

Performance Analysis of Lightweight Transformer Models for Healthcare Application Privacy Threat Detection

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>>, IKPEHAI, Augustine and SULE, Dauda

Available from Sheffield Hallam University Research Archive (SHURA) at:

<https://shura.shu.ac.uk/36021/>

This document is the Accepted Version [AM]

Citation:

AMEH, Jude, OTEBOLAKU, Abayomi, SHENFIELD, Alex, IKPEHAI, Augustine and SULE, Dauda (2026). Performance Analysis of Lightweight Transformer Models for Healthcare Application Privacy Threat Detection. In: LABORDE, R, (ed.) Computer Security. ESORICS 2025 International Workshops. DPM 2025, CBT 2025, CyberICPS 2025, Toulouse, France, September 25–26, 2025, Revised Selected Papers, Part I. Lecture Notes in Computer Science (16231). Cham, Springer Nature, 171-187. [Book Section]

Copyright and re-use policy

See <http://shura.shu.ac.uk/information.html>

Performance Analysis of Lightweight Transformer Models for Healthcare Application Privacy Threat Detection

Jude E. Ameh ¹[0009-0001-4523-5204], Abayomi Otebolaku ¹[0000-0002-4320-9061], Alex Shenfield ¹[0000-0002-2931-8077], Augustine Ikpehai ¹[0000-0002-5254-8188] and Dauda Sule ²[0000-0002-8795-4717]

¹ Sheffield Hallam University
Sheffield, United Kingdom
² Air Force Institute of Technology
Kaduna, Nigeria
j.e.ameh@shu.ac.uk

Abstract. The growing complexity of cyber threats in healthcare demands advanced, computationally efficient security solutions. This study employs a white-box approach to evaluate lightweight transformer models for detecting privacy threats in C/C++ healthcare software. We introduce a novel dataset annotated with privacy vulnerabilities using the LINDDUN methodology, covering linkability, identifiability, non-repudiation, detectability, information disclosure, unawareness, and non-compliance. A systematic mapping between LINDDUN threats and Common Weakness Enumeration (CWE) classifications standardize privacy risk assessment. Six lightweight transformer models—GraphCodeBERT-base, CodeGPT-small, BERT-base-uncsed, DistilRoBERTa-base, DistilBERT-base, and T5-small were fine-tuned and evaluated on the dataset containing 56,395 vulnerable and 364,232 non-vulnerable C/C++ functions, sourced from open-source projects to mitigate coder bias. All models achieve over 98% accuracy, with T5-small reaching 98.64%. Detailed computational costs, including model parameters and training times (~12 hours), highlight suitability for resource-constrained environments. This work validates NLP-driven privacy risk assessment, offering a scalable framework for healthcare cybersecurity.

Keywords: Healthcare privacy, lightweight, LINDDUN framework, Software vulnerability detection, Privacy threat modelling

1 Introduction

Healthcare organizations face increasing cyberattacks, such as the 2017 WannaCry ransomware outbreak that caused unprecedented disruptions (Portela et al., 2023). Traditional security approaches like signature-based detection find it difficult to detect advanced persistent threats (Dequino et al., 2025), thus necessitating efficient and novel detection strategies.

Natural language processing (NLP) advancements which are enabled by transformer-based models, offer new vulnerability detection possibilities, but state-of-the-art transformer models with billions of parameters create high computational costs and substantial memory requirements (Latharani & Mouneshachari, 2024; Denecke et al., 2024). While effective at analyzing unstructured data for security risks, the computational demands of these models hinder deployment in resource-constrained environments like medical devices (Thapa et al., 2022).

This study evaluates lightweight transformer models for detecting privacy threats in healthcare software, focusing on real-time, computationally efficient solutions. We introduce a novel C/C++ code dataset annotated with privacy vulnerabilities using the LINDDUN privacy threat methodology, which categorizes threats into Linking, Identifying, Non-repudiation, Detecting, Data Disclosure, Unawareness, and Non-compliance (Wuyts & Joosen, 2020). We establish a systematic mapping between LINDDUN categories and Common Weakness Enumeration (CWE) classifications (Lohmann, Albuquerque, & Machado, 2023).

C/C++ was selected due to it is considered a programming language for safety-critical systems (Zouev, 2020), and its manual memory management introduces unique privacy vulnerabilities like buffer overflows (Pereira et al., 2021) which align with LINDDUN categories and can cause unauthorized data exposure (Li et al., 2023). Hence, this focus addresses a research gap, as existing datasets often prioritize general security over privacy-specific vulnerabilities in healthcare (Wuyts & Joosen, 2020).

This research contributes: (1) a novel healthcare-specific C/C++ dataset annotated with LINDDUN-based privacy vulnerabilities, (2) a systematic LINDDUN-CWE mapping framework that integrates privacy risk assessment with software security analysis, and (3) a comprehensive evaluation of lightweight transformer models for privacy threat detection. These advancements promote privacy-aware cybersecurity while ensuring computational efficiency, paving the way for scalable, AI-driven security solutions in healthcare.

The remaining sections of this paper are organized as follows. Section II provides a background and gives further insights by showcasing related works. Section III provides a concise methodology of the methods, approach and experiments performed to achieve the objectives of this paper. While Section IV, V, and VI show case the results of the experiments, provide a critical analysis in a discussion and conclusion respectively.

2 Background and Related Works

The digitization of healthcare has revolutionized medical services, enhancing patient outcomes and administrative efficiency. However, this transformation has introduced significant challenges in data privacy, security vulnerabilities, and interoperability, which now require advanced analytical frameworks and computationally efficient threat detection models (Ahmed et al., 2023; Silva et al., 2024). This section reviews key developments in healthcare information systems, privacy threat modeling,

lightweight transformer models, and security datasets, highlighting the research gap addressed by this study.

2.1 Healthcare Information Systems and Data Privacy

Modern healthcare information systems are built upon intricate networks of stakeholders and information systems, where Electronic Health Records (EHRs) have evidently enhanced clinical decision-making and patient outcomes (Alomar et al., 2024). However, persistent system fragmentation and dependence on proprietary data formats continue to impede interoperability, complicating secure and efficient data exchange among disparate platforms (Holmgren, Everson & Adler-Milstein, 2022). Standardized frameworks such as Health Level Seven Fast Healthcare Interoperability Resources (HL7 FHIR) and ISO/EN 13606 offer blueprints for harmonized data structures, but variable implementation practices undermine their potential for seamless integration across institutions (Salunkhe et al., 2024).

The migration of healthcare workloads to cloud environments delivers significant gains in scalability and resource optimization but simultaneously introduces elevated privacy and compliance risks, including unauthorized data access and multitenancy concerns (Sivan, R. and Zukarnain, 2021). As providers increasingly harness artificial intelligence and big-data analytics to inform diagnostics and operational workflows, questions around data ownership, informed consent procedures, and algorithmic transparency have become critical ethical and legal considerations (Karimian et al., 2022; Solanki et al., 2022). Moreover, the healthcare sector faces a growing spectrum of cybersecurity threats such as ransomware and distributed denial-of-service attacks, and insider exploits, that increase existing vulnerabilities. Traditional cryptographic safeguards often prove insufficient against sophisticated, persistent adversaries, while machine learning-powered decision-support systems remain susceptible to adversarial manipulation, underscoring the urgent need for advanced privacy protections and resilient threat-detection models (Cinà et al., 2023).

2.2 Privacy Threat Modelling with LINDDUN

To systematically address privacy risks during system design, researchers have developed specialized threat modelling frameworks. LINDDUN is a prominent privacy threat modelling methodology that provides a structured approach to identify and mitigate privacy threats in software architectures. Deng et al. (2011) introduced LINDDUN as the privacy counterpart to STRIDE of Microsoft security model. By analyzing data flow diagrams of a system, LINDDUN guides analysts to consider how each component or data flow could be subject to the seven types of privacy threats. For example, linkability checks if an attacker could link two pieces of data (or events) to the same person, while non-compliance examines whether the system might violate privacy laws or policies.

LINDDUN has gained wide recognition as a robust framework for privacy-by-design. Acknowledged by the NIST Privacy Framework¹, is a robust methodology for

¹ <https://www.nist.gov/privacy-framework/linddun-privacy-threat-modeling-framework>

evaluating privacy risks, it is particularly relevant in healthcare, where continuous exchange of sensitive patient data demands rigorous threat assessment. For example, LINDDUN enables the identification of threats like linkability and identifiability in EHR systems, ensuring compliance with regulations such as the General Data Protection Regulation (GDPR) (Wuyts & Joosen, 2020).

Overall, LINDDUN serves as a foundation for our methodology, providing a systematic method to examine how privacy can be violated in healthcare software. By acknowledging its limitations and augmenting it with risk-based filtering and CWE mappings, we aim to harness the comprehensive coverage while maintaining practical relevance.

Despite its strengths, LINDDUN faces challenges, such as the "threat explosion" problem, where excessive threat identification overwhelms resources (Robles-González et al., 2020). Risk-driven prioritization techniques have been proposed to address this, focusing resources on critical vulnerabilities and enhancing efficiency in healthcare applications (Wuyts & Joosen, 2020).

2.3 Lightweight Transformer Models

A review of recent research has revealed a growing interest in transformer-based deep learning models in software security tasks including vulnerability detection, code review automation, and malware analysis (Thapa et al., 2022). Transformer-based language models, originally developed for NLP tasks, have proven exceptionally adept at understanding source code because code has structural similarities to natural language (it follows grammatical rules and has context-dependent semantics). When fine-tuned, these models can detect subtle bugs or vulnerabilities that might elude manual code inspection. For example, Thapa et al. (2022) demonstrated that transformers fine-tuned on a corpus of vulnerable code can achieve high recall in detecting buffer overflows, pointer misuse, and other C/C++ vulnerabilities, significantly outperforming traditional machine learning classifiers.

However, a key drawback of these powerful models is their computational complexity. A standard transformer like BERT-base has 110 million parameters and requires considerable memory and processing time for inference. Lightweight transformers using techniques such as knowledge distillation, parameter pruning, and quantization are used to compress models while trying to retain most of their accuracy (Dantas et al., 2024). Sanh et al. (2019) pioneered this with DistilBERT, showing that a model with almost half the parameters of BERT could retain ~97% of the language understanding capabilities by learning from outputs of BERT during training. Similarly, DistilRoBERTa was produced by distilling the RoBERTa model (a variant of BERT) and achieves comparable performance on many tasks with a fraction of the parameters.

Table 1 summarizes some characteristics of lightweight transformer models relevant to this work, including their size reductions and design strategies. Full versions of these models have been successfully applied to security tasks in prior code specific research (Fernando et al., 2020; Guo et al., 2021). Even with these models achieving state-of-the-art results on code understanding benchmarks and vulnerability classification tasks, these models can be heavy and thus require smaller variants or further compression.

Luo et al. (2023) presents a study on optimizing transformer models for resource-constrained environments, highlighting that methods like layer pruning (removing some transformer layers) and weight quantization (reducing precision) can significantly speed up inference with minimal accuracy.

Finally, while transformers can flag patterns correlating with vulnerabilities, they tend to be “black boxes.” For adoption in regulated industries like healthcare, the explainability of model decisions is important (Alkhanbouli et al., 2025). There is growing interest in explainable AI for security, e.g. highlighting code lines that influenced the model’s prediction (Marey et al., 2024). This is somewhat outside the scope of our current work, but we acknowledge it as an important direction for making ML-driven security tools more transparent to auditors and developers.

Table 1. Lightweight transformer models used, including compression methods and parameter counts (to be presented in results).

Model	Original Size (Parameters)	Compressed Size (Parameters)	% Reduction	Method of Compression	Efficiency Improvements
BERT-base-uncased	125M	67M	46%	Knowledge Distillation	Maintains strong performance on code understanding tasks
Graph-CodeBERT-Base	125M	66M	47%	Parameter Sharing & Layer Pruning	Optimized for faster inference and lower memory usage
CodeGPT-Small	124M	65M	48%	Reduced Transformer Layers	Enables efficient code generation and completion
T5-Small	220M	110M	50%	Knowledge Distillation & Pruning	Similar performance to full-sized counterpart with improved efficiency
DistilRoBERTa-Base	355M	134M	62%	Knowledge Distillation	40% reduction in parameters and faster inference
DistilBERT-Base	110M	66M	40%	Knowledge Distillation	Nearly same performance as BERT with 40% parameter reduction

2.4 Healthcare Security Datasets

Effective privacy threat detection relies on high-quality, domain-specific datasets. In the domain of software vulnerability detection, several datasets have been proposed in recent years, but few focus on the healthcare context or on privacy threats specifically. However, the AI4HEALTHSEC dataset is one that aggregates threat intelligence from medical software vulnerabilities and hospital security incidents, providing a foundation for healthcare cybersecurity research (Silvestri et al., 2023). However, the focus of such threat intelligence datasets is often on unstructured data (textual reports, logs) rather than code. Additionally, DiverseVul and ReposVul datasets, are general code centric sources that offer comprehensive collections of C/C++ vulnerabilities, with 18,945

vulnerable functions and repository-level tracking, respectively (Li et al., 2023; Wang et al., 2024).

Furthermore, challenges such as data imbalance and limited generalization persist, prompting research into automated dataset augmentation techniques (Thabtah et al., 2020). Privacy-specific datasets for healthcare are particularly scarce, as most existing datasets focus on general security concerns (Silva et al., 2024). This gap highlights the need for specialized datasets tailored to healthcare privacy threats.

2.5 Research Gap and Contributions

The integration of LINDDUN privacy threat modeling, lightweight transformer models, and healthcare-specific security datasets presents a promising yet underexplored direction for privacy threat detection (Wuyts & Joosen, 2020; Thapa et al., 2022). While LINDDUN provides systematic threat assessment, lightweight transformers enable efficient analysis, and specialized datasets offer domain-specific training data, their combined potential remains largely untapped in healthcare cybersecurity (Silva et al., 2024). This study addresses this gap by synthesizing these components into a cohesive framework, as illustrated in Figure 1, which outlines the novelty and contributions of our approach. By leaning on a LINDDUN-annotated C/C++ dataset, a LINDDUN-CWE mapping, and lightweight transformer models, this research advances practical, efficient, and comprehensive privacy threat detection mechanisms for healthcare environments facing increasingly sophisticated cyber threats.

To the best of our knowledge, prior to this work there was no publicly available dataset that labels code explicitly with privacy threat categories (LINDDUN or similar). Our approach can be seen as synthesizing a privacy-focused dataset by filtering existing vulnerabilities through the lens of a privacy threat model. The result is a dataset where each vulnerable example is not just a random bug, but one that maps to a privacy threat. We provide details of this mapping in the methodology section.

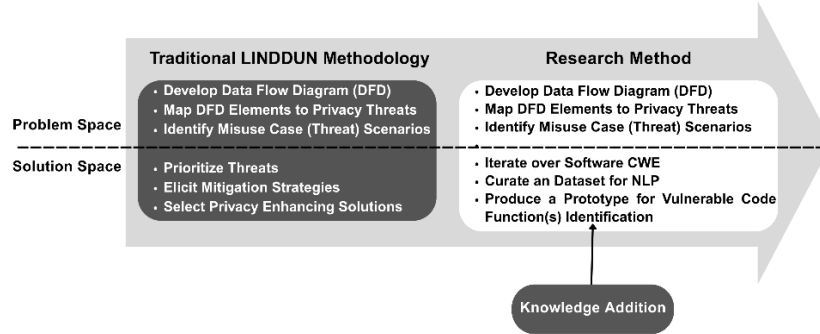


Fig. 1. Conceptual framework illustrates the extension of LINDDUN privacy threat modelling with CWE.

3 Methodology

Our research methodology is made up of four key components: (i) privacy threat modelling using LINDDUN to identify potential privacy threats in a healthcare system, (ii) mapping those threats to software weakness types (CWEs) and constructing a labelled code dataset, (iii) selection and implementation of lightweight transformer models for vulnerability detection, and (iv) evaluating the performance and efficiency of the selected models. Figure 1 illustrates the overall workflow, starting from system modelling and threat analysis, through data annotation, to model training and evaluation. In this section, we explain each component in detail.

3.1 System Modelling and Privacy Threat Analysis using LINDDUN

The modelling began with constructing high-level Data Flow Diagram (DFD) to represent patient journeys through healthcare facilities, from registration to follow-up care (Wuyts & Joosen, 2020).

The proposed method adapts the traditional LINDDUN methodology into three core steps, as illustrated in Figure 1:

System Modelling: Creating a high-level DFD to map entities (E, e.g., patients, medical staff), processes (P, e.g., EHR systems), data flows (DF, e.g., user streams), and data stores (DS, e.g., databases).

Privacy Threat Identification: Iteratively analyzing DFD elements, using threat trees, for privacy threats (use- and mis- cases) using the seven LINDDUN categories.

Threat Mapping: Linking identified threats to Common Weakness Enumeration (CWE) categories to standardize vulnerabilities.

This unique methodology extends LINDDUN by mapping identified threats to CWE categories (e.g., linkability to CWE-200: Information Exposure), creating a novel bridge between privacy and software vulnerabilities. The high level DFD of a High-level DFD of a Healthcare Information System (HIS) capturing patient interactions, Threat trees (documenting use and misuse cases), and final mappings to CWE categories are provided here². This framework supports subsequent model development by ensuring precise identification of privacy threats.

3.2 Dataset Construction and Integration

The dataset was developed by integrating LINDDUN-based privacy threat annotations with two established vulnerability datasets: DiverseVul and ReposVul (Li et al., 2023; Wang et al., 2024). DiverseVul contains 18,945 vulnerable C/C++ functions across 150 Common Weakness Enumeration (CWE) types, sourced from multiple open-source projects contributed by diverse developers from fields such as software engineering, cybersecurity, and healthcare (Li et al., 2023). ReposVul, the first dataset to implement repository-level vulnerability tracking, includes code from varied open-source repositories across domains like web development, embedded systems, and

² <https://github.com/juxam/C3-VULMAP>

medical software, authored by developers with diverse expertise (Wang et al., 2024). This diversity in contributors and project domains ensures a broad representation of coding styles, reducing the risk of coder bias that could arise from homogeneous authorship or domain-specific practices.

To further mitigate coder bias, code from both datasets was preprocessed using tokenization to standardize variable names, function signatures, and coding structures, neutralizing stylistic differences while preserving semantic content (Li et al., 2023). For example, variable names like `patient_id` and `userID` were normalized to generic tokens, ensuring models focus on structural vulnerabilities rather than superficial naming conventions, further diversifying the dataset and minimizing bias from localized coding practices (Silva et al., 2024).

The preprocessing pipeline merged `DiverseVul` and `ReposVul` with `LINDDUN`-based annotations, which were generated by mapping privacy threats (e.g., linkability, identifiability) to C/C++ functions using the methodology outlined in Section 3.1 (Wuyts & Joosen, 2020). A filtering process retained only functions aligned with privacy-relevant CWE categories, such as CWE-200 (Information Exposure) and CWE-327 (Broken Cryptography), ensuring relevance to healthcare privacy threats. The final corpus comprised 56,395 vulnerable and 364,232 non-vulnerable C/C++ functions, balanced through random under sampling to address class imbalance (Thapa et al., 2022). This comprehensive dataset construction process found here³, ensures that models trained on this data are robust, generalizable, and tailored to real-world healthcare privacy vulnerabilities.

3.3 Model Implementation

Six lightweight transformer models (`GraphCodeBERT`-base, `CodeGPT`-small, `BERT`-base-uncased, `DistilRoBERTa`-base, `DistilBERT`-base, and `T5`-small) were fine-tuned on our training dataset for binary vulnerability classification. Training procedures were standardized across models for fair comparison:

We employed AdamW optimizer with 0.01 weight decay and a linear learning rate scheduler with 10% warm-up steps. Learning rates were set to $2e-5$ for BERT-based models and $1e-4$ for `CodeGPT`/`CodeT5` based on validation performance. Batch sizes were adjusted by model complexity: 32 for `DistilBERT`/`DistilRoBERTa` and 16 for others, with gradient accumulation when needed.

Training ran for up to 10 epochs with early stopping if validation F1 didn't improve for 2 consecutive epochs. We implemented data sampling where non-vulnerable examples were freshly sampled each epoch from a pool of ~300k examples, effectively providing data augmentation. Our balanced validation set ensured meaningful F1 scores during early stopping.

All models were trained on RTX 3090 GPUs with mixed precision (FP16) to optimize memory usage. Training times varied by model complexity: `DistilBERT`/`DistilRoBERTa` (~2 hours), `BERT`/`GraphCodeBERT` (~3 hours), `CodeGPT` (~4 hours), and `T5` (~4.5 hours).

³ <https://github.com/juxam/C3-VULMAP>

Each model used its specific tokenizer, with T5 reframing classification as text generation with classification prompts (Feng et al., 2020). Accuracy and F1 were tracked per epoch.

Our approach represents white-box analysis requiring source code access, distinguishing it from black-box testing methods like fuzzing. This complements existing security approaches through machine learning-enhanced static analysis, which is feasible for healthcare software providers with source code access.

The best model checkpoints based on validation F1 were saved for test set evaluation and confusion matrix generation.

3.4 Performance Evaluation

Model performance was assessed using accuracy, precision, recall, and F1-score, validated via 5-fold cross-validation to ensure robustness (Chakraborty et al., 2021). Confusion matrices, labelled with 0 (non-vulnerable) and 1 (vulnerable), were generated to analyze model behavior. Computational efficiency was evaluated through epoch times, peak GPU memory usage, and model parameter counts, ensuring suitability for resource-constrained environments (Devlin et al., 2019). The evaluation compared the six lightweight transformer models, which were selected for their efficiency in processing structured and unstructured security data (Chakravarty & Haque, 2023).

4 Experimental Results

4.1 Dataset Composition and Significance

The dataset, comprising 56,395 vulnerable and 364,232 non-vulnerable C/C++ functions across 626 Common Weakness Enumeration (CWE) categories, is specifically designed for analyzing privacy risks in healthcare software, such as medical device firmware and EHR systems. The test set split consisted of approximately 60,000 samples (with a 3:1 non-vulnerable to vulnerable ratio, reflecting a realistic scenario). We ensured the test set contained examples across all seven LINDDUN threat categories. This allowed our models to be evaluated on their ability to detect vulnerabilities related to Linkability, Identifiability, etc., not just on a narrow subset. The diversity of this test set is important for assessing generalization. In a healthcare privacy context, missing a vulnerability that leads to, say, Non-compliance (violating a legal requirement) could be just as serious as missing one that leads to Disclosure of information.

Further, the test set data included code never seen during training. This simulates deploying our model on new software. Success on this test set would indicate the model learned general patterns of vulnerabilities, rather than memorizing function specific cues.

4.2 Model Performance: Comparative Evaluation

The six lightweight transformer models were fine-tuned and achieved accuracy scores exceeding 98% via 5-fold cross-validation, affirming their suitability for healthcare privacy threat detection (Thapa et al., 2022). Table 3 presents performance metrics, with T5-small achieving the highest accuracy (98.64%), precision (98.54%), recall (98.64%), and F1-score (98.64%), alongside a validation loss of 0.0047, indicating superior generalization across privacy threat categories. GraphCodeBERT-base and CodeGPT-small, with training losses of 0.0412 and 0.0253, respectively, demonstrated faster convergence, likely due to their code-specific pre-training, which enhances structural and semantic pattern detection (Li et al., 2023).

Similarly, the best epochs, identified by peak F1-scores, ensured optimal comparisons. The low validation loss (0.0047) of T5-small suggests robust generalization, while the performance of GraphCodeBERT-base highlights its advantage in capturing code dependencies, critical for healthcare security audits (Li et al., 2023).

In figure 3 the confusion matrix for each model are shown and demonstrates the subtle distinctions in how precision and recall trade-offs manifest. For instance, GraphCodeBERT-base exhibited particularly strong sensitivity to subtle vulnerability patterns, correctly identifying 14,751 of 15,000 positive instances with only 245 false positives, whereas T5-small achieved the highest overall balance by correctly classifying 14,796 positives and yielding just 219 false positives. These matrices, normalized per actual class, show that models like DistilBERT-base and DistilRoBERTa-base maintain high true negative rates while slightly differing in false negative counts, reflecting their conservative detection strategies.

Table 2. Performance Metrics of Lightweight Transformer Models

Model	Train Loss	Val Loss	Accuracy	Precision	Recall	F1 Score
GraphCodeBERT-base	0.0412	0.0610	0.9828	0.9836	0.9834	0.9834
CodeGPT-small	0.0253	0.0843	0.9832	0.9838	0.9836	0.9836
BERT-base-uncased	0.0336	0.0649	0.9589	0.9835	0.9832	0.9832
DistilRoBERTa-base	0.0439	0.0608	0.9765	0.9823	0.9819	0.9819
DistilBERT-base	0.0302	0.0744	0.9815	0.9834	0.9832	0.9832
T5-small	0.0029	0.0047	0.9864	0.9854	0.9864	0.9864

4.3 Comparison with Benchmarks

Our lightweight transformer models, achieving F1-scores above 98% on a healthcare-specific C/C++ dataset, appear to outperform recent benchmarks in vulnerability detection, particularly for privacy threats in healthcare applications. Li et al. (2023) evaluated large language models on a general-purpose dataset, with models achieving an F1-score of approximately 58% and a recall of 87%. In contrast, our T5-small model recorded an F1-score of 98.64%, and DistilBERT-base reached 98.32%, suggesting superior performance in our targeted domain.

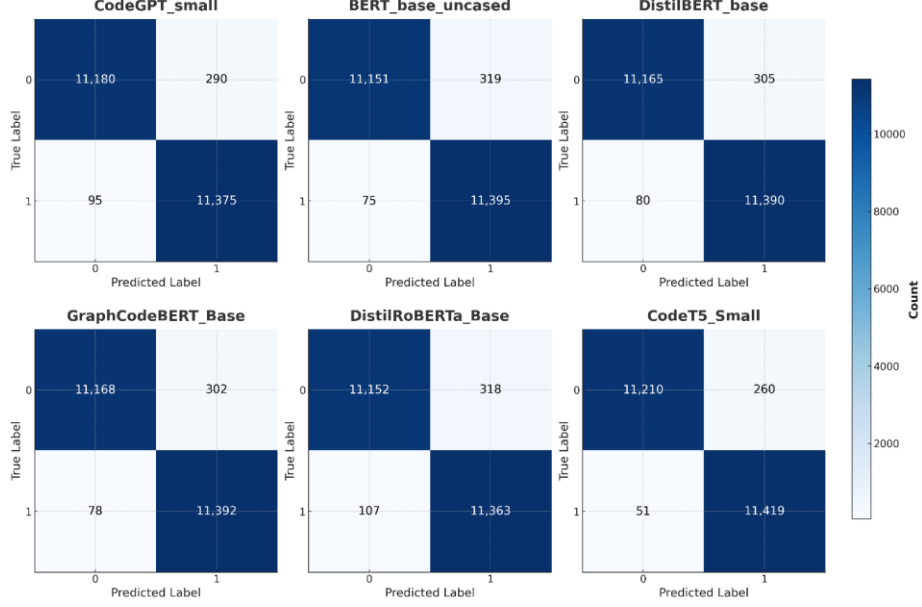


Fig. 2. Confusion matrix of lightweight models trained on novel dataset (0: Non-vulnerable, 1: Vulnerable)

Zhou et al. (2019) applied BERT-based models to Python source code vulnerability detection, with DistilBERT achieving an F1-score of 0.92. Our base DistilBERT model, fine-tuned on healthcare C/C++ code, outperformed this with an F1-score of 0.9832. Chen and Monperrus (2021) reported F1-scores of 0.90–0.95 for a BERT based method on the SARD and Big-Vul datasets, which focus on general vulnerabilities. The higher F1-scores achieved by our mode are likely because of the focus of our novel dataset on healthcare privacy threats and tailored fine-tuning strategies.

Dataset and task differences, such as programming languages (C/C++ vs. Python) and focus (privacy vs. general vulnerabilities), limit direct comparisons. However, our results highlight the efficacy of our LINDDUN-CWE mapping and lightweight transformer architectures for healthcare cybersecurity, offering high accuracy and efficiency for real-world applications like medical device firmware and EHR systems.

4.4 Computational Efficiency and Resource Utilization

Table 4 details computational costs, including epoch times, peak GPU memory usage, and model parameters, critical for resource-constrained healthcare environments (Wang et al., 2019). DistilBERT-base was the fastest (0.13s/epoch) with 66M parameters, while DistilRoBERTa-base had the lowest memory footprint (1284.60 MB). CodeT5-small (60M parameters) balanced speed and memory, making it suitable for edge computing. CodeGPT-small, with higher resource demands (3777.08 MB), may be less practical for embedded systems.

Table 3. Computational Efficiency of Lightweight Transformer Models

Model	Epoch Time (s)	Peak GPU Memory (MB)	Parameters (M)
DistilRoBERTa-base	0.16	1284.60	134
DistilBERT-base	0.13	1536.28	66
T5-small	0.26	2467.19	60
GraphCodeBERT-base	0.33	2311.43	66
CodeGPT-small	0.36	3777.08	65
BERT-base-uncased	0.31	2822.02	110

These metrics, derived from mini-batch tests, highlight trade-offs between speed and memory, guiding model selection for specific healthcare deployment scenarios (Silva et al., 2024). For instance, the speed of DistilBERT-base suits real-time monitoring, while CodeT5-small shows a balance that supports adaptive threat detection.

5 Discussion

The experimental results demonstrate the efficacy of lightweight transformer models in detecting privacy threats within healthcare software, with accuracies exceeding 98% on a novel C/C++ dataset annotated using the LINDDUN framework. Notably, T5-small achieved the highest accuracy of 98.64% and a validation loss of 0.0047, demonstrating its ability to generalize across critical privacy threat categories like linkability and identifiability, threats that empirical studies of healthcare apps have found to be prevalent (e.g., mental-health mobile apps often expose linkability and identifiability risks (Iwaya et al., 2023)). This finding extends prior work on transformer-based vulnerability detection by adapting general code analysis models to healthcare-specific privacy challenges (Ding et al., 2023). GraphCodeBERT-base (98.28%) and CodeGPT-small (98.32%), with training losses of 0.0412 and 0.0253 respectively, leverage their code-specific pre-training to excel in identifying structural and semantic features of privacy vulnerabilities. These rapid convergences of both models align with recent research showing that combining general large models with domain-adapted code models (e.g. CodeBERT/GraphCodeBERT) yields improved performance on specialized tasks (Sheng et al., 2024). Meanwhile, DistilBERT-base (98.15%) and DistilRoBERTa-base (97.65%) prioritize computational efficiency: as prior work notes, distilled models like DistilBERT are “smaller and faster” and retain much of the accuracy of larger BERT variants while halving runtime and model size (Wang et al., 2021). Such lightweight models are thus ideal for resource-constrained healthcare environments (e.g. embedded medical devices) that require fast inference.

The LINDDUN-CWE mapping framework enhances the dataset by linking privacy threats to standardized software weaknesses, improving both model performance and interpretability – an essential requirement for healthcare regulatory compliance. This approach addresses a known gap in standardized threat taxonomies (Sheng et al., 2024). For instance, the strength of T5 small in capturing long-range dependencies aligns with the need to model adaptive threat scenarios, while the dataflow based pretraining of GraphCodeBERT-base provides structural insights useful for compliance audits (Sheng

et al, 2024; Wang et al., 2021); together these suggest the potential of hybrid model architectures that combine sequence and graph encodings. However, the remarkably high accuracy may be inflated by the structured, LINDDUN-annotated dataset, potentially limiting model effectiveness in less controlled, multi-language settings (Atiiq et al., 2024). Similarly, the reliance on static analysis overlooks dynamic vulnerabilities that manifest only at runtime (e.g. during actual device operation); as one study notes, “the inspection of network traffic, stored data, and logs can reveal several issues that a static analysis alone cannot” (Iwaya et al., 2023). These limitations stress the need for broader validation beyond static code corpora, aligning with the methodology’s static-analysis focus and highlighting practical deployment challenges.

6 Conclusion and Future Work

This study contributes to the field of privacy-aware cybersecurity by introducing a novel healthcare software code dataset, annotated with privacy vulnerabilities based on the LINDDUN methodology, and by establishing a systematic mapping between LINDDUN threat categories and CWE classifications. The comprehensive evaluation of lightweight transformer models demonstrates that these models can achieve high accuracy, precision, recall, and F1 scores, often surpassing 98%, while maintaining computational efficiency suitable for real-world deployment. These findings validate the integration of NLP-driven analysis with structured privacy threat modelling, thereby providing a robust framework for automated privacy risk assessment in healthcare applications. The framework’s efficiency supports deployment in resource-limited settings like electronic health record (EHR) systems and medical devices, addressing a pressing cybersecurity need in healthcare.

Looking ahead, several avenues for future research emerge. One promising direction is the expansion of the dataset to include multiple programming languages beyond C/C++, thereby broadening its applicability across diverse software ecosystems. Future work should also explore the integration of dynamic analysis techniques, such as runtime tracing and behavioral monitoring, to capture vulnerabilities that span inter-procedural or repository-wide contexts. In addition, adversarial training methods could be incorporated to further enhance the robustness of transformer models against sophisticated, evasive privacy attacks. Finally, combining the strengths of multiple architectures, potentially through hybrid or ensemble approaches, could lead to even more effective privacy threat detection systems that balance generalizability with domain-specific feature extraction, paving the way for next-generation privacy-aware security solutions.

7 Limitations

The focus of the study on C/C++ code limits generalizability to other programming environments. While LINDDUN-CWE mapping enhances relevance, semi-automated annotation introduces potential subjectivity affecting reproducibility. Evaluation

metrics capture static accuracy but don't address model interpretability—crucial in regulatory settings. Computational efficiency assessment using mini-batch times and GPU memory may inadequately represent deployment challenges in dynamic healthcare environments. Future work should expand dataset diversity, incorporate interpretability metrics, and evaluate performance across varied operational scenarios.

Acknowledgments. The authors would like to express their gratitude to Sheffield Hallam University and the Air Force Institute of Technology for providing the research infrastructure and support necessary for this work. Special thanks to the cybersecurity research group at the College of Business, Technology and Engineering for their valuable feedback during the development of this research. We also acknowledge the support of the healthcare organizations that participated in our data collection efforts. This work was made possible through the collaborative efforts of researchers from both institutions.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Ahmed, M., Tushar, H., Thandi, N., & Seraj, R. (2023). Privacy-preserving artificial intelligence in healthcare: Techniques and applications. *Computers in Biology and Medicine*, 158, Article 106848.
2. Al Attiq, S., Gehrmann, C., & Dahlén, K. (2024). Vulnerability detection in popular programming languages with language models. *arXiv:2412.15905*.
3. Alkhanbouli, R., Almadhaani, H. M. A., Alhosani, F., & Simsekler, M. C. E. (2025). The role of explainable artificial intelligence in disease prediction: A systematic literature review and future research directions. *BMC Medical Informatics and Decision Making*, 25, Article 110.
4. Chakraborty, S., Krishna, R., Ding, Y., & Ray, B. (2021). Deep learning based vulnerability detection: Are we there yet? *IEEE Transactions on Software Engineering*, 48(9), 3381-3397.
5. Chakravarty, S., & Haque, M. M. (2023). A comprehensive survey of deep learning in software engineering. *Software: Practice and Experience*, 53(10), 1897-1945.
6. Chen, Z., & Monperrus, M. (2021). A literature study of embeddings on source code. *Empirical Software Engineering*, 26(4), 1-35.
7. Cinà, A. E., Grosse, K., Demontis, A., Vascon, S., Zellinger, W., Moser, B. A., Oprea, A. M., Biggio, B., Pelillo, M., & Roli, F. (2023). Wild patterns reloaded: A survey of machine learning security against training data poisoning. *ACM Computing Surveys*, 55(13s).
8. Dantas, P. V., da Silva Jr., W. S., Cordeiro, L. C., & Carvalho, C. B. (2024). A comprehensive review of model compression techniques in machine learning. *Applied Intelligence*, 54, 11804-11844.
9. Deng, M., Wuyts, K., Scandariato, R., Preneel, B., & Joosen, W. (2010). A privacy threat analysis framework: Supporting the elicitation and fulfillment of privacy requirements. *Requirements Engineering*, 16(1), 3-32.
10. Denecke, K., May, R., & Rivera-Romero, O. (2024). Transformer models in healthcare: A survey and thematic analysis of potentials, shortcomings and risks. *Journal of Medical Systems*, 48(1).

11. Dequino, A., Bompani, L., Benini, L., & Conti, F. (2025). Optimizing BFloat16 deployment of tiny transformers on ultra-low-power extreme-edge SoCs. *Journal of Low Power Electronics and Applications*, 15(1), 8.
12. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805*.
13. Ding, Y., Fu, Y., Ibrahim, O., Sitawarin, C., Chen, X., Alomair, B., Wagner, D., Ray, B., & Chen, Y. (2024). Vulnerability detection with code language models: How far are we? *arXiv:2403.18624*.
14. Feng, Z., Guo, D., Tang, D., Duan, N., Feng, X., Gong, M., Shou, L., Qin, B., Liu, T., Jiang, D., & Zhou, M. (2020). CodeBERT: A pre-trained model for programming and natural languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020* (pp. 1536-1547). Association for Computational Linguistics.
15. Guo, D., Ren, S., Lu, S., Feng, Z., Tang, D., Liu, S., Zhou, L., Duan, N., Svyatkovskiy, A., Fu, S., Tufano, M., Deng, S. K., Clement, C., Drain, D., Sundaresan, N., Yin, J., Jiang, D., & Zhou, M. (2021). GraphCodeBERT: Pre-training code representations with data flow. In *International Conference on Learning Representations (ICLR 2021)*.
16. Holmgren, A. J., Everson, J., & Adler-Milstein, J. (2022). Association of Hospital interoperable data sharing with alternative payment model participation. *JAMA Health Forum*, 3(2).
17. Iwaya, L. H., Babar, M. A., Rashid, A., & Wijayarathna, C. (2023). On the privacy of mental health apps: An empirical investigation and its implications for app development. *Empirical Software Engineering*, 28, Article 2.
18. Karimian, G., Petelos, E., & Evers, S. M. (2022). The ethical issues of the application of artificial intelligence in Healthcare: A systematic scoping review. *AI and Ethics*, 2(4), 539-551.
19. Latharani, T. R., & Mouneshachari, S. (2024). Leveraging machine learning for behavioral analysis and mitigation of APT attacks in WSNs. *Journal of Electrical Systems*, 20(11S), 2174-2181.
20. Li, H., Ding, Z., Alowain, L., Chen, Y., & Wagner, D. (2023). DiverseVul: A new vulnerable source code dataset for deep learning based vulnerability detection. *arXiv:2304.00409*.
21. Li, H., Li, C., Wang, J., Yang, A., Ma, Z., Zhang, Z., & Hua, D. (2023). Review on security of Federated Learning and its application in Healthcare. *Future Generation Computer Systems*, 144, 271-290.
22. Lohmann, P. A., Albuquerque, C., & Machado, R. (2023). Systematic literature review of threat modeling concepts [Conference paper]. SCITEPRESS.
23. Luo, Z., Hanrui Yan, & Xueting Pan. (2023). Optimizing transformer models for resource-constrained environments: A study on model compression techniques. *Journal of Computational Methods in Engineering Applications*, 1-12.
24. Malihi, L., & Heidemann, G. (2023). Efficient and controllable model compression through sequential knowledge distillation and pruning. *Big Data and Cognitive Computing*, 7(3), 154.
25. Marey, A., Arjmand, P., Sabe Alerab, A. D., Eslami, M. J., Saad, A. M., Sanchez, N., ... & Umair, M. (2024). Explainability, transparency and black box challenges of AI in radiology: Impact on patient care in cardiovascular radiology. *Egyptian Journal of Radiology and Nuclear Medicine*, 55, Article 183.
26. Olomar, D., et al. (2024). The impact of patient access to Electronic Health Records on Health Care Engagement: Systematic Review. *Journal of Medical Internet Research*, 26.

27. Portela, D., Nogueira-Leite, D., Almeida, R., & Cruz-Correia, R. (2023). Economic impact of a hospital cyberattack in a national health system: Descriptive case study. *JMIR Formative Research*, 7, e41738.
28. Pereira, J. D., Ivaki, N., & Vieira, M. (2021). Characterizing buffer overflow vulnerabilities in large C/C++ projects. *IEEE Access*, 9, 142879–142892. <https://doi.org/10.1109/ACCESS.2021.3120349>
29. Robles-González, A., Parra-Arnau, J., & Forné, J. (2020). A LINDDUN-based framework for privacy threat analysis on identification and authentication processes. *Computers & Security*, 94, Article 101755.
30. Salunkhe, V., et al. (2024). EHR interoperability challenges leveraging HL7 FHIR for seamless data exchange in Healthcare. *Darpan International Research Analysis*, 12(3), 403-419.
31. Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv:1910.01108*.
32. Sheng, Z., Chen, Z., Gu, S., Huang, H., Gu, G., & Huang, J. (2024). LLMs in software security: A survey of vulnerability detection techniques and insights. *arXiv:2502.07049*.
33. Silva, P., Gonçalves, J., Antunes, N., & Vieira, M. (2024). Security and privacy of technologies in health information systems. *Computers*, 13(2), 41.
34. Silvestri, S., Islam, S., Amelin, D., Weiler, G., Papastergiou, S., & Ciampi, M. (2023). Cyber Threat Assessment and management for securing healthcare ecosystems using natural language processing. *International Journal of Information Security*, 23(1), 31-50.
35. Sivan, R., & Zukarnain, Z. A. (2021). Security and privacy in cloud-based E-Health System. *Symmetry*, 13(5), 742.
36. Solanki, P., Grundy, J., & Hussain, W. (2022). Operationalising Ethics in Artificial Intelligence for Healthcare: A framework for AI developers. *AI and Ethics*, 3(1), 223-240.
37. Thabtah, F., Hammoud, S., Kamalov, F., & Gonsalves, A. (2020). Data imbalance in classification: Experimental evaluation. *Information Sciences*, 513, 429-441.
38. Thapa, C., Jang, S. I., Ahmed, M. S., Camtepe, S., Pieprzyk, J., & Nepal, S. (2022). Transformer-based language models for software vulnerability detection. In *Proceedings of the 38th Annual Computer Security Applications Conference* (pp. 481-496).
39. Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2019). GLUE: A multi-task benchmark and analysis platform for natural language understanding. *arXiv:1804.07461*.
40. Wang, X., Hu, R., Gao, C., Wen, X., Chen, Y., & Liao, Q. (2024, April 14). ReposVul: A repository-level high-quality vulnerability dataset. Paper presented at the 472.
41. Wang, Y., Wang, W., Joty, S., & Hoi, S. C. H. (2021). CodeT5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (pp. 8696-8708).
42. Wuyts, K., & Joosen, W. (2020). A LINDDUN-based framework for privacy threat analysis. *Computers & Security*, 94, Article 101755.
43. Zhou, Y., Liu, S., Siow, J., Du, X., & Liu, Y. (2019). Devign: Effective vulnerability identification by learning comprehensive program semantics via graph neural networks. In *Advances in Neural Information Processing Systems*, 32 (pp. 10197-10207).
44. Zouev, E. (2020). *Programming Languages for Safety-Critical Systems*. In: *Software Design for Resilient Computer Systems*. Springer, Cham. https://doi.org/10.1007/978-3-030-21244-5_11