An Intelligent Approach of Vulnerability Severity Assignment Based on Event Extraction

Xufan Zheng[1], Chang Liu[2], Tianci Li[3], Xiaoxue Wu[3]*

[1] College of Science, Northwest Agriculture and Forestry University, China
[2] School of Software, Northwestern Polytechnical University, China
[3] School of Information Engineering, Yangzhou University, China
2023014561@nwsuaf.edu.cn, chang_liu@mail.nwpu.edu.cn, 1557010767@qq.com, wuxiaoxue00@gmail.com

Abstract

The rapid increase in vulnerability reports presents significant challenges for manual analysis and risk management. To address this, we propose an innovative approach for vulnerability severity assignment using deep learning techniques focused on event extraction. Specifically, we develop an event-focused convolutional neural network that processes dual inputs—vulnerability descriptions and related events—to classify vulnerabilities into high, medium, or low risk. This framework improves classification accuracy by effectively incorporating relevant event details. Our approach aims to streamline vulnerability management, enabling organizations to prioritize their cybersecurity efforts more efficiently. Evaluations show notable improvements in accuracy, highlighting the potential of deep learning in vulnerability severity assignment.

Keywords: CNVD, Vulnerability severity assessment, Event extraction, Intelligent analysis

1 Introduction

The increasing complexity of software systems and the evolving landscape of cyber-attack vectors have made accurate vulnerability severity assignment a critical challenge in cybersecurity [1-3]. The growing number of vulnerability reports on platforms like the Common Vulnerabilities and Exposures (CVE) and the China National Vulnerability Database (CNVD) highlights the need for automated and efficient methods to assess and prioritize vulnerabilities. Traditional methods, such as the Common Vulnerability Scoring System (CVSS), provide a standardized approach to vulnerability scoring, but they often require extensive manual intervention and lack the ability to incorporate contextual information from vulnerability descriptions [4-5]. This manual process is not only time-consuming but also prone to human error, especially given the continuous influx of new vulnerabilities.

Recent advancements in Natural Language Processing (NLP) and deep learning have opened new avenues for automating vulnerability analysis. Information extraction techniques, particularly event extraction, have shown promise in transforming unstructured text into structured data, enabling more precise vulnerability analysis [6-7]. Event extraction involves identifying specific events, their triggers, types, and associated arguments, which can provide deeper insights into the nature and impact of vulnerabilities [8-9]. For instance, Nguyen et al. [10] demonstrated the effectiveness of Convolutional Neural Networks (CNNs) in event detection, while Chen et al. [11] proposed Dynamic Multi-Pooling CNNs (DMCNNs) for event extraction, achieving state-of-the-art performance in various NLP tasks.

In the context of vulnerability reports, event extraction can offer more nuanced insights than traditional Named Entity Recognition (NER) techniques. While NER focuses on identifying entities such as software names and version numbers, event extraction goes further by capturing the relationships and actions described in the text [12]. For example, in the sentence "*The XYZ software version 2.0 suffered from a buffer overflow vulnerability,*" event extraction not only identifies the entities "*XYZ software*" and "*version 2.0*" but also recognizes the event trigger "suffered from" and the event type "buffer overflow vulnerability." This additional layer of information can significantly enhance the accuracy of vulnerability severity assessments [13-14]. However, the integration of event extraction with deep learning models for vulnerability severity assignment is still in its infancy [15-16], with few studies exploring the potential of combining textual descriptions and event information for more accurate risk classification [17-19].

This paper addresses these gaps by proposing a novel approach to vulnerability severity assignment that leverages event extraction and deep learning techniques. Our contributions are as follows:

(1) Sequence Annotation for Event-Based Vulnerability Descriptions: We utilize Doccano to annotate vulnerability reports, identifying trigger words, event types, and arguments. The annotated samples are converted into JSON format, serving as input for machine learning models.

(2) Trigger Word Identification Using Advanced Models: We employ BERT and BiLSTM with GlobalPointer to define unique trigger words in

*Corresponding Author: Xiaoxue Wu; Email: wuxiaoxue00@gmail.com
DOI: https://doi.org/10.70003/160792642026032702007

vulnerability reports, achieving an F1 score of 0.85. This approach enhances the precision of event extraction.

(3) Improved Severity Assignment with Event Integration: We integrate event information into vulnerability severity assessment using TextCNN and fully connected neural networks. This method improves accuracy, highlighting the importance of event data in vulnerability analysis.

By combining these techniques, our approach aims to streamline vulnerability management, enabling organizations to prioritize their cybersecurity efforts more efficiently. The rest of this paper is organized as follows: Section 2 provides background information on entity and event extraction, CRF, and deep learning models. Section 3 details our proposed approach, including schema definition, event extraction methods, and the event-focused CNN model for severity assignment. Section 4 presents the experimental evaluation, and Section 5 discusses the results. Finally, Section 6 concludes the paper.

2 Background

2.1 Entity Extraction and Event Extraction

Entity extraction, or Named Entity Recognition (NER), is a core Natural Language Processing (NLP) task that involves identifying and categorizing named entities in text [20-22]. In the software domain, it focuses on extracting key entities from unstructured or semi-structured texts like vulnerability reports and bug descriptions, transforming them into structured data for analysis and application in areas such as automated security assessments and vulnerability management.

Event extraction, a subtask of entity extraction, identifies specific events or actions in the text along with their associated entities, roles, and attributes. It relies on first recognizing the entities involved before determining their relationships [23]. For example, in the sentence "*The XYZ software version 2.0 suffered from a buffer overflow vulnerability,*" entity extraction identifies "*XYZ software*" and "*version 2.0*," while event extraction recognizes "*suffered from*" as an event trigger and "*buffer overflow vulnerability*" as the event type affecting these entities. This process enhances understanding and utilization of textual information for improved cybersecurity practices.

2.2 Conditional Random Field (CRF)

CRF is a probabilistic model used for labeling or parsing structured data, particularly useful in tasks involving sequence prediction such as Named Entity Recognition (NER), part-of-speech tagging, and event extraction. Unlike Hidden Markov Models (HMMs) and Maximum Entropy Markov Models (MEMMs), CRF consider the entire input sequence simultaneously, which helps in mitigating the label bias problem and enhances prediction accuracy. The key characteristic of CRF is their ability to model dependencies between output labels, making them ideal for capturing contextual information within sequences.

A CRF model defines a conditional probability $P(y \mid x)$ over a set of output variables y given an input sequence x. This probability is computed using a feature function $f_j(x, y)$ that captures relevant features from the input and output:

$$P(y \mid x) = \frac{1}{Z(x)} \exp\left(\sum_j \lambda_j f_j (x, y) \right) \quad \textbf{(1)}$$

Where $Z(x)$ is the normalization factor known as the partition function, ensuring that the probabilities sum to 1. λ_j represents the weight parameters associated with each feature function f_j.

CRFs have been successfully applied in various NLP tasks, including event extraction. For example, Lafferty et al. [24] introduced CRFs for sequence labeling tasks, demonstrating their effectiveness in capturing long-range dependencies in text. In the context of vulnerability reports, CRFs can be used to extract event triplets (event type, role, and argument), which are crucial for accurate vulnerability severity assessment [25-27].

2.3 Deep Learning Models Highly-related in This Work

2.3.1 Bidirectional Encoder Representations from Transformers (BERT)

BERT has emerged as a leading pre-trained language model [28]. Its success is attributed to its unique approach to leveraging deep bidirectional contextual representations from unlabeled text, allowing it to effectively capture prior knowledge. Specifically, BERT uses transformers to encode context from both directions (left and right) around each word in a sentence.

For a given token t_i in a sequence, BERT computes its representation H_i using a transformer encoder:

$$H_i = Transformer\left(\{t_1, t_2, ...t_n\} \right) \quad \textbf{(2)}$$

Where $\{t_1, t_2, \ldots t_n\}$ represents the input token sequence, Transformer denotes the transformer encoder that processes the entire sequence to generate context-aware representations.

2.3.2 GlobalPointer

GlobalPointer is a span-based NER framework. It leverages a multiplicative attention mechanism that considers relative positions between tokens through Relative Position Embedding (ROPE). This allows the model to have a global view that takes into account both the beginning and end positions (i.e., head and tail information) for predicting entities.

Two modules are designed to identify the head and tail of an entity, enabling inconsistency between training and inference processes. This design aims to enhance the accuracy of identifying nested entities, where one entity may contain another.

An efficient variant of the GP is proposed to reduce the number of parameters required for each new entity type added. By sharing scores calculation under each entity type and treating NER as two subtasks—extraction and classification—the parameter increase per new entity

type is significantly reduced from 2vd to 4d, where v is the dimension of representation and d is typically much smaller.

2.3.3 Convolutional Neural Network (CNN)

CNN is a class of deep learning models designed to process data with grid-like topology. A typical CNN architecture consists of convolutional layers, pooling layers, and fully connected layers. Convolutional layers apply a series of filters to the input to detect local patterns, enabling the network to learn hierarchical feature representations. Pooling layers reduce the spatial dimensions of the data, helping to decrease computation and make the model invariant to small shifts and distortions. Fully connected layers then use these features to perform classification or regression tasks.

CNNs have been widely used in text classification tasks, including sentiment analysis and topic classification. In the context of vulnerability severity assessment, CNNs can be used to extract *n-gram* features from vulnerability descriptions, which are crucial for accurate risk classification. By incorporating CNNs into our severity assessment framework, we aim to improve the accuracy of vulnerability risk classification.

3 Our Approach

Vulnerability reports are semi-structured data primarily available as XML files. These files contain detailed information such as the ID, title, and description of each vulnerability. We focus on the unstructured content within the vulnerability descriptions. The framework of our vulnerability severity assignment approach is shown as Figure 1. The input of the approach is the vulnerability reports, and it then uses 3 major steps to achieve the severity intelligent assignment, namely, the schema definition, event extraction, and severity assignment phases, finally the severity of each vulnerability report is assigned.

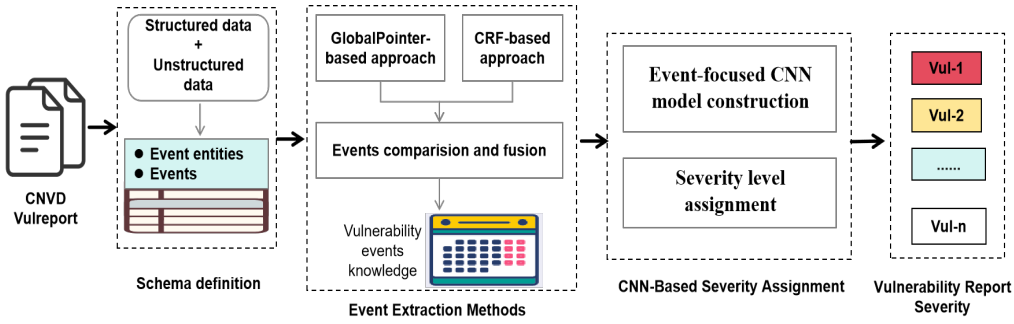

Figure 1. Framework of our approach

3.1 Schema Definition

Existing work focus on sentence-level extraction—assuming all arguments are contained within a single sentence. However, event understanding in bug report frequently requires context beyond a single sentence. Figure 2 provides an example from CVE entries, it describes a vulnerability with type "resource management error" for the product "*F5 BIG-IP VE*" that may span multiple sentences, necessitating a document-level extraction approach.

Our focus shifts towards event types and arguments rather than trigger words. In cases where trigger words are ambiguous or absent, we unify them with arguments. Specifically, vulnerability names (e.g., "*cross-site scripting vulnerability*") serve as default triggers, and when these are not available, other elements like vulnerability results are used instead. This method enhances applicability and prevents extraction failures due to missing trigger words. As illustrated in an example from Figure 3 where a vulnerability report lacking a specific name uses its result as the trigger word.

3.1.1 Data Annotation

Our data sourced from CNVD, is annotated based on the defined schema, focusing on event elements (as shown in Table 1). We pre-process the data to fit the deep learning model input format and then use Doccano to manually annotate it. Professionals with vulnerability knowledge annotated 600 samples covering 5 types of vulnerabilities (120 samples each).

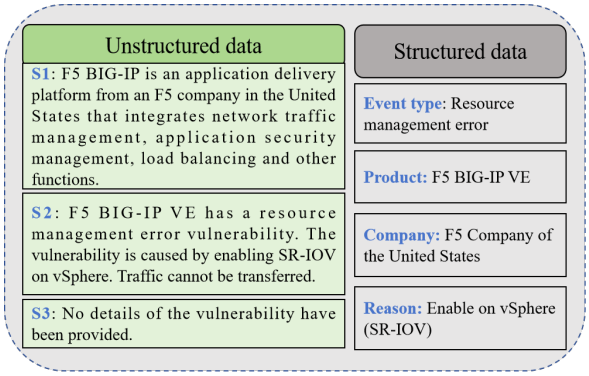

Figure 2. Example of vulnerability report

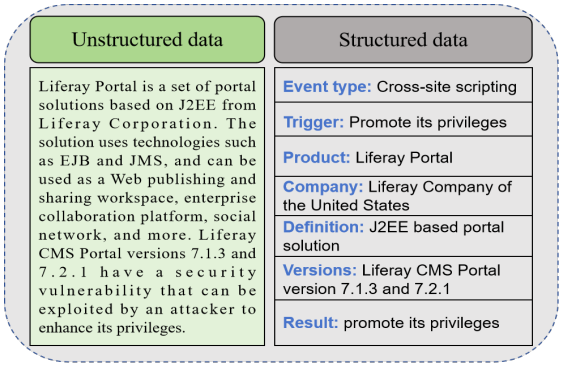

Figure 3. Example of vulnerability without name

Table 1. Event elements in schema

Event entities	Significance
Product	Name of the affected physical product
Company	Affected products are located in the company, community, group, etc
Definition	The essence of the product, or the attributes of the product; can be software or solutions
Version	The version number of the product where the vulnerability occurred
Vulnerability	The type of vulnerability that occurs in the product
Use pattern	The method by which an attacker attacks a vulnerability
Reason	The cause of the vulnerability
Result	The damage caused by the vulnerability

3.1.2 Vectorized Representation of Text Data

After data preprocessing, text data must be vectorized for input into models in the next phase. This process uses the RoBERTa model [29], an enhanced version of BERT, to convert text into vector representations. RoBERTa transforms the input text into a format suitable for deep learning by capturing its semantic features effectively.

3.2 Event Extraction Methods

3.2.1 GlobalPointer-based Event Extraction

The event extraction model based on GlobalPointer includes a vector representation layer, a BiLSTM lexical feature extraction layer, and an entity recognition output layer. Initially, the RoBERTa model processes input sentences (e.g., "*buffer overflow*") to generate word vectors, providing a vectorized text representation. For handling long sentences in document-level extraction, a BiLSTM network extracts features from RoBERTa's output to prevent gradient vanishing. These features are then passed to the GlobalPointer model, which identifies the head and tail positions (i, j) of subjects and objects and matches entity positions for each relation ($p(h_i, h_j)$ and $p(t_i, t_j)$). The intersection of these positions extracts the event information from the vulnerability report, as illustrated in Figure 4.

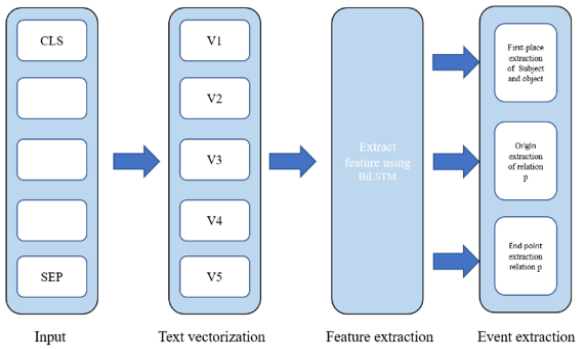

Figure 4. Process of GlobalPointer-based event extraction

3.2.2 CRF-based Event Extraction

CRF is effective for named entity recognition (NER) and simplifies event extraction to extracting triplets of event type, role, and argument. Matching these triplets determines the accuracy of predictions, framing the task as an entity annotation problem akin to NER, making CRF suitable for CNVD vulnerability reports.

As shown in Figure 5, we employ a RoBERTa-based CRF model. It loads CNVD vulnerability reports, segments the text with RoBERTa's tokenizer, and inputs the tokens into RoBERTa. All layers are fine-tuned without freezing, and the resulting word vectors are passed to the CRF layer. This layer uses a transition matrix to capture label correlations and outputs the extracted entities.

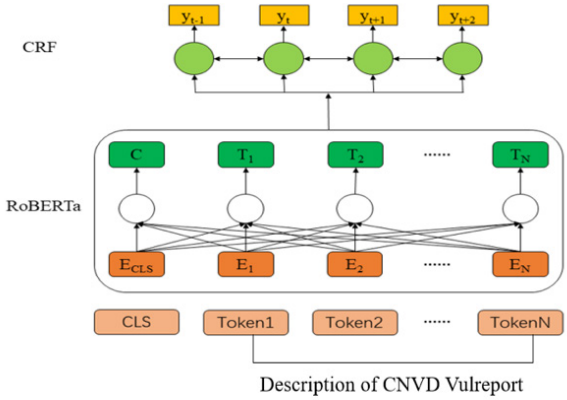

Figure 5. Process of CRF-based event extraction

3.2.3 Event Knowledge Comparison and Fusion

After extracting event knowledge with both the GlobalPointer approach and CRF-based approach, we then compare the results of the two methods to identify strengths and weaknesses in each. This comparison is aimed at leveraging the advantages of both techniques to achieve a more comprehensive and accurate understanding of events within vulnerability reports.

We implement a fusion strategy that combines the outputs from both models. Initially, we align the identified entities and events from both approaches based on their textual positions and semantic roles. Then, we apply a rule-based system to resolve discrepancies and integrate complementary information. For instance, if the CRF model accurately identifies an entity but fails to fully describe its role within an event, the richer contextual information provided by the GlobalPointer approach can be used to supplement this detail.

3.3 Event-focused CNN Model for Vulnerability Severity Assignment

We introduce an advanced deep learning model by leveraging the foundational architecture of TextCNN for severity assignment. This model creates a sophisticated dual-input, single-output framework, as shown in Figure 6.

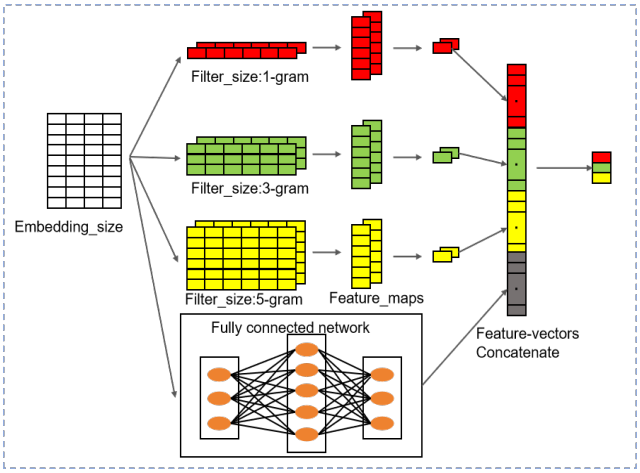

Figure 6. Event-focused CNN model for severity assignment

3.3.1 Event-focused CNN Model Construction

The event-focused CNN model is constructed to leverage both textual descriptions and event knowledge extracted from vulnerability reports.

In the input Layer, each vulnerability report is represented as a sequence of words, where each word is embedded into a vector using Word2Vec. Event Information details are also converted into vectors using a similar embedding technique. Multiple convolutional layers with different filter sizes (1-*gram,* 3-*gram, and* 5-*gram*) are applied to capture various n-gram features from the input sequences. This helps in identifying patterns at different granularities.

Max-pooling layers are used to select the most significant features from the feature maps generated by the convolutional layers. This reduces the dimensionality while retaining the most important information. In the fully Connected Layer, the feature maps from the pooling layers are concatenated and fed into a fully connected network. The fully connected layer uses multiple neurons to integrate the information from different convolutional layers and event information.

3.3.2 Severity Level Assignment

Within the event-focused CNN model, severity level assignment is executed adhering to the categorizations established by CNVD—high, medium, and low risk—based on comprehensive considerations of impact and environmental factors.

In the output layer, a softmax layer is used to classify the vulnerability into one of the three severity levels: high, medium, or low risk. The final output is a probability distribution over the three classes, indicating the likelihood of each severity level. Finally, to handle redundancy and improve classification accuracy, event fusion and deduplication techniques are applied. Redundant or duplicate event information is merged or removed, ensuring that only unique and informative data are used in the model.

4 Experimental Evaluation

We evaluate the effectiveness of our approach by assessing the severity of software vulnerabilities in CNVD.

4.1 Experiment Setting

Both GlobalPointer and CRF models utilized a RoBERTa base with a maximum input length of 300, covering 99.3% of vulnerability report descriptions without excessive GPU memory usage. The batch size was set to 16, and training involved 150 iterations at a learning rate of 2e-5.

In the severity assignment phase, the iteration count was also set to 150 to ensure convergence, as evidenced by stabilized loss in Figure 7. Batch sizes of 32, 64, and 128 were tested, with 32 providing the best validation performance (Table 2).

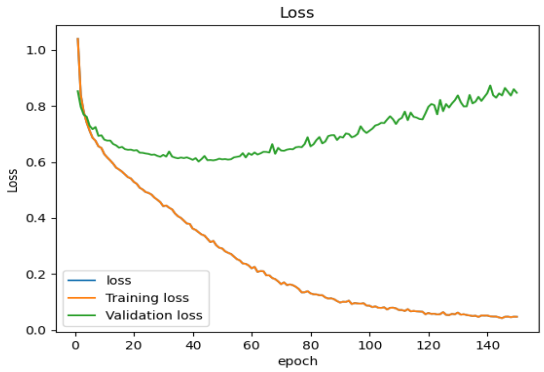

Figure 7. The relationship between epoch and loss

Table 2. Superparameter selection

Superparameter	Value	F1	Precision	Recall
batch_size	**32**	**0.730594**	**0.735384**	**0.732035**
	64	0.717702	0.726960	0.723053
	128	0.716173	0.720937	0.718562
Num_filters	32	0.701276	0.705361	0.705089
	64	0.705094	0.705094	0.700589
	128	**0.730594**	**0.735384**	**0.732035**

The CNN output feature vector dimension was set to 128, matching the number of filters per window size for optimal results. Throughout these experiments, only one hyperparameter was adjusted at a time while keeping others at their best-performing values. This meticulous tuning ensures robust and reliable model performance.

4.2 Result Analysis

We conduct three experiments: two for event extraction, followed by vulnerability severity assignment. We compare our approach with a baseline model.

4.2.1 Event Extraction Experiment

The GlobalPointer model was initially trained on the Baidu dataset. Utilizing transfer learning by loading pretrained weights (Table 3), the model showed improved F1 (+2.4%) and Recall (+5.1%), albeit with a slight decrease

in Precision (-0.94%). This method was also applied in subsequent experiments.

Table 3. Load the pre-training weight

Evaluation index	F1	Precision	Recall
Unloaded weight	0.84203	0.90525	0.78707
Load weight	0.86649	0.89581	0.83904

The experimental results presented in Table 4 indicate that the CRF model outperformed both the GlobalPointer and GlobalPointer-FGM models across all key evaluation metrics: F1 score, Precision, and Recall. Specifically, the CRF model achieved an F1 score of 0.91956, a Precision of 0.92551, and a Recall of 0.91369. These scores are notably higher than those of the GlobalPointer model (F1: 0.86649, Precision: 0.89581, Recall: 0.83904) and its variant, GlobalPointer-FGM (F1: 0.87680, Precision: 0.90773, Recall: 0.84791).

Despite these quantitative advantages, the CRF model faced challenges in processing Chinese texts and generating complete event information without the need for additional rules. This suggests that while CRF excels in terms of accuracy and recall, it may require supplementary mechanisms or manual intervention to fully interpret complex or nuanced textual data.

On the other hand, although the GlobalPointer and GlobalPointer-FGM models scored lower on the standard evaluation metrics, they provided more comprehensive outputs with respect to event extraction. For instance, the GlobalPointer models were better at avoiding the introduction of irrelevant entities during prediction, which is a common issue observed with the CRF model. This characteristic makes the GlobalPointer models particularly useful in contexts where completeness and relevance of extracted events are crucial, even if it comes at the cost of slightly lower precision and recall scores.

Table 4. Performance comparison

Evaluation index	F1	Precision	Recall
GlobalPointer	0.86649	0.89581	0.83904
GlobalPointer-FGM	0.87680	0.90773	0.84791
CRF	0.91956	0.92551	0.91369

4.2.2 Results of Severity Assignment

Table 5 highlight the performance comparison between our proposed approach and baseline models, specifically XLNET and BERT, for vulnerability severity assignment. Our approach demonstrates significant improvements over both baseline models across multiple evaluation metrics.

When compared to XLNET, there is a 2.1% increase in accuracy (from 0.715569 to 0.736526), a 1.3% enhancement in F1 score (from 0.717886 to 0.730594), a slight yet noteworthy 0.3% improvement in precision (from 0.732133 to 0.735384), and a 1.6% boost in recall (from 0.715569 to 0.732035). These gains indicate that our approach not only achieves higher accuracy but also maintains a balanced improvement across precision and recall, suggesting its robustness in handling various aspects of severity assignment.

In contrast with BERT, our approach exhibits even more substantial superiority. BERT scores lower than our method, showing a 9.3% reduction in precision (0.642576 vs. 0.735384) and a 9.5% decrease in recall (0.636879 vs. 0.732035). This considerable gap underscores the effectiveness of our approach in accurately identifying and assigning severity levels to vulnerabilities, particularly highlighting its advantage in capturing relevant details from complex or nuanced texts with the help of event knowledge.

Table 5. Performance comparison between EICNN and baseline model

Model	Acc	F1	Precision	Recall
XLNET	0.715569	0.717886	0.732133	0.715569
Our Approach	**0.736526**	**0.730594**	**0.735384**	**0.732035**
BERT	0.636879	0.638428	0.642576	0.636879

Figure 8 and Figure 9 indicate minor overfitting of our approach during training. Additionally, the study examined GlobalPointer and CRF models to refine event extraction for severity assignment. Our approach stands out for its simplicity and efficiency, leveraging only descriptions from vulnerability reports and pre-trained event information, thus avoiding complex scoring systems. This approach facilitates swift data generation without requiring post-processing, making it highly effective for analyzing vulnerability threats.

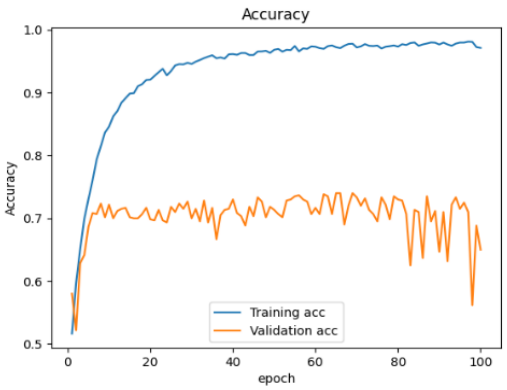

Figure 8. Our approach Accuracy (up), XLNET accuracy (down)

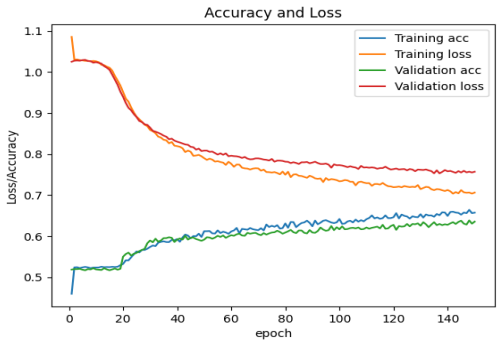

Figure 9. Accuracy and loss of BERT

5 Threats to Validity

Internal Validity: Data Bias and Imbalance: The manually annotated dataset (600 samples) may not fully represent the diversity of CNVD vulnerability reports, especially given the uneven distribution of severity levels. This imbalance could bias the model toward majority classes. Hyperparameter Tuning: While batch size and filter dimensions were selected experimentally, the lack of cross-validation or robustness checks raises concerns about overfitting, as evidenced by the overfitting observed in EICNN's training curves.

External Validity: Language and Domain Specificity: The method is tailored for vulnerability reports (CNVD). Its effectiveness on non-Chinese datasets (e.g., CVE/NVD) or other domains remains unverified, limiting generalizability. Data Source Dependency: The reliance on CNVDs semi-structured XML files may restrict applicability to other vulnerability databases with different formats or content styles.

6 Conclusion

This paper addresses the challenge of event extraction in the context of vulnerability reports from CNVD by proposing a specialized framework that combines deep learning techniques. We developed a schema to handle the high similarity among vulnerability texts and introduced an event extraction method using GlobalPointer and CRF with a RoBERTa pre-training model for dynamic text vector representation. The process involves trigger word extraction via GlobalPointer, followed by feature extraction using CRF. Entity positions are identified through GlobalPointer, while CRF integrates event features for enhanced fusion. Additionally, we proposed a novel method for assigning vulnerability severity by combining vulnerability descriptions with extracted event information, which outperformed baseline models in classification performance. This comprehensive approach effectively enhances the analysis and evaluation of vulnerability reports, contributing significantly to cybersecurity.

References

[1] D. Singh, S. Mohan, P. Dubey, Identifying Cyber Threats in Metaverse Learning Environment using Explainable Deep Neural Networks, *International Journal of Performability Engineering*, Vol. 20, No. 12, pp. 764-774, December, 2024.

[2] R. Kushwah, Navigating the Cybersecurity Landscape: Vulnerabilities, Mitigation Strategies and Future Outlooks, *International Journal of Performability Engineering*, Vol. 20, No. 11, pp. 688-698, November, 2024.

[3] Y. Wang, Y. Zhou, X. Zou, Q. Miao, W. Wang, The Analysis Method of Security Vulnerability Based on the Knowledge Graph, *Proceedings of the 2020 10th International Conference on Communication and Network Security*, Tokyo, Japan, 2020, pp. 135-145.

[4] S. Jiang, Y. Qi, H. Zhang, Z.-W. Bai, X. Lu, P. Wang, D3D: Dual 3-D Convolutional Network for Real-time Action Recognition, *IEEE Transactions on Industrial Informatics*, Vol. 17, No. 7, pp. 4584-4593, July, 2021.

[5] L. Su, J. Hu, W. Zheng, A Mutation-Based Data Enhancement Approach for Software Vulnerability Detection, *Journal of Internet Technology*, Vol. 25, No. 6, pp. 931-943, November, 2024.

[6] N. Meng, S. Nagy, D. Yao, W. Zhuang, G. A. Argoty, Secure coding practices in Java: Challenges and vulnerabilities, *Proceedings of the 40th International Conference on Software Engineering*, Gothenburg, Sweden, 2018, pp. 372-383.

[7] J. S. Qu, R. J. Zhang, Z. W. Zhang, N. Qiao, J. S. Pan, Image Sequence Facial Expression Recognition Based on Deep Residual Network, *Journal of Internet Technology*, Vol. 21, No. 6, pp. 1579-1587, November, 2020.

[8] S. Baltes, L. Dumani, C. Treude, S. Diehl, SoTorrent: Reconstructing and analyzing the evolution of Stack Overflow posts, *Proceedings of the 15th International Conference on Mining Software Repositories*, Gothenburg, Sweden, 2018, pp. 319-330.

[9] T. Nakamura, M. Conrad, Exploiting ferroptosis vulnerabilities in cancer, *Nature Cell Biology*, Vol. 26, No. 9, pp. 1407-1419, September, 2024.

[10] H. T. Nguyen, R. Grishman, Event detection and domain adaptation with convolutional neural networks, *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, Beijing, China, 2015, pp. 365-371.

[11] Y. Chen, L. Xu, K. Liu, D. Zeng, J. Zhao, Event extraction via dynamic multi-pooling convolutional neural networks, *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, Beijing, China, 2015, pp. 167-176.

[12] I. Keivanloo, J. Rilling, Y. Zou, Spotting working code examples, *Proceedings of the 36th International Conference on Software Engineering*, Hyderabad, India, 2014, pp. 664-675.

[13] S. Ananiadou, S. Pyysalo, J. I. Tsujii, D. B. Kell, Event extraction for systems biology by text mining the literature, *Trends in biotechnology*, Vol. 28, No. 7, pp. 381-390, July, 2010.

[14] T. Zhang, H. Ji, A. Sil, Joint entity and event extraction with generative adversarial imitation learning, *Data Intelligence*, Vol. 1, No. 2, pp. 99-120, April, 2019.

[15] J. Xie, H. R. Lipford, B. Chu, Why do programmers make security errors?, *2011 IEEE Symposium on Visual Languages and Human-Centric Computing*, Pittsburgh, PA, USA, 2011, pp. 161-164.

[16] A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet classification with deep convolutional neural networks, *Communications of the ACM*, Vol. 60, No. 6, pp. 84-90, June, 2017.

[17] K. Scarfone, P. Mell, An analysis of CVSS version 2 vulnerability scoring, *2009 3rd International Symposium on Empirical Software Engineering and Measurement*, Lake Buena Vista, FL, USA, 2009, pp. 516-525.

[18] C. Cortes, V. Vapnik, Support-vector networks, *Machine Learning*, Vol. 20, No. 3, pp. 273-297, September, 1995.

[19] R. Wang, L. Gao, Q. Sun, D. Sun, An improved CVSS-based vulnerability scoring mechanism, *2011 Third International Conference on Multimedia Information*

Networking and Security, Shanghai, China, 2011, pp. 352-355.

[20] M. Ahsan, K. E. Nygard, R. Gomes, M. Chowdhury, N. Rifat, J. F. Connolly, Cybersecurity threats and their mitigation approaches using Machine Learning—A Review, *Journal of Cybersecurity and Privacy*, Vol. 2, No. 3, pp. 527-555, September, 2022.

[21] O. O. Amoo, F. Osasona, A. Atadoga, B. S. Ayinla, O. A. Farayola, T. O. Abrahams, Cybersecurity threats in the age of IoT: A review of protective measures, *International Journal of Science and Research Archive*, Vol. 11, No. 1, pp. 1304-1310, February, 2024.

[22] S. Frei, M. May, U. Fiedler, B. Plattner, Large-scale vulnerability analysis, *Proceedings of the 2006 SIGCOMM Workshop on Large-Scale Attack Defense*, Pisa, Italy, 2006, pp. 131-138.

[23] A. Khazaei, M. Ghasemzadeh, V. Derhami, An automatic method for CVSS score prediction using vulnerabilities description, *Journal of Intelligent & Fuzzy Systems*, Vol. 30, No. 1, pp. 89-96, September, 2015.

[24] J. Lafferty, A. McCallum, F. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, *Proceedings of the 18th International Conference on Machine Learning (ICML)*, Williamstown, MA, USA, 2001, pp. 282-289.

[25] N. Sherje, Enhancing software development efficiency through AI-powered code generation, *Research Journal of Computer Systems and Engineering*, Vol. 5, No. 1, pp. 1-12, January to June, 2024.

[26] X. Gong, Z. Xing, X. Li, Z. Feng, Z. Han, Joint prediction of multiple vulnerability characteristics through multi-task learning, *2019 24th International Conference on Engineering of Complex Computer Systems (ICECCS)*, Guangzhou, China, 2019, pp. 31-40.

[27] J. Chen, C. Chen, J. Hu, J. Grundy, Y. Wang, T. Chen, Z. Zheng, Identifying smart contract security issues in code snippets from Stack Overflow, *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis*, Vienna, Austria, 2024, pp. 1198-1210.

[28] Y. Yu, X. Si, C. Hu, J. Zhang, A review of recurrent neural networks: LSTM cells and network architectures, *Neural computation*, Vol. 31, No. 7, pp. 1235-1270, July, 2019.

[29] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, X. Huang, Pre-trained models for natural language processing: A survey, *Science China Technological Sciences*, Vol. 63, No. 10, pp. 1872-1897, October, 2020.

Biographies

Xufan Zheng is an undergraduate student at Northwest Agriculture and Forestry University.

Chang Liu is currently a student of the Department of Software, University of Northwestern Polytechnical, for the Master degree. His research focus is on software security.

Tianci Li is currently a student of the School of Information Engineering, Yangzhou University, for the Master degree. His research focus is on software testing.

Xiaoxue Wu is currently an associate professor of Yangzhou University. She received the Ph.D degree in Cyberspace Security from Northwestern Polytechnical University.