Severity Prediction Based on Connectivity of Vulnerability Information via Related Product Information

Wataru Hiraiwa
Graduate School of Engineering
Kobe University
Kobe-shi, Japan
hiraiwa.wataru@gsuite.kobe-u.ac.jp

Hiroki Kuzuno
Graduate School of Engineering
Kobe University
Kobe-shi, Japan
https://orcid.org/0000-0003-2686-2541

Makoto Takita
Graduate School of Engineering
Kobe University
Kobe-shi, Japan
https://orcid.org/0000-0002-2532-6416

Thin Tharaphe Thein
Graduate School of Engineering
Kobe University
Kobe, 657-8501 Japan
https://orcid.org/0000-0002-1213-0393

Yoshiaki Shiraishi
Graduate School of Engineering
Kobe University
Kobe-shi, Japan
https://orcid.org/0000-0002-8970-9408

***Abstract*—There is a technique called vulnerability chaining in which an attacker does not just target a single vulnerability but combines multiple vulnerabilities to infiltrate a target. Therefore, when evaluating product safety, it is insufficient to check only one vulnerability. A search system had been proposed to enable comprehensive retrieval of information on multiple vulnerabilities by building an ontology of information related to vulnerabilities and products. The system obtains information about vulnerabilities and linked products to any desired extent. In this paper, we propose a method to predict the severity of newly discovered vulnerabilities using the base CVE scores of software with related vulnerabilities through the system. Using the constructed severity prediction model, we confirmed that severity can be predicted with an accuracy of 55% to 64% for severity classification.**

***Index Terms*—vulnerability chaining, CVE, CVSS, software supply chain, ontology, random forest**

I. Introduction

Software vulnerabilities are registered as Common Vulnerabilities and Exposures (CVEs) in a database, each with a unique CVE-ID. The National Vulnerability Database (NVD)[1] receives numerous vulnerability reports annually, as shown in Figure 1, with an increasing trend. Some cyberattacks exploit multiple vulnerabilities simultaneously, a technique known as vulnerability chaining [1], [2].

For example, a government agency attack exploited CVE-2020-1472[2] and other vulnerabilities, targeting software developed by related vendors. Future attacks may combine new and existing vulnerabilities. Ripple20[3], a set of 19 vulnerabilities in the Treck TCP/IP stack, highlights the risk of severe multi-vulnerability exploitation.

[1]https://nvd.nist.gov/general
[2]https://nvd.nist.gov/vuln/detail/cve-2020-1472
[3]https://www.jsof-tech.com/disclosures/ripple20/

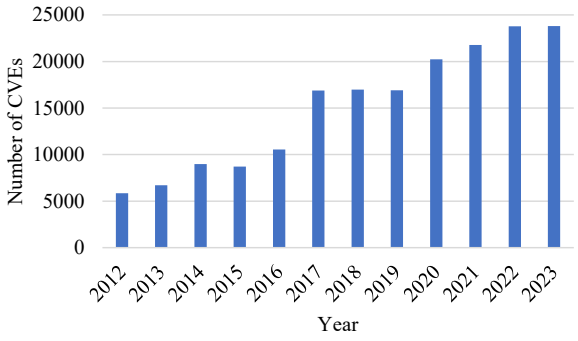

Fig. 1. The number of CVEs published to the NVD.

System administrators must manage vulnerabilities by predicting exploitation probabilities, applying mitigations, and utilizing ontology-based systems to assess risks comprehensively [3]–[6]. Research has also explored predicting CVE severity based on descriptions [7]–[10]. Tsutsui et al. [6] proposed an ontology-based search system to comprehensively retrieve vulnerability information by considering related vulnerabilities and products. The complexity of software supply chains and emerging techniques like vulnerability chaining highlight the necessity of analyzing relationships between components for accurate impact assessment.

In this study, we propose a model to predict the severity of vulnerabilities based on the relationships between vulnerabilities and related product information. The proposed method uses a search system constructed with an ontology to collect CVEs related to the CVE being predicted, and then calculates 11 features from these CVEs. These features are then fed into a machine learning model, Random Forest, to predict

the severity of the vulnerability. We evaluated the severity prediction model using CVEs published between 2012 and 2023, confirming that it achieves a prediction accuracy of 55% to 64% in each severity class.

The proposed method differs from traditional approaches by not relying on CVE descriptions for severity prediction. Instead, it predicts severity based on the relationships between CVEs, where products serve as mediating factors. Although the model's accuracy is lower compared to traditional description-based methods, it is advantageous for evaluating risks arising from connections between vulnerabilities. Additionally, since it does not depend on CVE descriptions, the method can also be used for product-specific risk assessment.

II. Research on Vulnerability Severity Prediction

A. Vulnerability Information

Software vulnerabilities are weaknesses that can be exploited in cyberattacks. To mitigate risks, it is essential to utilize publicly shared vulnerability information. Key sources include the Common Vulnerabilities and Exposures (CVE)[4], managed by Mitre Corporation, and the National Vulnerability Database (NVD), operated by NIST. CVEs provide unique identifiers for vulnerabilities, enabling standardized information exchange. NVD offers detailed descriptions, including affected products (CPE), vulnerability types (CWE), and severity scores (CVSS). CVSS, a widely used framework, evaluates vulnerability severity, with versions v2 and v3 being relevant to this study due to the analysis period starting in 2012.

B. Studies using vulnerability descriptions to predict severity

The CVSS base score for CVEs published in the NVD is determined by NVD analysts, so it takes time for the score to be assigned after the CVE and its related description are published. Therefore, a method to automatically assess the vulnerability risk is needed. An approach that mainly uses the vulnerability description (Description) has been proposed to estimate the CVSS base score and risk rating using machine learning [7]–[10].

In [7], a prediction model for the CVSS base score using Description was proposed. Text mining tools were used to extract feature vectors, and machine learning algorithms such as Support Vector Machine (SVM) and Random Forest were applied to predict the CVSS base score. Furthermore, when the model is used in combination with a fuzzy system, an accuracy of 88% has been achieved. In [7], only important information was extracted through principal component analysis from the feature vectors generated using the same method as in [8]. They report that the vulnerability risk was predicted using Extreme Gradient Boosting (XGBoost), acheving an accuracy of 87%. In [9], an accuracy of 81% was reported when they predicted the severity of vulnerabilities using the Convolutional Neural Network (CNN). In [10], a multifaceted machine learning algorithm is used that combines natural language processing (NLP), neural networks, and Bayesian optimization to learn and automatically predict the CVSS base score.

[4]https://cve.mitre.org/

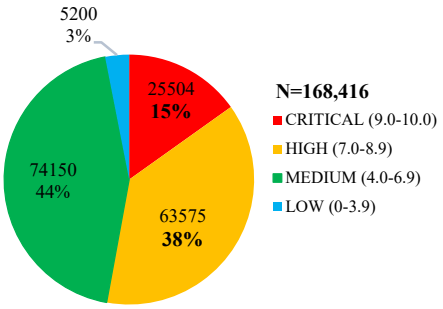

Fig. 2. Breakdown of base scores of CVEs.

C. Differences between Existing Research and Proposed Method

In the existing studies described in Section II-B, a machine learning model is used to predict the severity of CVEs by extracting words from the Description of CVEs and using them as feature vectors. In contrast, this study utilizes product information to search for related CVEs and predicts a base score based on the scores of relevant CVEs. In other words, we do not use the information on the vulnerability itself for machine learning. This means that our method can potentially offer more accurate predictions in cases where vulnerability descriptions may be vague, incomplete, or difficult to analyze manually. Additionally, by collecting related CVEs based on the product, it can also help predict the potential risk of the product itself.

III. Analysis of Vulnerability Information

We analyzed information from the NVD between 2012 and 2023, examining a total of 181,156 CVEs and 124,082 products associated with these CVEs over the 12-year period. To understand the characteristics of CVE severity, we examined the distribution of CVE base scores. CVEs with a base score of 0-3.9 were categorized as "LOW," 4.0-6.9 as "MEDIUM," 7.0-8.9 as "HIGH," and 9.0-10.0 as "CRITICAL." These results were summarized in Figure 2.

Our analysis revealed that 15% of CVEs are classified as CRITICAL and 38% as HIGH, indicating that a total of 53% of CVEs are considered dangerous. Next, we focused on CVEs associated with multiple products, which were analyzed separately. Figure 3 shows that of the 62,795 CVEs related to multiple products, 16% were CRITICAL and 44% were HIGH, making up a total of 60% of dangerous CVEs. In contrast, for CVEs linked to a single product (105,432 CVEs), 15% were CRITICAL and 34% were HIGH, leading to a total of 49% of dangerous CVEs.

Comparing Figure 3 and Figure 4, we observed that CVEs related to multiple products have a higher proportion of HIGH-severity classifications. This emphasizes the importance of considering information from both single and multiple related products when assessing CVE risk.

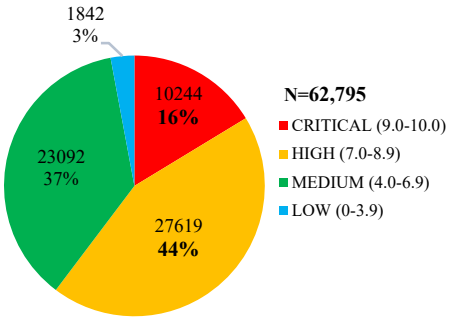

Fig. 3. Breakdown of base scores for CVEs related to multiple products.

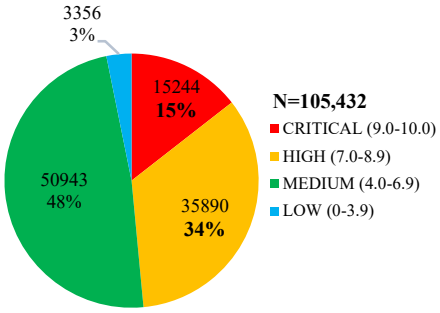

Fig. 4. Breakdown of base scores of CVEs related to a single product.

IV. Proposed Vulnerability Severity Prediction Model

A. Overview of Severity Prediction Model

As described in Section II-B, there is a delay between the disclosure of a CVE and its associated description and the determination of the base score. In this study, our objective was to predict the severity of newly discovered software vulnerabilities using the relationships in the software supply chain that contain the vulnerabilities and the CVE base scores that the software has. The software supply chain relationship refers to a situation in which multiple software components or libraries are combined, each coming from different sources. These relationships reflect the actual software development environment, where components depend on each other and a vulnerability in one may affect other components. A random forest is used to estimate the severity. Random forests are a popular machine learning algorithm that have been used in studies in [5], [7]. In this study, we collect relevant CVEs via product information associated with the CVEs to be used for severity prediction, create a relationship similar to a software supply chain, and use the data for learning and prediction. The dataset is constructed by efficiently extracting data from a graph database based on the ontology in [6], collecting the base scores of CVEs related to the CVEs to be predicted, and calculating various features.

B. Features used in the Predictive Model

This section describes the features used in the prediction model. In this study, we assume that the base scores of related CVEs are useful for predicting severity and we calculate the features accordingly. We define "CVE distance" to succinctly express the relationship between CVE nodes, and the CVE distance between CVE node 1 and CVE node 2 is defined as the number of product nodes on the path from CVE node 1 to CVE node 2. For example, node C1a is CVE nodes with a CVE distance of 1 from node C0 because they pass through a product node P1. Similarly, C2a is CVE nodes with CVE distance of 2 from C0 because it passes through product nodes P1 and P2a.

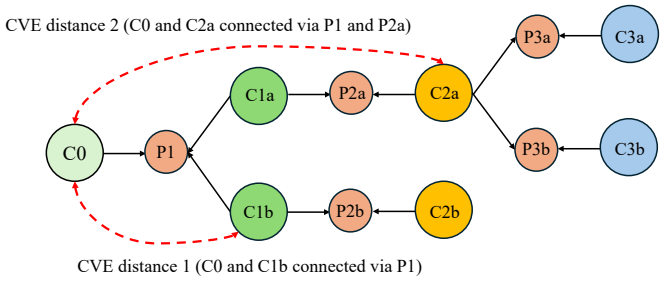

Fig. 5. CVE distance.

The following 11 CVE features are calculated using the base scores of the CVEs within a maximum CVE distance of 3.

1) Mean of the variance of the base score for CVEs at CVE distance 1
2) Mean of the variance of the base score for CVEs at CVE distance 2
3) Mean of the variance of the base score for CVEs at CVE distance 3
4) Mean of the variance of the base score for all CVEs up to the relevant CVE distance 3
5) Mean of the base scores for CVEs at CVE distance 1
6) Mean of the base scores for CVEs at CVE distance 2
7) Mean of the base scores for CVEs at CVE distance 3
8) Mean of the base scores for all CVEs up to the relevant CVE distance 3
9) Mean of the top 10% to 30% base scores for relevant CVEs
10) Mean of the bottom 10% to 30% base scores of the relevant CVEs
11) Mean of the variance excluding the top 10% and bottom 10% of base scores of the relevant CVEs

C. Dataset

The dataset consists of CVEs published over a 12-year period from 2012 to 2023, excluding those that satisfy the following conditions:

Condition 1: No CVE nodes exist with a CVE distance of 1.

Condition 2: There are 200 or more CVE nodes with a CVE distance of 1 (e.g., CVEs related to Windows and Linux).

Condition 3: The calculated features are matched; however, only one feature is selected from the matched pairs and left in the dataset.

Condition 1 is necessary because if there are no CVE nodes with a CVE distance of 1, all calculated features will have missing values, resulting in the generation of empty data. When there are approximately 200 or more CVE nodes with a CVE distance of 1, CVE nodes connected to a large number of nodes, such as Windows, Android, and Linux, are included. In this case, the subgraphs centered on the starting-point CVE node will have a similar shape, and Condition 2 is necessary to prevent the calculated features from having nearly identical values. This is also to prevent the search execution time from becoming excessively long during dataset creation. Condition 3 is necessary because even if Condition 2 does not apply, the feature values can be the same if the CVEs from the same product are used as starting points. If there are no CVE nodes corresponding to CVE distance 2 or CVE distance 3, the feature 2, 3, etc. become missing values, which are all replaced with 0

The training dataset consists of CVEs published over an 11-year period, from 2012 to 2022. A total of 6360 items were obtained, including 2000 CVEs each from the CRITICAL, HIGH, and MEDIUM categories, and 360 CVEs from the LOW category. As a result of excluding CVEs that satisfied the conditions, there was a small amount of data for vulnerabilities with low base scores as well as a small number of CVEs that fell into the LOW category. Similarly, the evaluation dataset includes CVEs published in 2023. For the evaluation dataset, we obtained data for 1000 CVEs each from the CRITICAL, HIGH, and MEDIUM categories, and 170 CVEs from the LOW category, for a total of 3170 data points.

D. Training of Severity Prediction Models

A severity prediction model was constructed using the training dataset. The proposed prediction model performs binary classification for CRITICAL, HIGH, MEDIUM, and LOW categories, respectively. For each classifier, 2000 items with positive labels and 2000 items with negative labels are selected from the other three categories, for a total of 4000 items in each binary classification. For example, when training a classifier to infer CRITICAL or NOT, 2000 items belonging to the CRITICAL category are extracted and assigned a positive label. Then, a total of 2000 items are extracted from the HIGH, MEDIUM, and LOW categories and assigned a negative label. The hyperparameters of the random forest are: **max_depth** = 40, **max_features** = 'auto', and **n_estimators** = 100.

E. Evaluation Indicators for Severity Prediction Models

In binary classification, if the actual label is positive and the predicted label is also positive, it is called a True Positive (TP). If the actual label is positive and the predicted label is negative, it is called a False Negative (FN). If the actual label is negative and the predicted label is negative, it is called a True Negative (TN). If the actual label is negative and the predicted label is positive, it is called a False Positive (FP). Table I shows the correctness and incorrectness rates of the classification results. The following formulas are used to calculate accuracy, precision, recall, and the F1 Score, which serve as evaluation metrics for classification results:

TABLE I
EVALUATING CLASSIFICATION.

		Actual	
		Positive	Negative
Predicted	Positive	TP (True Positive)	FP (False Positive)
	Negative	FN (False Negative)	TN (True Negative)

TABLE II
EVALUATION RESULT OF EACH MODEL.

	CRITICAL	HIGH	MEDIUM	LOW
Accuracy	0.6480	0.5607	0.5529	0.6200
Precision	0.6380	0.6371	0.5569	0.6022
Recall	0.6843	0.2823	0.5176	0.7066
F1 score	0.6603	0.3913	0.5365	0.6503

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}},$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}},$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}},$$

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.$$

Accuracy indicates the overall correctness of the classification results. Precision measures the accuracy of positive predictions, while recall measures the classifier's ability to identify positive examples. The F1 Score is the harmonic mean of precision and recall, providing a balance between them. The closer the F1 Score is to 1.0, the better the classification model performs in both precision and recall.

F. Evaluation of Severity Prediction Models

Table II presents the severity prediction results for each model using the evaluation dataset. Each column shows the results of binary classification, with the target category being the positive examples and the other three categories being the negative examples. Focusing on the CRITICAL and HIGH categories, the accuracy for CRITICAL is 64.80%, while the recall is 68.43%, which is relatively high. This indicates that the model misses fewer CRITICAL CVEs. On the other hand, the recall for HIGH is low, at 28.23%.

G. Feature Analysis

In the proposed prediction model, the features are heuristically defined based on the expectation that the base score of the relevant CVE is useful to predicting severity. The contribution of the features to the classification result is evaluated by examining their importance, which is obtained through a random forest algorithm. Figure 6 shows the importance of

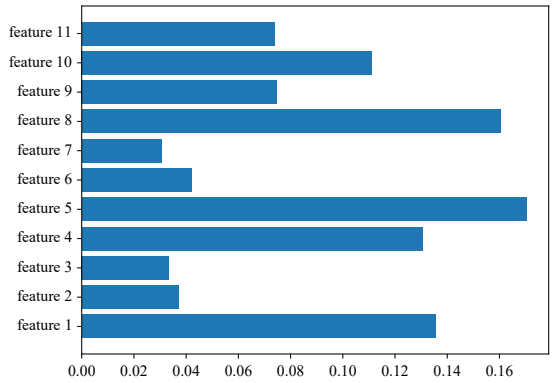

Fig. 6. Importance of features.

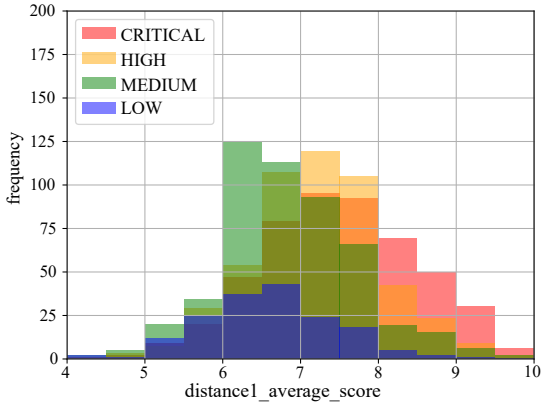

Fig. 7. Histogram of means of base scores for CVE distance 1.

the features based on the Gini impurity in the constructed random forest model. The figure indicates that the mean of the base score (feature 5) and the mean of the variance of the base score (feature 1) for CVEs at CVE distance 1 exceed 0.1 in relative values, while the mean of the base score (features 6 and 7) and the mean of the variance of the base score (features 2 and 3) for CVEs at CVE distances 2 and 3 are below 0.05. This suggests that the base scores for CVEs with closer CVE distances contribute more significantly to the prediction results.

Therefore, we randomly selected 500 CVEs from each category and created histograms of the mean of the base scores (features 5 and 6) for CVE distances 1 and 2, as shown in Figures 7 and 8, respectively. When comparing Figures 7 and 8, in the case where the CVE distance is 2, the mean of the base score is biased toward the 6.5 to 7.0 interval, and the differences between categories are smaller. This is thought to reduce its contribution to the prediction results.

The results of counting the number of relevant CVEs for each CVE distance are shown in Table III. The number of CVEs ais classified into three levels (1–10, 11–50, and 51 or more), and the percentage of each level is also calculated. The results show that for CVEs with CRITICAL severity, the percentages of 1-10 related CVEs are higher than those of other categories, at 48%, 21%, and 16% for CVEs at CVE distances 1, 2, and 3, respectively. In contrast, the percentages are smaller in the other categories. The greater the number of related CVEs, the more similar the calculated scores are, making the data less distinctive. Therefore, it can be inferred that the prediction accuracy of the CRITICAL category is nearly 10% higher than that of the other categories.

Therefore, Table IV shows the results of the prediction model evaluation by limiting the CVEs in the evaluation dataset to those with 10 or fewer related CVEs. Focusing on the CRITICAL and HIGH categories, the recall of the CRITICAL category improved from 68.43% to 73.25%, and the recall of the HIGH category improved from 28.23% to 32.25%. However, the other scores worsened. In the HIGH category, the recall was higher for data for which the features were calculated from fewer CVEs. Therefore, it was confirmed that the ratio of the number of relevant CVEs was one of the factors that increased the prediction accuracy of CRITICAL compared to the other categories. However, because precision was lower, the F1 score also decreased. More studies, including the addition of other features, are needed to improve the overall accuracy of the prediction.

H. Refining the Prediction Model

The results in Table II indicate that the recall for predicting whether a case is HIGH is low. It can be assumed that this is because a CRITICAL CVE, considered to have similar characteristics to HIGH due to its high risk, must be classified as a negative example, which causes a HIGH CVE to also be misclassified as negative. Therefore, the prediction model is a combination of three classifiers:

Classifier 1: Whether it is classified as CRITICAL.
Classifier 2: Whether it is classified as HIGH or above.
Classifier 3: Whether it is classified as MEDIUM or above.

Classifier 2 (HIGH+) treats the CRITICAL and HIGH categories as positive examples and the MEDIUM and LOW categories as negative examples. Classifier 3 (MEDIUM+) treats the CRITICAL, HIGH, and MEDIUM categories as positive examples and the LOW category as a negative example.

Table V shows the evaluation results for each prediction model. For classifier 2, the recall improved to 59.45%, resulting in fewer missed cases. The purpose of using the vulnerability severity prediction model in this study is to help prioritize countermeasures against high severity CVEs, particularly those classified as CRITICAL or HIGH. The improved prediction model can be used in the order of Classifier 1, Classifier 2, and Classifier 3 to more effectively prioritize and address vulnerabilities based on severity.

V. Conclusion

The proposed severity prediction model estimates the severity of a new vulnerability using the base score of a related CVE. It predicts CVE severity with 55% to 64% accuracy,

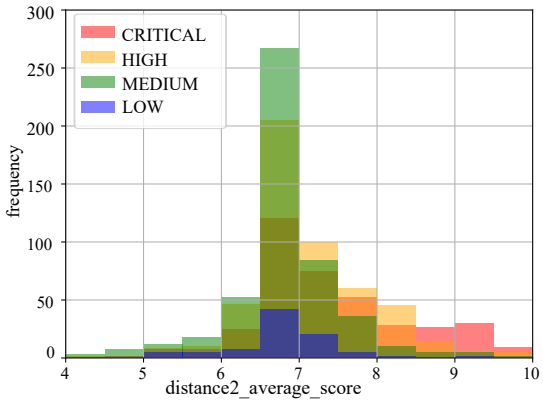

Fig. 8. Histogram of means of base scores for CVE distance 2.

TABLE III
NUMBER OF RELATED CVES PER CVE DISTANCE.

CRITICAL

Number of CVEs	distance 1		distance 2		distance 3	
1–10	512	48%	80	21%	49	16%
11–50	378	35%	80	21%	42	13%
51+	177	17%	227	59%	222	71%
All	1066		388		312	

HIGH

Number of CVEs	distance 1		distance 2		distance 3	
1–10	987	36%	216	16%	82	7%
11–50	991	36%	204	15%	81	7%
51+	747	27%	953	70%	1014	86%
All	2718		1371		1177	

MEDIUM

Number of CVEs	distance 1		distance 2		distance 3	
1–10	789	35%	130	12%	186	20%
11–50	725	32%	113	10%	67	7%
51+	739	33%	845	78%	692	73%
All	2249		1086		945	

LOW

Number of CVEs	distance 1		distance 2		distance 3	
1–10	62	36%	12	14%	2	3 %
11–50	65	38%	11	13%	3	4 %
51+	43	25%	63	73%	68	94%
All	170		86		72	

TABLE IV
EVALUATION RESULT OF EACH MODEL WHEN THE NUMBER OF RELEVANT CVES IS LIMITED TO 10 OR LESS.

	CRITICAL	HIGH	MEDIUM	LOW
Accuracy	0.5287	0.4950	0.5300	0.5875
Precision	0.5204	0.4930	0.5315	0.5813
Recall	0.7325	0.3525	0.5050	0.6250
F1 score	0.6085	0.4110	0.5179	0.6024

TABLE V
EVALUATION RESULT OF IMPROVED MODELS.

	CRITICAL	HIGH+	MEDIUM+
Accuracy	0.6480	0.5972	0.6200
Precision	0.6380	0.5977	0.6268
Recall	0.6843	0.5945	0.5933
F1 score	0.6603	0.5961	0.6096

with particularly strong performance for CRITICAL CVEs (64.80% accuracy and 68.43% precision). The model can help prioritize vulnerability mitigation efforts, especially for CRITICAL vulnerabilities. To improve recall for HIGH-severity classifications, three classifiers were combined, increasing recall to 59.45% and reducing False Negatives.

A key advantage of the proposed model is its independence from vulnerability descriptions. Unlike traditional methods that rely on time-consuming textual information, the model uses the base score of related CVEs and other features, allowing it to be deployed even before a full vulnerability description is available. This makes it valuable in the early stages of vulnerability detection.

The model predicts severity based on base scores and related CVEs, enabling efficient evaluation of software security. This approach allows for predicting the risk of potential vulnerabilities without extensive manual analysis, making it a valuable tool for proactively assessing software risks.

This study used 11 features in machine learning, but the relationships between CVEs were not fully utilized. Future research could improve the model by better incorporating these relationships, such as using a graph neural network to capture CVE connections.

ACKNOWLEDGMENT

This work was partially supported by JST K Program Japan Grant Number JPMJKP24K1.

REFERENCES

[1] N. Robinson, "Scoring vulnerabilities after seeing a chained vulnerability demonstration," *American Journal of Science & Engineering*, 2020.
[2] ——, "An exploratory study into vulnerability chaining blindness terminology and viability," 2022. [Online]. Available: https://arxiv.org/abs/2203.10403
[3] S. Wu, Y. Zhang, and W. Cao, "Network security assessment using a semantic reasoning and graph based approach," *Comput. Electr. Eng.*, vol. 64, no. C, p. 96–109, nov 2017.
[4] J.-b. Gao, B. Zhang, X.-h. Chen, and Z. Luo, "Ontology-based model of network and computer attacks for security assessment," *Journal of Shanghai Jiaotong University (Science)*, vol. 18, pp. 554–562, 10 2013.
[5] J. S. P. Salini, "Prediction and classification of web application attacks using vulnerability ontology," *International Journal of Computer Applications*, vol. 116, no. 21, pp. 42–47, April 2015.
[6] T. Tsutsui, Y. Shiraishi, and M. Morii, "Systemization of vulnerability information by ontology for impact analysis," in *2021 IEEE 21st International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, 2021, pp. 1126–1134.
[7] A. Khazaei, M. Ghasemzadeh, and V. Derhami, "An automatic method for cvss score prediction using vulnerabilities description," *J. Intell. Fuzzy Syst.*, vol. 30, pp. 89–96, 2015.
[8] P. Wang, Y. Zhou, B. Sun, and W. Zhang, "Intelligent prediction of vulnerability severity level based on text mining and xgbboost," in *2019 Eleventh International Conference on Advanced Computational Intelligence (ICACI)*, 2019, pp. 72–77.
[9] Z. Han, X. Li, Z. Xing, H. Liu, and Z. Feng, "Learning to predict severity of software vulnerability using only vulnerability description," in *2017 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, 2017, pp. 125–136.
[10] D. T. Vasireddy, D. S. Dale, and Q. Li, "Cvss base score prediction using an optimized machine learning scheme," in *2023 Resilience Week (RWS)*, 2023, pp. 1–6.