Hindawi
Security and Communication Networks
Volume 2023, Article ID 2118305, 14 pages
https://doi.org/10.1155/2023/2118305

Research Article

VulDistilBERT: A CPS Vulnerability Severity Prediction Method Based on Distillation Model

Shaofeng Kai, Fan Shi, and Jinghua Zheng

College of Electronic Engineering, National University of Defense Technology, Hefei 230037, China

Correspondence should be addressed to Fan Shi; shifan17@nudt.edu.cn

Received 1 September 2022; Revised 11 October 2022; Accepted 24 November 2022; Published 13 April 2023

Academic Editor: Yuanyuan Huang

With the vigorous development of the Internet, the ecosystem of cyber-physical systems is also developing at a high speed, but cyber-physical systems may be accompanied by unknown vulnerabilities in the process of concrete implementation. Thus, the number of vulnerabilities in cyber-physical systems has been increasing year by year. The vulnerability evaluation speed cannot keep up with the vulnerability exposure speed. The traditional manual evaluation method can no longer effectively deal with such large-scale vulnerabilities, resulting in a backlog of vulnerabilities. Therefore, the vulnerability evaluation results have a certain lag. To address this problem, the paper proposes a vulnerability severity assessment method based on the distillation model. The method first uses data augmentation and integration of optimal subsets to improve the amount of information in the vulnerability description text, then uses the DistilBERT model to characterize the text of the vulnerability description text, and then the characterized feature vectors are classified based on the linear layer to achieve the purpose of assessing vulnerability severity. Compared with the current method of manual assessment based on the CVSS metric system, this method can automate the assessment of vulnerabilities based on vulnerability description text, which improves the speed of vulnerability assessment, and the assessment accuracy and other metrics achieved by this method are improved compared with similar studies. This approach provides an automated solution for cyber-physical systems vulnerability assessment and can better address the current situation where cyber-physical systems vulnerabilities are being exposed at an accelerated rate.

1. Introduction

Cyber-physical systems (CPS) is a set of physical devices (hardware) controlled by computer algorithms (mostly software), and CPS are becoming increasingly important in the information society. However, with high heterogeneity, large-scale deployment, and high dependency on private and sensitive data, CPS is more vulnerable to threats from the network. In recent years, there have been many cases of attacks on CPS hardware, software, and data in the system using vulnerabilities in the CPS, so CPS security needs to be given high priority. This paper provides a priority basis for CPS to repair vulnerabilities by evaluating the severity of CPS vulnerabilities, so as to improve CPS security.

CPS vulnerabilities are flaws in specific implementations of hardware, software, protocols, etc., or system security policies that can enable an attacker to access or compromise a system without authorization. Severity assessment of vulnerabilities can be better for vulnerability severity rating so that according to the severity rating of vulnerabilities to prioritize those high-risk vulnerabilities that have a serious threat to the system, this operation can save the cost and time of vulnerability repair. According to the U.S. National Vulnerability Database (NVD) data, vulnerabilities are showing a trend of increasing year by year, as shown in Figure1. As of September 1, 2022, the NVD has accepted 194,532 vulnerabilities in total. Since the beginning of this year, a total of 16,506 vulnerabilities have been accepted, and an average of nearly 70 vulnerabilities is accepted every day [2]. In addition to the need for vulnerability assessors to assess the severity of newly-added vulnerabilities, some vulnerability exposed earlier due to changes in environmental factors, such as the increase in the number of patch installations, will also lead to changes in the severity of these

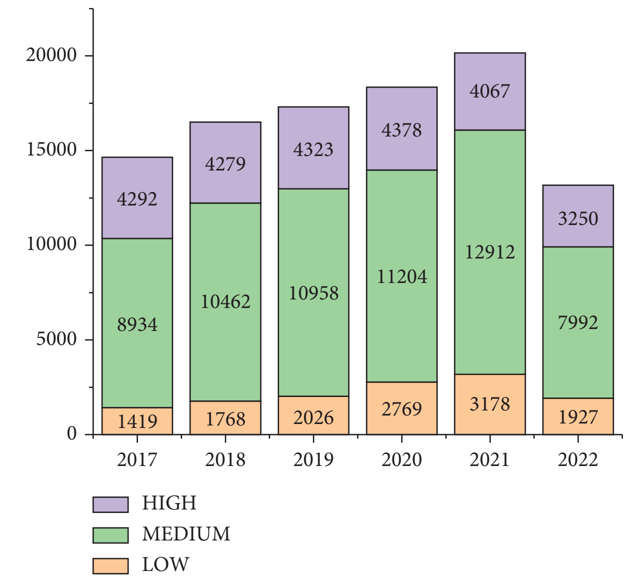

Figure 1: Vulnerability growth trend from January 1, 2017, to September 1, 2022. The data comes from the U.S. National Vulnerability Database [1].

vulnerabilities. The vulnerability assessors also need to reevaluate this part of the vulnerability. More vulnerability and fewer vulnerability assessors present a sharp contrast; in the long run; vulnerabilities will inevitably appear a backlog phenomenon. This leads to some vulnerability to determine the severity of the time lag behind the time of vulnerability exposure. Research by Chen et al. [3] and Ruohonen [4] show that the vulnerability assessors assess the severity of vulnerability lag behind the time of vulnerability exposure more than 130 days. During this time, unscrupulous individuals may be able to exploit these vulnerabilities to create threats and hazards to the system. There is an urgent need for a method that can quickly assess the severity of vulnerabilities, both to reduce the workload of vulnerability assessors and to improve network security. With the current methods of vulnerability assessment, the speed of vulnerability assessment can be improved by increasing the number of assessors, but this approach will increase the cost for security organizations, so a better way is to propose a method that can automatically assess vulnerabilities.

One idea is to characterize and classify vulnerability description text based on traditional machine learning methods, and some studies also deepen the model feature extraction capability based on deep learning algorithms, but this part of the research does not address the problem of multiple meanings of words and specialized terminology in the cybersecurity domain because algorithms using, for example, TF-IDF algorithms [5], rule-based statistical methods [6], and word embedding [7], cannot capture the contextual word relevance. For example, apple can mean both an apple in fruit and Apple Inc. However, the specific meaning of apple can be known if the context is combined. There are also some studies based on large models, such as BERT [8], but the parameters of models such as BERT are calculated in billions, and such a large model is bound to have an impact on the speed of the training and evaluation process. At the same time, the vulnerability description text also has a small amount of text information, creating the problem of data imbalance. According to statistics, the average description text contained in each vulnerability is about 40 words, and the NVD data show that the severity of the vulnerability there is also a serious imbalance, high-risk and medium-risk vulnerabilities occupy the vast majority of the sample, while low-risk vulnerabilities and critical-risk vulnerabilities occupy only a small part, if this part of the data used directly for training, it will inevitably lead to model bias and instability.

To address the above problems, this paper proposes a vulnerability severity assessment method based on the distillation model, which uses DistilBERT model to characterize the vulnerability description text, because DistilBERT model has been pretrained based on the general corpus, and the model itself already contains more knowledge, which can better extract the dependency relationship between features based on contextual information, thus solves the problem of multiple meanings of words and specialized terms, and then the features are further extracted using the LSTM [9] and CNN [10] algorithm, because for text classification tasks, the sentiment polarity of sentences often consists of individual words or phrases, and the positions of these decisive words and phrases in the sentences are not fixed, and it is difficult to capture such critical local information directly using the fully connected layer, and the data will be passed through the model in the process of the LSTM network can better solve the above problems. The article also addresses the problem of low data information and data imbalance and uses methods such as the data augmentation algorithm and optimal subset integration to describe the text for a better solution.

The experiments in this paper also validate the effectiveness of our proposed method. Compared with the current SOTA method, the accuracy of our evaluation is 96.62%, while the accuracy of SOTA is 93.95%, which is an improvement of 2.67%.

To address the challenge of vulnerability assessment, VulDistilBERT is proposed. The article has the following three main contributions:

(i) In this paper, we use the distillation model to better solve the problem that the text features are not strongly dependent and the word multiple meanings and technical terms cannot be recognized well.

(ii) This paper compiles a vulnerability assessment dataset based on NVD data and solves the problems of insufficient information and data imbalance based on data augmentation techniques and the optimal subset integration.

(iii) This paper proposes a vulnerability severity assessment framework based on the distillation model, which can automate the assessment of vulnerability severity levels. This method increases the speed of evaluation and reduces the workload of manual evaluation.

2. Background

2.1. The Common Vulnerability Scoring System. The Common Vulnerability Scoring System (CVSS) is an open framework developed by FIRST.Org, Inc. (FIRST) to characterize and quantify vulnerabilities. CVSS consists of three metric groups: the base metric group, the temporal metric group, and the environmental metric group. Since major vulnerability databases only provide base scores, the base metric group is the most used. The focus of this study is the base metric group. The base metric group reflects the inherent properties of vulnerabilities that remain unchanged over time and across user environments. The base metric group generates scores ranging from 0 to 10. The CVSS score can also be shown as a vector string, which is a textual way to represent the metric value compactly. The NVD, when using the CVSS, usually gives a string representation of the description and the corresponding vulnerability metric values. Table 1 shows the eight metrics and their meanings contained in the base metric group of CVSS V3.1, as well as the possible values of the metrics. The metric values are substituted into the CVSS quantification formula to obtain the base score. Finally, the base score can be converted into a vulnerability rating [12–14].

2.2. DistilBERT Model

2.2.1. The Meaning of Knowledge Distillation. Unlike pruning and quantization in model compression, knowledge distillation is a common method of model compression and knowledge migration. Knowledge distillation is done by constructing a small lightweight model and using the supervised information of a larger model with better performance to train the small model to achieve better performance and accuracy. As the name suggests, the teacher is responsible for teaching knowledge, and the student is responsible for receiving and digesting it. The final goal is that the student has successfully transferred most of the knowledge from the teacher's brain to his or her own brain, completing the knowledge distillation [15].

2.2.2. Purpose of Knowledge Distillation. It is not easy for humans to learn a subject well, let alone a model with no feelings and no brain. The only thing it can do is to use its big computing power to solve the problem; maybe a human can distinguish the emotional color of a sentence at a glance, while the machine will need constant trial and error corrections to achieve a closer result with humans. Therefore, this also leads to the large parameters of large models such as BERT. The parameters of the basic BERT model are about 115 million; however, this is still the only entry-level parameter in language models, which brings challenges for model training, evaluation, deployment, and application. If we can achieve or approach the effect that can only be achieved by the raw large model with a lightweight model, it can allow us to achieve two things in one. Experimental results demonstrate that DistilBERT reduces the model size by 40% and the inference operations are 60% faster while retaining 97% of the performance [16].

2.2.3. The Process of Knowledge Distillation. BERT is mainly based on a series of attention layers stacked on top of each other, so this means that the "hidden knowledge" learned by BERT is contained in these layers. The major difference between BERT models is the number of layers N, which is naturally proportional to the size of the model. It follows that the time taken to train the model and the time for forwarding propagation also depends on N and, of course, the memory used to store the model. Therefore, the logical conclusion of distillation BERT is to reduce N. DistilBERT is done by halving the number of layers and initializing the student's layer from the teacher's layer. For example, a distillation of the base BERT is done by distilling the 12-layer BERT to obtain a 6-layer DistilBERT, first pretraining the BERT, and then distilling the 12-layer BERT with a large-scale prefeed of the training BERT to obtain it. The knowledge distillation process is shown in Figure 2.

3. Methods

This study aims to design an efficient approach using vulnerability description text to predict vulnerability severity. This approach will help analysts quickly analyze the vulnerability severity levels. The paper proposes a learning method based on the DistilBERT model, which fine-tunes the DistilBERT model to improve the model's learning efficiency and prediction effect.

3.1. Methodology Overview. Figure 3 depicts the paper's two primary phases: The DistilBERT model transfer learning and severity prediction. By employing DistilBERT model transfer learning, a fine-tuned model will be developed to predict vulnerability severity. Severity prediction is a four-step process, i.e., data integration, text tokenization, token embedding, and vulnerability severity prediction. These steps are shown in Figure 3. First, data augmentation and optimal subset integration operations are performed on the raw data, and the integrated data is used as model input, then the vulnerability description content is divided into numerous tokens during the tokenization stage, and then the tokens are embedded by the fine-tuned DistilBERT model. Finally, the paper uses the linear function to predict the severity of vulnerabilities. The remainder of this section will provide a detailed description of the framework.

3.2. DistilBERT Transfer Learning. DistilBERT transfer learning fine-tunes the pretrained model utilizing the self-built corpus collected in the NVD. The pretrained DistilBERT is obtained by distilling the Bert model. DistilBERT transfer learning begins with downloading the appropriate pretrained DistilBERT model. In this study, the pretrained model is "DistilBERT-base-uncased." Then, the domain corpus is used to fine-tune the model. The domain corpus is constructed from 1999 to 2022 NVD vulnerability descriptions. The input and output relationships of the DistilBERT model are shown in the following equation:

$$\mathbf{e}^{[\mathbf{l}]} = f_{\text{DistilBERT}}\left(\Theta_{pre-\text{DistilBERT}}, \mathbf{t}\right) \tag{1}$$

Table 1: Description and possible values for base metric group of CVSS. The metric names in this table will be referred to by abbreviations and ID number in [11].

ID	Metric	Description	Possible values
0	Attack vector (AV)	This metric represents the conditions under which exploiting vulnerability is conceivable. The farther an attacker may exploit a susceptible component, the higher this metric	Physical Network Local Adjacent
1	Attack complexity (AC)	This metric reflects the attacker-uncontrolled circumstances needed to exploit the vulnerability	Low High
2	Privileges required (PR)	This metric represents an attacker's privilege before exploitation. The score is the highest when no privileges are necessary	None Low High
3	User interaction (UI)	This metric represents the necessity for a human user other than the attacker to be involved in the successful penetration of the susceptible component	Required None
4	Scope (S)	This metric measures whether one component's vulnerability impacts other components' resources	Unchanged Changed
5	Confidentiality (C)	This metric quantifies the confidentiality of a successfully exploited vulnerability on the component most directly and predictably affected by the attack	None Low High
6	Integrity (I)	This metric represents vulnerability's influence on integrity. Integrity means truthfulness and trustworthiness	None Low High
7	Availability (A)	This metric assesses a vulnerability's influence on a component's availability	None Low High

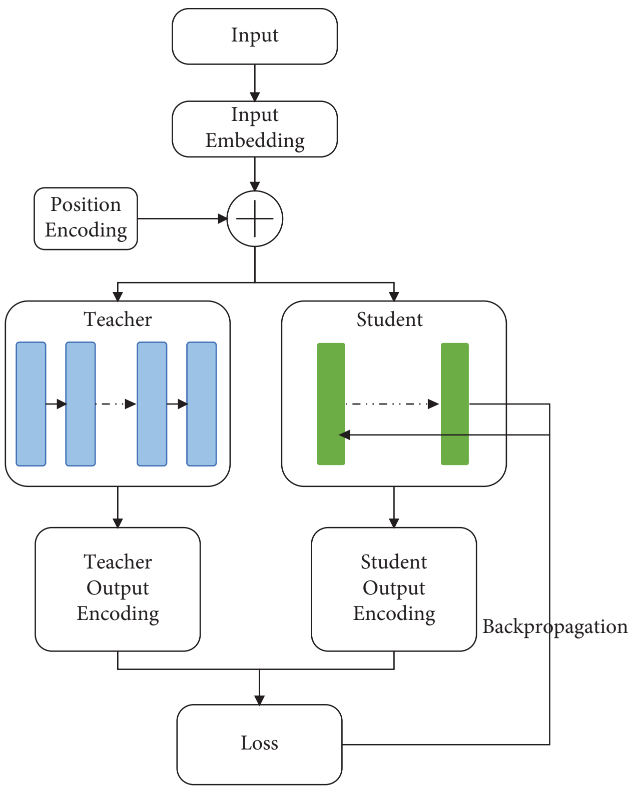

FIGURE 2: Distillation model working process. Students gain knowledge by learning from teachers.

where $\mathbf{t} = [t_1, t_2, \cdots, t_n]$ is a sequence of the token list with n tokens, which are tokenized based on the vulnerability description text; $\mathbf{e}^{[l]} = [e_1^{[l]}, e_2^{[l]}, \cdots, e_n^{[l]}]$ is the pretrained DistilBERT model's l th layer token embedding; $\Theta_{pre-\text{DistilBERT}}$ represents the parameters of the pretrained DistilBERT model; $f_{\text{DistilBERT}}(\cdot)$ is the conversion function of $\mathbf{t}$ and $\mathbf{e}^{[l]}$, determined by the DistilBERT structure; $e_i^{[l]}$ is the l th layer token embedding of the i th token t_i, $e_i^{[l]} \in \mathbb{R}^{H_{\text{DistilBERT}}^{[l]}}$, where $H_{\text{DistilBERT}}^{[l]}$ is the DistilBERT l th layer hidden layer size. By transfer learning, the parameters of DistilBERT are changed from its pretrained state $\Theta_{\text{pre-DistilBERT}}$ to its fine-tuned state $\Theta_{\text{fine-tuned DistilBERT}}$. Compared to training a DistilBERT model from scratch, using transfer learning on a DistilBERT model maintains a comprehensive model's high performance while avoiding the high training cost and lack of domain data [17].

3.3. Vulnerability Severity Prediction

3.3.1. Data Integration

(1) Data Augmentation. The statistics of the vulnerability description text showed that there was a serious imbalance in the data labels, and the results are shown in Figure 4. CRITICAL, HIGH, MEDIUM, and LOW accounted for 15.1%, 42.9%, 40.1%, and 1.9%, respectively. The low percentage of low-risk vulnerabilities may cause the trained model to fail to accurately predict low-risk vulnerabilities, which also affects the robustness of the model. In this paper, we adopt the method of data augmentation to solve the data imbalance problem. Data augmentation (DA) is a representation of processing more data from the raw data without substantially increasing the data and improving the quantity and quality of the raw data to approach the value generated by more data volume to improve the learning effect of the model. In this paper, we mainly adopt synonym replacement (SR) and random deletion (RD) to expand the sample with a small number of tags [18]. Synonym replacement refers to randomly selecting n words from the sentence that do not belong to the deactivated word set and randomly choosing their synonyms to replace them. Random deletion refers to randomly removing each word in a sentence with a probability of p.

(2) Optimal Subset. The analysis of the vulnerability description text shows that the average length of each vulnerability description statement is 43.77 words and 82.96% of the vulnerability description statements are less than 64 words, which is shown in Figures 5 and 6. When the model is trained to a certain level, the prediction effect of the model can no longer be improved, and even overfitting will occur.

In this paper, we adopt the method of extracting CVSS metrics and selecting the optimal subset (OS) of metrics to be incorporated into the vulnerability description text to improve the amount of textual information. The current latest version of CVSS is 3.1, whose base metric group contains 8 metrics, and we investigate the impact of these 8 metrics on the final vulnerability severity classification. In this study, CVSS metrics are first combined according to the

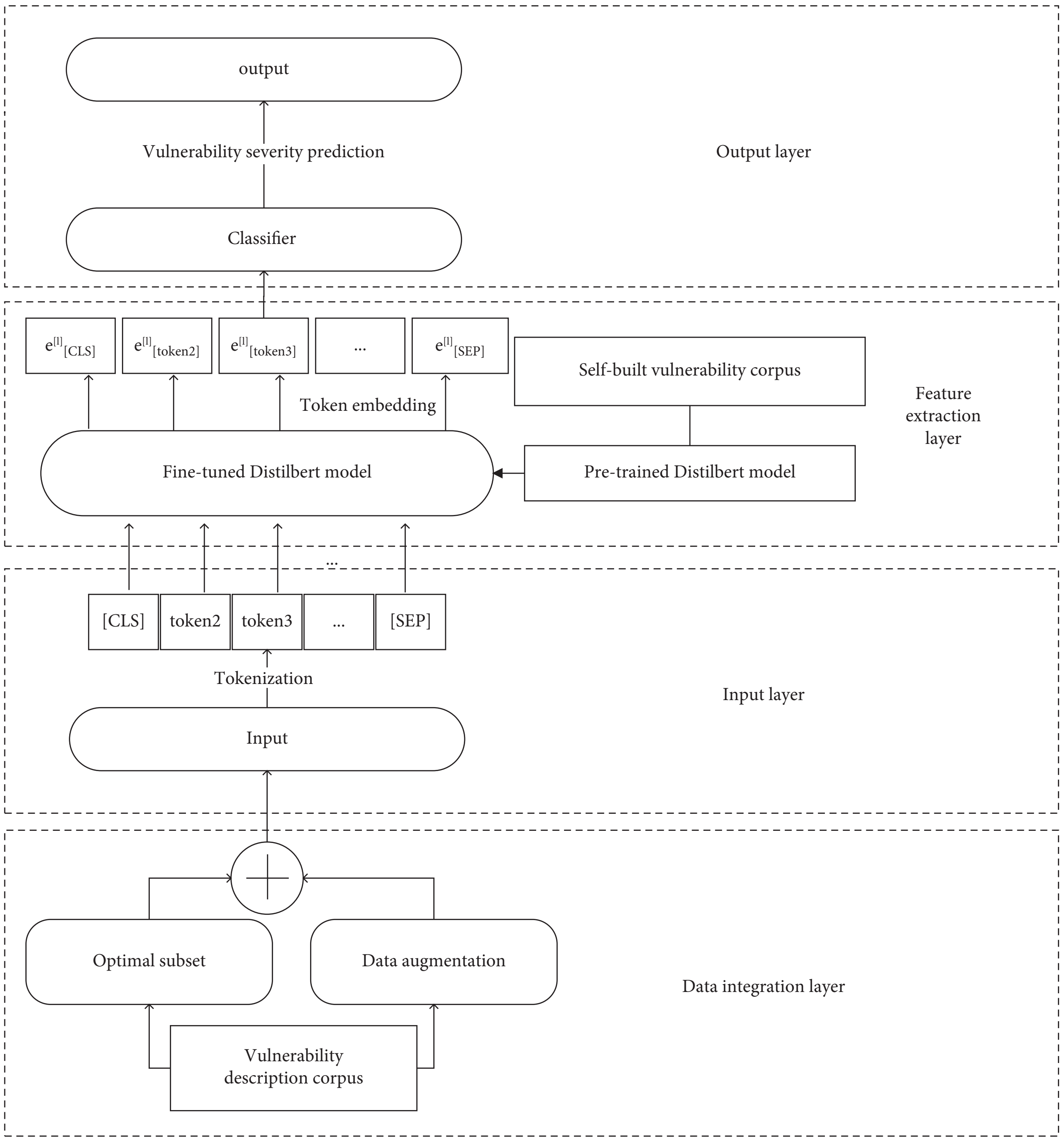

Figure 3: The framework structure of the algorithm in this paper. The framework mainly includes the data integration layer, input layer, feature extraction layer, and output layer.

full ranking method, and then a variety of typical machine learning methods are selected to select the optimal subset [19], and the relevant methods are random forest, decision tree, KNN, AdaBoost, SVM, logistic regression, multilayer perceptron, gradient boosting algorithm, and so on. Because the optimal subset is selected to improve the final vulnerability severity level classification effect, finally this paper lists the classification effect of related machine learning algorithms on the combination of metrics, due to the reason of space, only the first two combinations with better effect in each category are listed and the combination with the best effect is shown in bold. The results are shown in Tables 2 and 3. The numbers in the table refer to the metric names; please refer to Table 1 for the specific reference relationship. The effects of the algorithms are then combined to finally determine the optimal subset of each class of combinations, as shown in Table 4.

3.3.2. Model Building

(1) Input Layer. The main function of the input layer is text tokenization. Text tokenization is a data preprocessing process in which the description text $X = [V_1, V_2 \cdots, V_n]$ are turned into token sequences $T = [t^{(1)}, t^{(2)}, \cdots, t^{(n)}]$, $\mathbf{t}^{(i)} = [t_1, t_2, \cdots, t_k]$ is the token sequences obtained from the

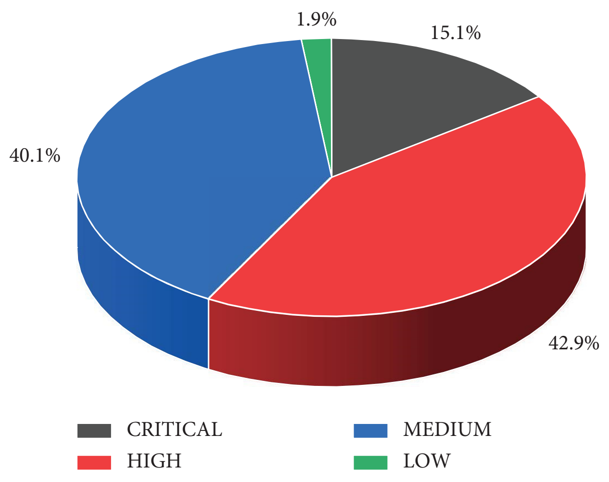

Figure 4: The proportion of each category of raw data.

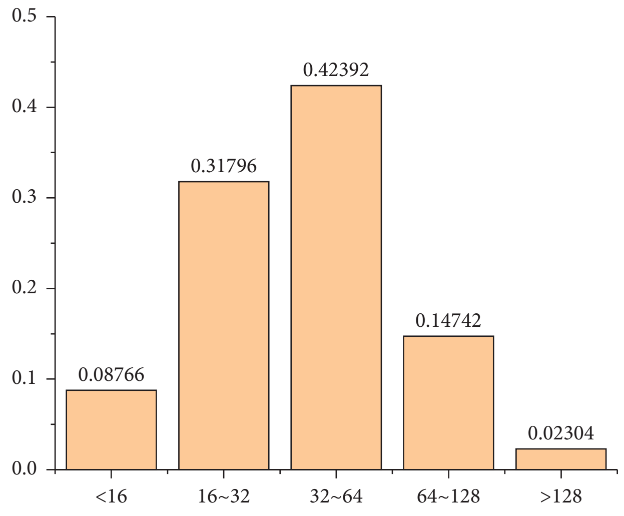

Figure 5: The distribution of the number of words in the vulnerability description text. The maximum number of words is in the range of 32–64.

description V_i; k is the maximum sequence length of the preset token. The symbol t_j denotes the j th token obtained from the characterization of the description text V_i, $i \in \{1, 2, \cdots, n\}$, $j \in \{1, 2, \cdots, k\}$.

(2) Feature Extraction Layer. When text tokenization is done, the result is what goes into token embedding. When the fine-tuned DistilBERT is given a token list **t**, different transfer layers will give different levels of token embedding. For instance, this is how the token embedding of the layer l from the fine-tuned DistilBERT is shown as follows:

$$\mathbf{e}^{[l]} = f_{\mathrm{DistilBERT}}\left(\Theta_{\text{fine-tuned DistilBERT}}, \mathbf{t}\right) \quad (2)$$

Similar to (1), $\mathbf{t} = [t_1, t_2, \cdots, t_k]$ is a token sequence that consists of k tokens, $\mathbf{e}^{[l]} = [e_1^{[l]}, e_2^{[l]}, \cdots, e_n^{[l]}]$ represents the l th layer of token embedding. $e_i^{[l]}$ is the i th token t_i, $e_i^{[l]} \in \mathbb{R}^{H^{[l]}_{\mathrm{DistilBERT}}}$, where $H^{[l]}_{\mathrm{DistilBERT}}$ is the DistilBERT l th layer hidden size.

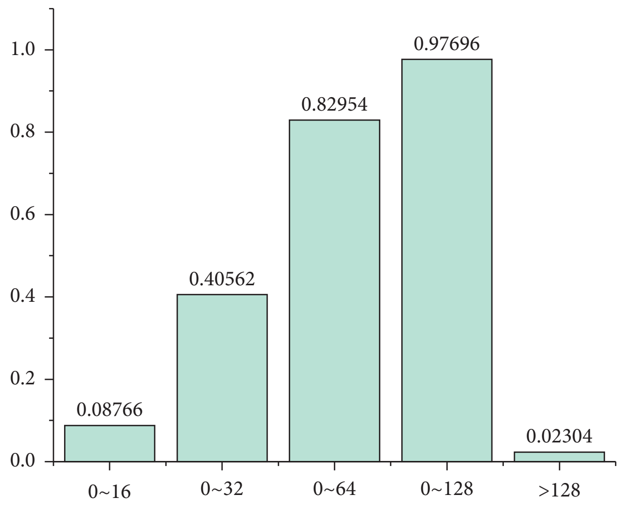

Figure 6: Cumulative distribution of the number of words in the vulnerability description text. Vulnerability description text less than 128 words accounted for the vast majority.

(3) Output Layer. In the research, we utilize the fully connected layer as a classifier. It can perform a linear transformation on the incoming data. The following is the formula to use: $\hat{y}_i = W^k \mathbf{e}^{[\mathbf{L}]} + b^k$, $\hat{y}_i$ is the possible category of the vulnerability severity, W^k and b^k are the functions' weights and biases, $\mathbf{e}^{[\mathbf{L}]}$ is the token embedding of the final layer output.

4. Experiments and Results

4.1. Experimental Data and Experimental Setup. The paper uses data from the US National Security Vulnerability Database [20], which contains all security vulnerabilities released from 1999 to 2022. The vulnerability description information in the web page is used as the dataset's text item, and the severity of vulnerabilities is processed to label items. There are four types of labels in the dataset, which correspond to the severity of four types of vulnerabilities, i.e., critical risk, high risk, medium risk, and low risk. The collected dataset is represented as $D = \{X, Y\}$. Examples of datasets are shown in the following Table 5. The vulnerability description item in the table is the input of the model, and the severity item is the label of the model. After processing, the dataset contains 105,984 vulnerability samples. The dataset was split into the training and test datasets in the following proportions: 85%: 15%. The statistics indicate that 99.8% of vulnerability descriptions are less than 256 words, with an average of 43.77 words per sample. The pretrained DistilBERT model used in the paper is the "DistilBERT-base-uncased" model [21]. In the data integration phase, the value of SR is set to 0.05 and the value of RD is set to 0.1. In the integration of the optimal subset, the integration is the optimal subset composed of four metrics: PR, UI, C, and I. All vulnerability descriptions were used to fine-tune the pretrained DistilBERT model. Two NVIDIA GeForce RTX 3090 GPUs were used for fine-tuning and training.

Table 2: Random forest, decision tree, AdaBoost, and KNN algorithms for evaluation results of metric combinations. Taking the first row of the random forest algorithm as an example, the vulnerability severity assessment using only one metric, 5, can still achieve an accuracy rate of 65.39%.

Random forest		Decision tree		AdaBoost		KNN	
Metric combination	Accuracy	Metric combination	Accuracy	Metric combination	Accuracy	Metric combination	Accuracy
5	0.6539	5	0.6539	5	0.6539	5	**0.6539**
6	**0.6687**	6	**0.6687**	6	**0.6687**	7	0.6523
0, 6	0.6749	0, 6	0.6749	0, 6	0.6749	0, 7	**0.6581**
5, 6	**0.6765**	5, 6	**0.6765**	5, 6	**0.6765**	3, 5	0.6578
2, 3, 5	0.7785	2, 3, 5	0.7785	2, 3, 5	**0.7785**	0, 2, 6	0.7637
2, 3, 6	**0.8238**	2, 3, 6	**0.8238**	2, 3, 6	0.7694	2, 3, 6	**0.8239**
2, 3, 5, 6	**0.8894**	2, 3, 5, 6	**0.8894**	2, 3, 5, 6	**0.8882**	2, 3, 5, 6	**0.8804**
2, 3, 5, 7	0.8841	2, 3, 5, 7	0.8841	2, 3, 5, 7	0.8841	2, 3, 5, 7	0.8646
0, 2, 3, 5, 6	**0.9280**	0, 2, 3, 5, 6	**0.9281**	0, 2, 3, 5, 7	**0.9225**	0, 2, 3, 5, 6	**0.9262**
0, 2, 3, 5, 7	0.9225	0, 2, 3, 5, 7	0.9225	1, 2, 3, 5, 7	0.9123	0, 2, 3, 5, 7	0.9219
0, 1, 2, 3, 5, 6	**0.9574**	0, 1, 2, 3, 5, 6	**0.9575**	0, 1, 2, 3, 5, 6	**0.9527**	0, 1, 2, 3, 5, 6	**0.9560**
0, 2, 3, 5, 6, 7	0.9525	0, 2, 3, 5, 6, 7	0.9523	0, 1, 2, 3, 5, 7	0.9480	0, 2, 3, 5, 6, 7	0.9493
0, 1, 2, 3, 4, 5, 6	0.9699	0, 1, 2, 3, 4, 5, 6	0.9699	0, 1, 2, 3, 5, 6, 7	**0.9850**	0, 1, 2, 3, 4, 5, 6	0.9685
0, 1, 2, 3, 5, 6, 7	**0.9852**	0, 1, 2, 3, 5, 6, 7	**0.9849**	0, 2, 3, 4, 5, 6, 7	0.9640	0, 1, 2, 3, 5, 6, 7	**0.9844**

Table 3: SVM, logistic regression, multilayer perceptron, and gradient boosting algorithms for evaluation results of metric combinations.

SVM		Logistic regression		Multilayer perceptron		Gradient boosting	
Metric combination	Accuracy	Metric combination	Accuracy	Metric combination	Accuracy	Metric combination	Accuracy
5	0.6539	5	0.6539	5	0.6539	5	0.6539
6	**0.6687**	6	**0.6687**	6	**0.6687**	6	**0.6687**
0, 6	0.6749	5, 6	**0.6765**	0, 6	0.6748	0, 6	0.6749
5, 6	**0.6765**	5, 7	0.6701	5, 6	**0.6765**	5, 6	**0.6765**
2, 3, 5	0.7785	2, 3, 6	**0.7976**	0, 3, 6	0.7352	2, 3, 5	0.7785
2, 3, 6	**0.8238**	2, 5, 7	0.7093	2, 3, 5	**0.7774**	2, 3, 6	**0.8238**
2, 3, 5, 6	**0.8892**	2, 3, 5, 7	**0.87412**	2, 3, 5, 7	**0.8823**	2, 3, 5, 6	**0.8887**
2, 3, 5, 7	0.8839	2, 3, 6, 7	0.8275	2, 3, 6, 7	0.8461	2, 3, 5, 7	0.8810
0, 2, 3, 5, 6	**0.9274**	1, 2, 3, 5, 7	**0.8962**	0, 2, 3, 5, 7	**0.9123**	0, 2, 3, 5, 6	**0.9244**
0, 2, 3, 5, 7	0.9218	2, 3, 5, 6, 7	0.8935	1, 2, 3, 5, 6	0.9069	1, 2, 3, 5, 6	0.9187
0, 1, 2, 3, 5, 6	**0.9575**	0, 2, 3, 5, 6, 7	0.9130	0, 1, 2, 3, 5, 7	0.9265	0, 1, 2, 3, 5, 6	0.9443
0, 2, 3, 5, 6, 7	0.9522	1, 2, 3, 5, 6, 7	**0.9271**	1, 2, 3, 5, 6, 7	**0.9424**	0, 2, 3, 5, 6, 7	**0.9488**
0, 1, 2, 3, 4, 5, 6	0.9691	0, 1, 2, 3, 5, 6, 7	**0.9486**	0, 1, 2, 3, 5, 6, 7	**0.9616**	0, 1, 2, 3, 5, 6, 7	**0.9792**
0, 1, 2, 3, 5, 6, 7	**0.9845**	1, 2, 3, 4, 5, 6, 7	0.9252	1, 2, 3, 4, 5, 6, 7	0.9470	0, 2, 3, 4, 5, 6, 7	0.9544

Table 4: The optimal subset selected for each combination.

Optimal subsets	Included metrics
Optimal subset with 1 metric	I
Optimal subset with 2 metrics	C, I
Optimal subset with 3 metrics	PR, UI, I
Optimal subset with 4 metrics	PR, UI, C, I
Optimal subset with 5 metrics	AV, PR, UI, C, I
Optimal subset with 6 metrics	AV, AC, PR, UI, C, I
Optimal subset with 7 metrics	AV, AC, PR, UI, C, I, A

4.2. Ablation Experiments. In this section, we set up multiple groups of control experiments to compare the classification effects of several pretrained models in fine-tuned and non-fine-tuned states, set up the effects of average pooling, maximum pooling, and no pooling on the classification results, and compare the differences in the effects of the linear layer, CNN, and LSTM methods as classifiers. It is demonstrated that the fine-tuned DistilBERT has a smaller size, faster inference speed, and top prediction results compared to other pretrained models.

4.2.1. Pretrained Model Effect Analysis. Pretrained models are a class of deep learning frameworks that have been trained on a large amount of data and can usually be used directly for downstream tasks, but since the data distribution at the time of pretraining may not be the same as that of the downstream tasks, the models may not be well adapted to the downstream tasks, so when pretrained models are used, they are usually fine-tuned on the existing data. In this study, four pretrained models were selected for the vulnerability assessment task, and their vulnerability classification effects were compared in the fine-tuned and non-fine-tuned states, respectively. The four pretrained models are BERT, XLNET [22], ROBERTA [23], and DistilBERT. The results are shown in the following table. Table 6 shows the effect of the test set on each pretrained model and Table 7 shows the effect of the test set on each fine-tuned model with the best number of

Table 5: Sample example of part of the dataset.

CVE ID	Vulnerability description	Severity
CVE-2022-40980	Potential unauthenticated file deletion vulnerability on Trend Micro Mobile Security for Enterprise 9.8 SP5 could allow an attacker with access to the Management Server to delete files. This issue was resolved in 9.8 SP5 Critical Patch 2	Critical
CVE-2022-3079	Festo control block CPX-CEC-C1 and CPX-CMXX in multiple versions allow unauthenticated, remote access to critical webpage functions which may cause a denial of service	High
CVE-2022-38846	EspoCRM version 7.1.8 is vulnerable to missing secure flag allowing the browser to send plain text cookies over an insecure channel (HTTP). An attacker may capture the cookie from the insecure channel using MITM attack	Medium
CVE-2022-39850	Improper access control in mum_container_policy service prior to SMR Oct-2022 release 1 allows allows unauthorized read of configuration data	Low

epochs taken as the number of epochs with the smallest validation loss. From the data in the table below, it can be seen that the DistilBERT model performs better on the test set in terms of accuracy and precision before fine-tuning, and requires fewer training epochs, while the performance after fine-tuning is generally on par with other models and exceeds them in some metrics. Table 8 gives a comparison of the performances.

4.2.2. Pooling Effect Analysis. The pooling layer can further aggregate the upstream features, which can effectively reduce the size of the parameter matrix, thereby reducing the number of parameters in the final connection layer, so adding a pooling layer can speed up the computation and prevent overfitting. We analyze the pooling effect for the fine-tuned DistilBERT models with maximum pooling, minimum pooling, and no pooling, respectively [24]. The results are shown in Table 9. From the data in the following table, it can be seen that the final classification effect is not much improved for the vectors obtained from the pretrained model and then pooled, and it is even not as good as the effect without pooling.

4.2.3. Data Augmentation Effect Analysis. In this paper, the vulnerability descriptions labeled low were amplified by 15 times, and the vulnerability descriptions labeled critical were amplified by 2 times, while the number of other category labels remained unchanged. The raw data sample size is 105,984, and after amplification, the sample size is 150,159. The percentage of each category is shown in Figure 7. To evaluate the impact and effect of augmented data, the integrated optimal subset of data, and both data augmentation and the integrated optimal subset of data on vulnerability severity prediction, we determined the token embedding model as a fine-tuned DistilBERT without pooling. The data is also validated using different classifiers under the same dataset. Table 10 gives the evaluation results of different classifiers in the vulnerability description text with data augmentation; Table 11 gives the evaluation results of different classifiers in the vulnerability description text with integrated optimal subset; and Table 12 gives the evaluation results of different classifiers in the vulnerability description text with both data augmentation and integrated optimal subset. Analysis of the data in the table below shows that the vulnerability description text with data augmentation improves the model better than the vulnerability description text incorporating the optimal subset. In the vulnerability description text with data augmentation, the accuracy of the pretrained model improves by 9% over the raw data and the accuracy of the fine-tuned model improves by 14% over the raw data with the same classifier. In the vulnerability description text incorporating the optimal subset, the accuracy of the pretrained model improves by 1% over the raw data, and the accuracy of the fine-tuned model improves 15% over the raw data with the same classifier, while the vulnerability description text with both data augmentation and incorporation of the optimal subset has better results than the single means, the accuracy of the pretrained model improves 19% over the raw data with the same classifier, and the accuracy of the fine-tuned model improves 19% over the raw data with the same classifier. The accuracy of the pretrained model improves by 19% over the raw data, 10% over the data-enhanced text, and 18% over the text incorporating the optimal subset. The accuracy of the fine-tuned model improved by 19% over the raw data, 5% over the data-augmented text, and 4% over the text incorporated into the optimal subset.

4.3. Comparison with Other Similar Works. In the study, the accuracy, precision, recall, and F1 score of fine-tuned DistilBERT are compared with those of similar works. The results of other works are taken from the original paper. Table 13 displays the pertinent data. The results demonstrate that our method enhances the performance of vulnerability severity prediction.

4.4. Analysis of Results. Tables 6 and 7 give the classification effects of different models in fine-tuning and pretraining states using a linear layer on the raw data, combined with training epochs, accuracy, and other data, DistilBERT model is selected as the text characterization model, Table 8 gives the improvement effect of DistilBERT model in the fine-tuning state compared with the pretraining state. From the data, it can be seen that in the fine-tuned model, the metrics are significantly improved, thus proving that the fine-tuning approach does have a significant improvement on the task.

Table 6: Results of evaluating the raw dataset with the pretrained model.

Models	Best epoch	Loss	Accuracy	Precision	Recall	F1 scores
BERT	26	1.038	0.54	**0.42**	0.33	0.32
XLNET	48	**0.978**	**0.56**	0.41	**0.36**	**0.35**
ROBERTA	74	1.043	0.54	0.34	0.32	0.29
DistilBERT	**24**	1.015	**0.56**	**0.42**	0.35	0.34

Table 7: Results of evaluating the raw dataset with the fine-tuned model.

Models	Best epoch	Loss	Accuracy	Precision	Recall	F1 scores
Fine-tuned BERT	**2**	0.580	0.77	0.75	0.62	0.66
Fine-tuned XLNET	4	0.570	0.78	0.74	0.65	0.68
Fine-tuned ROBERTA	4	**0.566**	0.78	0.76	0.65	0.68
Fine-tuned DistilBERT	4	0.571	**0.78**	**0.76**	**0.68**	**0.70**

Table 8: Performance comparison between pretrained model and fine-tuned model.

Models	Pretrained DistilBERT	Fine-tuned DistilBERT	Improvement (%)
Best epoch	24	4	500
Loss	1.015	0.571	43.74
Accuracy	0.56	0.78	28.21
Precision	0.42	0.76	43.59
Recall	0.35	0.68	48.53
F1 score	0.34	0.70	51.43

Table 9: Results of pooling effect comparison using fine-tuned DistilBERT model.

Models	Pool	Best epoch	Loss	Accuracy	Precision	Recall	F1 scores
	Mean	**3**	0.585	0.77	**0.78**	0.64	0.67
Fine-tuned DistilBERT	Max	4	0.588	0.77	0.75	0.64	0.67
	CLS	4	**0.571**	**0.78**	0.76	**0.68**	**0.70**

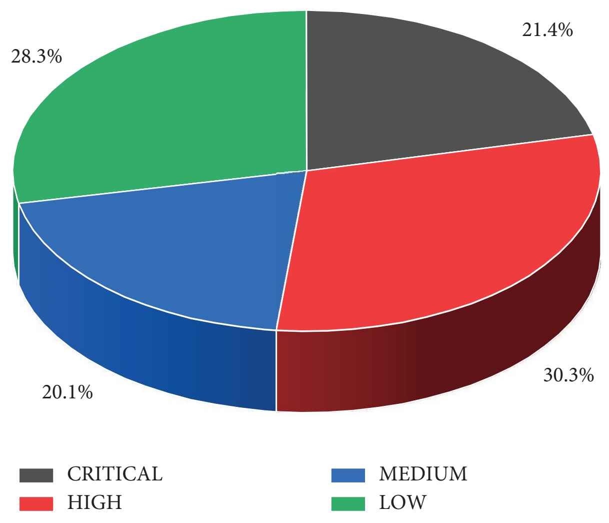

Figure 7: After data augmentation, the proportion of each category in the dataset.

The effects of different pooling effects on the model classification are given in Table 9. From the data in the table, it can be concluded that adding a pooling layer does not improve the model performance well, but increases the complexity of the model. Tables 10–12 give the control results of different information augmentation methods. From the data in the table, it can be analyzed that CNN, LSTM, and other complex networks do not improve the classification effect of the model well, probably because fine-tuned DistilBERT model has already characterized the features of the model well, even if a deeper network is used for feature extraction, its effect will not be better than the simple linear layer effect. Moreover, the data in Tables 10–12 can be analyzed to conclude that performing information

Table 10: Results of comparing the performance of different classifiers on a data-augmented dataset.

Models	Classifiers	Best epoch	Loss	Accuracy	Precision	Recall	F1 scores
Pretrained DistilBERT	Linear	32	0.927	0.65	0.66	0.65	0.65
Fine-tuned DistilBERT	Linear	3	0.221	0.92	0.92	0.92	0.92
Fine-tuned DistilBERT	CNN	4	0.225	0.91	0.92	0.92	0.92
Fine-tuned DistilBERT	LSTM	3	0.228	0.91	0.92	0.92	0.92

Table 11: Results of comparing the performance of different classifiers on the dataset integrated with the optimal subset.

Models	Classifiers	Best epoch	Loss	Accuracy	Precision	Recall	F1 scores
Pretrained DistilBERT	Linear	31	1.035	0.57	0.43	0.34	0.31
Fine-tuned DistilBERT	Linear	4	0.231	0.93	0.88	0.87	0.87
Fine-tuned DistilBERT	CNN	4	0.237	0.93	0.90	0.87	0.88
Fine-tuned DistilBERT	LSTM	4	0.234	0.93	0.86	0.90	0.88

Table 12: The results of comparing the performance of different classifiers on the dataset with the optimal subset integrated and data augmented.

Models	Classifiers	Best epoch	Loss	Accuracy	Precision	Recall	F1 scores
Pretrained DistilBERT	Linear	34	0.814	0.75	0.77	0.76	0.76
Fine-tuned DistilBERT	Linear	2	0.095	0.97	0.97	0.97	0.97
Fine-tuned DistilBERT	CNN	2	0.093	0.97	0.97	0.97	0.97
Fine-tuned DistilBERT	LSTM	2	0.092	0.97	0.97	0.97	0.97

Table 13: Comparison of results related to similar works.

Researchers	Feature extraction method	Classifiers	Accuracy (%)	Precision (%)	Recall (%)	F1 scores (%)
Khazaei et al. [25]	TF-IDF	Fuzzy system	88.37	—	—	—
Spanos et al. [26]	Document-term matrix	Decision tree	79.12	75.54	71.26	73.02
		Neural network	78.26	73.59	70.24	71.68
		SVM	79.53	78.49	68.21	71.50
Nakagawa et al. [27]	One-hot	CNN	72.50	—	—	—
Wang et al. [28]	Feature vector	PCA + XGBOOST	92.38	—	—	—
Han et al. [29]	Word embedding	1-Layer CNN	81.60	81.80	81.50	81.60
Liu et al. [30]	Text mining	XGBoost	87.30	—	—	—
		CNN	92.04	—	—	—
		LSTM	93.73	—	—	—
		TextRCNN	93.95	—	—	—
Our method	Fine-tuned DistilBERT with DA	Linear	90.64	91.92	91.92	91.83
	Fine-tuned DistilBERT with OS	Linear	92.80	88.32	87.26	87.07
	Fine-tuned DistilBERT with DA and OS	Linear	96.62	97.11	97.06	97.05

augmentation does have a better enhancement effect on the model compared to the raw dataset, where the dataset combining data augmentation and optimal subset integration has the best effect, reaching 97% accuracy, ahead of other current methods.

4.5. Discussion. The experimental results show that the fine-tuned DistilBERT model does have a better prediction capability for vulnerability severity, which benefits from both the power of the pretrained model itself and the knowledge provided by the fine-tuned corpus. Compared with traditional machine learning and deep learning, the DistilBERT model obtains a large amount of general knowledge from the large-scale corpus and domain knowledge from the fine-tuned corpus, which is the key to the successful application of the large model to downstream tasks. This knowledge compensates for the difficulties caused by insufficient data in downstream tasks, thus significantly improving them. In addition, the results show that pooling the fine-tuned DistilBERT model followed by a pooling layer and classification followed by a deep learning model do not further improve the model, which also indicates that the DistilBERT model has already performed a good feature extraction of the data, and even adding other modules cannot optimize the features further, but rather increases the complexity of the model, which is not conducive to the application of the model. After the optimal subset integration and data enhancement, the model effect has been improved. The key is that the data enhancement technique generates new data by perturbing the data itself to a certain extent, and the model improves its generalization ability by continuously learning a large amount of new data, and the optimal subset also further increases the information content of the data. However, the current model has problems, such as large number of parameters and low interpretability. In future research, we will further study the optimization and interpretability of the model in order to make the model have better application capability.

5. Related Work

Vulnerability assessment is one of the research topics in the field of cyberspace security, which is centered on threat assessment, risk level assessment, and vulnerability rating score for vulnerabilities. Current research on vulnerability assessment is mainly based on quantitative research and model-based research.

The CVSS is the de-facto vulnerability assessment standard, and many studies have been conducted based on it. Spanos et al. [31] has adjusted the weights and proposed a method called the Weighted Impact Vulnerability Scoring System (WIVSS). Based on this, Luo et al. [32] analyzed that CVSS cannot distinguish software vulnerabilities with the same score but different severity levels. A software vulnerability rating method, SVRA, was proposed, which uses a vulnerability database to analyze the frequency of CVSS metrics at different times, and then gives equations for exploitability and impact scores based on these frequencies. Wang et al. [33] considered that CVSS metrics ignore the impact of vulnerabilities on specific networks, i.e., the same vulnerabilities that exist in different network environments are assigned duplicate values, and the attack graph still suffers from scalability and readability issues. To address the above issues, the authors innovatively propose a vulnerability risk assessment method based on heterogeneous information networks. This method takes into account the exploitability of vulnerabilities, the impact of vulnerabilities on network components, and the importance of vulnerable components. The above three articles represent three directions of such research, i.e., focusing on the inadequacy of CVSS evaluation weights, the inadequacy of CVSS evaluation methods, and the inadequacy of CVSS evaluation metrics, but the above methods are mostly based on quantitative formulas to assess vulnerabilities, and the metric values required for the assessment process still need to be extracted manually, which does not significantly improve the speed of hole assessment.

The model-based approach, on the other hand, hopes to take advantage of machine learning algorithms to automatically extract features through algorithms to avoid the bias brought about by manual extraction of features, and the main idea is to characterize and classify vulnerability description statements based on machine learning methods. Khazaei et al. [25], Wang et al. [28], Han et al. [29], and Liu et al. [30] have characterized vulnerability descriptions using traditional word-vector algorithms. However, such methods do not consider contextual information, which may contain rich information that could enhance the final prediction. Based on these studies, Spanos et al. [26], Ali [34], Ameri et al. [35], and Kudjo et al. [36] apply traditional machine learning algorithms and deep learning algorithms to CVSS score prediction. The introduction of deep learning algorithms has improved feature extraction to a certain extent. These methods bring convenience to CVSS assessment work. Shahid and Debar [37], Gong et al. [38], and Costa et al. [39] applied pretrained models and deep learning algorithms to vulnerability metric prediction work. However, these methods do not give the final vulnerability severity level prediction. This can be used as a basis for more in-depth research.

At present, research on vulnerability assessment still has common problems to be studied, such as the unscientific establishment of the metric system, inadequate mining of text dependencies, complex model structure, etc. With the development of deep learning, some problems have been solved, but with the development of the network, some new problems have emerged in front of researchers, such as the problem of vulnerability hazard assessment, because the vulnerability hazard should not only consider the vulnerability itself attributes, such as vulnerability severity, but also need to consider the vulnerability external attributes, such as how much vulnerability affects assets. These issues also need to be studied.

6. Conclusion

This paper proposes an automatic vulnerability severity assessment method based on the distillation model to solve the problem of slow manual vulnerability assessment of CPS. The method first enhances the amount of text information using data augmentation and integration of optimal subsets to solve the data imbalance problem, then characterizes the vulnerability description text using the fine-tuned DistilBERT model to effectively extract the features of the text. Finally, the linear layer is used to classify the characterized vectors. In this paper, multiple groups of ablation experiments are conducted on data of 105,984 vulnerabilities, and the experimental results show that the current method achieves state-of-the-art performance in vulnerability severity assessment, and the method can achieve 97% assessment accuracy. In the future, we will consider CPS vulnerability assessment from different dimensions of vulnerability attributes, such as asset dimension and environment dimension, to build a set of metrics systems in line with vulnerability severity assessment and solve the problem of inaccurate vulnerability hazard assessment.

Data Availability

The dataset used in this paper is available from the corresponding author upon request. Researchers can also download from the NVD official website. Download links are in the 20th reference.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was funded by the National Key Research and Development Program of China under grant number 2021YFB3100500.

References

[1] nvd.nist.gov, "CVSS severity distribution over time," 2022, https://nvd.nist.gov/general/visualizations/vulnerability-visualizations/cvss-severity-distribution-over-time.

[2] nvd.nist.gov, "National vulnerability database," 2022, https://nvd.nist.gov/general/nvd-dashboard.

[3] H. Chen, J. Liu, R. Liu, N. Park, and V. V. E. S. T. Subrahmanian, "A system for vulnerability exploit scoring & timing," in *Proceedings of the International Joint Conferences on Artifical Intelligence*, pp. 6503–6505, Macau, China, Auguest 2019.

[4] J. Ruohonen, "A look at the time delays in CVSS vulnerability scoring," *Applied Computing and Informatics*, vol. 15, pp. 129–135, 2019.

[5] S. Qaiser and R. Ali, "Text mining: use of TF-IDF to examine the relevance of words to documents," *International Journal of Computer Application*, vol. 181, pp. 25–29, 2018.

[6] J. B. Awotunde, C. Chakraborty, and A. E. Adeniyi, "Intrusion detection in industrial internet of things network-based on deep learning model with rule-based feature selection," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 7154587, 17 pages, 2021.

[7] B. Wang, A. Wang, F. Chen, Y. Wang, and C. C. J. Kuo, "Evaluating word embedding models: methods and experimental results," *APSIPA transactions on signal and information processing*, vol. 8, 2019.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: pre-training of deep bidirectional transformers for language understanding," 2018, https://arxiv.org/abs/1810.04805.

[9] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Computation*, vol. 31, pp. 1235–1270, 2019.

[10] H. C. Shin, H. R. Roth, M. Gao et al., "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning," *IEEE Transactions on Medical Imaging*, vol. 35, pp. 1285–1298, 2016.

[11] first org, "Common vulnerability scoring system v3.1: specification document," 2022, https://www.first.org/cvss/v3.1/specification-document.

[12] P. Mell, K. Scarfone, and S. Romanosky, "A complete guide to the common vulnerability scoring system version 2.0," in *Proceedings of the Published by FIRST-Forum of Incident Response and Security Teams*, Seville, Spain, June 2007.

[13] P. Mell and K. Scarfone, "Improving the common vulnerability scoring system," *IET Information Security*, vol. 1, pp. 119–127, 2007.

[14] M. Schiffman, A. Wright, D. Ahmad, and G. Eschelbeck, *The Common Vulnerability Scoring System*, National Infrastructure Advisory Council, Vulnerability Disclosure Working Group, Washington, DC, USA, 2004.

[15] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: a survey," *International Journal of Computer Vision*, vol. 129, pp. 1789–1819, 2021.

[16] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," 2019, https://arxiv.org/abs/1910.01108.

[17] J. Yin, M. Tang, J. Cao, and H. Wang, "Apply transfer learning to cybersecurity: predicting exploitability of vulnerabilities by description," *Knowledge-Based Systems*, vol. 210, Article ID 106529, 2020.

[18] J. Wei and K. E. Zou, "Easy data augmentation techniques for boosting performance on text classification tasks," 2019, https://arxiv.org/abs/1901.11196.

[19] P. Somol, J. Novovicová, and P. Pudil, "Efficient feature subset selection and subset size optimization," *Pattern recognition recent advances*, vol. 1, 2010.

[20] nvd.nist.gov, "NVD data feeds," 2022, https://nvd.nist.gov/vuln/data-feeds.

[21] huggingface, "Distilbert-base-uncased," 2022, https://huggingface.co/distilbert-base-uncased.

[22] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: generalized autoregressive pretraining for language understanding," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[23] Y. Liu, M. Ott, N. Goyal et al., "A robustly optimized bert pretraining approach," 2019, https://arxiv.org/abs/1907.11692.

[24] F. Bieder, R. Sandkühler, and P. C. Cattin, "Comparison of methods generalizing max-and average-pooling," 2021, https://arxiv.org/abs/2103.01746.

[25] A. Khazaei, M. Ghasemzadeh, and V. Derhami, "An automatic method for CVSS score prediction using vulnerabilities description," *Journal of Intelligent and Fuzzy Systems*, vol. 30, pp. 89–96, 2016.

[26] G. Spanos, L. Angelis, and D. Toloudis, "Assessment of vulnerability severity using text mining," in *Proceedings of the 21st Pan-Hellenic Conference on Informatics*, pp. 1–6, Larissa, Greece, September 2017.

[27] S. Nakagawa, T. Nagai, H. Kanehara et al., "Character-level convolutional neural network for predicting severity of software vulnerability from vulnerability description," *IEICE - Transactions on Info and Systems*, vol. 102, pp. 1679–1682, 2019.

[28] P. Wang, Y. Zhou, B. Sun, and W. Zhang, "Intelligent prediction of vulnerability severity level based on text mining and XGBboost," in *Proceedings of the 2019 Eleventh International Conference on Advanced Computational Intelligence (ICACI)*, pp. 72–77, Guilin, China, June 2019.

[29] Z. Han, X. Li, Z. Xing, H. Liu, and Z. Feng, "Learning to predict severity of software vulnerability using only vulnerability description," in *Proceedings of the 2017 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pp. 125–136, Shanghai, China, September 2017.

[30] K. Liu, Y. Zhou, Q. Wang, and X. Zhu, "Vulnerability severity prediction with deep neural network," in *Proceedings of the 2019 5th International Conference on Big Data and Information Analytics (BigDIA)*, pp. 114–119, Kunming, China, July 2019.

[31] G. Spanos, A. Sioziou, and L. Angelis, "WIVSS: a new methodology for scoring information systems vulnerabilities," in *Proceedings of the Proceedings of the 17th Panhellenic Conference on Informatics*, pp. 83–90, New York, NY, USA, September 2013.

[32] J. Luo, K. Lo, and H. Qu, "A software vulnerability rating approach based on the vulnerability database," *Journal of Applied Mathematics*, vol. 2014, Article ID 932397, 9 pages, 2014.

[33] W. Wang, F. Shi, M. Zhang, C. Xu, and J. Zheng, "A vulnerability risk assessment method based on heterogeneous

information network," *IEEE Access*, vol. 8, pp. 148315–148330, 2020.

[34] M. Ali, "Character level convolutional neural network for Arabic dialect identification," in *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pp. 122–127, Santa Fe, NM, USA, August 2018.

[35] K. Ameri, M. Hempel, H. Sharif, J. Lopez, and K. C. Perumalla, "Cybersecurity claim classification by fine-tuning the BERT language model," *Journal of Cybersecurity and Privacy*, vol. 1, pp. 615–637, 2021.

[36] P. K. Kudjo, J. Chen, S. Mensah, R. Amankwah, and C. Kudjo, "The effect of Bellwether analysis on software vulnerability severity prediction models," *Software Quality Journal*, vol. 28, pp. 1413–1446, 2020.

[37] M. R. Shahid and H. Debar, "Cvss-Bert: Explainable natural language processing to determine the severity of a computer security vulnerability from its description," in *Proceedings of the 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 1600–1607, Pasadena, CA, USA, December 2021.

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

[39] J. C. Costa, T. Roxo, J. B. Sequeiros, H. Proença, and P. R. Inácio, *Predicting CVSS Metric via Description Interpretation*, IEEE Access, Piscataway, NJ, USA, 2022.