



Exposing Relational Identity Leakage in CAPEC Attack Pattern Classification with XGBoost

Li Xionglüe

National University of Defense Technology, Hunan, China

* lixio@nudt.edu.cn

Abstract

Machine learning models for cybersecurity taxonomy classification are frequently trained on relational identity features (such as CWE, Tactic, and Technique identifiers) that may encode dataset-specific associations rather than transferable attack semantics, yet this risk is rarely assessed alongside label formation and missing-value handling in a single, integrated design. This study evaluates how feature representation, CAPEC label formation strategy, and imputation method jointly affect the classification of CAPEC attack patterns using XGBoost on the VULDAT dataset (610 CVE samples, 21 candidate features). Eighteen experimental combinations, crossing three feature scenarios, three imputation methods, and two label formation strategies, were evaluated using five-fold stratified group cross-validation, dummy-classifier and label-shuffle sanity checks, gain-based feature importance, out-of-fold error analysis, and the Wilcoxon signed-rank test. Retaining relational identity features yielded 0.9398-0.9418 mean accuracy and 0.8793-0.8816 F1-Macro, whereas keeping only structural relational counts caused accuracy to collapse to 0.6136 (F1-Macro 0.4110). Relational identity features accounted for 65.2% of gain-based importance, indicating that reported performance depends substantially on deterministic taxonomy identifiers rather than solely on relational complexity. Fixed first-listed CAPEC labeling consistently outperformed frequency-based labeling, while the imputation method had a negligible effect. These findings offer a reusable, leakage-aware evaluation protocol for CVE-CWE-CAPEC-ATT&CK classification research.

Keywords: CAPEC classification; XGBoost; data leakage; feature representation; VULDAT dataset

How to cite:

Exposing Relational Identity Leakage in CAPEC Attack Pattern Classification with XGBoost. (n.d.). Journal of Systems Engineering and Information Technology (JOSEIT). <https://doi.org/10.29207/q7qmhe27>

Received by the Editor: 2026-08-24

Revised: 2026-09-25

Accepted: 2026-09-25

Available Online: 2026-09-31

SDGs contributed:



© The Author(s) 2025 | This is an open-access article under the [CC BY 4.0 License](#)
Publisher: [Ikatan Ahli Informatika Indonesia](#)



1. Introduction

Software vulnerabilities remain among the primary entry points exploited in cyberattacks, and the volume of vulnerabilities disclosed each year has grown beyond what manual security analysis can reasonably absorb [1]. Machine learning has therefore become a practical response to automating vulnerability identification and classification at scale [1]. Its effectiveness, however, depends not only on algorithm selection but equally on the rigor of feature representation and evaluation design [2].

Vulnerability and attack-pattern information is systematically documented across four interconnected taxonomies: Common Vulnerabilities and Exposures (CVE), Common Weakness Enumeration (CWE), Common Attack Pattern Enumeration and Classification (CAPEC), and MITRE ATT&CK. CVE catalogs disclosed vulnerabilities, CWE classifies the underlying weakness types, CAPEC documents attack patterns, and ATT&CK captures adversary tactics and techniques observed in practice [3]. The relational links among these taxonomies form a graph-like structure that has motivated a growing body of automated classification research [4].

Research on automating this mapping has progressed along a fairly consistent trajectory: from lexical and embedding-based text similarity, toward transformer-based semantic representations, and more recently toward feature-rich tabular classifiers that incorporate the taxonomies' own relational identifiers directly (Table 1). Kanakogi et al. used TF-IDF and Sentence-BERT to map CVE text to CAPEC entries, correctly identifying 48 of 58 relations in their best configuration [3], and later found that simple word-frequency methods remained competitive with denser embeddings [5]. Grigorescu et al. introduced CVE2ATT&CK, a BERT-based mapper from CVE descriptions to ATT&CK techniques [6].

Othman et al. built VULDAT, benchmarking multiple sentence-transformer models and releasing a relational dataset spanning CVE, CWE, CAPEC, and ATT&CK [4], and a subsequent extension of that line of work confirms that ATT&CK Technique-level text remains the strongest single semantic signal for this mapping task [7]. In parallel, Kawaminami et al. combined a CAPEC ontology with an ensemble machine-learning classifier in an Attack Classification and Response framework and reported a 99.13% F1-score [8]. Across these studies, a consistent pattern emerges: reported performance tends to improve as models are given more of the taxonomies' own relational identity content (CWE-ID, TacticID, TechniqueID) alongside or instead of free text.

Table 1. Software and Hardware Supporting Table

Study	Year	Method	Feature Type	Leakage / Validity Check
Kanakogi et al. [3]	2021	TF-IDF + Sentence-BERT similarity	CVE/CAPEC text	No
Kanakogi et al. [5]	2022	Word-frequency vs. embeddings	CVE/CAPEC text	No
Grigorescu et al. [6]	2022	BERT (CVE2ATT&CK)	CVE description text	No
Othman et al. [4]	2024	Sentence transformers (VULDAT)	Text + relational metadata	No
Othman [7]	2026	Sentence transformers	ATT&CK description text	No
Kawaminami et al. [8]	2026	Ensemble ML + CAPEC ontology	Ontology + behavioral features	No
Bonhomme & Dulaunoy [9]	2026	Multi-label classifier (gold set)	Curated text, avoids derived chain	Yes (quantifies chain artifacts)
This study	2026	XGBoost (ablation design)	Relational ID / Attribute / Structural	Yes (ablation + dummy/shuffle)

This trend invites critical scrutiny rather than uncritical adoption. Taxonomy identity features carry highly specific entity identifiers that can encode pre-established, dataset-wide associations rather than the semantic or behavioral cues a model is meant to learn. When the same relational pairs recur across training and test partitions, this can produce structural leakage, in which reported performance reflects memorized associations rather than genuine generalization [10], [11], [12]. This concern has recently been substantiated directly: an independent, contemporaneous study of CVE-to-ATT&CK mapping found that the CWE-to-CAPEC-to-ATT&CK derivation chain, when used to construct training labels, is dominated by table-expansion artifacts and is unsuitable as a learning target for exactly this reason [9]. Yet the near-perfect results reported for CAPEC/ATT&CK ontology-based classifiers, including the 99.13% F1-score noted above [8], have not been accompanied by an explicit test of dependence on relational identity features, leaving open whether such performance would hold once identifier information is withheld.

Beyond feature representation, label formation and missing-value handling can independently shape classification outcomes. Because a single CVE may be associated with more than one CAPEC entry, a strategy is required to reduce this one-to-many relationship to a single label [4]. Meanwhile, missing values present in several dataset attributes require an imputation method, and different methods can materially change the resulting data representation before modeling [13], [14]. Despite the individual importance of feature representation, label formation, and imputation, their combined influence on CAPEC classification validity has not previously been evaluated within a single, integrated experimental design [15].

In summary, prior CAPEC/ATT&CK classification research has (i) increasingly relied on relational identity features without directly testing their contribution to leakage, (ii) rarely evaluated label formation and imputation choices jointly with feature representation, and (iii) seldom paired reported accuracy with sanity checks capable of detecting spurious performance. This gap matters in practice: a classifier that appears highly accurate only because it has memorized taxonomy identifiers offers little value for classifying newly disclosed vulnerabilities that have not yet been fully annotated, which is precisely the setting such tools are meant to serve.

To address this gap, this study aims to: (1) quantify how far CAPEC classification performance depends on relational identity features by systematically ablating them from the input representation; (2) determine how

CAPEC label formation strategy and missing-value imputation method independently and jointly affect classification accuracy and stability; and (3) establish whether reported performance reflects genuine learned association rather than memorization, using sanity-check baselines and statistical testing. These objectives are addressed through eighteen experimental combinations that cross three feature representation scenarios, three imputation methods, and two label formation strategies, evaluated with XGBoost on the VULDAT dataset under five-fold stratified group cross-validation, together with dummy-classifier and label-shuffle sanity checks, gain-based feature importance analysis, out-of-fold error analysis, and Wilcoxon signed-rank testing. In doing so, this study contributes a reusable, leakage-aware evaluation framework for CVE-CWE-CAPEC-ATT&CK classification research and provides a more realistic picture of what actually drives model performance in this setting.

2. Methods

Figure 1 illustrates the ten-stage research pipeline, from data collection through interpretation and conclusion. The design compares three feature representation scenarios, three missing-value imputation methods, and two CAPEC label formation strategies, and evaluates the resulting eighteen models through sanity checks, feature importance analysis, error analysis, and Wilcoxon statistical testing.

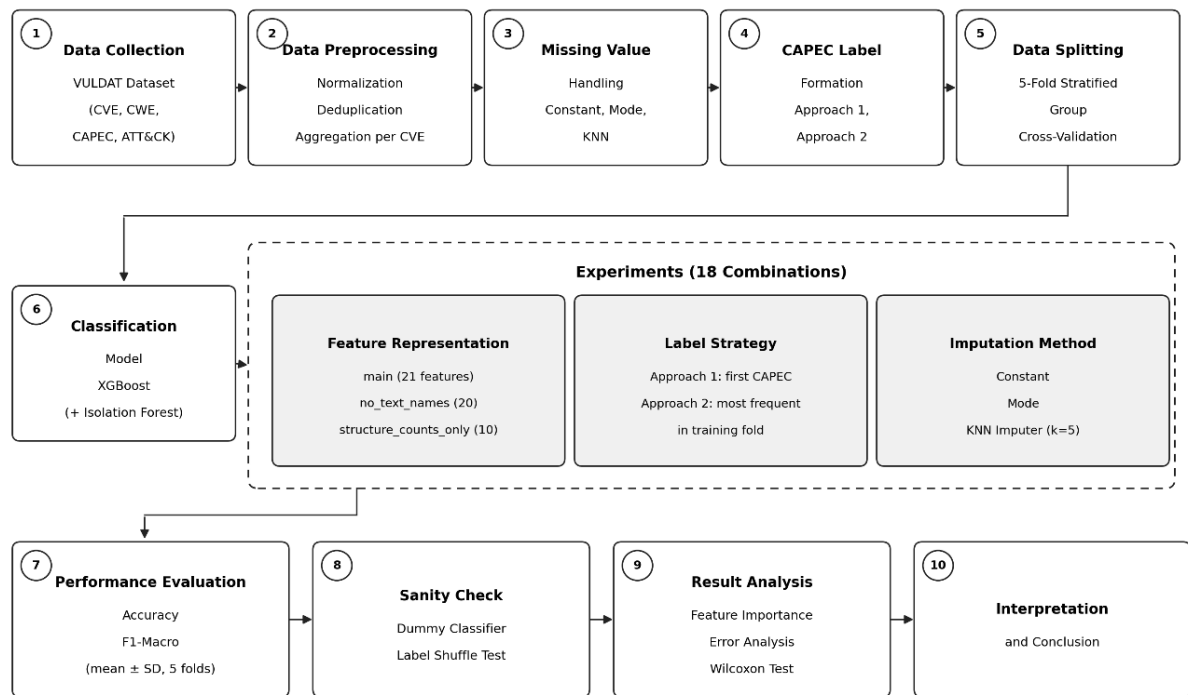


Figure 1. Research Methodology Pipeline

2.1 Preprocessing and Feature Engineering

This study uses the VULDAT dataset developed by Othman et al. and published through the Figshare repository [4]. The dataset contains 144,622 rows and 86 columns, where each row represents a relationship between a CVE vulnerability entry, a CWE weakness, a CAPEC attack pattern, and a MITRE ATT&CK tactic or technique; a single CVE may therefore appear across multiple rows. After preprocessing and aggregation described below, the dataset yields 610 unique samples, one per CVE. VULDAT was selected because it provides complete cross-taxonomy relational information from a single, publicly accessible source, which supports full reproducibility.

Preprocessing begins with text normalization across all string-type columns, including removal of excess whitespace and standardization of value formats. Feature selection then retains 15 columns containing relational information across the CVE-CWE-CAPEC-ATT&CK taxonomies from the original 86 [16]. The CWE-Likelihood of Exploit column is excluded because all its values are missing, and columns containing long descriptive text are omitted since this study focuses on non-textual feature representations. Selecting 15 of the 86 columns causes rows that previously differed only on excluded columns to become identical within the retained attribute space. Removing these duplicates reduces the dataset from 144,622 to 2,501 unique relational rows, which are then aggregated by CVE-ID so that each CVE is represented by a single row, producing 610 final samples.

The aggregation strategy varies by column type: taxonomy identity or name columns (CWE-ID, CAPECID, TacticID, TechniqueID, ATTACK_techniques_ID) are merged into sets of unique values separated by "|";

categorical attribute columns (Likelihood Of Attack, Typical Severity, CWE-Status) are filled with the mode of each CVE group [13]. Ten additional structural features are then constructed to capture the relational complexity of each CVE within the taxonomy network: `rel_row_count`, `num_unique_cwe`, `num_unique_attack_tech`, `num_unique_technique`, `num_unique_tactic`, `cve_year`, `TacticID_agg_count`, `TechniqueID_agg_count`, `ATTACK_techniques_ID_agg_count`, and `CWE-ID_agg_count`. Combining the 11 aggregated attributes with these 10 engineered features yields 21 input features for the full representation scenario. Table 2 lists the complete feature set under this grouping, which is the one used throughout the Results and Discussion sections.

Table 2. Complete Set of 21 Candidate Features Grouped by the Semantic Category

Category	Feature	Gain
Relational ID	TacticID_agg_count	0.0975
Relational ID	TechniqueName	0.0886
Relational ID	TechniqueID	0.0757
Relational ID	TacticID	0.0746
Relational ID	ATTACK_techniques_ID	0.0733
Relational ID	TacticName	0.0715
Relational ID	TechniqueID_agg_count	0.0680
Relational ID	CWE-Status	0.0372
Relational ID	CWE-Name	0.0202
Relational ID	CWE-Weakness Abstraction	0.0199
Relational ID	CWE-ID	0.0179
Relational ID	CWE-ID_agg_count	0.0046
Relational ID	ATTACK_techniques_ID_agg_count	0.0028
Structural	num_unique_tactic	0.0950
Structural	num_unique_attack_tech	0.0642
Structural	num_unique_technique	0.0453
Structural	num_unique_cwe	0.0085
Structural	rel_row_count	0.0011
Attribute	Typical Severity	0.0850
Attribute	Likelihood Of Attack	0.0446
Attribute	cve_year	0.0045

2.2 Classification Model and Evaluation Procedure

XGBoost (Extreme Gradient Boosting) is the primary classification model, chosen for its ability to handle tabular data with many features and imbalanced class distributions [18], [19]. A separate model is trained for each of the 18 combinations. Hyperparameters are held constant across all experiments: `n_estimators` = 300, `max_depth` = 6, `learning_rate` = 0.08, `subsample` = 0.85, `colsample_bytree` = 0.85, `min_child_weight` = 2, and `random_state` = 42. XGBoost builds decision trees sequentially, with each new tree correcting the residual error of the previous ensemble, governed by the regularized objective in Equation 1 [18]:

$$L(t) = \sum_{i=1}^n l(y_i, y_{\hat{i}}^{(t-1)} + f_t(x_i)) + \Omega(f_t) \quad (1)$$

with the loss function l being softmax cross-entropy (Equation 2):

$$l(y_i, y_{\hat{i}}) = - \sum_{k=1}^K 1[y_i = k] \log p_{ik} \quad (2)$$

where K is the number of CAPEC classes, $1[y_i = k]$ equals 1 if the true label is class k , and p_{ik} is the predicted probability of class k for sample i . The regularization term Ω controls tree complexity (Equation 3):

$$\Omega(f) = \gamma T + (1/2) \lambda \|w\|^2 \quad (3)$$

where T is the number of leaves, w is the vector of leaf weights, γ restricts the number of leaves that can be formed, and λ penalizes excessively large leaf weights to curb overfitting.

Isolation Forest (`n_estimators` = 100, `contamination` = 0.05) is applied as a supplementary, unsupervised analysis to flag CVEs with unusual relational profiles, without requiring labels [20], and has previously proven effective for detecting unusual relational patterns in vulnerability-related data [21]. The anomaly score is computed as in Equation 4:

$$s(x, n) = 2^{(-E[h(x)] / c(n))} \quad (4)$$

where $E[h(x)]$ is the average path length required to isolate sample x across all trees, compared against $c(n)$, the expected average path length for a dataset of size n (Equation 5):

$$c(n) = 2 * H(n-1) - 2 * (n-1) / n \quad (5)$$

where $H(i) = \ln(i) + 0.5772$ is the harmonic number. A score close to 1 indicates that a sample is easily isolated and is likely anomalous, while a score well below 0.5 indicates a normal sample [20].

Evaluation is carried out using five-fold Stratified Group Cross-Validation grouped by CVE-ID, ensuring that all rows originating from the same CVE always fall within the same partition [22], with Accuracy and F1-Macro as the reported metrics (Equations 6-7):

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN) \quad (6)$$

$$\text{F1-Macro} = (1/K) * \sum_{k=1}^K (2 * P_k * R_k) / (P_k + R_k) \quad (7)$$

where P_k and R_k are the precision and recall of class k . F1-Macro is treated as the primary metric because it weights every class equally regardless of frequency, making it more informative than accuracy for a highly imbalanced label space [23].

Two sanity checks are performed on all 18 combinations: a dummy classifier using the most_frequent strategy as a lower performance bound, and a label-shuffle test in which training labels are randomly permuted before retraining; a model that genuinely learns from its features should collapse under label shuffling [10]. Feature importance is computed from XGBoost's built-in gain score on the best-performing combination and aggregated by the three categories in Table 2. Configuration-level differences are tested pairwise with the Wilcoxon signed-rank test over the five per-fold accuracy values ($\alpha = 0.05$); with $n = 5$ folds the minimum attainable p-value is 0.0625, so statistical tests are treated as supporting evidence, with the main interpretation resting on the magnitude of the observed differences [24].

All experiments were implemented in Python using scikit-learn (cross-validation, KNN and mode imputation, dummy-classifier baseline, Isolation Forest), the official xgboost package (gradient boosting classifier), and scipy.stats (Wilcoxon signed-rank test), with the random seed fixed at 42 across all stochastic components. The VULDAT source data are publicly available through Figshare [4]; the preprocessing scripts and experiment configuration files used to produce the results below are available from the corresponding author upon reasonable request to support independent replication.

3. Results and Discussion

Table 3 and Figure 2 summarize accuracy and F1-Macro averaged across the six imputation-label combinations for each feature scenario. The main scenario (21 features) reaches 0.9398 accuracy and 0.8793 F1-Macro; removing only the CWE-Name column (no_text_names, 20 features) performs marginally better, at 0.9418 accuracy and 0.8816 F1-Macro, a gap below 0.003 in both metrics. Retaining only the ten structural count features (structure_counts_only) drops performance sharply, to 0.6136 accuracy and 0.4110 F1-Macro—a fall of roughly 33 accuracy points and 47 F1-Macro points relative to the other two scenarios.

Table 3. Mean Accuracy and F1-Macro

Feature Scenario	Mean Accuracy	Mean F1-Macro
main (21 features)	0.9398	0.8793
no_text_names (20 features)	0.9418	0.8816
structure_counts_only (10 features)	0.6136	0.4110

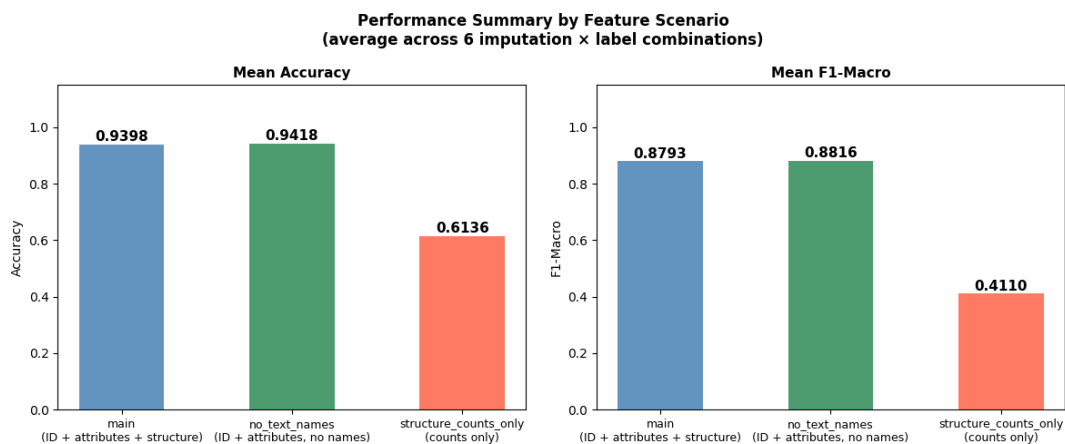


Figure 2. Mean Accuracy and F1-Macro

Table 4 reports the full 3 x 3 x 2 accuracy grid. Within the main and no_text_names scenarios, per-combination accuracy ranges from 0.9257 to 0.9523, with per-fold standard deviations of 0.0179-0.0222. Within structure_counts_only, accuracy is confined to 0.6058-0.6213 with higher standard deviations of 0.0372-0.0407, and-because the two imputed columns (Likelihood Of Attack, Typical Severity) are not part of this scenario's ten structural features-all three imputation methods yield numerically identical results within each label approach, which also serves as an internal consistency check on the experimental pipeline.

Table 4. Mean Accuracy

Scenario	Imputation	Approach 1: Acc (SD)	Approach 2: Acc (SD)
main	constant	0.9523 (0.0179)	0.9323 (0.0192)
main	mode	0.9490 (0.0205)	0.9257 (0.0211)
main	knn	0.9490 (0.0205)	0.9307 (0.0193)
no_text_names	constant	0.9473 (0.0222)	0.9357 (0.0213)
no_text_names	mode	0.9506 (0.0183)	0.9340 (0.0209)
no_text_names	knn	0.9490 (0.0213)	0.9340 (0.0209)
structure_counts_only	constant	0.6213 (0.0407)	0.6058 (0.0372)
structure_counts_only	mode	0.6213 (0.0407)	0.6058 (0.0372)
structure_counts_only	knn	0.6213 (0.0407)	0.6058 (0.0372)

3.1 Effect of CAPEC Label Formation Strategy

Approach 1 (fixed, first-listed CAPEC) outperforms Approach 2 (frequency-based, per-fold CAPEC) in all 18 combinations (Table 4). In the main scenario, the accuracy gap ranges from 0.0183 to 0.0233 depending on the imputation method, from 0.9523 vs. 0.9323 (constant) to 0.9490 vs. 0.9307 (KNN); the no_text_names scenario shows a similar 0.0116-0.0166 gap, and structure_counts_only shows a smaller but consistently directed gap (0.6213 vs. 0.6058). Stability, by contrast, shows no consistent winner: standard deviations for Approach 1 and Approach 2 differ by at most 0.0026 in the main and no_text_names scenarios, and the direction even reverses in structure_counts_only, where Approach 2 (SD = 0.0372) is more stable than Approach 1 (SD = 0.0407).

Across all scenario-approach pairs, accuracy differences among the three imputation methods never exceed 0.0066 points (Table 4). In the main scenario under Approach 1, constant (0.9523) marginally exceeds mode and KNN (both 0.9490); in no_text_names, the ordering shifts and mode (0.9506) exceeds constant (0.9473) and KNN (0.9490), showing that no single method is consistently superior. Standard deviations follow the same pattern, varying by at most 0.0039 within a given scenario-approach pair.

Table 5 summarizes the Wilcoxon signed-rank test across three comparison groups, using the five per-fold accuracy values as paired observations. None of the 45 tested pairs reached the conventional $\alpha = 0.05$ threshold, which is a mechanical consequence of $n = 5$ folds (minimum attainable $p = 0.0625$) rather than evidence of no difference. The direction of the test statistic is nonetheless informative: comparisons between the main/no_text_names scenarios and structure_counts_only produced $p = 0.0625$ in all 12 tested pairs, with the relational-ID-inclusive scenario winning every fold without exception; comparisons between Approach 1 and Approach 2 produced $p = 0.1250$ with Approach 1 winning every fold in the main and no_text_names scenarios; and comparisons among imputation methods produced p in the 0.5000-1.0000 range with no consistent winner.

Table 5. Wilcoxon Signed-Rank Test Summary

Comparison Group	Pairs Tested	p-value	Direction Across Folds
Q1: Approach 1 vs. Approach 2	12	0.1250 (all pairs)	Approach 1 higher in every fold
Q2: Imputation method vs. method	18	0.5000-1.0000	No consistent winner
Q3: (main / no_text_names) vs. structure_counts_only	12	0.0625 (all pairs)	Relational-ID-inclusive scenario higher in every fold

3.2 Feature Importance and Error Analysis

Feature importance was computed on the main + constant + Approach 1 combination (Figure 3). TacticID_agg_count ranks highest (0.0975 +/- 0.0382), followed by num_unique_tactic (0.0950 +/- 0.0121) and TechniqueName (0.0886 +/- 0.0060); eight of the ten highest-ranked features belong to the Relational ID category (Table 2), with the first Attribute-category feature (Typical Severity, 0.0850 +/- 0.0150) appearing fourth and CWE-ID appearing only sixteenth (0.0179 +/- 0.0027). Aggregated by category, Relational ID accounts for 65.2% of total gain-based importance, versus 21.4% for Structural and 13.4% for Attribute features.

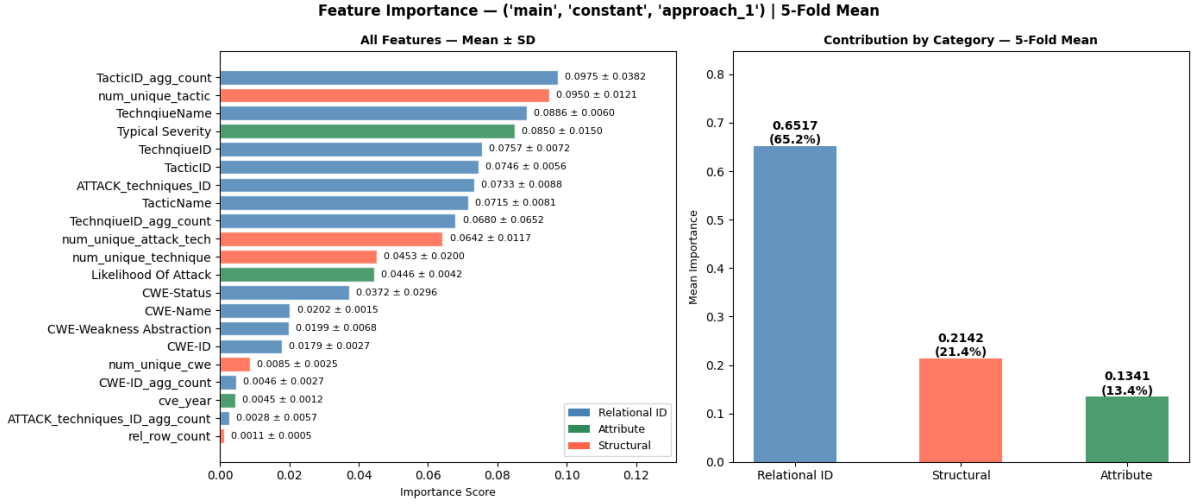


Figure 3. Five-fold Mean Feature Importance

Out-of-fold error analysis on the main + constant + Approach 1 combination (Figure 4) shows an overall error rate of 4.8%. Of the 38 CAPEC classes present in the test folds, 34 achieve $F1 > 0.80$, including several with a perfect $F1$ of 1.000; the two largest classes by support, CAPEC-31 (about 63 CVEs) and CAPEC-35 (about 54 CVEs), both reach $F1 = 1.000$. The lowest-performing classes all have small support: CAPEC-578, CAPEC-150, and CAPEC-457 (fewer than 10 test CVEs each) score $F1 = 0.667$, 0.600 , and 0.333 , respectively, and CAPEC-635 is never correctly predicted ($F1 = 0.000$).

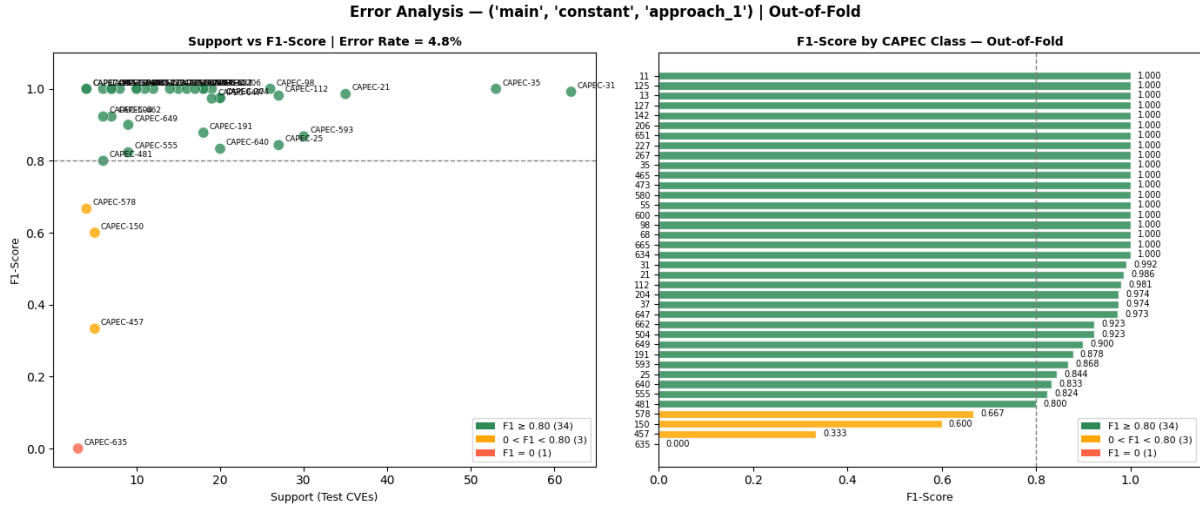


Figure 4. Out-of-Fold Error Analysis

Across all 18 combinations, XGBoost accuracy (0.6058-0.9523 depending on scenario) far exceeds both the dummy most_frequent baseline (0.10-0.19) and the label-shuffle test (0.07-0.12). In the main scenario specifically, XGBoost reaches 0.9257-0.9523 against a dummy baseline of 0.10-0.19 and a shuffle baseline of 0.07-0.12.

3.3 Deployment Implications and Limitations

The sanity checks confirm that XGBoost learns a genuine, non-trivial association between the input features and the CAPEC label, since performance collapses under both the dummy and label-shuffle baselines [10]. This is a necessary but not sufficient condition for deployment validity: it rules out the model simply guessing the majority class, but it does not rule out dependence on features that will be unavailable, or only partially available, for a newly disclosed CVE that has not yet been fully cross-referenced against CWE, CAPEC, and ATT&CK. The error analysis in Section 3.6 adds a further, complementary caveat: the classes with the weakest performance (CAPEC-578, CAPEC-150, CAPEC-457, and the entirely unpredicted CAPEC-635) are consistently the classes with the fewest test instances, indicating that the model's apparent overall strength is uneven across the CAPEC label space and concentrated in a comparatively small number of high-frequency classes [26]. Taken together, these results support treating relational-identity-driven accuracy figures in this line of research as an upper bound on deployment performance rather than as a direct estimate of it.

4. Conclusions

On the first objective, removing relational identity features and retaining only structural relational counts reduced mean accuracy from 0.9398-0.9418 to 0.6136 and F1-Macro from 0.8793-0.8816 to 0.4110, and feature importance analysis attributed 65.2% of gain-based importance to Relational ID features versus 21.4% for Structural and 13.4% for Attribute features. This confirms that current reported performance for this task depends substantially on deterministic taxonomy identifiers rather than on relational complexity alone. On the second objective, fixed first-listed CAPEC labeling (Approach 1) consistently outperformed frequency-based labeling (Approach 2) by 0.012-0.023 accuracy points across all scenarios, an effect attributable to label-target stability across folds rather than to attack-pattern relevance, while the choice of imputation method (constant, mode, or KNN) changed accuracy by no more than 0.0066 points in any configuration. On the third objective, dummy-classifier and label-shuffle sanity checks confirmed that the model learns a genuine, non-random association between features and labels, though this does not by itself guarantee generalization to newly disclosed, not-yet-annotated vulnerabilities.

The principal contribution of this work is a leakage-aware evaluation framework-combining feature ablation, dual sanity-check baselines, categorized feature importance, and out-of-fold error analysis-that can be applied to other CVE-CWE-CAPEC-ATT&CK classification studies to distinguish genuine behavioral learning from dependence on dataset-specific relational identifiers. Future evaluations of CAPEC and ATT&CK classifiers built on officially cross-referenced taxonomy data should report ablated, identity-free performance alongside full-feature results, so that reported accuracy can be interpreted as a realistic estimate of performance on new, unannotated vulnerabilities rather than as an upper bound driven by taxonomy identifiers.

References

- [1] A. Pinto, L.-C. Herrera, Y. Donoso, and J. A. Gutierrez, "Survey on Intrusion Detection Systems Based on Machine Learning Techniques for the Protection of Critical Infrastructure," *Sensors*, vol. 23, no. 5, p. 2415, Feb. 2023, doi: 10.3390/s23052415.
- [2] R. Kaur, D. Gabrijelčič, and T. Klobučar, "Artificial intelligence for cybersecurity: Literature review and future research directions," *Information Fusion*, vol. 97, p. 101804, Sep. 2023, doi: 10.1016/j.inffus.2023.101804.
- [3] K. Kanakogi et al., "Tracing CVE Vulnerability Information to CAPEC Attack Patterns Using Natural Language Processing Techniques," *Information*, vol. 12, no. 8, p. 298, Jul. 2021, doi: 10.3390/info12080298.
- [4] R. Othman, B. Rossi, and B. Russo, "Cybersecurity Defenses: Exploration of CVE Types Through Attack Descriptions," in *2024 50th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, IEEE, Aug. 2024, pp. 415-418, doi: 10.1109/SEAA64295.2024.00069.
- [5] K. Kanakogi et al., "Comparative Evaluation of NLP-Based Approaches for Linking CAPEC Attack Patterns from CVE Vulnerability Information," *Applied Sciences*, vol. 12, no. 7, p. 3400, Mar. 2022, doi: 10.3390/app12073400.
- [6] O. Grigorescu, A. Nica, M. Dascalu, and R. Rughinis, "CVE2ATT&CK: BERT-Based Mapping of CVEs to MITRE ATT&CK Techniques," *Algorithms*, vol. 15, no. 9, p. 314, Aug. 2022, doi: 10.3390/a15090314.
- [7] R. Othman, "Predicting Known Vulnerabilities from Attack Descriptions Using Sentence Transformers," Ph.D. dissertation, Free Univ. of Bozen-Bolzano, Bolzano, Italy, 2026, doi: 10.48550/arXiv.2602.22433.
- [8] I. Kawaminami, S. Shao, S. Hariri, and C. Tunc, "Attack Classification and Response Framework (ACR) Based on Machine Learning and CAPEC Ontology," *IEEE Access*, vol. 14, pp. 11560-11581, 2026, doi: 10.1109/ACCESS.2026.3653365.
- [9] C. Bonhomme and A. Dulaunoy, "Mapping CVEs to MITRE ATT&CK Techniques: A Curated Gold-Set Classifier and the Limits of LLM-Assisted Label Expansion," *arXiv:2607.25572*, Jul. 2026, doi: 10.48550/arXiv.2607.25572.
- [10] S. Kapoor and A. Narayanan, "Leakage and the reproducibility crisis in machine-learning-based science," *Patterns*, vol. 4, no. 9, p. 100804, Sep. 2023, doi: 10.1016/j.patter.2023.100804.
- [11] M. A. Bouke and A. Abdullah, "An empirical study of pattern leakage impact during data preprocessing on machine learning-based intrusion detection models reliability," *Expert Systems with Applications*, vol. 230, p. 120715, Nov. 2023, doi: 10.1016/j.eswa.2023.120715.
- [12] M. Rosenblatt, L. Tejaviulya, R. Jiang, S. Noble, and D. Scheinost, "Data leakage inflates prediction performance in connectome-based machine learning models," *Nature Communications*, vol. 15, no. 1, p. 1829, Feb. 2024, doi: 10.1038/s41467-024-46150-w.
- [13] T. Thomas and E. Rajabi, "A systematic review of machine learning-based missing value imputation techniques," *Data Technologies and Applications*, vol. 55, no. 4, pp. 558-585, Aug. 2021, doi: 10.1108/DTA-12-2020-0298.
- [14] J. Li et al., "Comparison of the effects of imputation methods for missing data in predictive modelling of cohort study datasets," *BMC Medical Research Methodology*, vol. 24, no. 1, p. 41, Feb. 2024, doi: 10.1186/s12874-024-02173-x.
- [15] L. Sasse et al., "Overview of leakage scenarios in supervised machine learning," *Journal of Big Data*, vol. 12, no. 1, p. 135, May 2025, doi: 10.1186/s40537-025-01193-8.
- [16] H. Zouhri, A. Idri, and A. Ratnani, "Evaluating the impact of filter-based feature selection in intrusion detection systems," *International Journal of Information Security*, vol. 23, no. 2, pp. 759-785, Apr. 2024, doi: 10.1007/s10207-023-00767-y.
- [17] S. K. Siahroudi and D. Kudenko, "An effective single-model learning for multi-label data," *Expert Systems with Applications*, vol. 232, p. 120887, Dec. 2023, doi: 10.1016/j.eswa.2023.120887.
- [18] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, New York, NY, USA: ACM, Aug. 2016, pp. 785-794, doi: 10.1145/2939672.2939785.
- [19] C. E. L. Asry, I. Benchaji, S. Douzi, and B. E. L. Ouahidi, "Enhancing cybersecurity: A high-performance intrusion detection approach through boosting minority class recognition," *PLoS ONE*, vol. 20, no. 3, p. e0317346, Mar. 2025, doi: 10.1371/journal.pone.0317346.
- [20] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation-Based Anomaly Detection," *ACM Transactions on Knowledge Discovery from Data*, vol. 6, no. 1, pp. 1-39, Mar. 2012, doi: 10.1145/2133360.2133363.
- [21] W. Chua et al., "Web Traffic Anomaly Detection Using Isolation Forest," *Informatics*, vol. 11, no. 4, p. 83, Nov. 2024, doi: 10.3390/informatics11040083.
- [22] J. Allgaier and R. Pryss, "Cross-Validation Visualized: A Narrative Guide to Advanced Methods," *Machine Learning and Knowledge Extraction*, vol. 6, no. 2, pp. 1378-1388, Jun. 2024, doi: 10.3390/make6020065.
- [23] S. Riyanto, I. S. Sitanggang, T. Djatna, and T. D. Atikah, "Comparative Analysis using Various Performance Metrics in Imbalanced Data for Multi-class Text Classification," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 6, 2023, doi: 10.14569/IJACSA.2023.01406116.
- [24] D. Berrar, "Using p-values for the comparison of classifiers: pitfalls and alternatives," *Data Mining and Knowledge Discovery*, vol. 36, no. 3, pp. 1102-1139, May 2022, doi: 10.1007/s10618-022-00828-1.

- [25] V. Lumumba, D. Kiprotich, M. Mpaine, N. Makena, and M. Kavita, "Comparative Analysis of Cross-Validation Techniques: LOOCV, K-folds Cross-Validation, and Repeated K-folds Cross-Validation in Machine Learning Models," *American Journal of Theoretical and Applied Statistics*, vol. 13, no. 5, pp. 127-137, Oct. 2024, doi: 10.11648/j.ajtas.20241305.13.
- [26] J. S. Aguilar-Ruiz and M. Michalak, "Classification performance assessment for imbalanced multiclass data," *Scientific Reports*, vol. 14, no. 1, p. 10759, May 2024, doi: 10.1038/s41598-024-61365-z.