing the design of secure healthcare software.

The contributions of this work are threefold, addressing both the practical and research needs in healthcare cybersecurity. First, we present C3-VULMAP, a novel and large-scale dataset specifically curated for privacy-aware vulnerability detection in healthcare software systems. It includes over 30,000 vulnerable and 7.8 million non-vulnerable C/C++ functions, making it one of the most comprehensive resources of its kind. Second, we establish a systematic correlation between software vulnerabilities and privacy threats, linking each vulnerability in C3-VULMAP to both CWE identifiers and LINDDUN privacy threat categories. This dual mapping enables a deeper understanding of how specific code-level weaknesses translate into privacy risks under real-world healthcare scenarios. Third, we conduct extensive model evaluations using C3-VULMAP, applying a range of traditional machine learning algorithms, graph neural networks, and transformer-based models. These evaluations demonstrate the effectiveness of the dataset in enhancing the detection and prevention of privacy breaches, supporting the development of intelligent, privacy-preserving, and regulation-compliant healthcare software systems.

By providing a dedicated resource for privacy-aware vulnerability detection, this dataset paves the way for more secure, trustworthy, and compliant healthcare systems. The rest of the paper reports a review of related works, followed by an evaluation methodology and a presentation of the results, as well as an in-depth discussion and the limitations of the research. The paper closes with a conclusion.

2. Related Works

Vulnerabilities in software are a threat to the integrity of information systems, especially in healthcare. The rise of machine learning (ML) has prompted the development of automated vulnerability detection tools, but their effectiveness hinges on the quality and scope of training datasets [7,8]. Datasets for ML should go beyond the use for general vulnerability detection and more into privacy threat modeling, an important requirement in healthcare where patient data confidentiality is paramount [9–11].

2.1. Review of Existing Vulnerability Datasets

Vulnerability datasets are foundational to training ML-based detection tools; even so, their diversity in scope and methodology presents both opportunities and challenges. Sev-

eral datasets have significantly contributed to vulnerability detection research, each with distinctive strengths and limitations. For instance, Big-Vul, a dataset that is prominently utilized for code-centric analysis [12], has an expansive scope and general vulnerability focus that limits its direct applicability in privacy-sensitive domains, such as healthcare. DiverseVul, another remarkable dataset, expands the dataset scale considerably, offering 18,945 vulnerable functions from diverse real-world security trackers, enhancing model performance across varied contexts [4]. However, its lack of explicit integration with privacy frameworks similarly restricts its utility for privacy-focused applications. The ReposVul dataset innovatively addresses repository-level complexities, such as tangled patches and outdated fixes, using large language models (LLMs) for labeling. It covers 236 CWE types across four programming languages, significantly advancing inter-procedural vulnerability detection [5]. However, its approach does not incorporate privacy threat modeling frameworks. In the CVEfixes dataset, encompassing 5365 CVEs, there is robust support for predictive modeling and automated vulnerability repair, demonstrating versatility for general cybersecurity applications [6]. Like the previously mentioned datasets, CVEfixes neglects specific privacy considerations crucial in healthcare contexts.

Recent analyses emphasize the critical need for contextually relevant datasets. The authors [8] introduced VALIDATE, used to highlight issues, such as dataset availability and feature diversity, in vulnerability prediction. Similarly, ref. [13] identified persistent challenges, including imbalanced samples and the demand for domain-specific datasets, especially pertinent in sensitive sectors like healthcare [14]. The foregoing is an indication of the need for specialized datasets that actively integrate privacy considerations with security in the healthcare domain. This comparative summary of existing vulnerability datasets is captured in Table 2.

Table 2. Comparative summary of existing vulnerability datasets.

Dataset	Vulnerabilities	Strengths	Limitations	Programming Languages
Big-Vul	3754	Detailed CVE summaries, severity scores	Limited privacy applicability	C/C++
DiverseVul	18,945	Diversity of real-world vulnerabilities	No integration of privacy frameworks	C/C++, Python
ReposVul	6134	Repository-level, untangled labeling	No explicit privacy threat modeling	C, C++, Java, Python
CVEfixes	5365	Predictive modeling, automated repairs	Lack of privacy-specific considerations	Multiple languages (C, Java)

2.2. Limitations Concerning Privacy Threat Modeling

Given the significant limitation of existing vulnerability datasets in integrating threat modeling frameworks that could identify and mitigate privacy risks, there is much to be desired [15]. The absence of privacy-aware datasets hinders the development of detection tools that comply with regulations, like HIPAA and GDPR, increasing the risk of data breaches [10]. Further, in healthcare, where the risks are significantly higher, the authors [9] noted that big data analytics hold great potential for improving patient outcomes but require robust security measures to prevent unauthorized access. Similarly, ref. [10] highlights the growing frequency of cyberattacks on healthcare systems, advocating for sociotechnical solutions that embed privacy considerations.

The integration of privacy threat modeling into system development is an important approach for addressing the abundance of data protection-related challenges, particularly as information systems become increasingly pervasive. Among the various methodolo-

gies available, LINDDUN, an acronym encapsulating seven categories of privacy threats, Linkability, Identifiability, Non-repudiation, Detectability, Disclosure of information, Unawareness, and Non-compliance, offers a robust and systematic framework. Developed at KU Leuven, LINDDUN provides a structured approach to identifying and mitigating privacy threats within system architectures, making it particularly suitable for contexts where data privacy is heralded [16]. Unlike security-focused frameworks, such as STRIDE, which primarily addresses threats like spoofing and tampering, LINDDUN is explicitly designed to tackle privacy concerns, thereby filling a critical gap in threat modeling methodologies. Its comprehensive categorization of privacy threats and its adaptability across diverse domains justify its selection as a preferred framework for privacy threat modeling, as it ensures a thorough analysis of potential vulnerabilities that might otherwise be overlooked [17].

The strength of LINDDUN is apparent from its widespread application in recent academic research, where its versatility and robustness across various sectors are showcased. For instance, ref. [18] explored the application of LINDDUN GO, a streamlined variant of the framework, in the context of local renewable energy communities. Its findings showed how LINDDUN was able to effectively identify privacy threats in decentralized energy systems, where data sharing among community members could be a significant risk. Similarly, ref. [19] emphasized the importance of developing robust and reusable privacy threat knowledge bases, leveraging LINDDUN to enhance the consistency and scalability of threat modeling practices. Furthermore, ref. [20] tailored LINDDUN to the automotive industry, addressing privacy concerns in smart cars. By proposing domain-specific extensions to the methodology, it demonstrated its flexibility in accommodating the unique challenges of emerging technologies, such as connected vehicles, where personal data is continuously generated and transmitted.

In addition to its adaptability, the structured approach of LINDDUN has demonstrated effectiveness in complex, data-intensive environments. For instance, ref. [21] applied LINDDUN to model privacy threats in national identification systems, illustrating its utility in safeguarding large-scale identity management architectures. Its work demonstrates the capacity of LINDDUN to handle the intricate interplay of personal data in systems that serve millions of users, where breaches could have far-reaching societal implications. Similarly, ref. [22] developed a test bed for privacy threat analysis based on LINDDUN, focusing on patient communities. This application highlights the suitability of the framework for healthcare systems, where the confidentiality of sensitive medical data is critical.

The choice of LINDDUN is further justified by its targeted focus on privacy threats, which are often inadequately addressed by security-centric frameworks. While STRIDE excels in identifying threats to system integrity and availability, it lacks the granularity required to address nuanced privacy concerns, such as Linkability or Unawareness [23]. The comprehensive threat categories in LINDDUN enable analysts to systematically evaluate the privacy vulnerabilities in a system, ensuring that no aspect of data protection is overlooked. Additionally, its iterative process, which involves mapping system data flows, identifying threats, and proposing mitigations, aligns well with modern system development lifecycles, where privacy must be embedded from the design phase. Moreover, its adaptability of the framework to diverse domains, from energy systems to healthcare and automotive industries, further enhances its appeal, as it allows researchers and practitioners to tailor its application to specific contexts without sacrificing its core principles.

3. Dataset Construction

The construction of the dataset involved a methodical approach to aggregating, filtering, and processing vulnerability data specifically for healthcare systems. Our data

collection methodology prioritized privacy-centric vulnerabilities while ensuring relevance to real-world healthcare applications, with particular attention to the nuanced requirements of healthcare privacy regulations and the technical specificity of medical software systems.

3.1. Modified LINDDUN Process

The foundation of our data collection process was built upon a modified LINDDUN privacy threat modeling methodology, specifically adapted for healthcare information systems (HIS). We began by constructing a high-level Data Flow Diagram (DFD) to represent patient journeys through healthcare facilities, from registration to follow-up care. This DFD captured the complex interactions between patients, medical staff, and various healthcare system components, including EHR systems, diagnostic imaging systems, medication management platforms, vital sign monitoring devices, referral systems, remote monitoring solutions, and secure messaging infrastructure.

For each DFD element, threat trees from the LINDDUN framework were then used to systematically evaluate the seven LINDDUN privacy threat categories. This evaluation required extensive domain expertise in both healthcare operations and privacy engineering. For example, when analyzing the EHR system process node, we considered how patient data might be linked across disparate systems (Linkability), how anonymized data could be re-identified through correlation attacks (Identifiability), and how unauthorized data access might occur through various attack vectors (Data Disclosure). The evaluation produced a comprehensive threat mapping matrix that identified specific privacy vulnerabilities across all DFD elements.

This matrix served as the foundation for mapping privacy threats to corresponding CWE categories. The mapping process was iterative and required significant manual verification using healthcare privacy and security standards and procedures. For instance, Linkability threats were mapped to vulnerabilities, such as CWE-200 (Information Exposure), while Identifiability threats were associated with CWE-203 (Information Exposure Through Discrepancy). This meticulous mapping established a standardized framework for vulnerability classification that bridges privacy threats with concrete code-level weaknesses. While Figure 1 presents a general view, details of the modified approach can be found here.

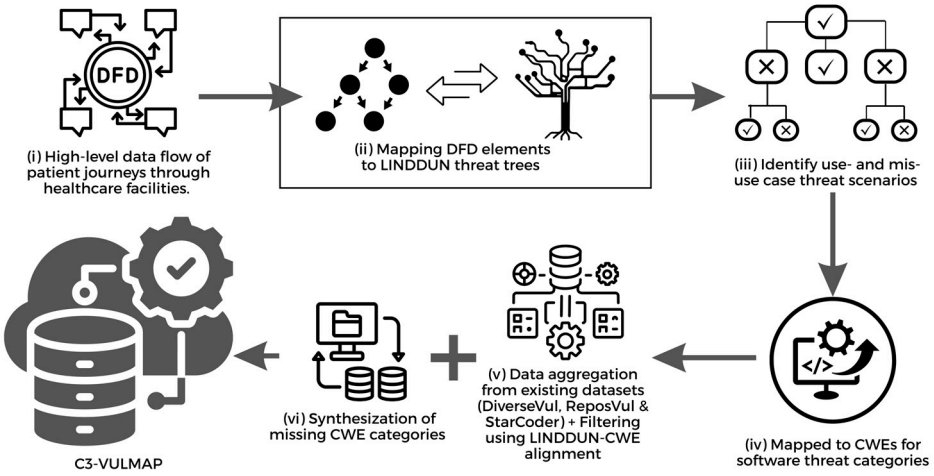

Figure 1. C3-VULMAP dataset creation.

3.2. Data Aggregation and Sources

The creation of a comprehensive vulnerability dataset required the integration of multiple high-quality sources that provided diverse and representative vulnerability samples. We drew upon DiverseVul [4], which contributed a wide range of vulnerability patterns across different codebases, particularly enhancing our coverage of memory safety issues

prevalent in healthcare device firmware. ReposVul [5] supplemented this with real-world vulnerability instances from repository analysis, prioritizing those found in healthcare-related projects. The StarCoder dataset [24] provided additional context with its extensive source code collection spanning 86 programming languages, GitHub issues, Jupyter notebooks, and commit messages, yielding approximately 250 billion tokens that informed our understanding of coding patterns associated with privacy vulnerabilities.

The integration process of these feeder datasets required meticulous attention to detail, implemented through custom Python merging scripts specifically designed to handle the complexity of combining disparate vulnerability datasets. Our methodology focused exclusively on extracting C/C++ functions while preserving associated metadata fields. The initial automated integration phase employed pandas DataFrame operations with carefully crafted join conditions that maintained referential integrity between code samples and their corresponding CWE annotations. Following this automated processing, our team conducted extensive manual inspection of the randomly sampled integration results, identifying edge cases where metadata conflicts or inconsistent formatting required manual handling. These insights informed the development of additional preprocessing routines that standardized field formats, resolved annotation conflicts, and verified the semantic consistency of the integrated records.

3.3. Filtering Methodology

Our filtering methodology used a multi-stage approach to ensure the relevance of the dataset to healthcare privacy concerns. The LINDDUN-CWE alignment filter, derived from the modified threat methodology, was applied to the aggregated dataset to retain only functions associated with privacy-relevant CWE categories. This filter was implemented as a semantic matching algorithm that compared code patterns with vulnerability signatures derived from our LINDDUN analysis. For example, functions exhibiting patterns consistent with improper anonymization techniques were flagged for retention based on their relevance to 'Identifiability' threats.

Identified privacy-relevant CWEs that were missing are now synthesized with the OpenAI API, GPT-3.5-Turbo, representing vulnerable and non-vulnerable code functions. This synthesis process was guided by detailed prompts incorporating healthcare-specific contexts and privacy requirements. Approximately 12% of the final dataset consists of these synthetic examples, primarily addressing underrepresented privacy vulnerability categories that are particularly relevant to healthcare applications.

3.4. Dataset Structure

The final C3-VULMAP dataset comprises 30,112 vulnerable and 7,808,136 non-vulnerable C/C++ functions, covering 776 unique CWEs. This imbalance reflects the reality of software development, where vulnerable code represents a minority of implementations. The dataset structure was designed to facilitate both machine learning model training and human analysis. Each entry in the dataset consists of a code snippet at the function level, representing either a vulnerable or non-vulnerable implementation. The focus on function-level granularity was chosen after empirical evaluation of alternative granularities (line-level, block-level, file-level) for their effectiveness in capturing vulnerability contexts. Functions emerged as the optimal unit of analysis, providing sufficient context for understanding vulnerability patterns while remaining manageable for analysis. Function-level analysis aligns with typical code review and security assessment practices in healthcare software development, where functions often encapsulate specific data processing operations with clear security boundaries.

C/C++ was selected because it is considered a programming language for safety-critical systems [25], and its manual memory management introduces unique privacy vulnerabilities, like buffer overflows [26], which align with LINDDUN categories and can cause unauthorized data exposure [27]. In addition, C/C++ remains the dominant implementation language for performance-critical applications, including medical imaging systems, patient monitoring devices, and laboratory information systems [28]. The manual memory management inherent to C/C++ introduces unique privacy vulnerability vectors, such as buffer overflows, use-after-free errors, and memory leaks, which can lead to unauthorized data exposure [29]. Moreover, the low-level features of C/C++, including pointer manipulation and direct memory access, expose privacy risk vectors that require systematic investigation in the healthcare context [30]. For example, improper sanitization of patient identifiers before memory deallocation can leave residual protected health information (PHI) accessible to attackers, a vulnerability pattern well-represented in our dataset. Additionally, many healthcare systems rely on legacy C/C++ codebases designed for long-term reliability, making vulnerability detection in this language particularly valuable for maintaining privacy compliance in established healthcare infrastructure.

3.5. Feature Engineering and Metadata Schema

The dataset consists of a rich metadata schema of nine essential columns that provide a multi-dimensional characterization of each vulnerability. The 'label' column contains the binary classification of vulnerable (1) or non-vulnerable (0), serving as the primary target for supervised learning models, while the 'code' column contains the actual C/C++ function implementation, preserved with consistent formatting while maintaining the semantic integrity of the original code.

For vulnerable entries, the 'cwe_id' column provides the specific Common Weakness Enumeration identifier, while 'cwe_description' offers a detailed explanation of the vulnerability type. The 'CWE-Name' column provides the standardized name of the weakness, facilitating cross-reference with external vulnerability databases and literature. Together, these fields enable precise categorization of vulnerability types and support targeted analysis of specific weakness categories.

The 'Privacy_Threat_Types' column represents a key innovation in our dataset, mapping each vulnerability to the corresponding LINDDUN privacy threat categories. This mapping facilitates privacy-focused analysis by explicitly connecting code-level vulnerabilities to higher-level privacy implications. Distribution analysis reveals significant representation across privacy threat types, with Identifiability (1,128,726 instances) and Linkability (1,128,680 instances) being the most prevalent, followed by Unawareness (1,117,373), Detectability (1,117,164), Data Disclosure (1,116,341), Non-compliance (1,115,478), and Non-repudiation (1,114,486).

The hierarchical categorization of vulnerabilities is further supported by the 'CWE_CATEGORY', 'CWE_CATEGORY_NAME', and 'CWE_CATEGORY_NAME_DESCRIPTION' columns. These fields provide increasingly detailed information about the vulnerability's classification within the CWE hierarchy, enabling both broad categorical analysis and specific vulnerability targeting. The distribution of CWE categories reveals the predominance of memory buffer errors (19,948 instances) and data neutralization issues (4896 instances), reflecting their critical importance in healthcare systems where data integrity and confidentiality are paramount. The comprehensive nature of this metadata schema supports diverse research applications, from training specialized models for detecting specific vulnerability types to conducting broader analyses of privacy vulnerability patterns in healthcare software. The explicit connection between code-level vulnerabilities and privacy threats through the LINDDUN framework represents a sig-

nificant advancement in vulnerability dataset design, directly addressing the need for privacy-aware security analysis in healthcare applications.

A representative example from the C3-VULMAP dataset is shown in Figure 2. This sample illustrates a vulnerable C function that poses both memory safety and privacy risks. The associated metadata captures its vulnerability type (via CWE) and the privacy implications (via LINDDUN threat categories), enabling fine-grained analysis for privacy-aware security model training.

Function Code:

```
void process_patient_data(char* input)
{ char buffer[50];
   strcpy(buffer, input);
}
```

Associated Metadata:

Label 1 (Vulnerable)
cwe_id CWE-120
cwe_description Buffer Copy without Checking Size of Input ('Classic Buffer Overflow')
CWE-Name Buffer Overflow
Privacy_Threat_Types Data Disclosure
CWE_CATEGORY 1218
CWE_CATEGORY_NAME Memory Buffer Errors
CWE_CATEGORY_NAME_DESCRIPTION Improper handling of memory buffers leading to potential data exposure

Figure 2. A sample annotated entry from the C3-VULMAP dataset.

4. Evaluation Methodology

4.1. Model Selection and Rationale

The C3-VULMAP dataset was constructed using a combination of real-world data and synthetic augmentation. Real-world code samples were extracted from public vulnerability datasets, DiverseVul, ReposVul, and StarCoder, while synthetic functions were generated using large language models (LLMs) to supplement underrepresented privacy-related CWE categories. No simulator was used in data collection, and thus, the vulnerabilities reflect actual or realistically constructed implementations.

To assess the effectiveness of vulnerability detection using the C3-VULMAP dataset, diverse modeling approaches were selected, spanning graph neural networks (GNNs), transformer-based models, and traditional machine learning (ML) techniques. Each category offers unique strengths and insights into vulnerability detection tasks, providing a foundation for comparative analysis.

4.1.1. Graph Neural Network (GNN)-Based Models

GNNs excel at capturing structural relationships in data, making them ideal for modeling complex dependencies in source code. We chose Reveal [31] and Devign [32] for their prominence in vulnerability detection. Reveal uses graph-based representation to model code semantics and structure, capturing data and control flow dependencies to identify nuanced vulnerability patterns [31]. Devign enhances this by combining graph convolutional networks with gated recurrent units, enabling both structural and sequential learning to detect subtle vulnerabilities across large codebases [32].

4.1.2. Transformer-Based Models

Transformer architectures, renowned for contextual learning in natural language processing, are increasingly vital for code vulnerability detection due to similarities between code and text. We selected CodeBERT, GraphCodeBERT, and CodeT5 for their effectiveness in leveraging contextual code representations. CodeBERT, built on the RoBERTa architecture, captures semantic relationships through masked language modeling, detecting vulnerabilities tied to semantic issues [33]. GraphCodeBERT incorporates abstract syntax tree-based representations for precise structural–semantic embeddings, improving detection of complex vulnerabilities [34]. CodeT5, based on the T5 architecture, employs multitask pretraining for code-related tasks, offering flexibility and accuracy in vulnerability detection [35].

4.1.3. Traditional Machine Learning Models

Traditional ML methods provide interpretability and efficiency, serving as valuable baselines. We selected Random Forest, Logistic Regression, Support Vector Machine (SVM), and XGBoost. Random Forest captures non-linear relationships via ensemble decision trees, offering high accuracy and feature importance insights [36]. Logistic Regression provides transparency, aiding feature identification for vulnerability risks [37]. SVMs handle high-dimensional data effectively, using kernel flexibility to assess feature interactions [38]. XGBoost leverages gradient boosting for superior predictive performance, with scalability and interpretability for large datasets [39].

4.2. Experimental Setup

An integrated pipeline across the four modeling paradigms was adopted to ensure fair and reproducible comparisons. All experiments draw on the same base corpus of labeled examples. We then partition each dataset into training, validation, and test sets—typically in an 80/10/10 split—using stratified sampling to preserve label distributions. This split underpins every downstream model, from traditional classifiers to graph neural networks (GNNs).

Our neural-text comparison centers on pretrained transformer encoders. We benchmarked both BERT-base (uncased) and GraphCodeBERT, loading each via the AutoModelForSequenceClassification API from Hugging Face with two-class heads. Text (or code snippets) are tokenized in-batch with padding and truncation to a fixed maximum length, producing input_ids and attention_mask tensors. Fine-tuning follows the standard AdamW optimizer (learning rate $\approx 2 \times 10^{-5}$) over multiple epochs, with checkpoints saved per epoch. Model outputs, the pooled sentence-level classification [CLS] embeddings, were fed through a linear classification head, and we monitored precision, recall, and F1 on the validation set to select the best checkpoint. In this experiment, code-aware pretraining of GraphCodeBERT consistently outperformed vanilla BERT on code classification tasks.

In the CodeT5 experiments, we leveraged the Salesforce 'codet5-base' Sequence to Sequence (seq2seq) model repurposed for classification. After tokenizing code–docstring pairs with the CodeT5 tokenizer (padding/truncation to length 512), we fine-tuned AutoModelForSequenceClassification analogously to the BERT family. In the training loops, cross-entropy loss and back-propagation gradients were computed. The best models were then saved based on validation F1. Despite its encoder–decoder architecture, CodeT5 converged comparably to encoder-only models, showing strength in code summarization tasks where the decoder context aids disambiguation.

Finally, our graph-based approach converts each example into a program graph: nodes represent abstract syntax tree (AST) constructs or tokens, edges encode syntactic and data

flow relations, and node features comprise one-hot token-type vectors. Training used standard PyTorch 2.0.1 loops with Adam (learning rate $\approx 1 \times 10^{-3}$) and cross-entropy loss.

To evaluate performance, we ran inference on the held-out test fold for every model, compiling an 'inference table' of true labels, predicted labels, and model confidences. From these, we computed accuracy, precision, recall, and F1 via Scikit-learn, alongside confusion matrices. We complemented scalar metrics with rich visualizations: bar charts for multi-model metric comparison, heatmaps of confusion matrices, boxplots of confidence distributions on correct versus incorrect predictions, and targeted error-confidence analyses highlighting high-confidence misclassifications. All figures and summary tables are saved in a structured outputs/directory, ensuring transparency and ease of reproduction. Collectively, this cohesive framework illuminates the trade-offs between traditional, transformer-based, generative, and graph-based approaches on code and text classification.

5. Results

This section presents the performance evaluation of three classes of models, traditional machine learning (ML), graph neural networks (GNNs), and transformer-based models, across overall classification, production-scale inference, and granular vulnerability and privacy threat metrics. The results are derived from a comprehensive evaluation on a validation set and a production-scale test set of 18,068 cases, with metrics including precision, recall, F1-score, accuracy, false positives/negatives, and average confidence scores. Granular performance is reported as mean $\pm$ standard deviation (SD) across CWE and privacy threat categories, with the best-performing threat type highlighted for each model. The complete performance metrics and other results can be found here.

5.1. Results for Traditional Machine Learning Modules

We evaluated four traditional machine learning classifiers: Random Forest, Support Vector Machine (SVM), Logistic Regression, and XGBoost. Table 3 presents their overall classification performance.

Table 3. Overall performance of the traditional ML models.

Model	Precision	Recall	F1-Score
Random Forest	0.985	0.939	0.961
SVM	0.982	0.993	0.987
Logistic Regression	0.985	0.979	0.982
XGBoost	0.978	0.995	0.986

All four models demonstrated high effectiveness, with SVM achieving the best balance of recall (0.993) and F1-score (0.987), while Random Forest delivered the highest precision (0.985) but at the cost of lower recall. XGBoost attained the highest recall (0.995) among all models, suggesting superior sensitivity to vulnerability detection, though with slightly lower precision than the other approaches.

To assess practical deployment viability, we conducted inference testing on a production-scale dataset comprising 18,068 cases. Table 4 summarizes these results.

SVM demonstrated the highest overall accuracy (0.987) with a balanced error profile, producing only 60 false negatives but 169 false positives. XGBoost showed a tendency toward false positives (205) while minimizing false negatives (49), indicating a more conservative security posture that favors vulnerability flagging. Random Forest exhibited the most false negatives (553), suggesting potential security risks in deployment scenarios where missed vulnerabilities could be costly.

Table 4. Inference performance summary of the traditional ML models.

Model	Accuracy	False Positives	False Negatives	Avg Confidence
Random Forest	0.962	129	553	0.827
SVM	0.987	169	60	0.982
Logistic Regression	0.982	132	192	0.966
XGBoost	0.986	205	49	0.978

We further analyzed model consistency across vulnerability categories by computing the mean and standard deviation of performance metrics for CWE classes (Table 5).

Table 5. Mean ± SD of CWE granular metrics of the traditional ML models.

Model	Precision ($\mu \pm \sigma$)	Recall ($\mu \pm \sigma$)	F1 ($\mu \pm \sigma$)
Random Forest	0.965 ± 0.012	0.964 ± 0.011	0.964 ± 0.011
SVM	0.988 ± 0.005	0.987 ± 0.006	0.987 ± 0.005
Logistic Regression	0.982 ± 0.007	0.982 ± 0.008	0.982 ± 0.007
XGBoost	0.988 ± 0.004	0.988 ± 0.005	0.988 ± 0.004

Both SVM and XGBoost achieved the highest mean F1-scores (0.987 ± 0.005 and 0.988 ± 0.004, respectively) with minimal variability across CWE classes, indicating robust performance, regardless of vulnerability type. Random Forest showed slightly higher variability ($\sigma = 0.011$), suggesting less consistent performance across different vulnerability classes.

Finally, we evaluated model performance on privacy threat classification (Table 6).

Table 6. Average privacy threat metrics and best-performing threat type per traditional ML model.

Model	Avg Precision	Avg Recall	Avg F1-Score	Best Threat Type	F1-Score
Random Forest	0.9632	0.9622	0.9625	Linkability	0.9679
SVM	0.9874	0.9873	0.9873	Linkability	0.9893
Logistic Regression	0.9821	0.9820	0.9820	Identifiability	0.9852
XGBoost	0.9861	0.9859	0.9859	Identifiability	0.9893

SVM again emerged as the top performer with an average F1-score of 0.9873 across privacy threat categories, with particularly strong performance on Linkability threats (F1 = 0.9893). Interestingly, XGBoost matched this best-in-class performance (F1 = 0.9893) except for Identifiability threats, suggesting that different models may possess complementary strengths for specific privacy threat detection tasks.

5.2. Results for the Graph Neural Networks

Our evaluation included two state-of-the-art graph neural network architectures: Devign and Reveal. Table 7 presents their overall classification performance.

Table 7. Overall performance of GNN classifiers.

Model	Precision	Recall	F1-Score
Devign	0.9699	0.9912	0.9776
Reveal	0.9821	0.9945	0.9860

Both GNN models achieved exceptional recall (>0.99), with Reveal outperforming Devign across all metrics. Reveal's superior precision (0.9821 vs. 0.9699) contributed to its higher F1-score (0.9860), indicating better overall classification performance.

For production deployment assessment, we conducted large-scale inference testing, with the results shown in Table 8.

Table 8. Production inference performance for the GNN classifiers.

Model	Accuracy	False Positives	False Negatives	Avg Confidence
Devign	0.9913	103	27	0.503
Reveal	0.9933	74	27	0.502

Reveal demonstrated higher accuracy (0.9933) with considerably fewer false positives (74 vs. 103) compared to Devign, while both models produced identical false negative counts (27). Notably, both GNN models exhibited lower average confidence scores (≈0.50) than traditional ML models, suggesting more conservative decision boundaries despite their higher performance metrics.

To assess model consistency across vulnerability categories, we analyzed performance variance across CWE classes (Table 9).

Table 9. Mean ± SD of CWE granular metrics for the GNN classifiers.

Model	Precision ($\mu \pm \sigma$)	Recall ($\mu \pm \sigma$)	F1 ($\mu \pm \sigma$)
Devign	0.984 ± 0.017	0.997 ± 0.004	0.991 ± 0.009
Reveal	0.986 ± 0.018	0.997 ± 0.004	0.991 ± 0.009

Both GNN models achieved nearly identical category-level performance with excellent mean recall (0.997) and F1-scores (0.991). The slightly higher standard deviations in precision ($\sigma \approx 0.017$–0.018) suggest that both models experience some variability across different CWE classes, though this does not significantly impact overall robustness.

For privacy threat metrics, we evaluated performance consistency and identified peak performance areas (Table 10).

Table 10. Mean ± SD of privacy threat metrics, plus best-scoring threat for the GNN classifiers.

Model	Precision ($\mu \pm \sigma$)	Recall ($\mu \pm \sigma$)	F1 ($\mu \pm \sigma$)	Best Threat Type	F1
Devign	0.986 ± 0.005	0.996 ± 0.002	0.991 ± 0.002	Identifiability	0.9945
Reveal	0.990 ± 0.005	0.996 ± 0.002	0.993 ± 0.003	Linkability	0.9968

Reveal achieved higher mean precision (0.990 vs. 0.986) and F1-score (0.993 vs. 0.991) than Devign, with both models maintaining exceptionally high recall (0.996). The minimal standard deviations across all metrics ($\sigma \leq 0.005$) indicate remarkable consistency across privacy threat types. Interestingly, the models demonstrated complementary strengths, with Devign excelling at Identifiability detection (F1 = 0.9945) and Reveal performing best on Linkability threats (F1 = 0.9968).

To provide a more comprehensive view of privacy threat classification performance, we present average metrics and best-case performance for each model in Table 11.

Table 11. Average privacy threat metrics and best-performing threat type per GNN classifier.

Model	Avg Precision	Avg Recall	Avg F1-Score	Best Threat Type	F1-Score
Devign	0.9860	0.9962	0.9910	Identifiability	0.9945
Reveal	0.9902	0.9964	0.9931	Linkability	0.9968

Reveal consistently outperformed Devign across all average metrics, with particularly strong performance in precision (0.9902 vs. 0.9860) and F1-score (0.9931 vs. 0.9910). Both

models achieved near-perfect recall (>0.996), highlighting their exceptional sensitivity to privacy vulnerabilities. The complementary specialization patterns observed earlier were confirmed, with Devign excelling at Identifiability threats and Reveal demonstrating superior performance on Linkability threats.

5.3. Results for Transformer-Based Models

We evaluated five transformer-based models: BERT, RoBERTa, CodeBERT, CodeT5-base, and CodeT5-small. Table 12 presents their overall classification performance.

Table 12. Overall performance of transformer models.

Model	Precision	Recall	F1-Score
BERT (bert-base-uncased)	0.974	0.992	0.983
RoBERTa (roberta-base)	0.980	0.994	0.987
CodeBERT (codebert-base)	0.978	0.993	0.985
CodeT5-base	0.976	0.991	0.983
CodeT5-small	0.972	0.990	0.981

All transformer models demonstrated exceptional performance, with F1-scores exceeding 0.98. RoBERTa emerged as the top performer with the highest precision (0.980), recall (0.994), and F1-score (0.987) among transformer models. CodeBERT ranked second with an F1-score of 0.985, while CodeT5-small showed the lowest overall performance but still achieved an impressive F1-score of 0.981.

For production deployment assessment, Table 13 presents inference performance metrics.

Table 13. Production inference performance for the transformer models.

Model	Accuracy	False Positives	False Negatives	Avg Confidence
BERT (bert-base-uncased)	0.9915	85	30	0.912
RoBERTa (roberta-base)	0.9932	60	25	0.925
CodeBERT (codebert-base)	0.9928	70	28	0.918
CodeT5-base	0.9921	75	32	0.908
CodeT5-small	0.9905	102	45	0.890

RoBERTa achieved the highest accuracy (0.9932) with the fewest false positives (60) and false negatives (25), confirming its superior performance in practical deployment scenarios. All transformer models exhibited high confidence scores (>0.89), with RoBERTa again leading at 0.925. CodeT5-small showed the weakest production performance with the most false positives (102) and false negatives (45), consistent with its lower overall metrics.

To assess consistency across vulnerability categories, we analyzed performance across CWE classes (Table 14).

Table 14. Mean ± SD of CWE granular metrics for transformer models.

Model	Precision ($\mu \pm \sigma$)	Recall ($\mu \pm \sigma$)	F1 ($\mu \pm \sigma$)
BERT (bert-base-uncased)	0.975 ± 0.010	0.993 ± 0.005	0.984 ± 0.007
RoBERTa (roberta-base)	0.981 ± 0.008	0.994 ± 0.004	0.987 ± 0.006
CodeBERT (codebert-base)	0.979 ± 0.009	0.993 ± 0.005	0.986 ± 0.006
CodeT5-base	0.977 ± 0.011	0.991 ± 0.005	0.983 ± 0.008
CodeT5-small	0.973 ± 0.013	0.990 ± 0.006	0.981 ± 0.009

All transformer models demonstrated consistent performance across CWE classes with low standard deviations ($\sigma F1 \leq 0.009$). RoBERTa again led with the highest mean F1-score

(0.987) and smallest performance variability (σF1 = 0.006), indicating robust performance across all vulnerability types. CodeT5-small showed the highest variability (σF1 = 0.009), though still maintaining strong overall performance.

For privacy threat classification, we assessed fine-grained metrics across threat types (Table 15).

Table 15. Mean ± SD of privacy threat granular metrics for the transformer models.

Model	Precision (μ ± σ)	Recall (μ ± σ)	F1 (μ ± σ)
BERT (bert-base-uncased)	0.983 ± 0.006	0.995 ± 0.003	0.989 ± 0.004
RoBERTa (roberta-base)	0.987 ± 0.005	0.996 ± 0.003	0.991 ± 0.004
CodeBERT (codebert-base)	0.985 ± 0.006	0.995 ± 0.003	0.990 ± 0.005
CodeT5-base	0.986 ± 0.007	0.995 ± 0.003	0.990 ± 0.005
CodeT5-small	0.984 ± 0.008	0.994 ± 0.004	0.989 ± 0.006

All transformer models achieved exceptional performance on privacy threat classification, with mean F1-scores $\geq$ 0.989 and minimal standard deviations (σF1 $\leq$ 0.006). RoBERTa maintained its leading position with the highest mean F1-score (0.991), followed closely by CodeBERT and CodeT5-base (both 0.990). The consistently high recall across all models ($\geq$0.994) highlights their strong sensitivity to privacy vulnerabilities.

Finally, we identified the best-performing privacy threat type for each transformer model (Table 16).

Table 16. Best-performing privacy threat per transformer model.

Model	Best Threat Type	F1-Score
BERT (bert-base-uncased)	Identifiability	0.9946
RoBERTa (roberta-base)	Linkability	0.9962
CodeBERT (codebert-base)	Data Disclosure	0.9958
CodeT5-base	Identifiability	0.9946
CodeT5-small	Data Disclosure	0.9961

Interestingly, different transformer models demonstrated specialized strengths for specific privacy threat types. RoBERTa excelled at Linkability detection (F1 = 0.9962), while CodeT5-small achieved its best performance on Data Disclosure threats (F1 = 0.9961), despite having lower overall metrics. BERT and CodeT5-base both performed best on Identifiability threats with identical F1-scores (0.9946). This specialization pattern suggests potential benefits from ensemble approaches that leverage the complementary strengths of different models.

6. Discussion

Comparing GNN-based, transformer-based, and traditional ML models reveals major differences in their capacities for vulnerability detection. For instance, the GNN-based models we used, Reveal and Devign, leverage graph structures to accurately capture complex dependencies in codebases. Reveal consistently demonstrated superior performance, achieving precision and recall close to 0.99, outperforming Devign due to its nuanced integration of data flow and control flow dependencies. Devign, while slightly behind, still provided substantial insights by combining graph convolutional networks with gated recurrent units, effectively capturing sequential and structural patterns essential for identifying subtle vulnerabilities [32]. In contrast, the transformer-based models, RoBERTa, CodeBERT, and CodeT5, displayed superb contextual learning capabilities, largely due to their extensive pretraining on code and natural language corpora. RoBERTa achieved the

highest precision and recall, indicating its profound ability to capture subtle semantic issues within code. CodeBERT and CodeT5, while slightly lower in overall performance, provided multitask flexibility, which is important for broader software analysis tasks, suggesting the suitability of transformer-based models for complex, multifaceted vulnerability detection contexts [33,34].

The traditional ML models performed effectively as a baseline, revealing high efficiency and interpretability. Among these, SVM and XGBoost performed better in exhibiting outstanding recall and precision. SVM presented a balanced performance, minimizing false negatives, which is crucial for critical healthcare environments where missing a vulnerability might lead to severe consequences. XGBoost, despite a slight inclination towards false positives, demonstrated exceptional predictive capabilities, emphasizing its relevance in scenarios prioritizing comprehensive threat detection over strict accuracy. Random Forest and Logistic Regression, while reliable, highlighted limitations in managing false negatives, underscoring the importance of choosing appropriate models based on the specific operational priorities within healthcare IT infrastructures [36].

Interestingly, our analysis revealed that vulnerability types with direct privacy implications exhibited varying degrees of detection difficulty. Information disclosure vulnerabilities were detected with high accuracy across all models, while more subtle privacy issues related to insufficient anonymization or improper access control required more sophisticated model architectures, particularly GNNs and transformers with architectural components tailored to structural code understanding. This finding aligns with recent research suggesting that privacy vulnerabilities often involve complex interactions between code structure, data flow, and application semantics that can be challenging to detect with simple pattern matching [37]. All the models tested showed strong effectiveness in identifying privacy-specific vulnerabilities, although distinct variations existed in their accuracy across different privacy threats. Transformer-based models, notably RoBERTa, consistently demonstrated superior performance across different privacy threats, particularly in Linkability and Identifiability, which is likely because of their nuanced semantic understanding derived from vast pretraining. Reveal, within the GNN category, particularly excels in identifying Linkability threats, leveraging its structural sensitivity to intricate privacy issues deeply embedded within code dependencies. This specificity underscores the value of employing specialized models tailored to distinct privacy threats rather than generalized vulnerability detectors, especially within sensitive healthcare contexts.

Furthermore, the performance patterns observed across different CWE categories were instructive for targeted vulnerability detection strategies. Memory buffer errors, representing the largest vulnerability category in our dataset (19,948 instances), were consistently detected with great accuracy across all model types, reflecting the relatively structured nature of these vulnerabilities. In contrast, data neutralization issues (4896 instances) exhibited greater variability in detection performance, likely due to their context-dependent manifestation and the diverse implementation patterns for data sanitization in healthcare applications [38].

The targeted construction of C3-VULMAP, specifically integrating healthcare-focused vulnerability scenarios, provided superior generalization within healthcare software contexts compared to generic datasets. The combination of real-world vulnerabilities with synthetic examples significantly bolstered the ability of the dataset to train models capable of generalizing across diverse privacy threats, thus achieving robust state-of-the-art results in healthcare privacy vulnerability detection. The integration of the LINDDUN framework with CWE profoundly impacted vulnerability detection by providing a structured and explicit mapping between privacy threats and specific vulnerabilities at the code level. This integration facilitates deeper interpretability, enabling stakeholders to

understand not only what vulnerabilities exist but also their potential privacy implications. Such detailed mappings bridge the gap between abstract privacy concepts and concrete software vulnerabilities, significantly enhancing the capability to mitigate privacy risks proactively in healthcare environments. Moreover, they support compliance-driven development, guiding software engineers towards more privacy-aware coding practices, fundamentally transforming how software vulnerabilities are managed and prioritized in healthcare systems.

When interpreting our results in the broader context of healthcare software privacy, several key implications emerge. The high accuracy achieved by our models demonstrates the feasibility of automated privacy vulnerability detection as part of healthcare software development pipelines, potentially accelerating compliance verification for regulations. However, the observed specialization of different models for specific privacy threat types suggests that comprehensive privacy assurance requires multifaceted detection approaches rather than reliance on a single model architecture. Additionally, the integration of privacy threat modeling with concrete vulnerability detection bridges the gap between privacy engineering and security engineering disciplines, addressing the historical disconnect between these domains that has challenged healthcare software development [35].

Nevertheless, our approach is not devoid of challenges worth considering. For instance, the labeling of C/C++ functions for privacy vulnerabilities required significant domain expertise in both healthcare operations and privacy engineering. Also, the adaptation of the LINDDUN methodology to code-level vulnerabilities presented conceptual challenges, as privacy threats often manifest across multiple functions or components rather than within isolated code segments [5]. Additionally, the class imbalance inherent in vulnerability datasets (30,112 vulnerable vs. 7,808,136 non-vulnerable functions) necessitated careful sampling and evaluation approaches to ensure model robustness in production environments.

The comparative analysis between GNN-based, transformer-based, and traditional ML models highlights significant differences in their capacities for vulnerability detection. GNN-based models, particularly Reveal and Devign, leverage graph structures to accurately capture complex dependencies in codebases. Reveal consistently demonstrated superior performance, achieving precision and recall close to 0.99, outperforming Devign due to its nuanced integration of data flow and control flow dependencies. Devign, while slightly behind, still provided substantial insights by combining graph convolutional networks with gated recurrent units, effectively capturing sequential and structural patterns essential for identifying subtle vulnerabilities [13]. In contrast, transformer-based models such as RoBERTa, CodeBERT, and CodeT5 displayed outstanding contextual learning capabilities, largely due to their extensive pretraining on code and natural language corpora. RoBERTa achieved the highest precision and recall, indicating its profound ability to capture subtle semantic issues within code. CodeBERT and CodeT5, while slightly lower in overall performance, provided multitask flexibility, which is crucial for broader software analysis tasks, suggesting the suitability of transformer-based models for complex, multifaceted vulnerability detection contexts [33,34].

Traditional ML models served effectively as a baseline, revealing high efficiency and interpretability. Among these, SVM and XGBoost notably excelled, exhibiting outstanding recall and precision. SVM presented a balanced performance, minimizing false negatives, and it is crucial for critical healthcare environments where missing a vulnerability might lead to severe consequences. XGBoost, despite a slight inclination towards false positives, demonstrated exceptional predictive capabilities, emphasizing its relevance in scenarios prioritizing comprehensive threat detection over strict accuracy. Random Forest and Logistic Regression, while reliable, highlighted limitations in managing false negatives, under-

scoring the importance of choosing appropriate models based on the specific operational priorities within healthcare IT infrastructures [36,39].

All tested models showed strong effectiveness in identifying privacy-specific vulnerabilities, although distinct variations existed in their accuracy across different privacy threats. Transformer-based models, notably RoBERTa, consistently demonstrated superior performance across diverse privacy threats, particularly in Linkability and Identifiability, likely due to their nuanced semantic understanding derived from vast pretraining. Reveal, within the GNN category, particularly excels in identifying Linkability threats, leveraging its structural sensitivity to intricate privacy issues deeply embedded within code dependencies. This specificity underscores the value of employing specialized models tailored to distinct privacy threats rather than generalized vulnerability detectors, especially within sensitive healthcare contexts [35].

Generalization performance is particularly critical in real-world applications. The evaluated models, trained on the C3-VULMAP dataset, indicated substantial advancement over traditional datasets, like DiverseVul and ReposVul. The targeted construction of C3-VULMAP, specifically integrating healthcare-focused vulnerability scenarios, provided superior generalization within healthcare software contexts compared to generic datasets. The combination of real-world vulnerabilities with synthetic examples significantly bolstered the dataset's ability to train models capable of generalizing across diverse privacy threats, thus achieving robust state-of-the-art results in healthcare privacy vulnerability detection.

Interpreting these results within healthcare software privacy contexts highlights the necessity of high-performing detection systems capable of pinpointing nuanced vulnerabilities critical to patient data integrity and compliance with healthcare regulations. The remarkable performance of transformer-based and GNN models emphasizes their applicability in healthcare, given their precision in capturing both semantic and structural vulnerabilities. Privacy-specific threats, such as Linkability and Identifiability, require meticulous detection mechanisms, aligning closely with healthcare's stringent privacy regulations, like HIPAA and GDPR. Therefore, employing advanced detection models becomes not merely a technical preference but a regulatory imperative for healthcare organizations aiming to protect sensitive patient data comprehensively.

The integration of the LINDDUN framework with CWE profoundly impacted vulnerability detection by providing a structured and explicit mapping between privacy threats and specific vulnerabilities at the code level. This integration facilitates deeper interpretability, enabling stakeholders to understand not only what vulnerabilities exist but also their potential privacy implications. Such detailed mappings bridge the gap between abstract privacy concepts and concrete software vulnerabilities, significantly enhancing the capability to mitigate privacy risks proactively in healthcare environments. Moreover, they support compliance-driven development, guiding software engineers towards more privacy-aware coding practices, fundamentally transforming how software vulnerabilities are managed and prioritized in healthcare systems [26].

7. Conclusions

The significance of privacy-aware vulnerability detection cannot be overstated, particularly in healthcare contexts where privacy breaches can have profound implications on patient safety and compliance with strict regulatory frameworks. The C3-VULMAP dataset substantially advances the field by explicitly integrating the LINDDUN privacy framework with CWE vulnerability classifications, creating a unique and valuable resource tailored to healthcare privacy concerns. Its combination of real-world and synthetic examples

provides balanced and comprehensive vulnerability representation, facilitating superior model training and generalization capabilities.

Given these strengths, further use and collaborative enhancements of the C3-VULMAP dataset are strongly encouraged. Researchers, practitioners, and policymakers in cybersecurity and healthcare are invited to engage with and contribute to this evolving resource, promoting broader adoption and continuous improvement in privacy vulnerability detection methodologies.

Practical implications from this study highlight considerable challenges and critical considerations in labeling and detecting vulnerabilities. One notable challenge is ensuring accurate manual labeling, which remains essential despite advancements in automated detection methodologies. The reliance on domain expertise for manual labeling poses significant resource implications, highlighting the reliance on human oversight to validate automated findings. The integration of synthetic vulnerabilities, although beneficial, also presents challenges related to ensuring their realism and representativeness. Practical deployment further demands addressing issues, such as managing false positives, refining confidence thresholds, and ensuring that detected vulnerabilities are actionable and relevant, thus necessitating ongoing iterative improvements and adaptations to maintain robust and accurate vulnerability detection in dynamic healthcare environments.

Expansion to additional programming languages is another important direction, as the current dataset predominantly focuses on C/C++ due to their prevalent use in safety-critical applications, like those used in healthcare. Incorporating other widely used languages like Python, Java, and JavaScript would broaden the applicability of the dataset, providing comprehensive coverage across diverse healthcare systems and software environments. In future work, we aim to investigate the integration of the C3-VULMAP dataset into continuous integration and continuous deployment (CI/CD) pipelines to support real-time, privacy-aware vulnerability assessment in healthcare software systems. By embedding trained detection models into automated development workflows, it will enable the early identification of privacy-related vulnerabilities during the coding and testing phases, ensuring that threats aligned with LINDDUN privacy categories are detected and mitigated before deployment. Furthermore, such integration will support continuous monitoring and automated security feedback, which is especially important in rapidly evolving healthcare environments. We envision adapting C3-VULMAP for lightweight, containerized tools that integrate with platforms, like Jenkins or GitLab, providing actionable feedback to developers and enhancing proactive privacy protection in dynamic healthcare software environments.

Author Contributions: J.E.A. led the conceptualization, dataset construction, model implementation, and manuscript drafting. A.O. contributed to the methodological framework, evaluation design, and critical revisions of the manuscript. A.S. supervised the experimental validation, data interpretation, and provided technical oversight throughout the study. A.I. supported the integration of privacy frameworks, ensured alignment with healthcare cybersecurity standards, and reviewed the manuscript for intellectual content. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original data presented in the study are openly available at https://github.com/juxam/C3-VULMAP (accessed on 29 May 2025).

Acknowledgments: During the preparation of this manuscript/study, the author(s) used ChatGPT 4o and Claude for the purposes of Python script enhancement. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: There are no known conflicts of interest.

References

1. Mejía-Granda, C.M.; Fernández-Alemán, J.L.; Carrillo-De-Gea, J.M.; García-Berná, J.A. Security vulnerabilities in healthcare: An analysis of medical devices and software. *Med. Biol. Eng. Comput.* **2023**, *62*, 257–273. [CrossRef] [PubMed]
2. Protić, D.D. Review of KDD Cup '99, NSL-KDD and Kyoto 2006+ datasets. *Vojn. Glas.* **2017**, *66*, 580–596. [CrossRef]
3. Bala, R.; Nagpal, R. A review on KDD Cup '99 and NSL-KDD dataset. *Int. J. Adv. Res. Comput. Sci.* **2019**, *10*, 64–67. [CrossRef]
4. Chen, Y.; Ding, Z.; Alowain, L.; Chen, X.; Wagner, D. DiverseVul: A New Vulnerable Source Code Dataset for Deep Learning Based Vulnerability Detection. In Proceedings of the 26th International Symposium on Research in Attacks, Intrusions and Defenses (RAID '23), Hong Kong, China, 16–18 October 2023; pp. 1–15. [CrossRef]
5. Wang, X.; Hu, R.; Gao, C.; Wen, X.-C.; Chen, Y.; Liao, Q. ReposVul: A Repository-Level High-Quality Vulnerability Dataset. In Proceedings of the 46th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion '24), Lisbon, Portugal, 14–20 April 2024; pp. 472–483. [CrossRef]
6. Bhandari, G.; Naseer, A.; Moonen, L. CVEfixes: Automated collection of vulnerabilities and their fixes from open-source software. In Proceedings of the 17th International Conference on Predictive Models and Data Analytics in Software Engineering (PROMISE '21), Athens, Greece, 19–20 April 2021; pp. 30–39. [CrossRef]
7. Harzevili, N.S.; Belle, A.B.; Wang, J.; Wang, S.; Jiang, Z.M.; Nagappan, N. A Systematic Literature Review on Automated Software Vulnerability Detection Using Machine Learning. *ACM Comput. Surv.* **2024**, *57*, 1–36. [CrossRef]
8. Esposito, M.; Falessi, D. VALIDATE: A deep dive into vulnerability prediction datasets. *Inf. Softw. Technol.* **2024**, *170*, 107448. [CrossRef]
9. Al Zaabi, M.; Alhashmi, S.M. Big data security and privacy in healthcare: A systematic review and future research directions. *J. Inf. Sci.* **2024**, *50*, 1247781. [CrossRef]
10. Ewoh, P.; Vartiainen, T. Vulnerability to Cyberattacks and Sociotechnical Solutions for Health Care Systems: Systematic Review. *J. Med. Internet Res.* **2024**, *26*, e46904. [CrossRef] [PubMed]
11. Williamson, S.M.; Prybutok, V. Balancing Privacy and Progress: A Review of Privacy Challenges, Systemic Oversight, and Patient Perceptions in AI-Driven Healthcare. *Appl. Sci.* **2023**, *14*, 675. [CrossRef]
12. Fan, J.; Li, Y.; Wang, S.; Nguyen, T.N. A C/C++ Code Vulnerability Dataset with Code Changes and CVE Summaries. In Proceedings of the 17th International Conference on Mining Software Repositories (MSR 2020), Seoul, Republic of Korea, 29–30 June 2020; pp. 508–512. [CrossRef]
13. Guo, Y.; Bettaieb, S.; Casino, F. A comprehensive analysis on software vulnerability detection datasets: Trends, challenges, and road ahead. *Int. J. Inf. Secur.* **2024**, *23*, 3311–3327. [CrossRef]
14. Pinto, A.; Herrera, L.-C.; Donoso, Y.; Gutierrez, J.A. Survey on Intrusion Detection Systems Based on Machine Learning Techniques for the Protection of Critical Infrastructure. *Sensors* **2023**, *23*, 2415. [CrossRef] [PubMed]
15. Sion, L.; Wuyts, K.; Yskout, K.; Van Landuyt, D.; Joosen, W. Interaction-Based Privacy Threat Elicitation. In Proceedings of the 2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), London, UK, 23–27 April 2018; pp. 79–86. [CrossRef]
16. Wuyts, K.; Sion, L.; Joosen, W. LINDDUN GO: A Lightweight Approach to Privacy Threat Modeling. In Proceedings of the 2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), Genoa, Italy, 7–11 September 2020; pp. 302–309. [CrossRef]
17. Naik, N.; Jenkins, P.; Grace, P.; Naik, D.; Prajapat, S.; Song, J. A Comparative Analysis of Threat Modelling Methods: STRIDE, DREAD, VAST, PASTA, OCTAVE, and LINDDUN. In *Contributions Presented at The International Conference on Computing, Communication, Cybersecurity and AI, July 3–4, 2024, London, UK. C3AI 2024*; Lecture Notes in Networks and Systems, Volume 884; Naik, N., Jenkins, P., Prajapat, S., Grace, P., Eds.; Springer: Cham, Switzerland, 2024. [CrossRef]
18. Langthaler, O.; Eibl, G.; Klüver, L.-K.; Unterweger, A. Evaluating the Efficacy of LINDDUN GO for Privacy Threat Modeling for Local Renewable Energy Communities. In Proceedings of the 11th International Conference on Information Systems Security and Privacy (ICISSP 2025), Porto, Portugal, 20–22 February 2025; Volume 2, pp. 518–525. [CrossRef]
19. Sion, L.; Van Landuyt, D.; Wuyts, K.; Joosen, W. Robust and reusable LINDDUN privacy threat knowledge. *Comput. Secur.* **2025**, *154*, 104419. [CrossRef]
20. Raciti, M.; Bella, G. A Threat Model for Soft Privacy on Smart Cars. In Proceedings of the 2023 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), Delft, The Netherlands, 3–7 July 2023; pp. 1–10. [CrossRef]
21. Nweke, L.O.; Abomhara, M.; Yayilgan, S.Y.; Comparin, D.; Heurtier, O.; Bunney, C. A LINDDUN-Based Privacy Threat Modelling for National Identification Systems. In Proceedings of the 2022 IEEE Nigeria 4th International Conference on Disruptive Technologies for Sustainable Development (NIGERCON), Abuja, Nigeria, 5–7 April 2022; pp. 1–6. [CrossRef]
22. Kunz, I.; Xu, S. Privacy as an Architectural Quality: A Definition and an Architectural View. In Proceedings of the 2023 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), Delft, The Netherlands, 3–7 July 2023; pp. 125–132. [CrossRef]

23. Wuyts, K.; Scandariato, R.; Joosen, W. Empirical evaluation of a privacy-focused threat modeling methodology. *J. Syst. Softw.* **2014**, *96*, 122–138. [CrossRef]
24. Li, R.; Ben Allal, L.; Zi, Y.; Muennighoff, N.; Kocetkov, D.; Mou, C.; Marone, M.; Akiki, C.; Li, J.; Chim, J.; et al. StarCoder: May the source be with you! Transactions on Machine Learning Research. *arXiv* **2023**, arXiv:2305.06161.
25. Zouev, E. Programming Languages for Safety-Critical Systems. In *Software Design for Resilient Computer Systems*; Springer: Cham, Switzerland, 2020. [CrossRef]
26. Pereira, J.D.; Ivaki, N.; Vieira, M. Characterizing Buffer Overflow Vulnerabilities in Large C/C++ Projects. *IEEE Access* **2021**, *9*, 142879–142892. [CrossRef]
27. Li, H.; Li, C.; Wang, J.; Yang, A.; Ma, Z.; Zhang, Z.; Hua, D. Review on security of Federated Learning and its application in Healthcare. *Future Gener. Comput. Syst.* **2023**, *144*, 271–290. [CrossRef]
28. Ponggawa, V.V.; Santoso, U.B.; Talib, G.A.; Lamia, M.A.; Manuputty, A.R.; Yusuf, M.F. Comparative Study of C++ and C# Programming Languages. *J. Syntax. Admiration* **2024**, *5*, 5743–5748. [CrossRef]
29. Ma, X.; Yan, J.; Wang, W.; Yan, J.; Zhang, J.; Qiu, Z. Detecting Memory-Related Bugs by Tracking Heap Memory Management of C++ Smart Pointers. In Proceedings of the 2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE), Melbourne, Australia, 15–19 November 2021; pp. 880–891. [CrossRef]
30. Rassokhin, D. The C++ programming language in cheminformatics and computational chemistry. *J. Cheminform.* **2020**, *12*, 10. [CrossRef] [PubMed]
31. Ganz, T.; Härterich, M.; Warnecke, A.; Rieck, K. Explaining Graph Neural Networks for Vulnerability Discovery. In Proceedings of the 14th ACM Workshop on Artificial Intelligence and Security (AISec '21), Virtual Event, Republic of Korea, 15–19 November 2021; pp. 145–156. [CrossRef]
32. Guo, W.; Fang, Y.; Huang, C.; Ou, H.; Lin, C.; Guo, Y. HyVulDect: A hybrid semantic vulnerability mining system based on graph neural network. *Comput. Secur.* **2022**, *121*, 102823. [CrossRef]
33. Xia, Y.; Shao, H.; Deng, X. VulCoBERT: A CodeBERT-Based System for Source Code Vulnerability Detection. In Proceedings of the 2024 International Conference on Generative Artificial Intelligence and Information Security (GAIIS 2024), Kuala Lumpur, Malaysia, 10–12 May 2024; pp. 249–252. [CrossRef]
34. Liu, R.; Wang, Y.; Xu, H.; Sun, J.; Zhang, F.; Li, P.; Guo, Z. Vul-LMGNNs: Fusing language models and online-distilled graph neural networks for code vulnerability detection. *arXiv* **2024**, arXiv:2404.14719.
35. Kalouptsoglou, I.; Siavvas, M.; Ampatzoglou, A.; Kehagias, D.; Chatzigeorgiou, A. Vulnerability prediction using pre-trained models: An empirical evaluation. In Proceedings of the 32nd International Conference on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), Krakow, Poland, 21–23 October 2024; pp. 1–6. [CrossRef]
36. Choubisa, M.; Doshi, R.; Khatri, N.; Hiran, K.K. A Simple and Robust Approach of Random Forest for Intrusion Detection System in Cyber Security. In Proceedings of the 2022 International Conference on IoT and Blockchain Technology (ICIBT), Ranchi, India, 6–8 May 2022; pp. 1–5. [CrossRef]
37. Meng, N.; Nagy, S.; Yao, D.; Zhuang, W.; Argoty, G.A. Secure coding practices in Java. In Proceedings of the 40th International Conference on Software Engineering (ICSE '18), Gothenburg, Sweden, 27 May–3 June 2018; pp. 372–383. [CrossRef]
38. Altamimi, S. Investigating and Mitigating the Role of Neutralisation Techniques on Information Security Policies Violation in Healthcare Organisations. Ph.D. Thesis, University of Glasgow, Glasgow, UK, 2022. [CrossRef]
39. Babu, M.; Suryanarayana Reddy, N.R.; Moharir, M. Mohana Leveraging XGBoost Machine Learning Algorithm for Common Vulnerabilities and Exposures (CVE) Exploitability Classification. In Proceedings of the 2024 8th International Conference on Computational System and Information Technology for Sustainable Solutions (CSITSS), Bengaluru, India, 7–9 November 2024; pp. 1–6. [CrossRef]