The Journal of Systems and Software 228 (2025) 112469

Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

FedMVA: Enhancing software vulnerability assessment via federated multimodal learning ☆

Qingyun Liu [a], Xiaolin Ju [a,*], Xiang Chen [a,*], Lina Gong [b]

[a] School of Artificial Intelligence and Computer Science, Nantong University, Nantong, 226019, Jiangsu, China
[b] School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, 211106, Jiangsu, China

ARTICLE INFO

Keywords:
Software vulnerability assessment
Federated learning
Multimodal fusion
Privacy-preserving

ABSTRACT

Software Vulnerability Assessment plays a crucial role in identifying and evaluating security vulnerabilities in software systems and prioritizing their resolution. However, as concerns about data privacy and security continue to grow, traditional vulnerability assessment methods struggle to balance effectiveness with privacy protection, particularly in heterogeneous data environments. To address this challenge, we propose a novel federated multimodal vulnerability assessment framework (FedMVA), designed with privacy preservation at its core. FedMVA leverages federated learning, enabling local model training without sharing data, thereby protecting sensitive information while ensuring efficient vulnerability evaluation. Our framework also incorporates multimodal data, including code structure, lexical features, and developer comments, fully utilizing the complementary nature of these modalities. We introduce a weighted variance minimization loss function to improve the alignment between local and global models and adopt a momentum-based weight allocation strategy with a dynamic learning rate mechanism to enhance the model's robustness and adaptability across diverse data environments. Extensive ablation studies demonstrate that FedMVA outperforms existing methods in multiple performance metrics, significantly improving the precision of vulnerability assessment. This work highlights the advantages of integrating multimodal data within a federated learning framework, providing an innovative and promising solution for effective and privacy-preserving vulnerability assessment in complex software systems.

Editor's note: Open Science material was validated by the Journal of Systems and Software Open Science Board.

1. Introduction

Software vulnerabilities pose significant threats to the security and reliability of computer systems, potentially leading to substantial economic losses. Effective automated detection methods (Liu et al., 2024; Lu et al., 2024; Cai et al., 2024) are crucial for mitigating these risks and ensuring the stability of digital ecosystems. Consequently, software vulnerability assessment (SVA) (Le et al., 2022; Dissanayake et al., 2022; Elder et al., 2024) has become an essential part of software development and maintenance, aiming to identify vulnerabilities, accurately assess their severity, and ensure timely remediation. However, due to the complexity of the software and the limited resources for remediation, it is not feasible to address all vulnerabilities immediately. Thus, it is necessary to prioritize the remediation of high-risk vulnerabilities. The Common Vulnerability Scoring System (CVSS) is a widely used assessment framework that generates severity scores based on various vulnerability attributes, helping developers prioritize remediation efforts. However, in handling many vulnerability reports, relying on expert knowledge for CVSS scoring often struggles to keep pace with the rate of new vulnerabilities. Zhou et al. (2021) and Feutrill et al. (2018) found that the time between the initial disclosure of a vulnerability and the update of its CVSS score could span several hundred days, preventing high-risk vulnerabilities from being addressed promptly. Therefore, there is an urgent need for automated assessment tools that assist developers in effectively prioritizing vulnerabilities, reducing response times, and mitigating risks.

Current SVA methods primarily rely on centralized data analysis models, which predict the severity of vulnerabilities using information such as vulnerability descriptions (Han et al., 2017; Sun et al., 2023b; Babalau et al., 2021; Le et al., 2019; Gong et al., 2019), source code (Le and Babar, 2022; Hao et al., 2023), or code commits (Le et al., 2021). Although centralized approaches improve model accuracy and generalization by aggregating data from multiple sources, they

☆ Editor: Nicole Novielli.

* Corresponding authors.
E-mail addresses: Lqy_1213@outlook.com (Q. Liu), ju.xl@ntu.edu.cn (X. Ju), xchencs@ntu.edu.cn (X. Chen), gonglina@nuaa.edu.cn (L. Gong).

https://doi.org/10.1016/j.jss.2025.112469
Received 18 December 2024; Received in revised form 8 April 2025; Accepted 17 April 2025
Available online 2 May 2025

face significant privacy protection and sharing challenges. Yang et al. (2024) pointed out that industrial data, especially source code, is often inaccessible to researchers due to strict privacy policies, significantly limiting the applicability of centralized methods in industrial settings. Furthermore, organizations are reluctant to share sensitive data out of concern for potential data breaches that could harm their commercial interests. These issues make centralized analysis unsuitable for distributed environments, mainly where data privacy and cross-organizational collaboration are critical.

To address these limitations, federated learning has emerged as a promising distributed machine learning paradigm. It enables participants to train models locally and share only model parameters, eliminating the need to transfer raw data. Federated learning effectively addresses the following challenges:

- **Data Privacy Protection** (Khraisat et al., 2024; Xu et al., 2023). Federated learning follows the principle of "data stays local, models move". Each client trains a local model and only shares model updates, reducing privacy risks.
- **Reducing Data Sharing Barriers.** Federatedlearning enables organizations to retain data privacy while leveraging patterns from other participants' data, enhancing model performance and reducing the barriers to data sharing.
- **Enhancing Model Generalizability.** By utilizing data from different organizations, federated learning can learn diverse vulnerability patterns, improving model generalizability, which is essential in vulnerability assessment.

Although federated learning mitigates centralized methods' privacy and data-sharing challenges, traditional SVA approaches often rely on single-modal data, such as vulnerability descriptions or source code. While vulnerability descriptions may not raise privacy concerns, source code and developer annotations can contain sensitive information. Relying on a single-modal approach fails to capture the multifaceted nature of vulnerabilities, limiting the accuracy and comprehensiveness of the assessments. To address this limitation, we propose an innovative federated multimodal vulnerability assessment framework (FedMVA), which combines federated learning with multimodal data fusion. FedMVA integrates lexical features of code, structured graph representations (e.g., control flow graphs Zeng et al., 2023, data flow graphs Guo et al., 2020 and abstract syntax trees Sun et al., 2023a), and developer annotations to exploit the complementary strengths of these modalities fully.

To support this framework, we constructed a high-quality SVA dataset aligned with the CVSS standard, incorporating three modalities: lexical features, structured graph information, and developer annotations. We removed empty lines, leading spaces, and irrelevant comments during data preprocessing to simplify the code structure (as detailed in Section 4.2). Experimental results demonstrate that FedMVA achieves a strong balance between privacy protection and assessment performance, offering an innovative and promising solution for practical vulnerability assessment in complex software systems.

We validated the effectiveness of FedMVA through systematic comparisons with several state-of-the-art baseline methods. These baseline methods include single-modal and multimodal models, representing mainstream vulnerability assessment approaches. Experimental results demonstrate that FedMVA achieves superior performance across all key metrics, including accuracy, recall, and F1-score, and consistently improves in various evaluation scenarios. To further examine the contributions of individual components, we conducted ablation studies by progressively removing the federated learning framework, multimodal inputs, or the feature fusion module. The results confirm that each component is crucial to the overall performance. Specifically, the federated learning framework ensures data privacy and enhances the model's generalization capabilities. The multimodal inputs, which integrate code structure, lexical features, and developer annotations, enable the model to capture vulnerability characteristics from multiple dimensions. Moreover, the feature fusion strategy effectively leverages the complementary strengths of these modalities, significantly improving assessment accuracy. The novelty and contributions of this study can be summarized as follows:

- **Methodology.** We propose the first SVA framework integrating tri-modal data fusion with federated learning. FedMVA leverages federated learning models and incorporates tri-modal information with a tailored fusion strategy to enhance SVA performance.
- **Dataset.** We constructed a high-quality SVA dataset that integrates structured information, lexical features, and annotations from source code. This dataset provides a robust foundation for multimodal research.
- **Evaluation.** We evaluated the proposed framework on the constructed dataset, demonstrating that FedMVA outperforms state-of-the-art SVA baseline models, with improvements of at least 9.19%, 8.66%, and 24.75% in accuracy, F1 score, and MCC, respectively. Ablation studies further validate the effectiveness of our customized FedMVA framework.

We share data, code, and detailed results at our project home to encourage follow-up studies for applying decentralized learning to software vulnerability assessment: https://github.com/Liuqy1213/FedMVA.

The remainder of this paper is organized as follows: Section 2 introduces the background and motivation for this study, focusing on Software Vulnerability Assessment (SVA) and federated learning. Section 3 details the proposed FedMVA framework and its components. Section 4 describes the experimental setup, including datasets, performance metrics, and baseline methods employed in this research. Section 5 reports the study results, accompanied by an analysis of the formulated research questions. Section 6 discusses key findings, their implications, and the limitations of the proposed approach. Section 7 reviews the related literature, emphasizing the novelty and contributions of this study. Finally, Section 8 concludes the research by summarizing the findings and proposing potential directions for future work.

2. Background

2.1. Software vulnerability assessment

As software systems become increasingly complex and interconnected, the number and complexity of software vulnerabilities (Le et al., 2022) have risen dramatically, posing unprecedented challenges to system security. To address these challenges, Software Vulnerability Assessment (SVA) has become a core tool for ensuring system security. SVA enables developers and security experts to optimize resource allocation by identifying, assessing, and prioritizing vulnerabilities, thereby reducing the risk of system attacks. Currently, SVA largely relies on publicly available database resources, such as the National Vulnerability Database (NVD)[1] and the Common Vulnerabilities and Exposures (CVE) repository. These databases provide detailed descriptions of vulnerabilities and use the Common Vulnerability Scoring System (CVSS)[2] to rate the severity of vulnerabilities, aiding developers and security teams in determining remediation priorities. CVSS is among the most widely applied frameworks in SVA, offering standardized evaluation criteria for developers by quantifying multiple vulnerability characteristics.

CVSS comprises three main components: Base, Temporal, and Environmental Metrics. The Base Metrics evaluate the intrinsic characteristics of a vulnerability and are regarded as the most critical part

[1] https://nvd.nist.gov/.
[2] https://www.first.org/cvss/.

of the scoring process. Temporal Metrics reflect changes in vulnerability characteristics over time, such as the availability of exploit code or patches. Environmental Metrics allow organizations to customize vulnerability assessments based on their specific operational contexts. Although these metrics provide a comprehensive perspective for assessing vulnerabilities, in practice, data for Temporal and Environmental Metrics are often difficult to extract directly from source code or textual descriptions. Consequently, most studies and applications primarily focus on Base Metrics. The Base Metrics of CVSS version 2 (CVSS v2) include seven key indicators: Access Vector, Access Complexity, Authentication, Confidentiality Impact, Integrity Impact, Availability Impact, and Exploitability. These indicators collectively determine the severity of a vulnerability, which is ranked into three levels: Low (0.1–3.9), Medium (4.0–6.9), and High (7.0–10.0). These categories assist developers and security teams in effectively prioritizing remediation efforts, enabling them to allocate limited resources to address the most critical vulnerabilities.

2.2. Federated learning model

With the widespread adoption of distributed systems and cross-organizational collaboration, achieving effective collaborative model training while safeguarding data privacy has become a critical challenge. To address this, Google introduced Federated Learning (FL) (Hanzely and Richtárik, 2020; Zhuo et al., 2019; Yu et al., 2020) in 2016 as an advanced distributed learning method designed to enable secure model training across devices without transferring raw data. FL facilitates local model training on multiple clients, sharing only model parameters to prevent data leakage. Due to its robust capabilities, FL has been widely adopted in fields such as autonomous driving, facial recognition, and system anomaly detection, attracting significant attention (Zhang et al., 2021) from both academia and industry for its potential applications.

However, the standard Federated Learning approach, FedAvg (McMahan et al., 2017), faces significant challenges when dealing with data heterogeneity (non-IID data) (Ma et al., 2022) across clients. This often leads to reduced generalization ability of the global model, making it challenging to meet the specific needs of individual participants. To address this limitation, Personalized Federated Learning (PFL) (Tan et al., 2022) was introduced. PFL aims to generate personalized models for each client by combining global collaborative learning with client-specific adaptation, thus better accommodating the unique data distributions of individual clients. Most PFL approaches adopt parameter decoupling, dividing the model into two components: a shared feature extractor and a client-specific classifier. For instance, the FedPer method (Arivazhagan et al., 2019) retains client-specific classifiers while sharing a common feature extractor, striking a balance between global knowledge sharing and local adaptation to enhance model personalization significantly.

Recent studies have highlighted the potential of Federated Learning in software engineering tasks (Li et al., 2020b; Yang et al., 2019; Abyane et al., 2023), including code analysis, requirements prediction, debugging, and refactoring. Yang et al. (2024) proposed a data aggregation strategy that dynamically adjusts based on data scale, class balance, and the learnability of minority classes, focusing on tasks including code clone detection and defect prediction to improve model performance on imbalanced and sensitive industrial datasets. Zhang et al. (2024) introduced a horizontally federated learning framework for vulnerability detection, addressing privacy concerns in multi-party data sharing while enhancing detection accuracy. Similarly, Yamamoto et al. (2023) developed a federated cross-project defect prediction model using logistic regression for distributed training, ensuring project data privacy by avoiding raw data sharing. Building on these advancements, our study develops a federated vulnerability assessment method based on FedPer. By integrating code structure, lexical features and comments, our approach offers a comprehensive capture of vulnerability characteristics, thereby enhancing the accuracy and efficiency of vulnerability assessment.

3. Methodology

The overall framework of our proposed FedMVA is presented in Fig. 1, which consists of three main stages: **data preprocessing stage**, **federated training and optimization phase**, and **vulnerability assessment stage**. ①In the data preprocessing stage, we constructed a high-quality software vulnerability assessment (SVA) dataset comprising source code, lexical information, and comment information. Graph-structured information was extracted to capture the syntax and dependency relationships of the code, while lexical and comment information was treated as textual data inputs. These were processed using the pre-trained model CodeT5 (Wang et al., 2021) to extract semantic features, which were then combined with label information to generate a unified multi-modal representation. ②In the federated training and optimization phase of the proposed FedMVA framework, we enhanced tri-modal data integration through self-attention weighted fusion. We introduced three key improvements, including a weighted variance minimization loss to align global and local models, a momentum-based weight allocation strategy to optimize client contributions, and a dynamic learning rate mechanism to handle heterogeneous data, all of which improve the robustness and accuracy of the global model. ③In the vulnerability assessment stage, we leveraged the complementary strengths of the tri-modal information, significantly improving the model's evaluation performance and practical applicability.

3.1. Data preprocessing stage

We preprocessed the data at this stage to construct a high-quality dataset suitable for software vulnerability assessment. Based on the federated learning framework, we emphasize data distribution and privacy protection. To this end, we applied de-identification techniques (Youm, 2020) to ensure sensitive information remained secure. We partitioned the data based on the code sources of different clients, simulating the real-world scenario of distributed data storage. To address the issue of data heterogeneity (Ye et al., 2023), we introduced a standardization step to unify data of varying formats into a structured representation. Furthermore, to ensure consistency and enhance model performance, each data instance was assigned a three-class label based on the CVSS v2 standard, reflecting the diversity of vulnerability types. This approach satisfies the decentralized learning framework's privacy protection and data decentralization requirements and provides high-quality input data for multimodal feature fusion and model training.

The primary objective of this phase is to map the lexical information, comments, and structural information of the code into a continuous vector space, enabling unified representation to support subsequent analysis and comparison. While both lexical information and comment information are textual in nature, they serve distinct roles in software vulnerability assessment. Lexical information refers to tokenized representations of source code, capturing syntactic patterns and function structures, whereas comment information consists of natural language annotations that provide additional explanations written by developers. Unlike lexical data, which directly reflects the structural composition of the code, comments often introduce domain-specific context but may not follow strict syntactic rules. To preserve their unique characteristics, we adopt different processing strategies. Lexical information is first tokenized and normalized to extract syntactic and symbolic features, ensuring consistency in representation. It is then processed using a pre-trained CodeT5 to extract semantic embeddings. Conversely, comment information is directly fed into CodeT5 as raw natural language text to capture its contextual meaning. By processing these inputs separately, we ensure that lexical and comment information contribute complementary yet distinct insights into the vulnerability assessment process.

In addition to textual representations, structural properties of the code also play a crucial role in vulnerability assessment. During the

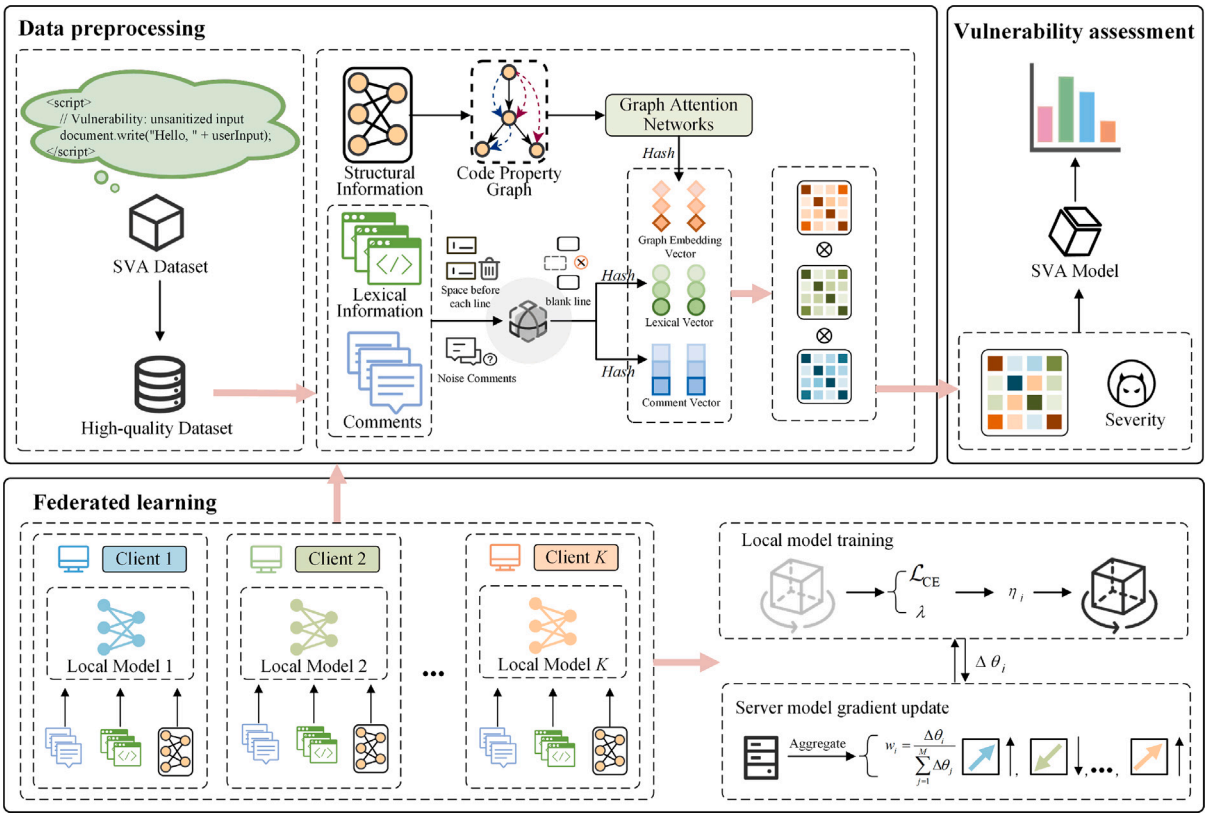


Fig. 1. The architecture of FedMVA, which mainly contains three stages: (A) Data preprocessing module; (B) Federated training model; (C) Vulnerability assessment module.

graph construction step, we use Joern[3] to generate the Code Property Graph (CPG) (Suneja et al., 2020), which encodes both syntactic and control-flow dependencies in the code. To effectively utilize these structural features, we employ a Graph Neural Network (GNN) (Ruiz et al., 2020) to aggregate information from adjacent nodes in the CPG. Additionally, we leverage a convolutional neural network (CNN) to extract global structural representations of the code, complementing the local feature aggregation of the GNN. Finally, we employ binary hash encoding to align multimodal features, establishing an initial semantic unification. We then refine feature representations through a self-attention mechanism, enabling deep fusion and enhancing cross-modal interactions. This mechanism dynamically adjusts the contribution of each modality, effectively mitigating conflicts and inconsistencies among text, code, and comments. By prioritizing the most relevant modalities while suppressing noise from less informative ones, each data source meaningfully contributes to the overall evaluation.

We first use Joern to generate the CPG, which captures the syntactic structures and dependencies within the code. Specifically, each node v_i in the CPG represents a program element in the code (such as a function, variable, or statement), and an edge (v_i, v_j) represents the dependency relationship between node v_i and node v_j. Next, we use GNN to aggregate information from neighboring nodes, extracting local syntactic features. The following equation gives the feature update for each node:

$$\mathrm{h}_i^{(k+1)} = \mathrm{AGGREGATE}\left(\{\mathbf{h}_j^{(k)} | v_j \in \mathcal{N}(v_i)\}\right) + \mathbf{h}_i^{(k)}, \tag{1}$$

where $h_i^{(k)}$ represents the feature of node v_i at the kth iteration, $\mathcal{N}(v_i)$ is the set of neighboring nodes of v_i, and AGGREGATE$(\cdot)$ is the aggregation operation (such as sum, average, etc.). To further capture global structural information, we integrate CNNs on top of the GNN-aggregated node features. CNNs are utilized to model the global syntactic structure of the code by processing the local features extracted by GNNs. Simultaneously, lexical and comment information from the code is treated as text data and processed by the pre-trained Transformer model CodeT5 to extract semantic features. To align the features from different modalities, we use binary hash encoding to reduce semantic discrepancies between the graph structure, lexical features, and comments. This alignment step ensures that the modalities are compatible and can be effectively fused. Finally, we use a self-attention mechanism to perform weighted fusion of the features from all modalities, which is mathematically represented as:

$$\mathbf{h}_i^{\text{final}} = \mathrm{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}, \tag{2}$$

where $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$ are the query, key, and value vectors for the graph structure and other modalities. The final feature representation $\mathbf{h}_i^{\text{final}}$ is obtained by applying the self-attention mechanism. The self-attention mechanism dynamically adjusts the contribution of each modality based on its relevance to the task, ensuring that conflicts or inconsistencies are resolved and that each modality contributes meaningfully to the final representation. Our ablation experiments support this process, demonstrating that the fusion of all three modalities (code structure, lexical information, and comments) significantly improves model performance compared to single or dual-modal inputs.

To enhance the model's ability to concentrate on essential information and minimize extraneous noise, we implemented a data preprocessing step that removed redundant comments, blank lines, and other irrelevant elements. Essential code information was preserved, ensuring that the functionality and integrity of the code remained unaffected. These optimizations significantly improved data quality and relevance, laying a solid foundation for subsequent model training.

In the federated learning scenario, the original SVA dataset was randomly split into training and testing sets at an 8:2 ratio, ensuring

[3] https://github.com/joernio/joern

both sets contained vulnerability samples for reliable model evaluation. To eliminate errors caused by random experiments, we conducted 10 repeated trials during the federated learning and vulnerability assessment phases, using the average results as the final performance metrics. This approach ensured the model's stability and generalization ability in a continual learning setting.

3.2. Federated training and optimization phase

During the training phase of the proposed FedMVA framework, our goal is to enhance the performance and adaptability of the global model in heterogeneous data environments. Federated learning, as implemented in FedMVA, is built on the foundational principles of federated learning, where multiple clients collaboratively train a global model without sharing their local data. Each client trains the model using its local data and only sends model updates (such as gradients or weight updates) to the server, aggregating the updates from all clients to update the global model. This approach ensures data privacy while enabling the global model to learn from diverse data distributions across clients. In FedMVA, we further enhance this process by addressing key challenges such as data heterogeneity and model alignment gaps, which are critical for effective federated learning. To address these challenges, we propose three key enhancements based on the traditional FedPer (Arivazhagan et al., 2019) method: First, we introduce a weighted variance minimization loss function to optimize the consistency between the global and local models, reducing the alignment gap. Second, we design a momentum-based weight allocation strategy to dynamically adjust the significance of clients, balancing their influence on the global model. Finally, we incorporate a dynamic learning rate mechanism to improve clients' adaptability to heterogeneous data. These enhancements are grounded in the principles of federated learning, ensuring that the global model not only integrates knowledge from diverse clients but also maintains stability and generalization in heterogeneous environments.

In federated training, the global model must integrate client updates with diverse data distributions. However, the heterogeneity of client data often results in significant alignment gaps between the global and local models. To mitigate this issue, we extend the standard cross-entropy loss by incorporating a weighted variance minimization term to enhance consistency across all clients. The proposed loss function is defined as:

$$\mathrm{L} = \mathcal{L}_{\mathrm{CE}} + \lambda \cdot \sum_{i=1}^{M} w_i \|\theta_{\mathrm{global}} - \theta_{\mathrm{local},i}\|_2^2, \tag{3}$$

here $\mathcal{L}_{\mathrm{CE}}$ is the cross-entropy loss used for classification tasks, and λ is a regularization coefficient balancing the trade-off between classification accuracy and alignment consistency. The dynamic client weight w_i captures the relative contribution of client i to the global model and is computed as:

$$\mathrm{w}_i = \frac{\Delta\theta_i}{\sum_{j=1}^{M} \Delta\theta_j}, \tag{4}$$

where $\Delta\theta_i$ measures the parameter update magnitude of client i. By prioritizing clients with larger updates, this mechanism ensures that critical clients with unique data distributions play a more significant role in global model optimization, enhancing its robustness and stability.

Second, we introduce a momentum-based weight adjustment strategy to balance the influence of clients on the global model. This strategy combines historical weights with current updates to ensure smooth and dynamic weight adjustments. The updated formula is given as:

$$\mathrm{w}_i^{(t+1)} = \beta \cdot w_i^{(t)} + (1-\beta) \cdot \frac{\Delta\theta_i^{(t)}}{\sum_{j=1}^{M} \Delta\theta_j^{(t)}}, \tag{5}$$

where β is the momentum coefficient (typically between 0.8 and 0.9), and $\Delta\theta_i^{(t)}$ represents the parameter update magnitude of client i at iteration t. This approach ensures smoother weight changes while emphasizing clients with substantial contributions.

Finally, we propose a dynamic learning rate mechanism to adapt learning rates to client-specific data distributions. The learning rate is updated as:

$$\eta_i = \eta_0 \cdot \left(1 + \frac{\|\mathbf{h}_i^{\mathrm{global}} - \mathbf{h}_i^{\mathrm{local}}\|_2}{\max_{j\in[1,M]} \|\mathbf{h}_j^{\mathrm{global}} - \mathbf{h}_j^{\mathrm{local}}\|_2}\right), \tag{6}$$

here η_0 is the base learning rate, and $\mathbf{h}_i^{\mathrm{global}}$ and $\mathbf{h}_i^{\mathrm{local}}$ denote the global and local feature representations of client i. Clients with greater discrepancies receive higher learning rates to accelerate convergence, while clients with smaller discrepancies maintain lower rates to ensure stability.

These three improvements collectively address the challenges of federated environments, demonstrating improvements in training efficiency and model generalization. Experimental results show that the FedMVA framework outperforms traditional methods in evaluation performance.

3.3. Vulnerability assessment stage

The vulnerability assessment stage represents the final step in the FedMVA framework, focusing on utilizing the aligned and fused tri-modal features — graph structure, lexical, and comment information — produced during the earlier stages for precise vulnerability classification. Building on the outcomes of the federated training and feature fusion phases, this stage ensures that the processed multi-modal representations are effectively employed to classify vulnerabilities into three severity levels: high, medium, and low.

In this stage, the tri-modal features, aligned and fused during the previous phases, are input into a multi-layer perceptron (MLP) classifier designed for three-class classification. The classifier leverages the combined strengths of the structural, semantic, and contextual information captured in the tri-modal data. A key distinguishing feature of this stage is its ability to refine and exploit the complementary relationships among the modalities. By integrating the optimized feature representations from the training stage with the model's dynamic adaptation capabilities, the assessment module achieves a robust balance between precision and generalizability. The enhanced attention mechanism introduced earlier plays a critical role here, dynamically focusing on the most relevant modal contributions for accurate predictions.

Through this comprehensive approach, the vulnerability assessment stage not only ensures reliable classification performance across diverse data distributions but also highlights the practical applicability of the FedMVA framework in real-world software vulnerability evaluation tasks. This stage demonstrates how the unified representations and optimized model weights generated in the earlier phases translate into tangible improvements in the model's predictive power and reliability.

4. Experimental setup

In this section, we first present the research questions we designed along with their underlying motivations. Then, we introduce the experimental subjects, performance evaluation metrics, baseline methods, and detailed experimental settings.

4.1. Research questions

To evaluate the effectiveness of our proposed FedMVA, we design the following five research questions (RQs):

RQ1: How does FedMVA perform compared to state-of-the-art baselines in SVA?

Motivation: The purpose of RQ1 is to demonstrate the competitiveness of our proposed method FedMVA compared to current state-of-the-art SVA baselines. Additionally, to ensure a comprehensive evaluation,

we considered five automatic evaluation metrics: accuracy, precision, recall, F1 score, and MCC. These metrics provide a holistic measure of performance across various aspects of vulnerability assessment.

RQ2: How do different input modality combinations impact the performance of FedMVA?

Motivation: Our FedMVA method incorporates capturing the structural information of code by generating its CPG and combining it with lexical information and comment information to create tri-modal inputs for SVA model training. Therefore, in RQ2, we aim to investigate whether this tri-modal input design achieves optimal performance for FedMVA. Additionally, we aim to compare different modality configurations to determine which input modality contributes most significantly to the performance of FedMVA.

RQ3: How does federated learning influence the effectiveness of SVA compared to conventional training paradigms?

Motivation: In RQ3, we aim to analyze the impact of federated learning on the effectiveness of SVA compared to conventional training paradigms. Federated learning introduces both benefits and challenges: while it enhances privacy preservation and allows federated collaboration, it also faces issues such as client heterogeneity and communication overhead. To systematically evaluate these factors, we conduct ablation studies to compare different FL aggregation strategies, assess their influence on model performance, and discuss the trade-offs introduced by federated learning in software vulnerability assessment.

RQ4: How do different federated learning settings affect the performance of FedMVA?

Motivation: Existing federated learning methods have limitations in dealing with data heterogeneity and customer importance differences, which can lead to poor model fitting. To address these challenges, we reduce model variance by minimizing weighted variance, using momentum-based updates to highlight key customers, and using dynamic learning rate strategies to accelerate convergence, enhancing model robustness and overall performance. This RQ will analyze whether this adjustment can help FedMVA achieve the best performance.

RQ5: How does the proposed multimodal fusion strategy impact the performance of FedMVA and the interaction across modalities?

Motivation: In this RQ, we aim to explore the impact of the multi-modal fusion strategy on FedMVA's performance and cross-modal interactions. Multi-modal information is critical in vulnerability assessment; however, existing methods often fall short of effectively fusing modal features and facilitating information exchange across modalities. To address these shortcomings, we employ a self-attention mechanism for the weighted fusion of modal features to enhance cross-modal interactions and improve overall model performance.

4.2. Experimental subject

To construct a high-quality SVA dataset, we aggregated data from multiple sources, including the CVE database and associated Git repositories. We pre-processed data by removing blank lines, leading spaces, and irrelevant comments to simplify the code structure. The resulting dataset comprises 9,941 samples, each containing tri-modal information: structural information from CPG, lexical features, and comment information. Compared to existing datasets such as SARD (SARD, 2020), Devign (Zhou et al., 2019), and BigVul (Fan et al., 2020), our dataset offers significant advantages in multi-modal characteristics. It strictly adheres to the CVSS v2 standard, providing a robust foundation for advanced SVA research.

During the data preprocessing stage, we applied de-identification techniques to safeguard sensitive information and removed irrelevant elements, such as blank lines and excessive comments, to optimize model performance. This process not only enhanced data privacy but also improved the robustness of the dataset, making it suitable for federated learning applications. In the data processing phase, we used Joern to extract CPGs, capturing the source code's syntactic structures and dependency relationships. Additionally, lexical and comment information was treated as textual data and processed using the pre-trained CodeT5 model to generate semantic embeddings. Each data sample was ultimately labeled with one of three severity levels — low (0), medium (1), or high (2) — according to the CVSS v2 standard, ensuring consistent and reliable labeling. The final dataset includes 2,138 low-severity samples (21.5%), 3,587 medium-severity samples (36.1%), and 4,216 high-severity samples (42.4%).

Table 1
A tri-modal example including CPG information, lexical information, comment information, and severity.

Code Property Graph (CPG):
$Node1$: **Function** *process_request*
$Node2$: **Variable** *buffer[128]*
$Edge1$: *Defines* $\rightarrow$ *Node2*
$Edge2$: *Calls* $\rightarrow$ *process()*
...

Lexical Information:
```
static void process_request(Request *req)
{
  char buffer[128];
  if(req->size > 128){
    return;
  }
  memcpy(buffer, req->data, req->size);
  process(buffer);
}
```

Comment Information:
```
/* Functiontoprocessrequestssafely */
/* Ensureinputsizedoesnotexceedbufferlimit */
/* memcpyisusedforcopyingdataintobuffer */
```

Severity: 2 (High)

By integrating structural, lexical, and semantic features, this tri-modal dataset provides a comprehensive foundation for the FedMVA framework, enabling in-depth and reliable vulnerability assessments. Table 1 illustrates a tri-modal data sample, including its CPG, lexical information, comment information, and severity label. This example highlights the composition of the tri-modal data and its potential value in vulnerability evaluation tasks.

4.3. Baseline methods

To validate the effectiveness of our proposed FedMVA framework, we compared it against six representative baselines. These baselines encompass a range of mainstream methods, from feature-level processing to model design, representing the forefront of research in the SVA domain. We briefly describe these baseline methods as follows.

CWM (Character-Word Model) (Le et al., 2019) addresses the challenge of concept drift in SVA by combining character and word features from vulnerability descriptions. Depending on the classifier used, CWM is further divided into three baselines:

- CWM_{NB}. Employing the Naïve Bayes (NB) (Russell and Norvig, 2016) classifier, this approach is rooted in Bayesian decision theory. NB assumes conditional independence among features, which may limit performance when the assumption is violated. Its simplicity and efficiency make NB well-suited for large-scale classification tasks. No hyperparameter optimization was performed for this baseline.
- CWM_{SVM}. Utilizing the Support Vector Machine (Cortes, 1995) classifier, this baseline maps input features into a high-dimensional space and identifies the optimal hyperplane to separate different classes. Regularization parameters were fine-tuned within a predefined range to enhance classification performance.
- CWM_{XGB}. This baseline uses Extreme Gradient Boosting (XGB) (Chen, 2015), an ensemble learning method that integrates multiple tree models to form a robust classifier. Key hyperparameters were optimized to improve accuracy, including the number of trees, tree depth, and leaf nodes.

Fun (Function-Level SVA) (Le and Babar, 2022) focuses on function-level vulnerability assessment by analyzing contextual information surrounding vulnerable statements. It extracts features from code to develop robust models for SVA. Depending on the classifier used, Fun is divided into two baselines:

- Fun_{RF}. This baseline employs Random Forest (RF) (Ho, 1995), an ensemble model combining multiple decision trees. Predictions are finalized through majority voting, enhancing robustness and predictive accuracy. Key hyperparameters were carefully tuned, such as the number of trees, tree depth, and leaf nodes.
- Fun_{LGBM}. Leveraging the Light Gradient Boosting Machine (LGBM) (Ke et al., 2017) classifier, this method is characterized by high efficiency and scalability. LGBM shares similar hyperparameter structures with FuncRF but incorporates additional optimizations to handle large-scale data.

DeepCVA (Deep Commit-Level SVA) (Le et al., 2021) is a deep learning framework for commit-level vulnerability assessment, originally designed for multi-task learning to predict multiple CVSS metrics such as confidentiality, integrity, and availability. To align with our evaluation framework, we adapted DeepCVA into a three-class classification model based on the CVSS v2 standard, reformulating its outputs to match our single-task learning setting. We simplified its feature extraction by retaining only code change features, excluding repository-specific metadata, and ensuring consistency across all baselines. These modifications allow DeepCVA to be fairly evaluated in the same classification setting while preserving its commit-based assessment capability.

CodeBERT (Sahar et al., 2024) is a bi-modal pre-trained transformer model designed for natural language and programming language understanding. It is trained using a masked language modeling objective and a replaced token detection task, enabling it to learn rich semantic representations of source code. In our evaluation, we utilize the pre-trained CodeBERT model to extract function-level code representations and assess its effectiveness in vulnerability identification.

CodeT5 (Wang et al., 2021) is a pre-trained encoder–decoder model tailored for code-related tasks, including code generation, completion, and translation. Unlike CodeBERT, which primarily focuses on code understanding, CodeT5 captures syntactic structures and semantic relationships within source code.

By comparing FedMVA with these baseline methods, we aim to comprehensively demonstrate its competitiveness and advantages in multi-modal vulnerability assessment. To ensure fairness, all baselines were trained and evaluated under identical experimental conditions and dataset splits.

4.4. Performance metrics

To comprehensively evaluate the performance of our proposed FedMVA framework, we employ five commonly used metrics: Accuracy (ACC), Precision (PR), Recall (RC), F1 Score (F1), and Matthews Correlation Coefficient (MCC). These metrics collectively provide a multidimensional evaluation, covering overall classification accuracy and positive and negative class prediction capabilities. These metrics were chosen because they capture different aspects of model performance, particularly in vulnerability assessment, where class imbalance and varying severity levels are common.

For the SVA task, the first four metrics are standard, while MCC is widely used for datasets with class imbalance. Since we need to predict three severity levels (low, medium, and high), we use macro-averaged metrics to present the final results. In the rest of the subsection, we show the details of calculating these performance metrics.

- TP (True Positive): The number of positive class samples correctly classified. The SVA task represents cases where the model accurately identifies different severity levels.
- TN (True Negative): The number of negative class samples correctly classified. For each severity level, it represents samples correctly classified as not belonging to that severity level.
- FN (False Negative): The number of positive class samples incorrectly classified as negative. Each severity level represents samples wrongly classified as not belonging to that severity level.
- FP (False Positive): The number of negative class samples incorrectly classified as positive. Each severity level represents samples wrongly classified as belonging to that severity level.

For each severity category i, these statistics can be represented as TP_i, TN_i, FN_i, and FP_i.

Accuracy: Accuracy represents the proportion of correctly classified samples among all samples, calculated as:

$$Accuracy = \frac{\sum_i TP_i + \sum_i TN_i}{\sum_i (TP_i + TN_i + FP_i + FN_i)}. \tag{7}$$

Precision: Precision indicates the proportion of samples predicted as positive that are truly positive. The macro-average precision is calculated as:

$$Precision_i = \frac{TP_i}{TP_i + FP_i}, \tag{8}$$

$$Precision_{macro} = \frac{1}{k} \sum_{i=1}^{k} \text{Precision}_i. \tag{9}$$

Recall: Recall measures the proportion of actual positive samples correctly predicted as positive. The macro-average recall is calculated as:

$$Recall_i = \frac{TP_i}{TP_i + FN_i}, \tag{10}$$

$$Recall_{macro} = \frac{1}{k} \sum_{i=1}^{k} \text{Recall}_i. \tag{11}$$

F1-score: F1-score is the harmonic mean of Precision and Recall. The macro-average F1 Score is calculated as:

$$F1\text{-}score = 2 \times \frac{Precision_i \times Recall_i}{Precision_i + Recall_i}, \tag{12}$$

$$F1\text{-}score_{macro} = \frac{1}{N} \sum_i F1_i. \tag{13}$$

MCC: MCC considers TP, TN, FP, and FN and is effective for datasets with the class imbalance problem. For the multi-class classification problem, we calculate the macro average of each class's MCC as follows.

$$MCC_i = \frac{TP_i \cdot TN_i - FP_i \cdot FN_i}{\sqrt{(TP_i + FP_i)(TP_i + FN_i)(TN_i + FP_i)(TN_i + FN_i)}}, \tag{14}$$

$$MCC_{macro} = \frac{1}{k} \sum_{i=1}^{k} \text{MCC}_i. \tag{15}$$

In the experiments, we conducted multiple independent runs for each baseline method and the FedMVA framework to ensure the stability of the results. The results of the five metrics were calculated on the test set, and the final scores were averaged over all runs. This design ensures a reliable evaluation of the FedMVA framework's competitiveness in multi-modal-based vulnerability assessment and provides a comprehensive analysis of its performance.

4.5. Implementation details

Our experiments and baseline methods were conducted on a computer with a 3.50 GHz CPU and a GeForce RTX 4090 GPU with 24 GB of graphics memory running on Windows 10.

In our study, we maintained consistent experimental settings across all methods to ensure a fair comparison. The initial learning rate was set to 5e−5 and progressively reduced using a linear decay strategy

Table 2
Performance comparison between FedMVA and SVA baselines, with the best results for each metric highlighted in bold.

Method	Accuracy	Precision	Recall	F1	MCC
CWM_{NB}	0.654	0.683	0.547	0.515	0.501
CWM_{SVM}	0.691	0.627	0.577	0.551	0.465
CWM_{XGB}	0.703	0.693	0.639	0.635	0.492
Fun_{RF}	0.672	0.645	0.508	0.549	0.424
Fun_{LGBM}	0.718	0.707	0.526	0.541	0.484
DeepCVA	0.67	0.575	0.572	0.583	0.473
CodeBERT	0.721	0.69	0.621	0.634	0.523
CodeT5	0.745	0.716	0.672	**0.692**	0.573
FedMVA	**0.784**	**0.739**	**0.712**	0.690	**0.625**

Table 3
Class-specific metrics and confusion matrix for the FedMVA model.

(a) Class-wise performance metrics of FedMVA

Class	Precision	Recall	F1
Low	0.672	0.633	0.652
Medium	0.721	0.693	0.707
High	0.805	0.774	0.789

(b) Confusion matrix of FedMVA on the test set

Actual\Pred	Low	Med	High
Low	412	153	53
Medium	85	867	137
High	28	136	983

throughout training. A batch size of 16 was adopted to strike a balance between computational efficiency and model stability. The AdamW optimizer was employed, with a weight decay parameter of 1e−2, to reduce the risk of overfitting. Each training round consisted of 50 iterations, providing sufficient learning opportunities for the models while maintaining reasonable computational demands.

For the federated learning phase, we conducted five communication rounds, during which global model parameters were aggregated and updated on the server after each round. This setup preserved data privacy at the client level and ensured high training efficiency in federated learning scenarios.

To guarantee the reliability of our results, we conducted 10 independent runs for both the FedMVA framework and baseline methods. The performance metrics for each method were calculated based on the test set, and the final results were averaged across all runs. This design allowed us to robustly evaluate the competitive performance of the FedMVA framework in multi-modal vulnerability assessment tasks.

5. Experimental results

5.1. RQ1: How does FedMVA perform compared to state-of-the-art baselines in SVA?

Approach. To evaluate the effectiveness of our proposed method FedMVA for the SVA task, we selected a diverse set of baseline methods, including CWM_{NB}, CWM_{SVM}, CWM_{XGB} (Le et al., 2019; Russell and Norvig, 2016; Cortes, 1995; Chen, 2015), Fun_{RF}, Fun_{LGBM} (Le and Babar, 2022; Ho, 1995; Ke et al., 2017), DeepCVA (Le et al., 2021), CodeBERT (Sahar et al., 2024), and CodeT5 (Wang et al., 2021). These baselines represent state-of-the-art methods in SVA, spanning different approaches ranging from traditional machine learning to deep learning-based models.

For FedMVA, we followed the experimental setup outlined in Section 4.5 and assessed the method's performance using the evaluation metrics introduced in Section 4.4. These methods represent a variety of approaches in SVA and utilize different data modalities to extract vulnerability information. The CWM category is based on vulnerability descriptions, extracting character- and word-level textual features, making it suitable for scenarios with comprehensive vulnerability reports. The Fun category focuses on function-level source code analysis, capturing patterns within vulnerable functions and their context. DeepCVA is designed for commit-level vulnerability assessment, integrating contextual information from code changes and metadata to predict vulnerability properties through a multi-task learning framework. CodeBERT is designed for code understanding tasks, leveraging a masked language modeling objective to learn representations from source code and paired natural language descriptions. And CodeT5 extends beyond code understanding by incorporating generative capabilities, capturing both syntactic structures and semantic relationships within code. In this study, we apply both models directly to function-level vulnerability classification to assess their effectiveness in the SVA task.

Result. The performance comparison between our proposed method FedMVA and the baselines is shown in Table 2, with the best results for each metric highlighted in bold. It is clear from the table that our method outperforms all baselines. Specifically, compared to the baselines, FedMVA demonstrates improvements in MCC ranging from 9.08% to 47.41%, F1-score from −0.28% to 33.98%, accuracy from 5.96% to 19.88%, precision from 3.21% to 28.52%, and recall from 5.95% to 40.16%. These results demonstrate the advantages of FedMVA in capturing multi-faceted vulnerability characteristics that are often overlooked in single-modality baselines. Traditional statistical learning approaches, such as CWM and Fun, which rely solely on handcrafted features, struggle to model the complex structural and contextual dependencies in source code, leading to lower performance. DeepCVA shows limited performance gains due to its reliance on commit-level information, which lacks structural code representations and lexical details. CodeBERT achieves notable improvements over traditional baselines, benefiting from its pre-trained contextual embeddings. However, its bi-modal nature (code + natural language) does not fully exploit the structural properties of code, resulting in lower recall compared to FedMVA. CodeT5, designed for code generation and understanding, performs better than CodeBERT as it captures richer syntactic and semantic relationships. CodeT5 achieves a slightly higher F1-score than FedMVA, likely due to its token-level modeling approach excels at capturing localized vulnerability patterns, leading to a more balanced trade-off between precision and recall.

In contrast, FedMVA integrates structural, lexical, and comment-based features through a self-attention fusion mechanism, allowing it to extract complementary information from multiple modalities dynamically. This approach improves model robustness and generalization, particularly in scenarios with heterogeneous data distributions. Moreover, incorporating momentum mechanisms and dynamic learning rates enables FedMVA to better adapt to non-IID (non-independent and identically distributed) client data, improving model stability and convergence. Overall, the combination of multimodal feature fusion and adaptive learning mechanisms allows FedMVA to outperform baselines in most metrics, demonstrating its effectiveness in SVA.

To further assess the fairness and robustness of FedMVA, we provide class-wise performance metrics and the confusion matrix in Table 3. The class-wise results show that FedMVA achieves the highest recall (0.774) and F1-score (0.789) on the High severity class, which is critical in real-world applications where severe vulnerabilities must be prioritized. While the Medium class exhibits balanced performance, the Low severity class shows relatively lower precision and recall. This is largely attributed to class imbalance and the subtle characteristics of low-risk vulnerabilities, which can be easily confused with medium-risk ones. As shown in the confusion matrix, most misclassifications occur between adjacent severity levels (e.g., Medium vs. High), whereas severe misclassifications (e.g., Low vs. High) are relatively rare. These findings confirm that FedMVA not only performs well in overall metrics but also maintains consistent effectiveness across different severity levels, reinforcing its practical utility in triage and prioritization scenarios.

Table 4
Performance comparison of FedMVA with different input modality combinations.

Settings	Accuracy	Precision	Recall	F1	MCC
FedMVA+S	0.632	0.503	0.549	0.519	0.584
FedMVA+L	0.653	0.582	0.616	0.598	0.497
FedMVA+L+C	0.621	0.643	0.559	0.592	0.463
FedMVA+S+C	0.647	0.529	0.537	0.523	0.570
FedMVA+S+L	0.712	0.689	0.623	0.664	0.591
FedMVA+S+L+C	**0.784**	**0.739**	**0.712**	**0.690**	**0.625**

Summary for RQ1: Our proposed method FedMVA outperforms baseline methods across most performance metrics. For example, FedMVA achieves an improvement of 9.08% to 47.41% in MCC and 5.95% to 40.16% in recall.

5.2. RQ2: How do different input modality combinations impact the performance of FedMVA?

Approach. To investigate the impact of different input modalities on the performance of FedMVA, we conducted an ablation study by combining three types of input information. Our study involved three modalities: (1) code structure information, (2) lexical information from the code, and (3) comment information. We denote code structure information as S, lexical information as L, and comment information as C. By evaluating various combinations of these modalities, we assessed the contribution of each modality to the model's performance.

Result. Based on the experimental results shown in Table 4, we observe a significant improvement in the performance of FedMVA as the number of information modalities increases. Specifically, when using single-modal information and dual-modal combinations, the performance of our method, FedMVA, slightly decreased across all metrics. However, FedMVA performs significantly better by incorporating three modalities. The accuracy improved by 10.11% to 26.25%. Precision saw the largest increase, rising by 7.26% to 46.92%. Recall improved by 14.29% to 32.59%, the F1-score increased by 3.92% to 32.95%, and the MCC increased by 5.75% to 32.14%. These results suggest that the model's performance improves as the number of information modalities increases.

This improvement is primarily attributed to the effective combination of code structure and lexical information. When using code structure information alone, the performance was relatively moderate. However, when lexical information was added, the model significantly improved across all metrics, particularly precision and MCC. While including comment information led to slight performance regression in some cases (e.g., with the FedMVA+L+C setting), it still provided valuable supplementary information to the model, especially when combined with code structure and lexical information. The fusion of all three modalities further enhanced the model's ability to make comprehensive assessments, enabling it to evaluate vulnerabilities from multiple levels — syntax, lexical features, and the semantic content of comments — thereby improving the performance of the SVA. We did not conduct experiments using comments alone, mainly because comment information did not significantly improve the model's performance when used in isolation. Its value was more apparent when combined with code structure and lexical information, which served as a complementary factor rather than an independent contributor.

Summary for RQ2: Integrating all three modalities–code structure, lexical information, and comments–yields better performance than single-modal and dual-modal inputs. This indicates that these three modalities provide complementary information to each other.

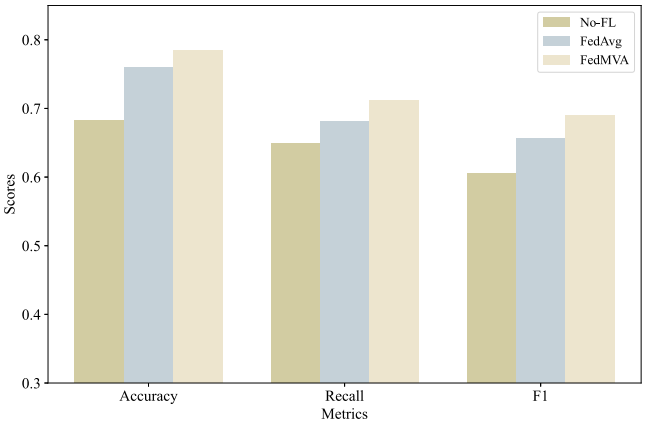


Fig. 2. The impact of federated learning for FedMVA.

5.3. RQ3: How does federated learning influence the effectiveness of SVA compared to conventional training paradigms?

Approach. To investigate how the federated learning framework enhances the effectiveness of FedMVA in the SVA task, we designed three experimental setups: a centralized setting without federated learning, a standard federated learning setup using the FedAvg algorithm, and our proposed FedMVA method. In the centralized setting, all data was stored on a single server for training, with no collaboration or parameter sharing between clients, serving as the baseline for comparison. The FedAvg approach enables clients to independently train local models and send their parameters to a central server for aggregation, facilitating distributed learning but struggling to handle data heterogeneity. In contrast, FedMVA incorporates momentum mechanisms and dynamic learning rates, optimizing parameter updates and dynamically adjusting client contributions through adaptive weight allocation. We employed three evaluation metrics, accuracy, recall, and F1-score, to compare the performance of each method and identify the most effective approach.

Result. The performance comparison of the three methods is shown in Fig. 2. The figure shows FedMVA demonstrates improved overall performance under the federated learning setting, compared to both the basic FedAvg method and the model without federated learning. Specifically, accuracy increased by 3.29% to 14.79%, recall improved by 4.25% to 9.71%, and the F1-score rose by 5.18% to 14.05%. These results demonstrate that the federated learning framework effectively facilitates information sharing and collaborative optimization between clients, enabling the integration of local knowledge into the global model and significantly enhancing its generalization ability. Furthermore, FedMVA leverages momentum mechanisms and dynamic learning rates to mitigate the challenges posed by data heterogeneity, improving both convergence speed and model stability. Notably, precision and MCC metrics are not presented here due to their minimal improvement under federated learning. Instead, we focus on accuracy, recall, and F1-score, which provide more meaningful insights into the performance gains.

Summary for RQ3: Federated learning demonstrates improvements over the basic FedAvg method and models without federated learning. This suggests that federated learning helps enhance collaboration between models, thereby improving the effectiveness of vulnerability assessment.

5.4. RQ4: How do different federated learning settings affect the performance of FedMVA?

Approach. To address RQ4, we conducted ablation experiments to evaluate the impact of different configurations within the federated learning framework on model performance. These experiments focused on three key enhancements to the FedPer method: the weighted variance minimization loss function, the momentum-based weight allocation strategy, and the dynamic learning rate mechanism. The experiments were designed to progressively assess the effect of each modification and examine their role in improving the model's performance in vulnerability assessment tasks.

In addition, to further analyze the effectiveness of our proposed loss function, we conducted a comparative study using several widely adopted federated learning loss functions. Specifically, we evaluated FedAvg (McMahan et al., 2017), FedProx (Li et al., 2020a), FedDyn (Acar et al., 2021), and MOON (Li et al., 2021), alongside our proposed Loss Function. These loss functions were chosen due to their effectiveness in addressing different challenges in federated optimization, including model drift, data heterogeneity, and feature alignment. To ensure a fair comparison, all experiments were conducted within the same federated learning framework, keeping hyperparameters and other experimental conditions constant.

Result. Fig. 3 presents the results obtained from our experiments. The five configurations in the figure represent different variants of FedMVA: FedMVA w/o All refers to the version without any federated enhancements, including dynamic learning rate, momentum-based weighting, and the weighted variance minimization (WVM) loss; FedMVA w/ LR, FedMVA w/ Momentum, and FedMVA w/ WVM enable only one of these components respectively; while FedMVA (Full) activates all enhancements. The model shows improvements across all evaluation metrics based on the experimental results from different federated learning settings. Specifically, accuracy increased by 4.81% to 10.89%, recall improved by 3.04% to 6.43%, and F1-score increased by 3.04% to 6.43%. These improvements suggest that the enhancements made to the federated learning framework have contributed to the model's overall performance.

Further, Fig. 4 presents the accuracy comparison across different loss functions over multiple communication rounds. The results demonstrate that our proposed loss function consistently outperforms FedAvg, FedProx, FedDyn, and MOON, particularly during the initial training phases. The faster convergence observed in the early rounds suggests that our method effectively mitigates local model drift while maintaining stable and efficient global model updates. Since accuracy directly reflects the model's predictive performance and is widely regarded as a primary evaluation metric in federated learning, we focus on its trends to compare the effectiveness of different loss functions.

In detail, the weighted variance minimization loss function effectively reduced the alignment gap between local and global models, enhancing the global model's generalization ability and performance across diverse data distributions. The momentum-based weight allocation strategy dynamically adjusted client contributions, ensuring that the most important clients were appropriately weighted, which improved the model's adaptability to heterogeneous data. Finally, the dynamic learning rate mechanism optimized the training process by fine-tuning adjustments based on each client's specific data distribution. This led to improved accuracy and recall in vulnerability assessment tasks. These enhancements worked synergistically, collectively boosting the performance of the federated learning framework in vulnerability evaluation.

> **Summary for RQ4:** The experimental results demonstrate that the carefully designed federated adjustments contribute to model performance improvement. The enhancements in the approach have strengthened the overall generalization capability of the model, leading to better adaptability and performance across various evaluation metrics.

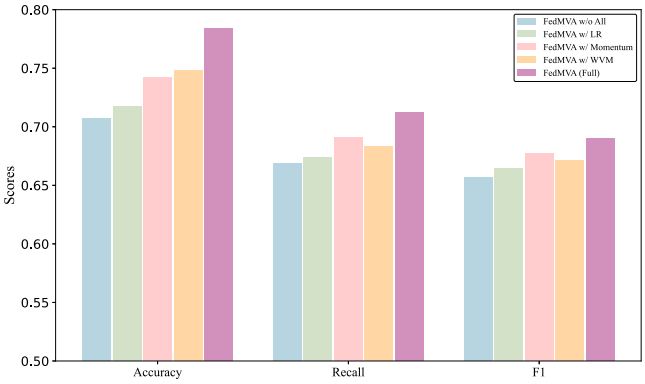


Fig. 3. The impact of different federated learning settings for FedMVA.

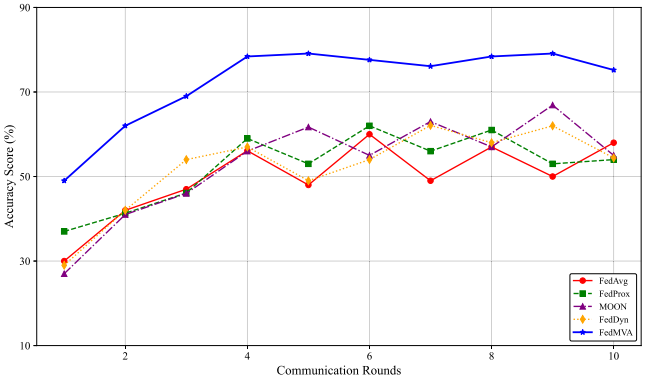


Fig. 4. Accuracy comparison across different loss functions over communication rounds.

5.5. RQ5: How does the proposed multimodal fusion strategy impact the performance of FedMVA and the interaction across modalities?

Approach. To address RQ5, we conducted ablation experiments to assess the impact of the proposed multimodal fusion strategy on the performance of FedMVA and the interaction across modalities. Our approach aligns multimodal features through binary hash encoding, achieving an initial semantic unification. We further refine the feature representations using a self-attention mechanism, enabling deep fusion and enhancing the interaction between different modalities. To validate the effectiveness of this fusion strategy, we compared it with four alternative strategies: feature addition, feature concatenation, weighted averaging, and element-wise multiplication.

Result. Fig. 5 presents the experimental results of different multimodal fusion strategies. The results indicate that the proposed multimodal fusion strategy leads to noticeable improvements in model performance. Specifically, the Accuracy metric increased by 22.12%–39.25%, the F1-score improved by 20.42%–30.93%, and MCC showed an increase of 18.37%–43.02%. We focused on these three metrics as Precision and Recall did not show significant changes across the fusion strategies. These improvements highlight the advantages of the proposed strategy in addressing the challenges of dimensionality reduction when integrating multimodal data within a federated learning framework. The differences in performance can be attributed to how information is retained during the feature integration process. In the Addition, Concatenation, Weighted Averaging, and Element-wise Multiplication strategies, particularly for high-dimensional data, information loss often occurs during feature merging, which weakens the interaction between modalities. In contrast, the proposed fusion strategy leverages the self-attention mechanism to effectively minimize information loss and enhance the integration of features across modalities.

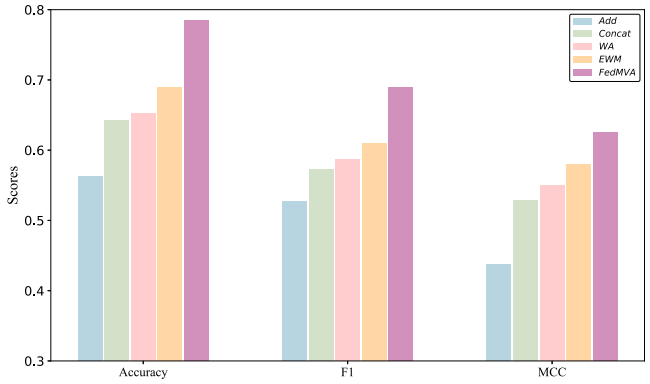


Fig. 5. The impact of different fusion methods for FedMVA.

This leads to better performance in combining cross-modal information and significantly improves the overall model capabilities. These findings demonstrate the clear advantages of the deep fusion strategy in multimodal learning tasks, particularly for complex vulnerability assessment scenarios. The proposed strategy effectively captures subtle relationships across modalities, resulting in more accurate and reliable evaluation outcomes.

Summary for RQ5: The proposed multimodal fusion strategy outperforms traditional methods by better integrating cross-modal information, reducing information loss, and improving model performance. Its self-attention mechanism enhances modality interaction, making it effective for complex tasks like vulnerability assessment.

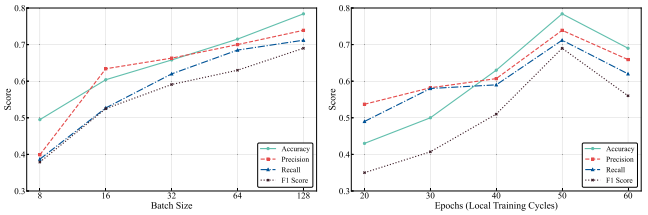
6. Discussion

6.1. Parameter analysis

This section discusses three key factors affecting model performance in federated learning: hyperparameter settings, computational cost, and data heterogeneity. By systematically analyzing these aspects, we evaluate the effectiveness and robustness of our proposed approach under different experimental conditions.

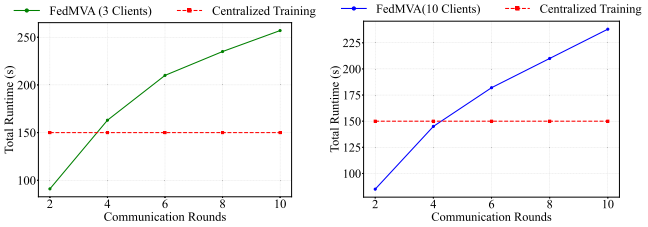
To investigate the influence of hyperparameters, we varied the batch size and the number of epochs. The batch size was set to 8, 16, 32, 64, 128, and the epoch values were set to 20, 30, 40, 50, 60. The experimental results are shown in Fig. 6, which presents the model's performance across different metrics under various configurations.

The results suggest that both batch size and epochs significantly impact model performance. Specifically, the model performed best when the batch size was 128. Larger batch sizes typically improve training efficiency, particularly with large datasets, by enabling faster processing without significantly increasing computational cost. However, excessively large batch sizes can lead to higher memory consumption, limiting scalability. For the batch size in our experiments, smaller values (e.g., 8 and 16) led to slower convergence and more significant fluctuations in evaluation metrics, which can be attributed to the limited amount of gradient information processed in each iteration, resulting in unstable updates during training. A batch size of 128 appeared to offer the best trade-off while avoiding the memory overhead associated with excessively large batches. In terms of local epochs, the model achieved the best performance when trained for 50 epochs. As the number of epochs increased, the model's accuracy improved steadily. However, the performance gains became negligible, and overfitting tendencies emerged beyond this point. Training for 50 epochs struck a balance between learning capacity and generalization, making



(a) batch size (b) local epochs

Fig. 6. The figures illustrate the impact of batch size and local epochs on model performance.



(a) 3 Clients (b) 10 Clients

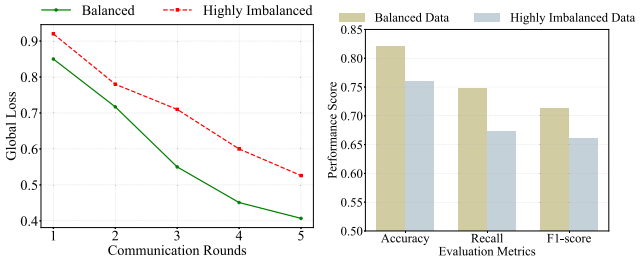
Fig. 7. Runtime comparison of FedMVA and centralized training under different client settings.

it the most appropriate configuration for this task. The results indicate that our method maintains solid performance across different batch sizes and epoch values, showcasing its robustness. In particular, when the batch size is set to 128 and the epochs to 50, the model achieves optimal performance across all evaluated metrics. This shows that while parameter settings influence model performance, our approach remains effective and reliable across various configurations.

Beyond hyperparameter optimization, another critical factor in real-world applications is computational efficiency. In federated learning, computational cost is influenced not only by local training but also by communication overhead, distinguishing it from centralized training approaches. To better understand the practical feasibility of FedMVA, we analyze its total runtime under different communication rounds and compare it with a traditional centralized method.

As shown in Fig. 7, FedMVA's total runtime increases with the number of communication rounds due to iterative model synchronization. However, compared to a centralized approach, FedMVA remains computationally competitive when using a moderate number of rounds (e.g., 4–6 rounds). And we examine how the number of clients impacts computational efficiency by comparing FedMVA with 3 clients and 10 clients. With fewer clients (3 clients), each client is assigned a larger portion of data, leading to longer local training time per round. However, this also results in lower communication overhead, making the overall runtime increase less pronounced compared to the 10-client setting, where communication is more frequent but each client processes less data per iteration. This trade-off underscores the scalability of FedMVA, demonstrating that client allocation can be adjusted to optimize computational efficiency.

Another key challenge in federated learning is the heterogeneity of data distribution across clients, which affects both global model convergence and final model performance. To assess this, we conducted a sensitivity analysis by comparing balanced and highly imbalanced client data distributions. As shown in Fig. 8, when client data distributions are highly imbalanced, the model demonstrates slower convergence, maintaining a higher loss over multiple communication rounds, indicating that local training instability impacts global aggregation. Furthermore, the data heterogeneity negatively affects key evaluation metrics, particularly recall, indicating that the model struggles to generalize when learning from minority class samples. These findings highlight the



(a) Global Model Convergence under Different Data Distributions (b) Final Model Performance under Different Data Distributions

Fig. 8. Impact of Data Heterogeneity on Model Convergence and Performance.

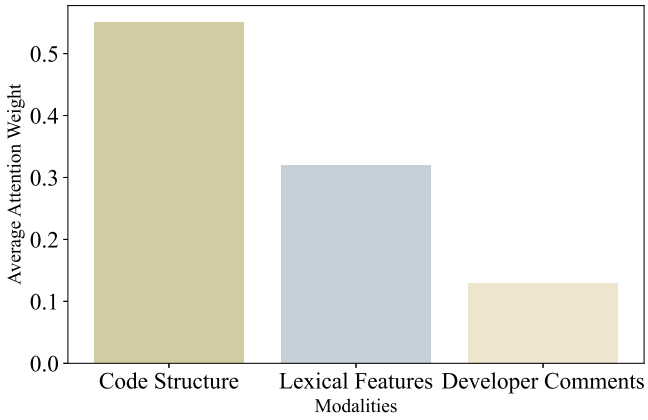


Fig. 9. Importance of Different Modalities.

importance of accounting for data heterogeneity in federated learning frameworks. In scenarios where client data is highly skewed, strategies such as adaptive weighting mechanisms or personalized learning approaches may be required to mitigate imbalance-induced biases and ensure stable model convergence.

Unlike conventional classification tasks, SVA involves highly imbalanced data, complex code structures, and ambiguous vulnerability patterns, making it difficult for models to achieve near-perfect performance. Despite these challenges, FedMVA consistently outperforms baseline methods across most evaluation metrics, demonstrating its effectiveness in leveraging multimodal representations. Furthermore, its federated learning framework enables privacy-preserving collaboration across distributed clients, an essential feature for real-world deployment scenarios where centralized data collection is infeasible. These results highlight the practical feasibility of FedMVA for SVA, as it balances privacy preservation, computational efficiency, and model performance in federated environments.

6.2. Threats to validity

In this subsection, we primarily outline the potential threats to the validity of our research.

Internal threat. One internal threat to the validity of our findings could be related to the quality of the data used in our experiments. While we have applied rigorous preprocessing steps, there may still be latent biases or noise in the dataset that could affect model performance. Factors such as parameter tuning and initialization could also influence the model's optimization process, which might lead to suboptimal training or overfitting in specific configurations. To mitigate this, we ensured our experiments' reproducibility and validated our model's robustness using different settings.

External threat. An external threat to the validity of our study could stem from the generalizability of our results. The dataset used in our experiments is specific to the context of vulnerability assessment, and our findings may not be directly applicable to other domains or datasets without further validation. Moreover, the experimental setup may not represent all real-world use cases, including the specific choice of models, features, and evaluation metrics. Therefore, while our results demonstrate strong performance in the context of our research, additional studies are necessary to evaluate the applicability of our approach in diverse scenarios and to broader populations of software systems.

Construct threat. A key construct-related threat in our study is whether the selected evaluation metrics and experimental design fully capture the complexity of vulnerability assessment. While accuracy, precision, recall, F1-score, and MCC provide a solid assessment of model performance, they primarily reflect classification effectiveness rather than deeper characteristics such as a model's ability to generalize across diverse vulnerability types or handle rare cases.

Another potential limitation lies in the contribution of different input modalities within our multimodal framework. Our ablation study confirmed that integrating multiple modalities improves performance, but it does not explicitly quantify how much each modality contributes to the final prediction. To further investigate this, we conducted an attention weight analysis, which offers an interpretability perspective by revealing how the model allocates importance across different modalities. As illustrated in Fig. 9, the analysis shows that code structure information is the dominant modality, while lexical features play a secondary role, and developer comments contribute the least. These findings reinforce our previous experimental results: structural and lexical information are the primary drivers of vulnerability prediction, whereas comments provide limited independent value.

Despite the insights provided by attention-based attribution, it is important to acknowledge its limitations. Attention weights indicate how the model distributes focus across features but do not necessarily imply causal importance. Furthermore, they do not fully capture potential interactions between modalities, meaning that features with lower individual weights, such as developer comments—might still have indirect influence through their interplay with structural and lexical elements.

Adversarial Threats. While FedMVA enhances data privacy by preventing direct data sharing, federated learning frameworks remain susceptible to adversarial threats, including data poisoning attacks (Steinhardt et al., 2017), model inversion attacks (Song and Namiot, 2022), and membership inference attacks (Shokri et al., 2017). Addressing these risks is critical to ensuring the security and practical adoption of FedMVA.

One critical threat is data poisoning, where malicious clients introduce adversarial updates to degrade model performance. FedMVA mitigates this risk at the design level by employing robust aggregation mechanisms that limit the influence of individual client updates on the global model. While we do not empirically evaluate this aspect in the current work, prior studies (Blanchard et al., 2017) have shown that robust aggregation strategies such as Krum and Multi-Krum can effectively defend against Byzantine or poisoned updates. FedMVA's architecture is compatible with such mechanisms, and future work will incorporate and benchmark these defenses under simulated poisoning scenarios. Another potential risk is model inversion attacks (Fredrikson et al., 2015), where adversaries attempt to reconstruct private data from model updates. FedMVA inherently reduces this risk by limiting the granularity of shared updates and ensuring that no raw data is ever transmitted. Additional protection could be achieved by integrating gradient clipping (Chen et al., 2020) and differential privacy techniques (Hassan et al., 2019), which have been demonstrated to obscure sensitive patterns in model updates. Membership inference attacks (Hu et al., 2022), which aim to determine whether a specific data sample was part of the training set, may also compromise client privacy. Although federated learning already provides a

degree of resistance by decentralizing training, this can be further enhanced through adversarial regularization (Nasr et al., 2018) or gradient masking techniques.

In summary, while the current version of FedMVA focuses on architectural-level mitigation strategies, we acknowledge the importance of empirical validation and plan to explore it in future work. Specifically, we aim to integrate poisoning simulations and privacy-preserving defenses into the FedMVA framework to enable systematic robustness evaluation.

7. Related work

Software vulnerability assessment (Elder et al., 2024; Le et al., 2022) involves identifying, analyzing, and prioritizing vulnerabilities within software systems to maintain their security and reliability. Effective SVA methods can assist developers in prioritizing high-risk vulnerabilities and allocating remediation resources more efficiently. With advances in network and software technologies, the number of vulnerabilities in software systems has surged, making traditional manual methods of vulnerability detection and patching inefficient and increasingly impractical. Relying solely on expert knowledge for individual assessments is time-consuming and insufficient in addressing complex and evolving attack scenarios, thus emphasizing the need for automated vulnerability assessment methods and models. Previous SVA studies have introduced effective methods based on vulnerability descriptions, source code, or code commits.

Vulnerability description-based SVA. A vulnerability description provides a detailed account of software flaws, including their nature, potential impact, and the conditions under which they can be exploited. Prior research indicates that vulnerability descriptions hold valuable information and are practical tools for predicting vulnerability traits in SVA tasks. For example, Han et al. (2017) transformed the SVA task into a text classification task by capturing word-level features through word embeddings, then designed a shallow CNN to capture sentence-level features for severity classification. Sun et al. (2023b) argued that some elements in vulnerability descriptions are extraneous, proposing instead using essential vulnerability elements rather than full descriptions for severity assessment. Babalau et al. (2021) introduced a deep learning approach that predicts severity scores and metrics solely based on textual vulnerability descriptions. Their method employs a multi-task learning framework and a pre-trained BERT model to generate word vector representations, enhancing the prediction of vulnerability severity. Le et al. (2019) proposed a systematic method that combines character and word features, using time-based k-fold cross-validation for model selection and vulnerability description-based severity classification. Gong et al. (2019) suggested a multi-task machine learning approach that jointly predicts multiple vulnerability features from descriptions, eliminating the need for balanced data.

Source code-based SVA. In addition to using vulnerability descriptions, recent studies have also explored approaches to analyzing source code related to vulnerabilities. For example, Le and Babar (2022) investigated machine learning (ML) models for automating function-level SVA tasks. Their approach used fine-grained vulnerable code statements in the assessment model, combining vulnerable and non-vulnerable statements to capture context around vulnerable code and gather additional information about vulnerabilities from surrounding code. Hao et al. (2023) focused on constructing function call graphs centered on vulnerable functions. By leveraging a graph attention neural network algorithm, they extracted key vulnerability features from function call graphs and vulnerability attribute graphs, utilizing information from both graphs to classify vulnerability severity.

Commit-based SVA. In recent years, commit-based vulnerability assessment has gained attention. Le et al. (2021) proposed the DeepCVA model, a deep multi-task learning approach that directly predicts CVSS metrics from changes in vulnerability-contributing commits. DeepCVA extracts features from code changes and their surrounding context by integrating attention mechanisms and convolutional gated recurrent units, eliminating reliance on often-delayed vulnerability reports. This approach enhances assessment efficiency and enables developers to prioritize vulnerability remediation tasks more effectively at earlier stages of the development lifecycle.

Traditional approaches to software vulnerability assessment often rely on centralized models that require aggregating sensitive code or vulnerability descriptions in a single location for analysis. This practice increases the risk of data breaches and violates stringent data privacy regulations, particularly in industrial settings where source code and vulnerability data are highly confidential. To address these privacy concerns, federated learning paradigms have emerged as a promising solution. By enabling collaborative training across organizations without sharing raw data, federated learning ensures data privacy while facilitating cross-organizational collaboration for vulnerability assessment.

Recent advancements in privacy-preserving federated learning have further enhanced its applicability in SVA tasks. Yazdinejad et al. (2024a) proposed a privacy-preserving FL framework that integrates Additive Homomorphic Encryption and a Gaussian Mixture Model to detect malicious gradients with low computational overhead, enhancing robustness against adversarial attacks. Beyond cryptographic approaches, Yazdinejad et al. (2024b) introduced a hybrid FL framework designed to handle irregular user participation and varying data quality in next-generation IoT environments. By combining synchronous and asynchronous update mechanisms, their approach improves FL performance in highly dynamic network settings. Meanwhile, Namakshenas et al. (2024) focused on enhancing FL explainability in industrial cyber–physical systems. Their interpretation-based FL model employs Shapley values alongside Additive Homomorphic Encryption, enabling both privacy preservation and fairness in federated model training. These studies highlight the importance of integrating security, adaptability, and explainability in privacy-preserving FL frameworks for SVA. However, existing methods primarily address communication efficiency and data privacy, leaving optimization challenges in heterogeneous FL environments underexplored.

In addition to privacy challenges, existing SVA methods often focus on single modalities, such as vulnerability descriptions or source code, neglecting the structural and contextual information embedded within code. These limitations hinder the model's ability to capture the complex characteristics of software vulnerabilities fully. Recent advancements in multimodal learning have shown promise in addressing these limitations. For instance, as highlighted by Shiri Harzevili et al. (2024), hybrid data sources and graph-based input representations are increasingly used in software vulnerability detection, with 39.1% of studies utilizing hybrid sources and 57.2% employing graph-based techniques. These approaches demonstrate the importance of combining multiple modalities, such as source code, graph structures, and textual features, to improve vulnerability assessment accuracy.

To address these challenges, we propose a multimodal approach that integrates structural, lexical, and comment-based features. Specifically, we use a graph attention network (GAT) to capture local structural relationships, a convolutional neural network (CNN) to extract global structural patterns, and the pre-trained language model CodeT5 to provide lexical insights by treating code as plain text. Developer comments are also incorporated to enrich contextual understanding. To combine these modalities effectively, we employ a self-attention weighted fusion mechanism that dynamically adjusts the contribution of each modality based on its relevance. This comprehensive integration enables a deeper understanding of vulnerabilities, improving the accuracy and efficiency of vulnerability assessment while addressing the dual challenges of privacy protection and feature representation.

8. Conclusion

This study proposes a multimodal vulnerability assessment method based on a federated learning framework (FedMVA), effectively improving performance by integrating graph structure information, lexical features of code, and code comments. In the design of the method, we introduce a weighted variance minimization loss function to optimize the alignment between local and global models and incorporate a momentum-based weight allocation strategy along with a dynamic learning rate mechanism to further enhance the model's robustness and adaptability in heterogeneous data environments. Through ablation experiments, we have validated that our multimodal fusion strategy significantly improves model performance within the federated learning framework, demonstrating its advantages in handling complex vulnerability assessment tasks.

In the future, we aim to improve our proposed method by integrating advanced federated learning techniques, including incorporating large language models (Achiam et al., 2023; Su et al., 2021) to enhance the model's ability to process and understand complex textual data. Furthermore, we will focus on further validating FedMVA in more challenging distributed environments to assess its scalability and robustness in real-world applications.

CRediT authorship contribution statement

Qingyun Liu: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Data curation. **Xiaolin Ju:** Writing – review & editing, Supervision, Methodology, Investigation, Conceptualization. **Xiang Chen:** Writing – review & editing, Validation, Supervision, Methodology, Investigation, Conceptualization. **Lina Gong:** Writing – review & editing, Supervision, Methodology, Investigation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

References

Abyane, A.E., Zhu, D., Souza, R., Ma, L., Hemmati, H., 2023. Towards understanding quality challenges of the federated learning for neural networks: a first look from the lens of robustness. Empir. Softw. Eng. 28 (2), 44.

Acar, D.A.E., Zhao, Y., Navarro, R.M., Mattina, M., Whatmough, P.N., Saligrama, V., 2021. Federated learning based on dynamic regularization. arXiv preprint arXiv:2111.04263.

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al., 2023. Gpt-4 technical report. arXiv preprint arXiv:2303.08774.

Arivazhagan, M.G., Aggarwal, V., Singh, A.K., Choudhary, S., 2019. Federated learning with personalization layers. arXiv preprint arXiv:1912.00818.

Babalau, I., Corlatescu, D., Grigorescu, O., Sandescu, C., Dascalu, M., 2021. Severity prediction of software vulnerabilities based on their text description. In: 2021 23rd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing. SYNASC, IEEE, pp. 171–177.

Blanchard, P., El Mhamdi, E.M., Guerraoui, R., Stainer, J., 2017. Machine learning with adversaries: Byzantine tolerant gradient descent. Adv. Neural Inf. Process. Syst. 30.

Cai, Z., Cai, Y., Chen, X., Lu, G., Pei, W., Zhao, J., 2024. CSVD-TF: Cross-project software vulnerability detection with TrAdaBoost by fusing expert metrics and semantic metrics. J. Syst. Softw. 213, 112038.

Chen, T., 2015. Xgboost: extreme gradient boosting. R Packag. Version 0. 4-2 1 (4).

Chen, X., Wu, S.Z., Hong, M., 2020. Understanding gradient clipping in private sgd: A geometric perspective. Adv. Neural Inf. Process. Syst. 33, 13773–13782.

Cortes, C., 1995. Support-vector networks. Mach. Learn.

Dissanayake, N., Jayatilaka, A., Zahedi, M., Babar, M.A., 2022. Software security patch management-A systematic literature review of challenges, approaches, tools and practices. Inf. Softw. Technol. 144, 106771.

Elder, S., Rahman, M.R., Fringer, G., Kapoor, K., Williams, L., 2024. A survey on software vulnerability exploitability assessment. ACM Comput. Surv. 56 (8), 1–41.

Fan, J., Li, Y., Wang, S., Nguyen, T.N., 2020. AC/C++ code vulnerability dataset with code changes and CVE summaries. In: Proceedings of the 17th International Conference on Mining Software Repositories. pp. 508–512.

Feutrill, A., Ranathunga, D., Yarom, Y., Roughan, M., 2018. The effect of common vulnerability scoring system metrics on vulnerability exploit delay. In: 2018 Sixth International Symposium on Computing and Networking. CANDAR, IEEE, pp. 1–10.

Fredrikson, M., Jha, S., Ristenpart, T., 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. pp. 1322–1333.

Gong, X., Xing, Z., Li, X., Feng, Z., Han, Z., 2019. Joint prediction of multiple vulnerability characteristics through multi-task learning. In: 2019 24th International Conference on Engineering of Complex Computer Systems. ICECCS, IEEE, pp. 31–40.

Guo, D., Ren, S., Lu, S., Feng, Z., Tang, D., Liu, S., Zhou, L., Duan, N., Svyatkovskiy, A., Fu, S., et al., 2020. Graphcodebert: Pre-training code representations with data flow. arXiv preprint arXiv:2009.08366.

Han, Z., Li, X., Xing, Z., Liu, H., Feng, Z., 2017. Learning to predict severity of software vulnerability using only vulnerability description. In: 2017 IEEE International Conference on Software Maintenance and Evolution. ICSME, IEEE, pp. 125–136.

Hanzely, F., Richtárik, P., 2020. Federated learning of a mixture of global and local models. arXiv preprint arXiv:2002.05516.

Hao, J., Luo, S., Pan, L., 2023. A novel vulnerability severity assessment method for source code based on a graph neural network. Inf. Softw. Technol. 161, 107247.

Hassan, M.U., Rehmani, M.H., Chen, J., 2019. Differential privacy techniques for cyber physical systems: A survey. IEEE Commun. Surv. Tutor. 22 (1), 746–789.

Ho, T.K., 1995. Random decision forests. In: Proceedings of 3rd International Conference on Document Analysis and Recognition, vol. 1, IEEE, pp. 278–282.

Hu, H., Salcic, Z., Sun, L., Dobbie, G., Yu, P.S., Zhang, X., 2022. Membership inference attacks on machine learning: A survey. ACM Comput. Surv. 54 (11s), 1–37.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.-Y., 2017. Lightgbm: A highly efficient gradient boosting decision tree. Adv. Neural Inf. Process. Syst. 30.

Khraisat, A., Alazab, A., Singh, S., Jan, T., Gomez, Jr., A., 2024. Survey on federated learning for intrusion detection system: Concept, architectures, aggregation strategies, challenges, and future directions. ACM Comput. Surv. 57 (1), 1–38.

Le, T.H.M., Babar, M.A., 2022. On the use of fine-grained vulnerable code statements for software vulnerability assessment models. In: Proceedings of the 19th International Conference on Mining Software Repositories. pp. 621–633.

Le, T.H., Chen, H., Babar, M.A., 2022. A survey on data-driven software vulnerability assessment and prioritization. ACM Comput. Surv. 55 (5), 1–39.

Le, T.H.M., Hin, D., Croft, R., Babar, M.A., 2021. Deepcva: Automated commit-level vulnerability assessment with deep multi-task learning. In: 2021 36th IEEE/ACM International Conference on Automated Software Engineering. ASE, IEEE, pp. 717–729.

Le, T.H.M., Sabir, B., Babar, M.A., 2019. Automated software vulnerability assessment with concept drift. In: 2019 IEEE/ACM 16th International Conference on Mining Software Repositories. MSR, IEEE, pp. 371–382.

Li, L., Fan, Y., Tse, M., Lin, K.-Y., 2020a. A review of applications in federated learning. Comput. Ind. Eng. 149, 106854.

Li, Q., He, B., Song, D., 2021. Model-contrastive federated learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10713–10722.

Li, T., Sahu, A.K., Talwalkar, A., Smith, V., 2020b. Federated learning: Challenges, methods, and future directions. IEEE Signal Process. Mag. 37 (3), 50–60.

Liu, C., Chen, X., Li, X., Xue, Y., 2024. Making vulnerability prediction more practical: Prediction, categorization, and localization. Inf. Softw. Technol. 171, 107458.

Lu, G., Ju, X., Chen, X., Pei, W., Cai, Z., 2024. GRACE: Empowering LLM-based software vulnerability detection with graph structure and in-context learning. J. Syst. Softw. 212, 112031.

Ma, X., Zhu, J., Lin, Z., Chen, S., Qin, Y., 2022. A state-of-the-art survey on solving non-IID data in federated learning. Future Gener. Comput. Syst. 135, 244–258.

McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A., 2017. Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics. PMLR, pp. 1273–1282.

Namakshenas, D., Yazdinejad, A., Dehghantanha, A., Parizi, R.M., Srivastava, G., 2024. IP2FL: Interpretation-based privacy-preserving federated learning for industrial cyber-physical systems. IEEE Trans. Ind. Cyber-Phys. Syst.

Nasr, M., Shokri, R., Houmansadr, A., 2018. Machine learning with membership privacy using adversarial regularization. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. pp. 634–646.

Ruiz, L., Gama, F., Ribeiro, A., 2020. Gated graph recurrent neural networks. IEEE Trans. Signal Process. 68, 6303–6318.

Russell, S.J., Norvig, P., 2016. Artificial Intelligence: A Modern Approach. Pearson.

Sahar, S., Younas, M., Khan, M.M., Sarwar, M.U., 2024. DP-CCL: A supervised contrastive learning approach using CodeBERT model in software defect prediction. IEEE Access 12, 22582–22594.

SARD, S., 2020. Software assurance reference dataset project.
Shiri Harzevili, N., Boaye Belle, A., Wang, J., Wang, S., Jiang, Z.M., Nagappan, N., 2024. A systematic literature review on automated software vulnerability detection using machine learning. ACM Comput. Surv. 57 (3), 1–36.
Shokri, R., Stronati, M., Song, C., Shmatikov, V., 2017. Membership inference attacks against machine learning models. In: 2017 IEEE Symposium on Security and Privacy. SP, IEEE, pp. 3–18.
Song, J., Namiot, D., 2022. A survey of the implementations of model inversion attacks. In: International Conference on Distributed Computer and Communication Networks. Springer, pp. 3–16.
Steinhardt, J., Koh, P.W.W., Liang, P.S., 2017. Certified defenses for data poisoning attacks. Adv. Neural Inf. Process. Syst. 30.
Su, J., Cao, J., Liu, W., Ou, Y., 2021. Whitening sentence representations for better semantics and faster retrieval. arXiv preprint arXiv:2103.15316.
Sun, W., Fang, C., Miao, Y., You, Y., Yuan, M., Chen, Y., Zhang, Q., Guo, A., Chen, X., Liu, Y., et al., 2023a. Abstract syntax tree for programming language understanding and representation: How far are we?. arXiv preprint arXiv:2312.00413.
Sun, X., Ye, Z., Bo, L., Wu, X., Wei, Y., Zhang, T., Li, B., 2023b. Automatic software vulnerability assessment by extracting vulnerability elements. J. Syst. Softw. 204, 111790.
Suneja, S., Zheng, Y., Zhuang, Y., Laredo, J., Morari, A., 2020. Learning to map source code to software vulnerability using code-as-a-graph. arXiv preprint arXiv:2006.08614.
Tan, A.Z., Yu, H., Cui, L., Yang, Q., 2022. Towards personalized federated learning. IEEE Trans. Neural Netw. Learn. Syst. 34 (12), 9587–9603.
Wang, Y., Wang, W., Joty, S., Hoi, S.C., 2021. Codet5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation. arXiv preprint arXiv:2109.00859.
Xu, Z., Zhang, Y., Andrew, G., Choquette-Choo, C.A., Kairouz, P., McMahan, H.B., Rosenstock, J., Zhang, Y., 2023. Federated learning of gboard language models with differential privacy. arXiv preprint arXiv:2305.18465.
Yamamoto, H., Wang, D., Rajbahadur, G.K., Kondo, M., Kamei, Y., Ubayashi, N., 2023. Towards privacy preserving cross project defect prediction with federated learning. In: 2023 IEEE International Conference on Software Analysis, Evolution and Reengineering. SANER, IEEE, pp. 485–496.
Yang, Y., Hu, X., Gao, Z., Chen, J., Ni, C., Xia, X., Lo, D., 2024. Federated learning for software engineering: A case study of code clone detection and defect prediction. IEEE Trans. Softw. Eng..
Yang, Q., Liu, Y., Chen, T., Tong, Y., 2019. Federated machine learning: Concept and applications. ACM Trans. Intell. Syst. Technol. (TIST) 10 (2), 1–19.
Yazdinejad, A., Dehghantanha, A., Karimipour, H., Srivastava, G., Parizi, R.M., 2024a. A robust privacy-preserving federated learning model against model poisoning attacks. IEEE Trans. Inf. Forensics Secur.
Yazdinejad, A., Dehghantanha, A., Srivastava, G., Karimipour, H., Parizi, R.M., 2024b. Hybrid privacy preserving federated learning against irregular users in next-generation Internet of Things. J. Syst. Archit. 148, 103088.
Ye, M., Fang, X., Du, B., Yuen, P.C., Tao, D., 2023. Heterogeneous federated learning: State-of-the-art and research challenges. ACM Comput. Surv. 56 (3), 1–44.
Youm, H.Y., 2020. An overview of de-identification techniques and their standardization directions. IEICE Trans. Inf. Syst. 103 (7), 1448–1461.
Yu, H., Liu, Z., Liu, Y., Chen, T., Cong, M., Weng, X., Niyato, D., Yang, Q., 2020. A fairness-aware incentive scheme for federated learning. In: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society. pp. 393–399.
Zeng, C., Yu, Y., Li, S., Xia, X., Wang, Z., Geng, M., Bai, L., Dong, W., Liao, X., 2023. degraphcs: Embedding variable-based flow graph for neural code search. ACM Trans. Softw. Eng. Methodol. 32 (2), 1–27.
Zhang, C., Xie, Y., Bai, H., Yu, B., Li, W., Gao, Y., 2021. A survey on federated learning. Knowl.-Based Syst. 216, 106775.
Zhang, C., Yu, T., Liu, B., Xin, Y., 2024. Vulnerability detection based on federated learning. Inf. Softw. Technol. 167, 107371.
Zhou, Y., Liu, S., Siow, J., Du, X., Liu, Y., 2019. Devign: Effective vulnerability identification by learning comprehensive program semantics via graph neural networks. Adv. Neural Inf. Process. Syst. 32.
Zhou, J., Pacheco, M., Wan, Z., Xia, X., Lo, D., Wang, Y., Hassan, A.E., 2021. Finding a needle in a haystack: Automated mining of silent vulnerability fixes. In: 2021 36th IEEE/ACM International Conference on Automated Software Engineering. ASE, IEEE, pp. 705–716.
Zhuo, H.H., Feng, W., Lin, Y., Xu, Q., Yang, Q., 2019. Federated deep reinforcement learning. arXiv preprint arXiv:1901.08277.

Qingyun Liu is currently pursuing the Master degree at the School of Artificial Intelligence and Computer Science, Nantong University. Her research interests include vulnerability assessment and federated learning.

Xiaolin Ju (Member, IEEE) was born in April 1976. He received the B.S. degree in information science from Wuhan University, in 1998, the M.Sc. degree in computer science from Southeast University, in 2004, and the Ph.D. degree in computer science from the Chinese University of Mining Technology, in 2014. He is currently an Associate Professor with the School of Information Science and Technology, Nantong University, Nantong, China. His current research interests include software testing, such as collective intelligence, deep learning testing and optimization, and software defects analysis.

Xiang Chen received the B.Sc. degree in the school of management from Xi'an Jiaotong University, China in 2002. Then he received his M.Sc., and Ph.D. degrees in computer software and theory from Nanjing University, China in 2008 and 2011 respectively. He is currently an Associate Professor at the Department of Information Science and Technology, Nantong University, Nantong, China. He has authored or co-authored more than 120 papers in refereed journals or conferences, such as IEEE Transactions on Software Engineering, ACM Transactions on Software Engineering and Methodology, Empirical Software Engineering, Information and Software Technology, Journal of Systems and Software, IEEE Transactions on Reliability, Journal of Software: Evolution and Process, Software - Practice and Experience, Automated Software Engineering, Journal of Computer Science and Technology, IET Software, Software Quality Journal, Knowledge-based Systems, International Conference on Software Engineering (ICSE), The ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE), International Conference Automated Software Engineering (ASE), International Conference on Software Maintenance and Evolution (ICSME), International Conference on Program Comprehension (ICPC), and International Conference on Software Analysis, Evolution and Reengineering (SANER). His research interests include software engineering, in particular software testing and maintenance, software repository mining, and empirical software engineering. He received two ACM SIGSOFT distinguished paper awards in ICSE 2021 and ICPC 2023. He is the editorial board member of Information and Software Technology. More information about him can be found at: https://smartse.github.io/index.html.

Lina Gong is an associate professor at the School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China. She received her Ph.D. degree (thesis: Study on Cross-Project Defect Prediction Based on Transfer Learning) in Computer Software and Theory from China University of Mining and Technology in December 2020. From September 2019 to September 2020, she studied as a visiting researcher in the Software Analysis and Intelligence Lab (SAIL), School of Computing, Queen's University, Canada, under the supervision of Prof. Dr. Ahmed Hassan. Her research focuses on helping developers gain insights from messy data recorded in software repositories and developing machine learning (ML)-enabled software development techniques (e.g., ML for software defects, ML for identifier normalization, ML for issue classification) to improve the efficiency and productivity of developers. These techniques aim to help developers find defects more precisely and quickly, as well as make better data-informed decisions. The foundation of her research has been published in prestigious journals and conferences, including IEEE Transactions on Software Engineering (TSE), ACM Transactions on Software Engineering and Methodology (TOSEM), International Conference on Automated Software Engineering (ASE), IEEE Transactions on Reliability, and Journal of Software: Evolution and Process.