

RESEARCH ARTICLE OPEN ACCESS

Resilient Multimodal Fusion for Robust Vulnerability Exploitability Prediction Under Data Degradation

Mona Dolati¹ | Seyedeh Leili Mirtaheeri¹  | Reza Shahbazian²  | Andrea Pugliese¹ 

¹Department of Computer Engineering, Modeling, Electronics and Systems Engineering, University of Calabria, Rende (CS), Italy | ²Department of Humanities, University of Palermo, Palermo, Italy

Correspondence: Seyedeh Leili Mirtaheeri (leili.mirtaheeri@dimes.unical.it)

Received: 14 April 2026 | **Revised:** 24 August 2026 | **Accepted:** 26 August 2026

Keywords: knowledge distillation | missing modality learning | multimodal fusion | resilient fusion | robust AI | vulnerability exploitability

ABSTRACT

Automated assessment of software vulnerability exploitability is essential for intelligent cyber defense, yet its effectiveness is often hindered by unstable, delayed, or degraded security data. Existing models typically rely on either unstructured descriptions or structured metadata, making them fragile to information gaps. We present SecureFusion-X, a resilient multimodal framework designed for reliable prediction under adverse data conditions. The system utilizes decision-level fusion with two specialized branches: TextGuard-Lite, an efficient encoder based on a distilled MiniLM model, and TabShield-Net, a TabTransformer-based metadata encoder. A leakage-free training protocol simulating modality perturbations supports robustness. Evaluation on 8549 curated NVD-aligned vulnerability records shows that SecureFusion-X improves F_1 from 0.9766 for the structured metadata baseline to 0.9798, while reducing the Brier score from 0.0202 to 0.0182 under clean-test conditions. Under representative data degradation with 30% missing or noisy text, SecureFusion-X remains stable, with less than 0.1% variation in fused performance, whereas the text-only baseline drops by 9.5% under missing-text conditions. SecureFusion-X provides a computationally efficient and robustness-oriented framework for vulnerability exploitability prioritization under controlled and temporally shifted evaluation settings.

1 | Introduction

The relentless expansion of the global software ecosystem has led to a dramatic surge in publicly disclosed vulnerabilities, placing an immense burden on cybersecurity practitioners. Thousands of new Common Vulnerabilities and Exposures (CVEs) are cataloged annually; however, empirical evidence suggests that only a small fraction of these are ever successfully weaponized in real-world attacks [1, 2]. While few in number, these exploited vulnerabilities represent a disproportionate threat to organizational stability, often resulting in severe financial loss and operational disruption. Given the sheer volume of disclosures, manual triaging is no longer feasible, necessitating the development of intelligent, automated systems capable of accurately forecasting exploitability to prioritize remediation efforts.

The Common Vulnerability Scoring System (CVSS) remains the industry standard for quantifying the technical severity of software defects [3]. However, while CVSS metadata provides essential structural insights, technical severity is frequently a misleading proxy for actual exploitability [4]. Attackers may prioritize lower- or medium-severity vulnerabilities when they are easier to exploit, chain, operationalize, or keep under the radar of traditional defense mechanisms [5]. Consequently, a robust defensive posture requires predictive models that look beyond static severity scores to capture the dynamic nuances of threat landscapes.

1.1 | Problem Statement

Current automated risk assessment tools, such as the Exploit Prediction Scoring System (EPSS) [6] and research-driven

This is an open access article under the terms of the [Creative Commons Attribution](https://creativecommons.org/licenses/by/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2026 The Author(s). *Advanced Intelligent Systems* published by Wiley-VCH GmbH.

Natural Language Processing (NLP) models [7, 8], have significantly advanced the field. Despite these improvements, a critical limitation persists: most existing frameworks are unimodal, relying exclusively on either unstructured textual descriptions or structured metrics. In real-world security operations, data is often imperfect; NVD-derived structured metadata, such as vector components and base severity information, may be unavailable, incomplete, or revised after the initial disclosure, while NVD descriptions may be sparse, noisy, generic, or temporarily incomplete.

Unimodal models are inherently fragile in these scenarios. A text-centric model becomes ineffective if the description is missing, while a metric-based model fails when the metadata is unavailable. Furthermore, existing systems often struggle to maintain performance under “modality degradation,” where the quality of one data source is compromised. There is an urgent need for a unified framework that fuses disparate data signals and explicitly evaluates resilience when textual evidence is missing or corrupted while also being designed to tolerate perturbations in structured metadata during training.

1.2 | Novelties and Contributions

To address these challenges, we introduce SecureFusion-X, a resilient multimodal framework designed for high-fidelity exploitability prediction under adverse data conditions. Our architecture employs a decision-level fusion strategy that integrates two specialized branches: TextGuard-Lite, a distilled MiniLM-based student model optimized for the semantic understanding of vulnerability descriptions, and TabShield-Net, a TabTransformer-based encoder designed to extract complex patterns from structured vulnerability metadata.

The primary contributions of this work are as follows:

1. **Robust Multimodal Fusion:** We propose a framework that harmonizes semantic textual features with structured metadata, providing a comprehensive risk profile that improves F_1 , recall, Matthews Correlation Coefficient (MCC), and calibration over the strongest single-modality baseline under the validation-calibrated decision threshold.
2. **Resilience via Modality Perturbation:** We introduce a specialized training protocol that utilizes modality masking and noise injection. This encourages the system to remain operational and accurate under the representative 30% textual degradation setting, while sensitivity analysis further evaluates missing-text and noisy-text ratios of up to 50%. The model thereby learns to rely on the available signals without sacrificing the overall performance.
3. **High-Efficiency Design:** By leveraging knowledge distillation, SecureFusion-X achieves strong leakage-free performance ($F_1 = 0.9798$) while maintaining a lightweight computational footprint. This computational profile supports its potential integration into high-throughput vulnerability management workflows, while operational deployment under naturally imbalanced threat distributions remains an important direction for further validation.

The remainder of this paper is organized as follows: Section 2 reviews related work in vulnerability prediction and multimodal

learning. Section 3 describes the dataset construction, feature extraction, data complexity considerations, proposed architecture, and training methodology. Section 4 presents the experimental setup and results, including performance, robustness, ablation, temporal transfer, practical, and comparative analyses. Finally, Section 5 concludes the paper and outlines future research directions.

2 | Related Work

Early exploitability prediction focused on handcrafted features and classical machine learning over National Vulnerability Database (NVD)-style metadata and descriptions [9–11]. These approaches typically used severity metrics, basic text representations, and balanced (or heuristically rebalanced) datasets but did not model robustness to noisy or incomplete information [12]. With neural language models, DarkEmbed [13] and FastEmbed [14] introduced contextual and fastText-based embeddings for vulnerability text, improving performance under class imbalance but still depending heavily on complete textual data.

To move beyond severity-only prioritization, the Exploit Prediction Scoring System (EPSS) integrates large-scale behavioral and community-driven signals to produce probabilistic exploit scores [6, 15]. Subsequent work incorporated authority alerts as early real-world danger indicators [16] and cost-sensitive learning to better handle skewed exploit labels [17]. However, most such systems treat text, CVSS metadata, and external threat indicators as separate feature sources rather than truly multimodal components, and they generally assume that all key fields are present at prediction time [11, 18].

Transformer-based models for vulnerability analysis have shown strong semantic understanding of CVE reports and related security text [7, 8, 19, 20]. Large architectures, however, incur significant computational and memory overhead, limiting their suitability for high-throughput operational pipelines [21]. Knowledge distillation and compression methods such as DistilBERT, TinyBERT, MobileBERT, quantization, and parameter sharing [22–26] have enabled lightweight deployments; Green-AI perspectives like VuldistilBERT further emphasize energy efficiency in vulnerability analysis [27]. Similar lightweight transformer designs with multi-task learning and dynamic weighting have been explored for vulnerability prediction [28], and for other cyber-tasks such as phishing detection and email classification [29–31].

Beyond text-only modeling, recent work highlights the value of structural and relational context. Heterogeneous graph-based frameworks such as ExPreHet [32] and relational vulnerability-vendor networks [15] capture multi-attribute interactions and co-exploitation behavior, outperforming purely feature-based methods and EPSS-like baselines on large-scale datasets. In parallel, multimodal architectures for vulnerability and severity prediction combine code, graph structure, and textual metadata using models like GraphCodeBERT, gradient boosting, and ordinal regression [33, 34]. These works demonstrate the benefit of fusing heterogeneous information but typically target severity or detection rather than explicit exploitability classification and only rarely consider robustness when modalities are missing or delayed.

The broader multimodal learning community has begun to systematically address missing-modality robustness. Surveys and methods on “multimodal learning with missing modalities” and parameter-efficient adaptation show that standard multimodal networks degrade sharply when some inputs are absent and propose lightweight adaptation layers to mitigate this [35, 36]. Other work studies modality dropout and modality bias using knowledge distillation and adaptive decision strategies to maintain performance under partial observations [37, 38]. These advances provide generic strategies, such as modality dropout, distribution approximation, and test-time adaptation, that are highly relevant to cyber-physical and security settings but have not yet been explicitly applied to vulnerability exploitability prediction.

Large language models (LLMs) and generative models have also been explored for vulnerability assessment. Lightweight generative-AI architectures augment NVD data and fine-tune compact Bidirectional Encoder Representations from Transformers (BERT) variants for automatic CVSS scoring [7, 39], while zero-shot ChatGPT evaluations reveal both potential and limitations of general-purpose LLMs for predicting CVSS, Common Weakness Enumeration (CWE), and Common Platform Enumeration (CPE) from descriptions at scale [18]. Multi-objective BERT-based classifiers support joint prediction of severity and vulnerability type for large-scale CVE triage [20], and higher-level impact analysis frameworks use LLMs to map CVEs to ATT&CK techniques [40]. Yet, these systems generally operate on single-modality inputs (text) and do not incorporate robustness under degraded or missing metadata.

Overall, the state-of-the-art offers powerful text-based, graph-based, and multimodal models for vulnerability analysis and exploit scoring, along with separate lines of work on missing-modality robustness. However, existing exploitability prediction approaches typically presume complete, clean inputs and do not jointly optimize for multimodal fusion, resilience to data degradation, and computational efficiency. The proposed SecureFusion-X framework is positioned at this intersection, drawing on advances in early exploitability prediction [11], community-driven exploit scoring [6], efficient transformers [22–24], and robust multimodal learning with missing

modalities [35–37] to deliver resilient exploitability prediction under real-world data loss and delay. Recent progress in adjacent infrastructure-monitoring domains also highlights the broader relevance of transformer-based representation learning, attention mechanisms, and teacher–student knowledge distillation for robust multimodal analysis. Xu et al. [41] proposed a transformer-based large vision model for universal structural damage segmentation, combining a pre-trained transformer-based frozen backbone with a fine-tuned CNN-based segmentation head and student–teacher contrastive learning. Although this work targets structural damage segmentation rather than cybersecurity, it demonstrates how transformer backbones and teacher–student learning can improve generalization and robustness under complex conditions. Li et al. [42] introduced an adaptive attention-based image-captioning framework for bridge damage description, where multi-level visual features are selectively attended to generate natural-language damage descriptions. This study further illustrates how attention mechanisms can align heterogeneous evidence with textual outputs for interpretable decision support. These advances motivate the methodological direction of SecureFusion-X, which adapts compact representation learning and modality-aware fusion to exploitability prediction by combining a distilled MiniLM-based textual branch with a structured-metadata branch.

To further expand the literature on vulnerability knowledge bases and graph-based reasoning, recent studies have explicitly modeled relational security knowledge from public vulnerability repositories. Shi et al. [43] constructed threat knowledge graphs integrating CVE, CWE, and CPE entities to uncover hidden associations among vulnerabilities, weaknesses, and affected products. Host et al. [44] built vulnerability knowledge graphs from textual descriptions in the National Vulnerability Database using named entity recognition, relation extraction, and knowledge graph embeddings. More directly related to exploitability prediction, Xu et al. [32] proposed ExPreHet, a heterogeneous graph-based framework that models associated vulnerabilities and neighboring entities to predict exploitability. As shown in Table 1, these studies demonstrate the value of the relational vulnerability context; however, they often require explicit graph construction and external relation modeling. In contrast, SecureFusion-X adopts a

TABLE 1 | Comparison of SecureFusion-X against state-of-the-art vulnerability prediction and assessment frameworks.

Work	Year	Modalities	Fusion strategy	Resilience ^a	Efficiency	Objective
Bozorgi et al. [9]	2010	Text, tabular	Early fusion (support vector machine, SVM)	No	Low	Exploitability
Jacobs et al. [6]	2023	Behavioral, tabular	Probabilistic	No	High	Exploitability
Kai et al. [27]	2023	Text	Unimodal	No	High (distilled)	Severity
Iannone et al. [11]	2024	Text, tabular	Feature concatenation	No	Moderate	Exploitability
Lim et al. [33]	2024	Text, code, tabular	Decision-level	No	Moderate	Severity
Mahbub et al. [15]	2025	Graph, metadata	Relational embedding	No	Moderate	Exploitability
Xu et al. [32]	2025	Text, graph	Heterogeneous graph	No	Low	Exploitability
Mirtaheiri et al. [45]	2025	Text	Unimodal	No	High (green AI)	Exploitability
SecureFusion-X	2026	Text, tabular	Decision-level meta-classifier	Yes	High (distilled)	Exploitability

Note: Bold indicates the proposed method.

^aRefers to robustness against modality degradation. In this work, empirical robustness evaluation is conducted under missing and noisy NVD descriptions, while structured-metadata perturbations are used during training as a robustness regularization mechanism.

lightweight multimodal design that combines NVD descriptions and structured metadata through robust decision-level fusion, making it practical when textual or structured inputs are incomplete or degraded.

3 | Proposed Architecture

The general architecture of SecureFusion-X is presented in Figure 1, where we propose a robustness-oriented multimodal architecture engineered to predict the likelihood of software vulnerability exploitation in operational environments. The core design philosophy hinges on synthesizing the semantic depth of unstructured NVD descriptions with the standardized predictive signals inherent in NVD-derived structured metadata. By conceptualizing these two modalities as complementary perspectives of a unified threat profile, SecureFusion-X maintains stable performance even when inputs are fragmented, noisy, or delayed conditions that represent the “norm” rather than the exception in real-world security triage.

Formally, each vulnerability instance is modeled as a triplet $(x^{\text{text}}, x^{\text{tab}}, \text{and } y)$, where x^{text} denotes the NVD textual description, x^{tab} represents the vector of NVD-derived numerical and categorical structured features, and $y \in \{0, 1\}$ indicates whether publicly documented exploit evidence is present in the adopted GreenExBert label source. SecureFusion-X learns a robust predictive mapping $f: (x^{\text{text}}, x^{\text{tab}}) \mapsto \hat{y} \in [0, 1]$.

3.1 | Dataset and Feature Extraction

The experimental dataset is constructed by combining publicly accessible vulnerability records from the NVD with curated CVE-level binary exploit-evidence labels adopted from the GreenExBert dataset [45].

The raw vulnerability records used for feature extraction were obtained from the official NVD Data Feeds provided by the National Institute of Standards and Technology (NIST).

Specifically, we downloaded the NVD JSON 2.0 yearly CVE feeds for 2016–2024 from the official NVD Data Feeds portal [46]. These feeds provide NVD vulnerability data in the JSON 2.0 format and include yearly CVE records. NVD data are publicly accessible for retrieval, display, and analysis under the NVD terms of use [47]. Therefore, no additional permission was required for using and analyzing the NVD records beyond compliance with the NVD terms and proper attribution.

The binary exploit-evidence labels were adopted from the curated labeled dataset introduced in the GreenExBert study [45]. The GreenExBert-labeled dataset was obtained directly from its authors upon request, with permission for use in the present research. In that work, exploit and non-exploit labels were constructed by merging public exploit records from Exploit-DB with NVD metadata and applying targeted sampling to obtain a class-balanced dataset containing exploitable and non-exploitable vulnerabilities. In the present work, this prior-labeled dataset is used only as the source of curated CVE-level binary exploit-evidence labels. The textual and structured input features used by SecureFusion-X were extracted independently from the official NVD JSON 2.0 feeds. The rationale for adopting the GreenExBert label set is that the NVD provides standardized vulnerability descriptions and structured metadata. Although the NVD contains exploit-related indicators for some CVEs, these signals do not constitute a complete and uniformly available CVE-level binary exploit-evidence label across the study cohort. A separate curated source of task-specific supervision was therefore required for our supervised classification formulation. GreenExBert was selected because it provides previously curated CVE-level binary labels derived from public exploit evidence, can be directly aligned with official NVD records through CVE identifiers, and allows the predictive inputs to be reconstructed independently from the NVD. Using an existing published label construction also avoids introducing a new study-specific labeling heuristic solely for the present experiments. In SecureFusion-X, GreenExBert is used only as the source of the prediction target.

Accordingly, the two data sources play distinct roles. The NVD is the standardized public source of vulnerability descriptions and structured attributes, including CVSS, CWE, CPE, vendor/

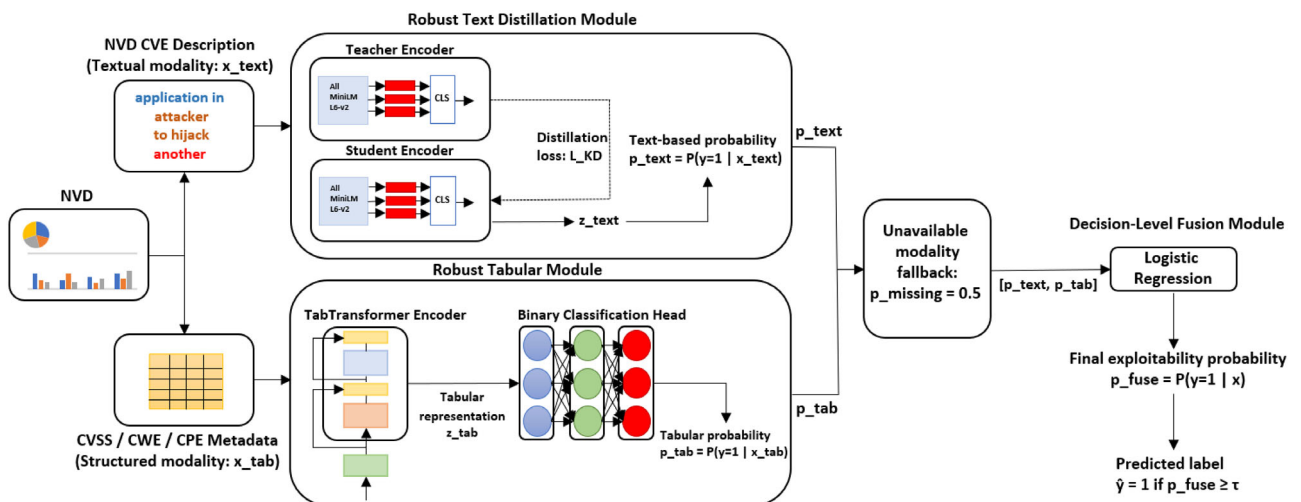


FIGURE 1 | Overview of the SecureFusion-X architecture. TextGuard-Lite and TabShield-Net generate branch-level probabilities p_{text} and p_{tab} , which are combined through a decision-level logistic-regression fusion module to produce the final exploitability probability p_{fuse} .

product, and temporal information, whereas GreenExBert supplies only the curated binary exploit-evidence labels.

The GreenExBert study employed targeted sampling to address the strong imbalance between exploit and non-exploit cases. Accordingly, the primary stratified evaluation in this work should be interpreted as a controlled assessment of multimodal exploitability prediction rather than as an estimate of exploitation prevalence in the complete NVD population. To provide a more challenging assessment of generalization, we additionally perform a forward temporal transfer evaluation across disclosure years, in which models trained on 2016–2022 CVEs are calibrated on 2023 CVEs and evaluated on 2024 CVEs.

After downloading the NVD feeds, we parsed the original JSON files and transformed the raw vulnerability records into a flat tabular representation. The extracted fields included CVE identifiers, English vulnerability descriptions, publication and modification dates, CVSS vector components, CWE metadata, CPE-derived vendor and product information, vulnerability status, and auxiliary reference metadata. The initial extraction from the 2016–2024 NVD feeds produced 202,095 CVE records. These NVD-derived records were then matched with the adopted GreenExBert exploit evidence labels using normalized CVE identifiers. When duplicate NVD records existed for the same CVE identifier, the most recently modified record was retained. Records with missing, non-binary, or inconsistent labels were removed.

To prevent target leakage, all exploit-reference-derived variables, reference-tag indicators, post-hoc exploitability and impact score fields, and constant fields were removed before model training, validation, and testing. The final 8549 instances should therefore be interpreted as a curated CVE cohort obtained by aligning the NVD-derived feature representation with the available GreenExBert exploit-evidence labels rather than as a random downsample of the complete NVD population.

Following this filtering and metadata normalization process, the resulting corpus contains 8549 labeled CVE instances. Each instance is assigned $y=1$ when publicly documented exploit evidence is present in the adopted GreenExBert label source and $y=0$ otherwise. Importantly, $y=0$ denotes the absence of documented exploit evidence in the adopted source and should not be interpreted as proof of intrinsic or permanent non-exploitability. The resulting cohort is approximately class-balanced, with 4275 positive and 4274 negative instances. This balance facilitates controlled model comparison but does not

reproduce the natural prevalence of exploitation in operational vulnerability populations.

A stratified split is utilized to generate 5812 training samples, 1454 validation samples, and 1283 testing samples, corresponding approximately to 68%, 17%, and 15% for training, validation, and testing, respectively. All dataset-derived preprocessing statistics, including normalization parameters, imputation values, and categorical encodings, were estimated exclusively from the training set. The validation split is used for neural checkpoint selection, fitting and tuning the fusion meta-classifier, and decision threshold calibration, while the test set remains untouched until final evaluation. The transformer tokenizer uses its fixed pre-trained vocabulary and is not fitted to the test set.

Textual Modality: We extract the raw “Description” field from the NVD records. As depicted in Figure 2, while the majority of descriptions span 20–80 tokens, the distribution shows a right-skewed long tail, with the x -axis truncated at 200 tokens for readability. To maintain computational efficiency without compromising the semantic coverage, sequences are truncated at 384 tokens. The 384-token limit was selected as a practical compromise between semantic coverage and computational efficiency: it covers the vast majority of NVD descriptions in our corpus while avoiding the higher memory and latency cost of the standard 512-token transformer limit. Minimal normalization is applied to preserve critical technical tokens, such as API names, file paths, and exploit-specific terminology.

Structured Modality: The tabular features include CVSS v3 vector components, CWE taxonomy identifiers, CPE indicators, non-exploit structural count features, and temporal attributes. After the leakage audit and removal of exploit-reference-derived, reference-tag, post-hoc exploitability and impact, and constant fields, the final structured representation consists of 41 retained features, including 25 categorical and 16 numerical features, as summarized in Table 2. It is important to note that $y=0$ represents the absence of publicly documented exploitation evidence in the adopted label source rather than absolute non-exploitability. This epistemic uncertainty motivates our use of robustness-aware multimodal learning: TextGuard-Lite provides resilience against generic text while TabShield-Net extracts latent patterns from standardized metadata.

3.2 | Data Complexity and Practical Challenges

Exploitability prediction involves several data-related challenges that extend beyond the size of the training corpus. Vulnerability

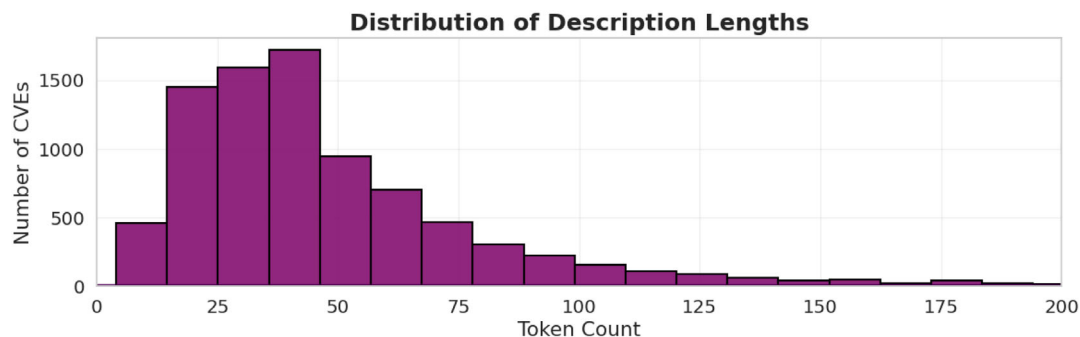


FIGURE 2 | Distribution of NVD description lengths in token counts.

TABLE 2 | Structured feature groups utilized in the tabular modality after leakage removal.

Feature group	Count	Type	Examples
CVSS v3 vector components	8	Categorical	AV, AC, PR, UI, S, C, I, A
CWE identifiers	2	Categorical	CWE main, CWE all
CPE-derived identifiers	1	Categorical	CPE-derived category indicator
Vendor and product identifiers	2	Categorical	Vendor and product identifiers
Other categorical metadata	12	Categorical	Vulnerability status, CVSS version, source identifier, temporal categories
Total categorical	25	—	—
CVSS numeric metric	1	Numerical	Base score
Temporal features	10	Numerical	Age, publication year, modification year, time intervals
Non-exploit structural count features	5	Numerical	CWE count, CPE count, vendor count, product count, reference-count metadata after leakage removal
Total numerical	16	—	—
Total structured features	41	—	25 categorical + 16 numerical
Removed leakage-prone fields	—	Excluded	Exploit-reference-derived variables, reference-tag indicators, post-hoc exploitability and impact fields, and constant fields

Note: The bold is on total numbers to highlight also the separator.

records are heterogeneous, temporally evolving, partially observed, and subject to uncertainty in both their attributes and exploit-evidence labels. In the present study, these issues are explicitly considered in the construction, preprocessing, and evaluation of the proposed multimodal learning pipeline.

First, vulnerability information is inherently heterogeneous. In addition to the unstructured NVD vulnerability description, the structured modality incorporates 41 NVD-derived attributes covering CVSS vector components, CWE identifiers, CPE-derived information, vendor and product identifiers, temporal attributes, and non-exploit structural count features. The use of these complementary modalities enables the model to capture both the semantic information contained in vulnerability descriptions and structural characteristics represented by standardized vulnerability metadata.

Second, vulnerability records may contain incomplete, delayed, or unavailable information, particularly during the early stages of the vulnerability disclosure lifecycle. To address this issue, missing numerical values are imputed using statistics estimated exclusively from the training partition, while categorical attributes use a dedicated placeholder representation for the missing or previously unseen values. Missing textual descriptions are represented as empty inputs. Furthermore, stochastic perturbations are applied to both textual and structured modalities during training to reduce excessive dependence on complete and clean input information.

Third, exploitability labels are subject to observational uncertainty. In this study, the adopted binary labels indicate the presence or absence of publicly documented exploit evidence in the GreenExBert label source. Therefore, a negative label should not be interpreted as definitive evidence that a vulnerability is intrinsically non-exploitable or will never be exploited. This distinction is important because public exploit evidence may emerge after the initial vulnerability disclosure.

Fourth, vulnerability data are affected by temporal distribution shift. Vulnerability characteristics, affected software products,

reporting practices, and attacker behavior may change over time, meaning that the performance obtained under a random stratified split may not fully represent future deployment conditions. To explicitly examine this issue, we complement the primary stratified evaluation with a forward temporal transfer experiment in which CVEs from 2016 to 2022 are used for training, CVEs from 2023 are used for calibration, and CVEs from 2024 are used for evaluation. The substantial performance reduction observed under this setting demonstrates that temporal shift represents a significant practical challenge for exploitability prediction.

Finally, vulnerability exploitability datasets are particularly susceptible to target leakage because some metadata fields may directly or indirectly encode information associated with publicly available exploit evidence. To mitigate this risk, exploit-reference-derived variables, reference-tag indicators, post-hoc exploitability and impact fields, and constant fields are removed prior to model training, validation, and testing. All preprocessing statistics and encodings are estimated exclusively from the training partition and subsequently applied unchanged to the validation and test sets.

Taken together, these considerations clarify that the difficulty of exploitability prediction is not determined solely by the dataset size. Instead, it arises from the interaction of heterogeneous information sources, incomplete and degraded observations, uncertainty in exploit-evidence labels, leakage-prone metadata, and temporal distribution shift. The proposed experimental design therefore focuses not only on predictive performance under a controlled labeled cohort but also on the robustness and transfer limitations that emerge under less ideal data conditions.

3.3 | Data Preprocessing

The preprocessing pipeline was designed to enforce operational realism. We remove all features that could introduce target leakage, including exploit-reference indicators, reference-tag variables,

aggregate exploit-reference-derived fields, post-hoc exploitability and impact score fields, and constant fields.

These variables were retained only during the raw NVD parsing and auditing stage and were excluded before model training, validation, and testing because they directly or indirectly overlap with the exploit-evidence source used to construct the adopted binary labels.

CVSS vectors were decomposed into atomic components. Numerical features undergo Z-score standardization, while categorical attributes are label-encoded with a dedicated categorical placeholder index to accommodate unavailable or previously unseen categorical values. Missing descriptions are replaced with empty strings, and missing numerical values are imputed using training set medians. All preprocessing parameters are estimated exclusively on the training split and then applied unchanged to the validation and test splits.

3.4 | TextGuard-Lite: Efficient Textual Encoding

TextGuard-Lite extracts semantic exploitability cues using a lightweight, distilled transformer. A high-capacity teacher model is first fine-tuned using weighted binary cross-entropy to mitigate the class imbalance. A compact student model (MiniLM-based) is then distilled by retaining a reduced set of layers, significantly lowering the inference latency.

Let z^T and z^S denote the teacher and student logits, respectively. A temperature parameter, T , softens both distributions:

$$q^T = \sigma(z^T/T), \quad q^S = \sigma(z^S/T) \quad (1)$$

The student is optimized via a hybrid loss function:

$$\mathcal{L}_{\text{text}} = \lambda_{\text{CE}} \mathcal{L}_{\text{CE}} + \lambda_{\text{KD}} \mathcal{L}_{\text{KD}} \quad (2)$$

where $\mathcal{L}_{\text{CE}}$ ensures classification accuracy and $\mathcal{L}_{\text{KD}}$ aligns the student's decision boundaries with the teacher's softened predictions. To enhance robustness, descriptions are perturbed during training (fully masked or token-replaced with UNK) with probability p_{text} .

3.5 | Robustness Objective

We define robustness as the stability of exploitability predictions under realistic modality degradation. Let $\mathcal{M}_{\text{text}}$ and $\mathcal{M}_{\text{tab}}$ denote perturbation operators applied to the textual and structured modalities, respectively. The goal is to reduce the discrepancy between the prediction obtained from clean inputs and the prediction obtained after perturbing one or both modalities.

$$\mathbb{E}[|f(x^{\text{text}}, x^{\text{tab}}) - f(\mathcal{M}_{\text{text}}(x^{\text{text}}), \mathcal{M}_{\text{tab}}(x^{\text{tab}}))|] \approx 0 \quad (3)$$

This objective should be interpreted as a stability regularization criterion rather than an absolute mathematical guarantee. In practical vulnerability triage, missing or noisy inputs can lead to unstable risk estimates. For example, an early-stage CVE may contain only a generic or incomplete NVD description,

causing a text-only model to underestimate the exploitability. Conversely, token corruption or parsing noise may remove critical terms such as “remote,” “authentication,” or “privilege escalation,” leading to an artificially lower or higher risk score. Similarly, delayed or partially unavailable structured metadata may obscure important CVSS, CWE, or CPE signals. Such shifts can either hide truly exploitable vulnerabilities or generate excessive false positives for security analysts.

For decision-level fusion, this stability criterion can be related to the sensitivity of the fused probability. SecureFusion-X computes the final probability as

$$p^{\text{fuse}} = \sigma(w_1 p^{\text{text}} + w_2 p^{\text{tab}} + b) \quad (4)$$

where p^{text} and p^{tab} are the branch-level probabilities. Let $\tilde{p}^{\text{text}}$ and $\tilde{p}^{\text{tab}}$ denote the corresponding probabilities after modality perturbation. Since the sigmoid function is $\frac{1}{4}$ -Lipschitz, the change in the fused probability is bounded by

$$|p^{\text{fuse}} - \tilde{p}^{\text{fuse}}| \leq \frac{1}{4} (|w_1| |p^{\text{text}} - \tilde{p}^{\text{text}}| + |w_2| |p^{\text{tab}} - \tilde{p}^{\text{tab}}|) \quad (5)$$

This bound shows that reducing the sensitivity of each branch to missing or noisy inputs directly limits the sensitivity of the final fused risk score. In other words, if perturbations cause only small changes in the branch-level probabilities, then the decision-level fusion output is also constrained to vary smoothly. SecureFusion-X encourages this behavior by training the neural branches with corrupted samples while retaining the original ground truth labels and by calibrating the fusion threshold on the validation split. The stability claim is further verified empirically in the robustness experiments, where SecureFusion-X maintains stable F_1 scores across missing-text and noisy-text ratios, whereas the text-only branch degrades substantially under the missing textual evidence.

3.6 | TabShield-Net: Structured Feature Modeling

TabShield-Net utilizes a transformer-based architecture to model dependencies between categorical- and numerical-structured vulnerability features. Categorical features are embedded as $e_i = \mathbf{E}_{e_i}$, while numerical features x^{num} are projected as:

$$h^{\text{num}} = W_{\text{num}} x^{\text{num}} + b_{\text{num}} \quad (6)$$

The input sequence $H_0 = [h^{\text{CLS}}, h^{\text{num}}, e_1, \dots, e_{d_c}]$ is processed through self-attention layers:

$$\text{Attn}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \quad (7)$$

The final tabular prediction is derived from the CLS token: $p^{\text{tab}} = \sigma(W_{\text{cls}} h_L^{\text{CLS}} + b_{\text{cls}})$. Robustness in TabShield-Net is implemented through stochastic tabular perturbation during training. With probability $p_{\text{tab}} = 0.30$, a subset of tabular inputs is degraded to simulate delayed or missing structured metadata. Since numerical features are Z-score standardized, setting a numerical feature to zero corresponds to replacing it with its training-set mean, which acts as a neutral missing-value representation rather than an arbitrary value. For categorical attributes, each

feature vocabulary includes a dedicated categorical placeholder index. During perturbation, the original categorical value is replaced with this index, allowing the embedding layer to learn a representation for unavailable or previously unseen categorical metadata. These perturbations are applied only during training while preserving the original ground-truth label, encouraging the tabular branch to avoid excessive dependence on any single structured field and to remain stable when CVSS, CWE, or CPE metadata are incomplete or delayed at inference time.

3.7 | SecureFusion-X: Resilient Late Fusion

Rather than feature-level concatenation, SecureFusion-X employs decision-level fusion. Let p^{text} and p^{tab} be the branch probabilities. The fusion module is implemented as a regularized logistic regression meta-classifier operating on the two branch probabilities:

$$z_{\text{fuse}} = w_1 p^{\text{text}} + w_2 p^{\text{tab}} + b \quad (8)$$

$$p^{\text{fuse}} = \sigma(z_{\text{fuse}}) \quad (9)$$

The regularization strength and decision threshold were selected on the validation split. In the final tuned configuration, the fusion model used $C=1.0$ and a decision threshold of 0.74. The training procedure is summarized in Algorithm 1.

3.8 | Training and Calibration

The neural branch training stages are stabilized via gradient clipping, while the fusion stage is trained using a regularized logistic regression meta-classifier. Decision thresholds are not fixed at 0.5; instead, they are calibrated on the validation split to maximize the F_1 score, ensuring alignment with operational security priorities. The final fusion configuration and decision threshold were selected based on the optimal validation F_1 score. In the final configuration, the selected regularization parameter is $C=1.0$ and the decision threshold is 0.74.

4 | Experimental Setup

The neural branches were implemented in PyTorch, while the decision-level fusion meta-classifier was implemented using scikit-learn logistic regression. Hugging Face Transformers were used for text encoding. All experiments were conducted in a Google Colab GPU environment running Linux 6.6.122 on a single NVIDIA Tesla T4 GPU with CUDA 12.8 acceleration and mixed-precision training. Early stopping was applied based on the validation F_1 .

All experiments were implemented in Python 3.12.13. Neural models were implemented using PyTorch 2.11.0+cu128 and Hugging Face Transformers 5.0.0. Data preprocessing, feature engineering, model calibration, and evaluation were performed using pandas 2.2.2, NumPy 2.0.2, and scikit-learn 1.6.1. For the primary stratified experiments, random seeds were fixed to 42 for Python, NumPy, and PyTorch to improve reproducibility. The

temporal-transfer adaptation experiments were separately repeated across ten random seeds, as described in the corresponding evaluation section.

Following the preprocessing procedure in Section 3.1, the dataset was partitioned using a stratified split into 5812 training samples, 1454 validation samples, and 1283 test samples.

4.1 | Training Configuration

Table 3 summarizes the hyperparameters for each component of the SecureFusion-X. TextGuard-Lite shares the tokenizer and maximum sequence length with the transformer teacher. The student encoder is obtained by distilling a reduced set of teacher layers. TabShield-Net is trained independently on the structured modality, while the fusion module is a lightweight logistic-regression meta-classifier operating on unimodal probability outputs.

The hyperparameters in Table 3 were selected using the validation split. In addition to these settings, student distillation used a temperature of $T=3.0$ and a hybrid cross-entropy–distillation objective. Text perturbation was applied with probability $p_{\text{text}}=0.15$, while tabular perturbation was applied with probability $p_{\text{tab}}=0.30$. Teacher, student, and TabShield-Net checkpoints were selected based on validation F_1 with early stopping where applicable. The final fusion decision threshold was also calibrated on the validation split to maximize F_1 .

4.2 | Robustness-Oriented Training and Evaluation

To reflect realistic deployment conditions—where textual descriptions may be incomplete, generic, redacted, or noisy—SecureFusion-X incorporates stochastic modality perturbations during training. TextGuard-Lite is exposed to corrupted input descriptions with probability $p_{\text{text}}=0.15$, implemented either by fully masking the entire text (missing text) or by replacing a random subset of tokens with the UNK token (noisy text). Similarly, TabShield-Net is trained under tabular perturbations applied with probability $p_{\text{tab}}=0.30$, including zeroing numerical features, replacing categorical attributes with dedicated categorical placeholder indices, or injecting Gaussian noise into normalized numerical fields. These perturbations enable both unimodal branches to learn stable representations that generalize robustly to the missing or unreliable metadata.

The primary robustness evaluation is conducted under three representative conditions: clean inputs, a 30% missing-text setting in which selected descriptions are fully masked, and a 30% noisy-text setting in which selected descriptions undergo token-level corruption. In addition, a sensitivity analysis is conducted using the missing-text and noisy-text ratios from 0% to 50% in 10% increments. During the evaluation, perturbations are applied exclusively to the textual modality to isolate their effect on the fusion mechanism. For all components of SecureFusion-X, decision thresholds are tuned on the validation split to maximize F_1 , and the final model configuration is selected according to the validation performance.

Require: Training set $\mathcal{D}$, validation set $\mathcal{D}_{\text{val}}$, perturbation probabilities p_{text} and p_{tab} , and distillation temperature T .

Ensure: Trained text branch f_{text} , tabular branch f_{tab} , and fusion model f_{fuse} .

1: **Stage 1: Teacher Fine-tuning**

2: **for** each minibatch (x^{text}, y) sampled from $\mathcal{D}$ **do**

3: Encode x^{text} using the teacher model and compute teacher logits z^T .

4: Update teacher via weighted BCE loss.

5: **end for**

6: Select the best teacher checkpoint based on validation F_1 .

7: **Stage 2: Student Distillation (TextGuard-Lite)**

8: **for** each minibatch (x^{text}, y) sampled from $\mathcal{D}$ **do**

9: $x^{\text{text}} \leftarrow \mathcal{M}_{\text{text}}(x^{\text{text}})$ with probability p_{text} .

10: Compute q^T and q^S as $q^T = \sigma(z^T/T)$ and $q^S = \sigma(z^S/T)$.

11: Update student via $\mathcal{L}_{\text{text}} = \lambda_{\text{CE}} \mathcal{L}_{\text{CE}} + \lambda_{\text{KD}} \mathcal{L}_{\text{KD}}$.

12: **end for**

13: Select the best student checkpoint based on validation F_1 and set it as f_{text} .

14: **Stage 3: Robust Tabular Training (TabShield-Net)**

15: **for** each minibatch (x^{tab}, y) sampled from $\mathcal{D}$ **do**

16: $x^{\text{tab}} \leftarrow \mathcal{M}_{\text{tab}}(x^{\text{tab}})$ with probability p_{tab} .

17: Compute the tabular prediction $p^{\text{tab}} = f_{\text{tab}}(x^{\text{tab}})$.

18: Update TabShield-Net via BCE loss.

19: **end for**

20: Select the best tabular checkpoint based on validation F_1 and set it as f_{tab} .

21: **Stage 4: Late-Fusion Meta-learning**

22: Compute text-branch validation probabilities:

23: $p_{\text{val}}^{\text{text}} \leftarrow f_{\text{text}}(x_{\text{val}}^{\text{text}})$.

24: Compute tabular-branch validation probabilities:

25: $p_{\text{val}}^{\text{tab}} \leftarrow f_{\text{tab}}(x_{\text{val}}^{\text{tab}})$.

26: Construct the fusion input matrix $X_{\text{fuse}} \leftarrow [p_{\text{val}}^{\text{text}}, p_{\text{val}}^{\text{tab}}]$.

27: Fit a regularized logistic-regression meta-classifier on $(X_{\text{fuse}}, y_{\text{val}})$.

28: Tune the regularization parameter C and decision threshold τ on the validation split to maximize F_1 .

29: Set the selected meta-classifier as f_{fuse} and retain τ for inference.

30: **return** $(f_{\text{text}}, f_{\text{tab}}, f_{\text{fuse}})$.

4.3 | Overall Exploitability Prediction Performance

TabShield-Net was retrained and re-evaluated using the final leakage-free structured representation containing 41 retained features after the removal of exploit-reference-derived variables, reference-tag indicators, post-hoc exploitability and impact fields, and constant fields. SecureFusion-X was subsequently refitted and evaluated using the predictions produced by this leakage-free TabShield-Net branch, together with TextGuard-Lite.

TextGuard-Lite operates exclusively on NVD vulnerability descriptions and therefore was not affected by the structured feature removal procedure. Table 4 summarizes the performance of the three evaluated models under clean test conditions (1283 CVEs). The TextGuard-Lite branch achieves an F_1 of 0.8608 and a Receiver Operating Characteristic - Area Under the Curve (ROC-AUC) of 0.9422, confirming that NVD descriptions contain meaningful semantic cues related to attack feasibility and adversarial capability. The structured TabShield-Net branch—leveraging CVSS vectors, CWE metadata, CPE

TABLE 3 | Training configuration and hyperparameters for each component.

Component	Learning rate or key parameter	Epochs and batch size
Text encoder (Teacher MiniLM)	3×10^{-5}	10 epochs, batch = 8
Text encoder (Student)	3×10^{-5}	6 epochs, batch = 8
TabShield-Net (TabTransformer)	5×10^{-4}	50 epochs, batch = 256
SecureFusion-X (Fusion Meta-Model)	$C = 1.0$	Threshold = 0.74
Optimizer	AdamW with gradient clipping for neural branches; regularized logistic regression for fusion	

indicators, and temporal and numerical attributes—shows a substantial performance advantage over the text-only branch. It achieves strong scores across nearly all metrics, including $F_1 = 0.9766$, ROC-AUC = 0.9978, balanced accuracy = 0.9766, and MCC = 0.9532. These results highlight the strong predictive signal embedded in standardized vulnerability metadata. SecureFusion-X integrates both modalities through a lightweight decision-level fusion meta-model. It is important to note that clean-test performance is primarily driven by the structured metadata branch: TabShield-Net already achieves a strong stand-alone F_1 score of 0.9766, while SecureFusion-X improves this to 0.9798. Therefore, the contribution of TextGuard-Lite under clean and complete metadata conditions should be interpreted as complementary rather than dominant. The fused model provides modest but consistent gains in F_1 , recall, MCC, and Brier score, indicating improved classification balance and calibration under the selected validation-calibrated decision threshold. Although TabShield-Net retains the highest ROC-AUC and Precision-Recall Area Under the Curve (PR-AUC), SecureFusion-X benefits from the additional semantic signal provided by vulnerability descriptions. Thus, SecureFusion-X should be viewed as a metadata-strong multimodal framework in which textual evidence provides complementary support, especially for operational settings where textual descriptions may be incomplete, generic, redacted, or noisy, while structured metadata remain available.

These branch-level results provide contribution analysis of the two specialized components. TextGuard-Lite evaluates the contribution of the distilled MiniLM textual encoder alone, TabShield-Net evaluates the contribution of the structured-metadata branch alone, and SecureFusion-X measures the effect of combining both branches through decision-level fusion.

In all performance tables, bold values indicate the best result in each column. Higher values are better for accuracy, balanced accuracy, precision, recall, specificity, F_1 , ROC-AUC, PR-AUC, MCC, and Kappa, whereas lower values are better for Brier score, False Positive Rate (FPR), and adaptation time. For Δ columns, the smallest absolute change indicates the most stable behavior. Ties are bolded for all tied models.

Figure 3 compares the ROC curves of the three evaluated prediction components under clean test conditions: the text-only TextGuard-Lite branch, the structured-metadata TabShield-Net branch, and the fused SecureFusion-X model. TextGuard-Lite achieves a ROC-AUC of 0.942, whereas TabShield-Net and

SecureFusion-X reach substantially higher ROC-AUC values of 0.998 and 0.995, respectively. This confirms that the structured and fused models provide stronger discrimination than the text-only branch, while SecureFusion-X remains close to the structured model. At the validation-calibrated threshold, both TabShield-Net and SecureFusion-X maintain a low FPR of 0.0218, which is particularly valuable in operational Security Operations Center (SOC) environments, where excessive false positives increase analyst workload and delay response times.

4.4 | Fusion Strategy Ablation

To address the effect of alternative multimodal fusion strategies, we conducted an additional ablation comparing the proposed decision-level logistic-regression fusion with two feature-level alternatives: embedding concatenation followed by a Multilayer Perceptron (MLP) and an attention-based feature-level fusion module. All variants were evaluated under the same leakage-free split using clean, 30% missing-text, and 30% noisy-text test settings.

As shown in Table 5, the attention-based feature-level model achieves the highest clean F_1 score of 0.9813, compared with 0.9798 for decision-level fusion and 0.9783 for feature-level concatenation. Under degraded text conditions, all fusion strategies remain stable. Decision-level fusion achieves $F_1 = 0.9806$ under both missing-text and noisy-text settings, showing comparable robustness despite its simpler architecture.

These results indicate that feature-level attention can provide marginal gains in clean-test performance; however, the decision-level strategy remains attractive for practical vulnerability management settings because it is simpler, modular, easier to calibrate, and compatible with degraded or unavailable textual inputs and independently updated model branches. Therefore, SecureFusion-X retains decision-level fusion as its main design, while deeper cross-attention and multi-level fusion mechanisms are left as promising future extensions.

4.5 | Training Dynamics

Figure 4a,b show the optimization behavior of the two neural branches. TextGuard-Lite converges rapidly, with the best validation performance reached at epoch 4, benefiting from softened teacher predictions during distillation. TabShield-Net also shows

TABLE 4 | Leakage-controlled performance of the three in-house models under clean test conditions.

Model	Accuracy	Bal. Acc.	Precision	Recall	Specificity	F_1	ROC-AUC	PR-AUC	MCC	Brier	Kappa	FPR
TextGuard-Lite	0.8636	0.8636	0.8797	0.8427	0.8846	0.8608	0.9422	0.9399	0.7279	0.1072	0.7272	0.1154
TabShield-Net	0.9766	0.9766	0.9781	0.9751	0.9782	0.9766	0.9978	0.9979	0.9532	0.0202	0.9532	0.0218
SecureFusion-X	0.9797	0.9797	0.9783	0.9813	0.9782	0.9798	0.9949	0.9948	0.9595	0.0182	0.9595	0.0218

Note: TabShield-Net was retrained using the final 41-feature leakage-free structured representation, and SecureFusion-X was refitted using the corresponding leakage-free branch outputs. Bold indicates the best values.

stable convergence and reaches its best validation performance at epoch 12, indicating effective learning of interaction patterns across numerical and categorical vulnerability attributes.

These training dynamics confirm the efficiency of the staged optimization pipeline. To substantiate the lightweight deployment claim quantitatively, we report parameter counts, model size, estimated computational cost, and measured inference time in Table 6. TextGuard-Lite contains 19.16M parameters, corresponding to approximately 73.11 MB in 32-bit Floating-Point (FP32) format. At a sequence length of 384, its counted computational cost is at least 2.72 Giga-Floating Point Operations (GFLOPs) per sample, and its measured single-sample inference time is 6.84 ms on the Tesla T4 environment. TabShield-Net contains 4.11M parameters, corresponding to approximately 15.67 MB in FP32 format, with an analytical Multiply-Accumulate (MAC)-style estimate of approximately 0.024 GFLOPs per sample. The decision-level fusion meta-classifier contains only three trainable parameters and has negligible scalar computational cost. Together with the three-parameter fusion meta-classifier, the complete framework contains approximately 23.27M trainable parameters and occupies approximately 88.78 MB in FP32 format. Full batched inference over the 2024 transfer set required 2.9479 s. These measurements quantitatively demonstrate the compact computational footprint of SecureFusion-X and the negligible overhead introduced by its decision-level fusion and lightweight adaptation procedures. Because computational results reported in prior studies were obtained under different hardware, software, sequence-length, and batching conditions, direct cross-publication latency and FLOP comparisons are not considered reliable.

The reported TextGuard-Lite FLOPs (Floating Point Operations) should be interpreted as a lower-bound estimate because some optimized Transformer operations, such as scaled dot-product attention and embedding lookup, are not fully counted by the tracing tool. The TabShield-Net operation count is an analytical MAC-style estimate based on the architecture.

4.6 | Robustness Under Missing or Noisy Text

To simulate degradation commonly observed early in the vulnerability disclosure lifecycle, we evaluate model stability when NVD descriptions are either fully masked (missing-text) or perturbed with token-level replacement noise (noisy-text). To further examine robustness beyond a single degradation level, we conduct a sensitivity analysis using missing-text and noisy-text ratios from 0% to 50% in 10% increments. Table 7 reports the resulting F_1 scores.

TextGuard-Lite exhibits clear performance degradation under missing-text conditions. Its F_1 decreases from 0.8608 under clean inputs to 0.7793 at 30% missing text and further to 0.7393 at 50% missing text. Under noisy-text conditions, the decline is more limited, with F_1 decreasing from 0.8608 to 0.8549 at 50% noisy text. This sensitivity highlights the dependence of text-only models on semantic completeness, especially when descriptions are sparse, generic, or partially corrupted.

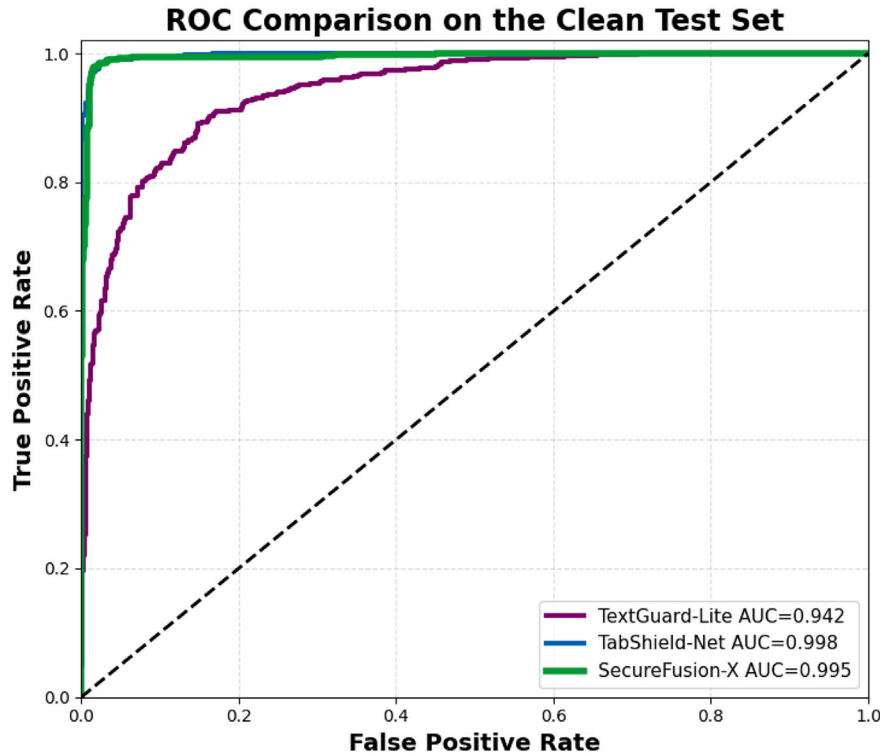
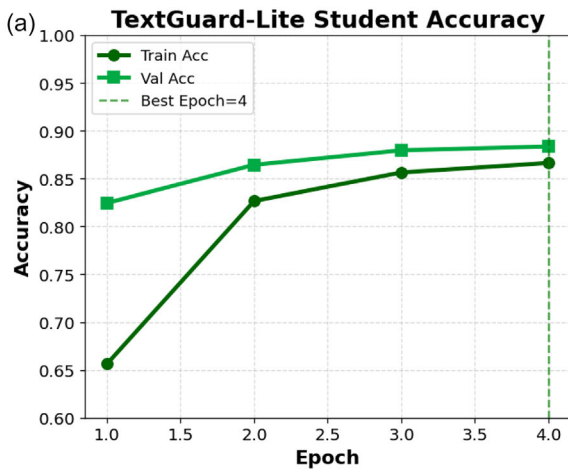


FIGURE 3 | ROC comparison of the three evaluated prediction components on the clean test set: TextGuard-Lite, TabShield-Net, and SecureFusion-X. TabShield-Net and SecureFusion-X show substantially stronger discrimination than TextGuard-Lite, with SecureFusion-X remaining close to the structured branch. The dashed diagonal line represents random classification.

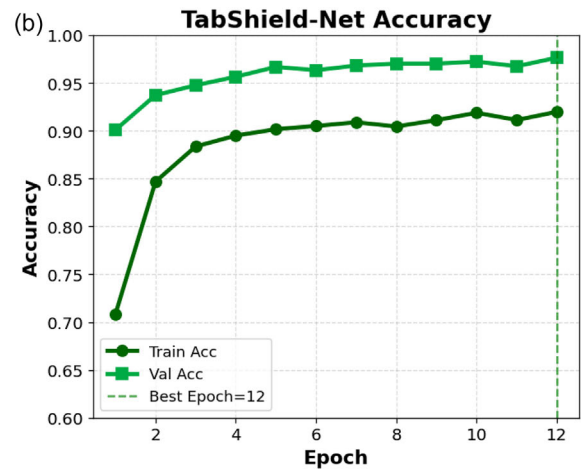
TABLE 5 | Fusion strategy ablation under clean and degraded-text settings.

Fusion strategy	Clean F_1	30% Missing F_1	30% Noisy F_1	Δ Missing	Δ Noisy
Decision-level LR	0.9798	0.9806	0.9806	+0.0008	+0.0008
Feature-level attention	0.9813	0.9820	0.9821	+0.0007	+0.0008
Feature-level concatenation	0.9783	0.9806	0.9798	+0.0023	+0.0015

Note: Bold values indicate the best result in each column; for Δ columns, the smallest absolute change indicates the most stable behavior.



Training and validation accuracy for TextGuard-Lite. The best validation checkpoint is reached at epoch 4.



Training and validation accuracy for TabShield-Net. The best validation checkpoint is reached at epoch 12.

FIGURE 4 | Training and validation accuracy of the two neural branches. (a) TextGuard-Lite student accuracy and (b) TabShield-Net accuracy.

TABLE 6 | Computational efficiency summary of SecureFusion-X components.

Component	Parameters	Model size	Estimated operations	Inference time
TextGuard-Lite student	19.16M	73.11 MB	≥ 2.72 GFLOPs per sample	6.84 ms per sample
TabShield-Net	4.11M	15.67 MB	≈ 0.024 GFLOPs per sample	—
Fusion meta-classifier	3	≈ 0 MB	Negligible	≈ 0.000001 ms per sample
Full 2024 inference	—	—	—	2.9479 s total
Threshold-only adaptation	0 trainable	—	—	0.1545 s
Fusion-only refit	3 trainable	—	—	2.5703 s

Note: Single-sample latency is reported for TextGuard-Lite, whereas the full 2024 inference time is measured using batched inference over the transfer set.

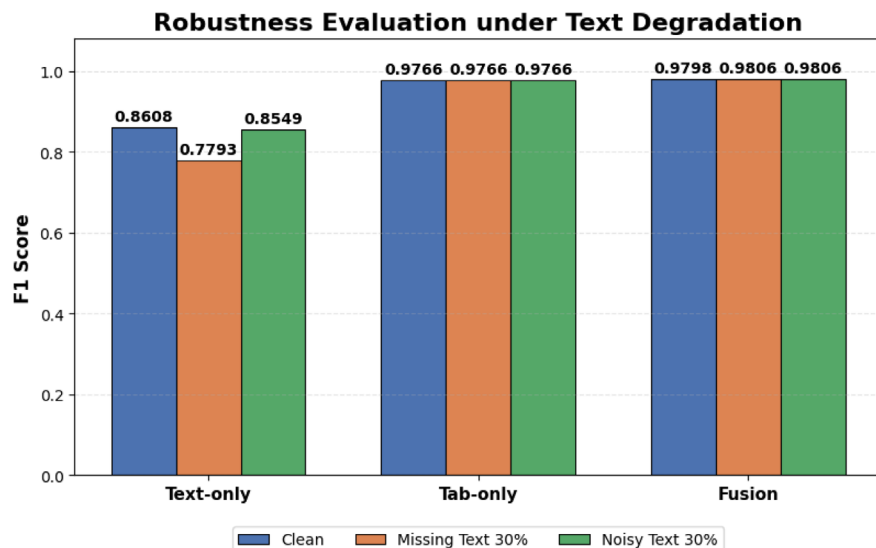
TABLE 7 | Robustness sensitivity analysis under varying missing-text and noisy-text ratios (F_1 scores).

Degradation	Ratio	TextGuard-Lite	TabShield-Net	SecureFusion-X
Missing text	0%	0.8608	0.9766	0.9798
Missing text	10%	0.8381	0.9766	0.9798
Missing text	20%	0.8107	0.9766	0.9798
Missing text	30%	0.7793	0.9766	0.9806
Missing text	40%	0.7635	0.9766	0.9798
Missing text	50%	0.7393	0.9766	0.9798
Noisy text	0%	0.8608	0.9766	0.9798
Noisy text	10%	0.8590	0.9766	0.9798
Noisy text	20%	0.8603	0.9766	0.9798
Noisy text	30%	0.8549	0.9766	0.9806
Noisy text	40%	0.8505	0.9766	0.9805
Noisy text	50%	0.8549	0.9766	0.9798

Note: Bold indicates the best values.

By contrast, TabShield-Net remains unaffected by text perturbations. Since the structured metadata are not altered in this robustness evaluation, its F_1 score remains identical across all degradation ratios at 0.9766.

SecureFusion-X maintains stable performance under both missing-text and noisy-text conditions. Across the full degradation range, its F_1 value remains between 0.9798 and 0.9806. These results confirm that the fusion model provides robust

**FIGURE 5** | Robustness under 30% missing-text and noisy-text conditions. SecureFusion-X remains stable, while TextGuard-Lite degrades under textual perturbation.

fallback behavior across a broad range of textual degradation levels rather than only at a single 30% setting. The small positive fluctuations observed under some degraded settings should be interpreted as threshold-level variation rather than systematic improvement caused by degradation.

Figure 5 visualizes the representative 30% degradation setting. The text-only model shows a clear drop under missing-text conditions, while the tabular branch remains unchanged because its structured inputs are not perturbed. SecureFusion-X remains stable across both missing-text and noisy-text settings. This stability is consistent with the strong structured branch remaining unaffected by textual perturbations and continuing to provide reliable input to the fixed decision-level fusion model. No dynamic gating or inference-time modality-switching mechanism is used; the logistic regression fusion model applies its fixed learned coefficients to the two branch probabilities.

4.7 | Temporal Transfer Evaluation

To evaluate transferability under a scenario different from the original stratified setting, we conducted a forward temporal transfer experiment. The neural models were trained on CVEs published during 2016–2022 and calibrated using 2023 CVEs. We first evaluated the resulting models on the 2024 transfer cohort without updating model parameters, which we report as the zero-shot setting. Separately, we investigated two lightweight few-shot adaptation strategies using only 10% of the 2024 transfer data. These adaptation experiments were repeated across ten random seeds and are reported as mean ± standard deviation.

For each adaptation run, the selected 10% subset of the 2024 transfer cohort was used exclusively for adaptation. All post-adaptation performance metrics were then computed on the disjointed remaining 90% of the 2024 cohort, which was not used for threshold selection or fusion refitting. This procedure was repeated across ten random splits. For consistency across adaptation runs, zero-shot performance was also evaluated on the corresponding held-out 90% evaluation subset for each of the ten random splits; therefore, zero-shot results are likewise reported as mean ± standard deviation.

Table 8 reports the temporal transfer results. TextGuard-Lite shows limited transferability under this shifted 2024 scenario, achieving an F_1 score of 0.1662 ± 0.0174 . TabShield-Net transfers more effectively, with $F_1 = 0.3880 \pm 0.0219$ and $\text{ROC-AUC} = 0.9098 \pm 0.0178$, indicating that structured metadata provides more stable cross-year signals. SecureFusion-X achieves the best zero-shot F_1 score of 0.3970 ± 0.0323 and the lowest FPR of 0.0511 ± 0.0033 , showing that decision-level fusion remains competitive under the forward temporal shift.

We also evaluated the transfer efficiency through two lightweight adaptation strategies using only 10% of the 2024 data. Threshold-only adaptation updates no model parameters and requires 0.1545 ± 0.0208 s. Fusion-only refitting updates only the three parameters of the logistic regression meta-classifier and requires 2.5703 ± 0.3575 s. Both TextGuard-Lite and TabShield-Net remain frozen during transfer. Full batched inference over the 2024 transfer set required 2.9479 s. These results show that SecureFusion-X

TABLE 8 | Repeated temporal transfer evaluation on 2024 CVEs under a train-2016–2022 and calibration-2023 protocol.

Setting	Model	Adaptation	F_1	ROC-AUC	PR-AUC	MCC	FPR	Adapt. time, s
Zero-shot transfer	TextGuard-Lite	None	0.1662 ± 0.0174	0.7961 ± 0.0205	0.1233 ± 0.0105	0.1637 ± 0.0239	0.1141 ± 0.0047	—
Zero-shot transfer	TabShield-Net	None	0.3880 ± 0.0219	0.9098 ± 0.0178	0.3824 ± 0.0313	0.4217 ± 0.0269	0.0668 ± 0.0030	—
Zero-shot transfer	SecureFusion-X	None	0.3970 ± 0.0323	0.8634 ± 0.0250	0.3304 ± 0.0343	0.4093 ± 0.0365	0.0511 ± 0.0033	—
Few-shot transfer	SecureFusion-X	Threshold-only	0.3296 ± 0.0846	0.8634 ± 0.0250	0.3304 ± 0.0343	0.3525 ± 0.0615	0.0871 ± 0.0576	0.1545 ± 0.0208
Few-shot transfer	SecureFusion-X	Fusion-only refit	0.3939 ± 0.0318	0.8629 ± 0.0247	0.3293 ± 0.0339	0.4067 ± 0.0360	0.0519 ± 0.0036	2.5703 ± 0.3575

Note: Values are reported as mean ± standard deviation over ten 10% adaptation splits. Adaptation time is reported only for methods that perform threshold or fusion adaptation; “—” indicates that no adaptation step is performed. Bold indicates best values.

can be evaluated and lightly recalibrated for new temporal scenarios with negligible computational overhead; however, in this experiment, threshold-only and fusion-only adaptation did not improve F_1 over the zero-shot SecureFusion-X setting.

4.8 | Practical Implications

Real-world vulnerability triage requires timely exploitability estimation under incomplete and evolving information. The computational efficiency and modality-robust design of SecureFusion-X suggest potential utility for vulnerability management workflows in which textual or structured information may be incomplete or degraded. In particular, the lightweight decision-level fusion and compact textual branch make the framework computationally suitable for further investigation in high-throughput settings such as SIEM, SOAR, and enterprise vulnerability management platforms. However, because the primary labeled cohort is curated and approximately class-balanced, additional validation under larger naturally imbalanced operational populations is required before making deployment-level claims.

4.9 | Comparison With State-of-the-Art Models

To contextualize the performance of SecureFusion-X, we compare it with previously reported exploitability prediction methods. Transformer-based approaches such as BERT-LSTM, all-mpnet, sentence-bert-base, and GreenExBert primarily rely on textual vulnerability representations. However, these methods were not evaluated under the same missing- and noisy-text protocol considered in the present study.

Cost-sensitive sequence models address class imbalance, whereas graph-based approaches such as ExPreHet and RGCN-based vendor-relational models exploit relational vulnerability information. Because these studies use different datasets, feature representations, and evaluation protocols, their reported results are used here only for literature-level contextualization rather than as controlled head-to-head comparisons. Table 9 reports the metrics

available from prior publications together with complete in-house results for the three main models implemented in this study. Among the three main in-house models reported in Table 9, SecureFusion-X achieves the strongest overall classification balance, with the highest accuracy, balanced accuracy, recall, F_1 , and MCC, as well as the lowest Brier score. Although TabShield-Net retains the highest ROC-AUC and PR-AUC, SecureFusion-X remains close in ranking performance while improving the validation-calibrated classification outcome. Most notably, SecureFusion-X maintains stable F_1 performance under 30% textual degradation, whereas the text-only branch is more sensitive to missing descriptions.

These results indicate that multimodal fusion and robustness-aware training can provide complementary benefits for exploitability prediction when vulnerability information is incomplete or degraded compared with relying on a single information source.

To ensure a fair interpretation of Table 9, we distinguish between controlled in-house comparisons and literature-based external comparisons. The three in-house models are evaluated using the same leakage-free dataset, identical training, validation, and test partitions, the same leakage control procedure, and the same validation-only model selection and threshold calibration protocol. Because the models process different modalities and use different architectures, each model applies the appropriate modality-specific preprocessing and hyperparameter configuration. Different hyperparameter settings may lead to different performance values; therefore, all reported in-house results correspond to configurations and checkpoints selected using validation performance, primarily validation F_1 , without access to the test set. Early stopping is applied where appropriate to reduce overfitting and select the best validation checkpoint; however, it is not interpreted as a guarantee against local optima. For external state-of-the-art methods, the reported values are taken from the original publications and may reflect different datasets, label definitions, class-balancing strategies, preprocessing procedures, hyperparameter search spaces, and evaluation protocols. These external results are therefore presented as contextual comparisons with the literature rather than as fully controlled same-dataset benchmark results.

TABLE 9 | Comparison with state-of-the-art exploitability prediction models.

Model	Acc	Bal. Acc	Prec	Recall	F_1	ROC-AUC	PR-AUC	MCC	Brier
BERT-LSTM [45]	0.925	—	0.915	0.937	0.926	—	—	—	—
All-mpnet [45]	0.932	—	0.934	0.930	0.932	—	—	—	—
Sentence-bert-base [45]	0.934	—	0.938	0.928	0.933	—	—	—	—
GreenExBert (Student) [45]	0.926	—	0.923	0.931	0.927	—	—	—	—
Hoque et al. (Cost-LSTM) [17]	0.920	—	0.890	0.980	0.940	0.970	—	—	—
ExPreHet [32]	0.8324	—	0.8322	0.8328	0.8325	0.8324	—	—	—
RGCN-VulVendorNet [15]	0.8062	—	0.8144	0.8345	0.8240	—	—	—	—
TextGuard-Lite (Ours)	0.8636	0.8636	0.8797	0.8427	0.8608	0.9422	0.9399	0.7279	0.1072
TabShield-Net (Ours)	0.9766	0.9766	0.9781	0.9751	0.9766	0.9978	0.9979	0.9532	0.0202
SecureFusion-X (Ours)	0.9797	0.9797	0.9783	0.9813	0.9798	0.9949	0.9948	0.9595	0.0182

Note: External results are literature-reported values and may not include all metrics, whereas our models are evaluated using the same leakage-free dataset, fixed data partitions, and validation-based model-selection protocol. Bold indicates best values.

5 | Conclusion

This study introduced SecureFusion-X, a resilient multimodal framework for vulnerability exploitability prediction. By integrating unstructured NVD descriptions with structured metadata from CVSS, CWE, and CPE sources, the framework addresses data instability and missing-modality conditions relevant to vulnerability management settings. The architecture utilizes an efficient student–teacher distillation process for textual analysis and a transformer-based metadata encoder, which are integrated through a decision-level logistic-regression meta-classifier.

A primary contribution of this work is the systematic integration of robustness into the learning objective. Through stochastic modality perturbations during training, SecureFusion-X is designed to maintain stable predictions when textual descriptions are sparse, redacted, noisy, or unavailable. Our experimental results on 8,549 curated NVD-aligned CVE records demonstrate that the framework achieves a strong clean-test F_1 -score of 0.9798 and maintains stable fused performance under the representative 30% textual degradation setting. These results demonstrate robustness within the adopted controlled benchmark, while the forward temporal transfer analysis highlights the additional difficulty of generalization across disclosure periods.

Furthermore, the distilled nature of the text encoder gives the framework a compact computational profile, while the results show that combining semantic and structured vulnerability signals provides modest complementary gains and stable performance under textual degradation. At the same time, the substantial performance reduction observed in the forward temporal transfer experiment highlights the difficulty of generalizing exploitability predictors across disclosure periods and emphasizes the need for continued evaluation under changing vulnerability distributions.

Future research will therefore focus on larger naturally imbalanced exploitability cohorts, additional independent label sources, real-time threat intelligence, and continual learning strategies for temporal adaptation. SecureFusion-X should consequently be viewed as a robustness-oriented multimodal foundation for vulnerability prioritization rather than as a fully deployment-validated operational predictor.

Acknowledgments

This work was supported in part by Project Security and Rights in the Cyberspace (SERICS) through the Ministry of University and Research (MUR) National Recovery and Resilience Plan funded by the European Union–NextGenerationEU under Grant PE00000014 (for authors S. L. Mirtaheri and A. Pugliese). During the preparation of this manuscript, the authors used Gemini solely for language polishing and assistance with LaTeX formatting. The tool was not used to generate or analyze research data or to determine the scientific results or conclusions. All AI-assisted text was reviewed, verified, and edited by the authors, who take full responsibility for the final content.

Open access publishing facilitated by Universita della Calabria, as part of the Wiley - CRUI-CARE agreement.

Funding

The authors have nothing to report.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The NVD records used in this study are publicly available through the official NVD Data Feeds portal. The GreenExBert-labeled dataset was obtained directly from its authors with permission for use in the present research and is not redistributed through our repository. Requests for access to the GreenExBert dataset should be directed to its original authors. The implementation, trained models, preprocessing and feature-construction scripts, and experimental outputs developed in this study are publicly available at: <https://github.com/maryamdolati/SecureFusion.git>.

References

1. S. Elder, M. R. Rahman, G. Fringer, K. Kapoor, and L. Williams, “A Survey on Software Vulnerability Exploitability Assessment,” *ACM Computing Surveys* 56, no. 8 (2024): 1–41.
2. E. Fedorchenko, N. Busko, and E. Novikova, “Automated Assessment of the Exploits Using Deep Learning Methods,” in *International Conference on Risks and Security of Internet and Systems* (Springer, 2025), 509–524.
3. M. Aggarwal, “A Study of CVSS v4. 0: A CVE Scoring System,” in *2023 6th International Conference on Contemporary Computing and Informatics (IC3I)*, vol. 6 (IEEE, 2023), 1180–1186.
4. B. Jung, Y. Li, and T. Bechor, “CAVP: A Context-Aware Vulnerability Prioritization Model,” *Computers & Security* 116 (2022): 102639.
5. F. R. Parente, E. B. Rodrigues, and C. L. C. Mattos, “Frape: A Framework for Risk Assessment, Prioritization and Explainability of Vulnerabilities in Cybersecurity,” *Journal of Information Security and Applications* 89 (2025): 103971.
6. J. Jacobs, S. Romanosky, O. Suciu, B. Edwards, and A. Sarabi, “Enhancing Vulnerability Prioritization: Data-Driven Exploit Predictions With Community-Driven Insights,” in *2023 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)* (IEEE, 2023), 194–206.
7. S. L. Mirtaheri, A. Pugliese, N. Movahedkor, and A. H. Majd, “Advanced Automated Vulnerability Scoring: Improving Performance With a Fine-Tuned Bert-CNN Model,” in *2024 11th International Symposium on Telecommunications (IST)* (2024), 109–113.
8. J. Yin, M. Tang, J. Cao, and H. Wang, “Apply Transfer Learning to Cybersecurity: Predicting Exploitability of Vulnerabilities by Description,” *Knowledge-Based Systems* 210 (2020): 106529.
9. M. Bozorgi, L. K. Saul, S. Savage, and G. M. Voelker, “Beyond Heuristics: Learning to Classify Vulnerabilities and Predict Exploits,” in *KDD* (2010), 105–114.
10. M. Edkrantz and A. Said, “Predicting Cyber Vulnerability Exploits With Machine Learning,” in *Thirteenth Scandinavian Conference on Artificial Intelligence (IOS Press, 2015)*, 48–57.
11. E. Iannone, G. Sellitto, E. Iaccarino, F. Ferrucci, A. De Lucia, and F. Palomba, “Early and Realistic Exploitability Prediction of Just-Disclosed Software Vulnerabilities: How Reliable Can It Be?,” *ACM Transactions on Software Engineering and Methodology* 33, no. 6 (2024): 1–41.
12. R. Croft, Y. Xie, and M. A. Babar, “Data Preparation for Software Vulnerability Prediction: A Systematic Literature Review,” *IEEE Transactions on Software Engineering* 49 (2021): 1044.
13. N. Tavabi, P. Goyal, M. Almukaynizi, P. Shakarian, and K. Lerman, “Darkembed: Exploit Prediction With Neural Language Models,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32 (2018).

14. Y. Fang, Y. Liu, C. Huang, L. Liu, and L. H. Son, "Fastembed: Predicting Vulnerability Exploitation Possibility," *PLOS ONE* 15, no. 2 (2020): e0228439.
15. M. Mahbub, M. S. A. Khan, T. Hamid, and M. S. Mia, "A Novel Vulnerability Exploit Prediction System Using the Relational Vulnerability-Vendor Network," *Digital Threats: Research and Practice* 6, no. 2 (2025): 1–17.
16. R. Watanabe, T. Matsunaka, A. Kubota, and J. Urakawa, "Machine Learning Based Prediction of Vulnerability Information Subject to a Security Alert," in *ICISSP* (2023), 313–320.
17. M. S. Hoque, N. Jamil, N. Amin, and K.-Y. Lam, "An Improved Vulnerability Exploitation Prediction Model With Novel Cost Function and Custom Trained Word Vector Embedding," *Sensors* 21, no. 12 (2021): 4220.
18. A. Costin, H. Turtiainen, N. Yousefnezhad, V. Bogulean, and T. Hämäläinen, "Evaluating Zero-Shot ChatGPT Performance on Predicting CVE Data From Vulnerability Descriptions," in *European Conference on Cyber Warfare and Security* (2024).
19. P. Kühn, D. N. Relke, and C. Reuter, "CVSS Prediction Based on Open Source Intelligence," *Computers & Security* 131 (2023): 103286.
20. H. Tiwari, "Advancing Vulnerability Classification With Bert: A Multi-Objective Learning Model," arXiv preprint arXiv:2503.20831 (2025).
21. H. Kheddar, "Transformers and Large Language Models for Efficient Intrusion Detection Systems: A Comprehensive Survey," *Information Fusion* 124 (2024): 103347.
22. V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert: Smaller, Faster, Cheaper and Lighter," arXiv preprint arXiv:1910.01108 (2019).
23. X. Jiao, Y. Yin, L. Shang, et al., "Tinybert: Distilling Bert for NLP," arXiv preprint arXiv:1909.10351 (2019).
24. Z. Sun, H. Yu, X. Song, R. Liu, Y. Yang, and D. Zhou, "Mobilebert: A Compact Bert for Resource-Limited Devices," arXiv preprint arXiv:2004.02984 (2004).
25. O. Zafrir, G. Boudoukh, P. Izsak, and M. Wasserblat, "Q8Bert: Quantized 8-Bit Bert," in *EMC2-NIPS* (2019), 36–39.
26. Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A Lite Bert," arXiv preprint arXiv:1909.11942 (2019).
27. S. Kai, F. Shi, and J. Zheng, "Vuldistilbert: CPS Vulnerability Severity Prediction," *Security and Communication Networks* 2023 (2023): 2118305.
28. L. Liu, Z. Hui, G. Chen, T. Cai, and C. Zhou, "A Lightweight Transformer Based Multi Task Learning Model With Dynamic Weight Allocation for Improved Vulnerability Prediction," *Scientific Reports* 15 (2025): 28176.
29. R. Meléndez, M. Ptaszynski, and F. Masui, "Comparative Investigation of Traditional Machine-Learning Models and Transformer Models for Phishing Email Detection," *Electronics* 13 (2024): 4877.
30. M. A. Uddin and I. H. Sarker, "An Explainable Transformer-Based Model for Phishing Email Detection: A Large Language Model Approach," arXiv preprint arXiv:2402.12345 (2024).
31. U. P. Sajad, "Explainable Transformer-Based Email Phishing Classification With Adversarial Robustness," arXiv Preprint (2025).
32. G. Xu, X. Chen, X. Cai, and D. Yu, "Exploitability Prediction of Vulnerabilities Based on Heterogeneous Graphs," *Knowledge-Based Systems* 330 (2025): 114517.
33. S. Lim and J.-C. Kim, "Multi-Modal Machine Learning for Automated CVSS Severity Prediction for Conservative Vulnerability Assessment," *IEEE Access* 13 (2024): 189344.
34. S. Chavan and G. Pappas, "From Detection to Decision: Transforming Cybersecurity With Deep Learning and Visual Analytics," *AI* 6 (2025): 214.
35. R. Wu, H. Wang, H.-T. Chen, and G. Carneiro, "Deep Multimodal Learning With Missing Modality: A Survey," arXiv preprint (2024).
36. M. K. Reza, A. Prater-Bennette, and M. S. Asif, "Robust Multimodal Learning With Missing Modalities via Parameter-Efficient Adaptation," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 47 (2023): 742.
37. Y. Dai, H. Chen, J. Du, et al., "A Study of Dropout-Induced Modality Bias on Robustness to Missing Video Frames for Audio-Visual Speech Recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2024).
38. Y. Yamashita, A. Taya, and Y. Tobe, "Robust Inference Under Modality Missing in Multimodal Split Learning," in *Proceedings of the Federated Learning and Edge AI for Privacy and Mobility* (2025).
39. S. L. Mirtaheri, A. Pugliese, and V. Pascucci, "Automated Vulnerability Score Prediction Through Lightweight Generative AI," *Knowledge-Based Systems* 329 (2025): 114406.
40. A. M. Høst, P. Lison, and L. Moonen, "A Systematic Approach to Predict the Impact of Cybersecurity Vulnerabilities Using Llms," arXiv e-prints (2025), arXiv-2508.
41. Y. Xu, C. Zhang, and H. Li, "Transformer-Based Large Vision Model for Universal Structural Damage Segmentation," *Automation in Construction* 176 (2025): 106256.
42. S. Li, M. Dang, Y. Xu, A. Wang, and Y. Guo, "Bridge Damage Description Using Adaptive Attention-Based Image Captioning," *Automation in Construction* 165 (2024): 105525.
43. Z. Shi, N. Matyunin, K. Graffi, and D. Starobinski, "Uncovering CWE-CVE-CPE Relations With Threat Knowledge Graphs," *ACM Transactions on Privacy and Security* 27, no. 1 (2024): 1–26.
44. A. Høst, P. Lison, and L. Moonen, "Constructing a Knowledge Graph From Textual Descriptions of Software Vulnerabilities in the National Vulnerability Database," in *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)* (2023), 386–391.
45. S. L. Mirtaheri, M. Yousefikia, A. Majd, et al., "Towards Green Generative Ai for Exploit Prediction in Resource-Limited Systems," in *IEEE Cyber Science and Technology Congress* (2025).
46. National Institute of Standards and Technology, "National Vulnerability Database: Data Feeds," (2025), accessed June 2, 2026, <https://nvd.nist.gov/vuln/data-feeds>.
47. National Institute of Standards and Technology, "National Vulnerability Database: Terms of Use," (2025), accessed June 2, 2026, <https://nvd.nist.gov/developers/terms-of-use>.