Using Federated Learning to Predict Vulnerability Exploitability

Wangyuan Jing*, Lingbo Wei†, Hao Jin‡, Chi Zhang*, Wenxiang Dong†, Yangyang Li‡

*School of Cyberspace Science and Technology, University of Science and Technology of China, Hefei, Anhui, China

†Institute of Dataspace, Hefei Comprehensive National Science Center, Hefei, Anhui, China

‡National Engineering Research Center for Risk Perception and Prevention, CAEIT, Beijing, China

jwy123@mail.ustc.edu.cn, {lingbowei,chizhang}@ustc.edu.cn, {jinhao18,liyangyang}@cetc.com.cn, dyjavin@iat.ustc.edu.cn

2022 5th International Conference on Hot Information-Centric Networking (HotICN) | 979-8-3503-3520-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/HOTICN57539.2022.10036231

***Abstract*—The dramatic increase in the number of vulnerabilities and threats prompts the development of vulnerability exploitability prediction research. However, the existing vulnerability exploitability prediction research directly aggregates all vulnerability data without considering the security of vulnerability information, which leads to some problems such as data leakage and data island. In this paper, we propose a method for vulnerability exploitability prediction based on federated learning, which aims to achieve vulnerability exploitability prediction while protecting the security of vendor vulnerability data. Specifically, we first construct a vulnerability exploitability prediction model in a federated learning environment and classify the collected vulnerability data by vendors. Second, we evaluate multiple vulnerability exploitability prediction models and improve existing models. Finally, extensive experiments demonstrate that our proposed model achieves good results in the federated learning environment.**

***Index Terms*—Federated Learning, Vulnerability Exploitability Prediction, Natural Language Processing, Deep Learning**

I. INTRODUCTION

In recent years, the number of discovered and publicly disclosed vulnerabilities has grown very rapidly, and the threats posed by vulnerabilities are also increasing day by day, which brings enormous challenges to information security and network platform maintenance. However, only a small fraction of all disclosed vulnerabilities is widely exploited, and the value of these vulnerabilities varies. In order to find the most likely exploitable vulnerabilities and patch them in a limited time, the research of vulnerability exploitability prediction is carried out. Over the years, there have been studies on the problem of vulnerability exploitability prediction, which have achieved certain results. For example, the Common Vulnerability Scoring System (CVSS) score contains the exploitability score of the vulnerability, which is used to measure the probability that the vulnerability can be exploited. However, because the CVSS score measurement standard is too simple, some researchers further build a vulnerability exploitability prediction model [1]–[8]. They collect open-source vulnerability information and aggregate this vulnerability information. By using more effective natural language processing algorithms and classification prediction algorithms to further improve the model, good results have been achieved.

However, for network platforms, vulnerability information is extremely sensitive. Once leaked, it will bring very serious disasters. The existing vulnerability exploitability prediction methods simply aggregate the vulnerability information of all platforms and train a machine learning model, without considering the security and sensitivity of vulnerability data, which result in data islands and affect vulnerability exploitability prediction in actual use. In recent years, federated learning has emerged, aiming to address data locality and privacy issues across various devices and scenarios [9]–[14]. In the face of massive private data, different from traditional machine learning methods, federated learning does not need to centralize all private data to train a model. On the contrary, federated learning can ensure that private data is local to the data holder while jointly training a model, thereby protecting the privacy and security of the data.

Therefore, in view of the security problem of data leakage that may occur in the process of vulnerability exploitability prediction, we propose a method of vulnerability exploitability prediction based on federated learning, which aims to achieve vulnerability exploitability prediction while protecting the security of relevant vendor vulnerability data. We collect data from multiple open-source datasets, classify vulnerability information according to different vendors, and construct a federated learning environment. We propose a federated learning and deep learning approach for vulnerability exploitability prediction. Based on the existing research work, we compare the performance of various vulnerability exploitability prediction models with and without federated learning, and select the model with the best performance as the prediction model.

Our contributions are as follows:

- First, we propose a vulnerability exploitability prediction method based on federated learning. This method considers the security problem in model training for the first time, which protects the security of vulnerable data, and solves the problem of data island.
- Second, we evaluate multiple exploitability prediction models and compare their performance.
- Finally, we improve the existing vulnerability exploitability model and propose a new FastText+DNN model, which is applied to the vulnerability exploitability pre-

This work was supported by the Natural Science Foundation of China (NSFC) under Grants 61871362, 62072426, U19B2023, and U21B2026; and Innovation and Application Demonstration of Social Management Information Technology of Hainan Free Trade Port under Grant ZDKJ2019008.

diction in the federated learning environment, improving the prediction performance of the model.

The rest of this paper is structured as follows. Section II presents the background and related work. Section III presents our constructed federated learning-based vulnerability exploitability prediction model. Section IV presents our experimental data and experimental results. Section V concludes.

II. Background and Related Work

In this section, we mainly introduce the evaluation criteria of vulnerability exploitability (exploitability score in CVSS) and some current research methods of vulnerability exploitability prediction. At the same time, we also briefly describe the concept of federated learning.

A. *CVSS*

At present, the most official evaluation of vulnerability exploitability is the CVSS score, which contains the exploitability score of the vulnerability. CVSS constructs a fixed calculation formula based on parameters such as AccessComplexity, Authentication and AccessVector of the vulnerability, and then performs weighted calculation to score the exploitability of the vulnerability [15]. However, the CVSS score considers too few vulnerability features, and the constructed formula is too simple, resulting in an inaccurate evaluation.

B. *Vulnerability Exploitability Prediction Method*

At present, some researches on vulnerability exploitability prediction have been carried out. The main idea of these researches is to collect open source data sets of vulnerability information and exploit information, extract the characteristics of vulnerabilities, and then use machine learning algorithms to predict vulnerability exploitability. For example, Han et al. [1] used convolutional neural network (CNN) to extract vulnerability text description features, and then used SVM to classify and predict. Due to too few vulnerability features considered, the accuracy rate is only 81.6%. Sabottke et al. [8] used a linear SVM classifier to predict whether Twitter related to CVE vulnerabilities on Twitter would lead to exploits, but it was limited to Twitter, and the field of consideration was too narrow. Jay Jacobs [6] et al. trained more than 20 parameters in the CVSS score of vulnerabilities through XGBoost decision tree. Such methods only consider the parameters in the CVSS score, and consider too few features of the vulnerability, which is not conducive to a comprehensive study of vulnerability exploitability. Huang et al. [2] considered the text features of vulnerabilities and combined other features of vulnerabilities such as CVSS and OVAL, and achieved vulnerability exploitability prediction through the FastText + LightGBM algorithm, with an accuracy rate of 91%; By comparing 48 supervised and unsupervised algorithms, Xiang Chen et al. [5] demonstrated that supervised machine learning algorithms can achieve better results in practical scenarios of vulnerability prediction.

However, these studies do not take into account the security of vulnerability information and the data island problem that may be caused by the privacy of vulnerability information. Moreover, the existing models also have problems such as too few extracted vulnerability features and insufficient optimization of the model, resulting in an unsatisfactory performance of the prediction model.

C. *Federated Learning*

Federated Learning was first proposed by Google, and the goal is to build a joint machine learning model based on data distributed on multiple devices. Federated learning is a distributed training method by introducing a variety of privacy protection technologies and using data scattered in multiple participants to collaboratively build a global machine learning model [9]–[14]. When training the model, the model ensures that the data of each holder participating in federated learning will not leave itself. It realizes the joint modeling of multiple participants while ensuring data security and improves the performance of the participants training the model independently, addressing data breaches and data silos

III. Model Construction

In this section, we introduce the constructed federated learning model and vulnerability exploitability prediction model.

A. *Federated Learning Model Construction*

We adopt the traditional horizontal federated learning architecture to build our model. Figure 1 briefly describes the federated learning model we built.

1) Federated Learning Server: The federated learning server is mainly responsible for uniting all participants to jointly train a global model. The federated learning server receives the gradient information of the participants, updates the gradient information of the model through secure aggregation, and then returns the gradient information to each participant.

2) Federated Learning Participants: The federated learning participants used in this paper are network companies, which contain their own vulnerability information. The federated learning participants train their own models locally, send the trained gradient information to the federated learning server, and continuously update their models according to the gradient information returned by the federated learning server.

B. *Vulnerability Exploitability Prediction Model Construction*

According to the existing vulnerability exploitability prediction research [2], we construct the model as shown in Figure 2.

We extract the textual description of the vulnerability and other features of the vulnerability separately. For the text description of the vulnerability, we use related natural language processing algorithms to extract text features from the text description of the vulnerability. According to the existing research, we use FastText to process the text description. Since there is no research on using BERT to extract text features, we also use BERT to extract text features. We evaluate the performance of the two algorithms in our experiments. For some other features of vulnerabilities, we use encoding and

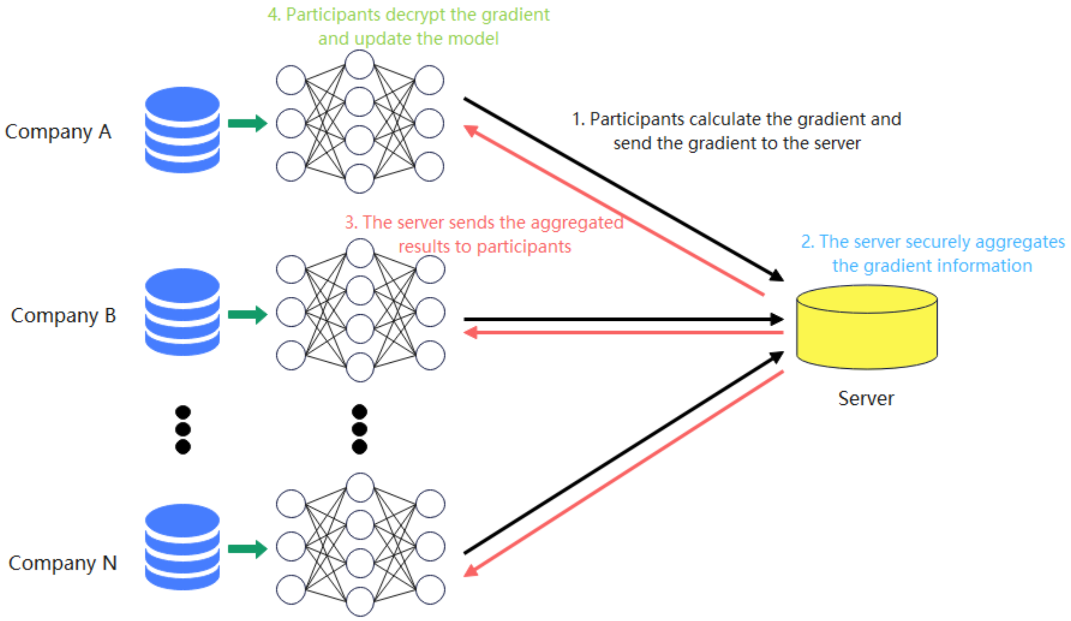

4. Participants decrypt the gradient and update the model
Company A
1. Participants calculate the gradient and send the gradient to the server
3. The server sends the aggregated results to participants
2. The server securely aggregates the gradient information
Company B
Server
Company N

Fig. 1. Federated Learning Model

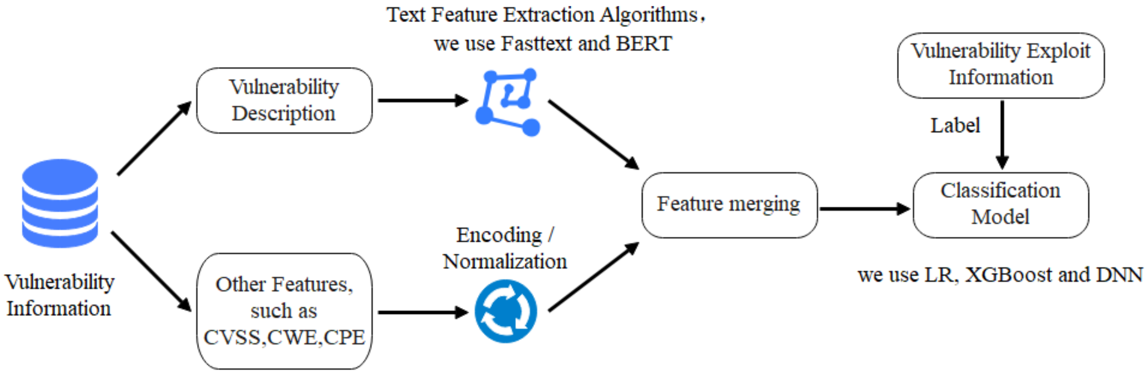

Text Feature Extraction Algorithms, we use Fasttext and BERT
Vulnerability Description
Vulnerability Exploit Information
Label
Feature merging
Classification Model
Encoding / Normalization
Vulnerability Information
Other Features, such as CVSS,CWE,CPE
we use LR, XGBoost and DNN

Fig. 2. Vulnerability Exploitability Prediction Model

normalization methods. deal with these features. After completion, we merge these features, classify and predict by means of machine learning, we consider three popular algorithms: logistic regression (LR), decision tree (XGBoost) and deep learning (DNN).

We extract some other features of vulnerabilities, such as CVSS score, CPE information [16], CWE information [17], and OVAL [18], etc.

Table 1 shows our approach to vulnerability features. For vulnerability text description, we use FastText and BERT methods to extract text features according to previous research. For some other features of the vulnerability, such as AccessVectors and AccessComplexity, we use one-hot encoding to process them. For digital features such as Basic Scores, we use a normalized method to scale these numbers between 0 and 1 to facilitate subsequent model training. For OVAL vectors that are originally 0 or 1, we directly use them as Binary vector for input.

IV. EXPERIMENT

A. *Experimental Data*

We collected vulnerability data on multiple open source platforms. In general, the sources of datasets can be mainly divided into two categories: Vulnerability Information Disclosure Platform and Vulnerability Exploitation Information Platform. All of the data we use is a combination of vulnerability information and exploit tags using the Common Vulnerability and Exposure Identifier (CVE-ID) as a unified locator.

1) Vulnerability Information Disclosure Platform: We mainly select the US National Information Security Vulnerability Database (NVD) as the vulnerability information disclosure platform, which is a repository of standards-based vulnerability management data. The vulnerabilities in the NVD database are indexed by CVE-ID, which provides a download channel for vulnerability information. For all vulnerability information, NVD combines the vulnerability information of each year in json format. Each entry contains a lot of

TABLE I
METHOD OF HANDLING VULNERABILITY CHARACTERISTICS

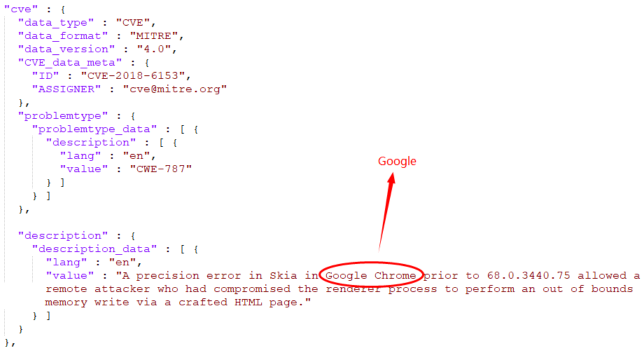
Feature Name	Method	Feature Name	Method
Vulnerability Description	BERT/FastText	Availability Impact	One-hot Encoding
CWE-ID	One-hot Encoding	BaseScore	Scaling to N(0,1)
AccessVector	One-hot Encoding	Severity	Scaling to N(0,1)
AccessComplexity	One-hot Encoding	ExploitabilityScore	Scaling to N(0,1)
Authentication	One-hot Encoding	ImpactScore	Scaling to N(0,1)
Confidentiality Impact	One-hot Encoding	Product Count	Scaling to N(0,1)
Integrity Impact	One-hot Encoding	OVAL&CPE	Binary Vector

information such as a descriptive text summary of the vulnerability, CVSS scores and associated metrics, information about affected products and vendors, vulnerability categories based on the Common Weakness Enumeration (CWE) system, and reference URLs.

2) Vulnerability Exploitation Information Platform: We mainly select: Securityfocus, GreyNoise, Exploit-DB and some exploit web pages in Github as exploit information platforms. There is exploit information of vulnerabilities in these platforms, which is the basis for this experiment to determine whether the vulnerability is exploited, that is, the label of the sample when performing classification prediction. Exploited vulnerabilities have a label of 1, and unexploited vulnerabilities have a label of 0. Table 2 presents the data we collected.

TABLE II
VULNERABILITY DATA STATISTICS

Data Set	Sum of Samples
NVD	147565
Greynoise	75
Securityfocus	33820
Github	35
Exploit-DB	8701
CVE-ID(Has label)	16793



```
"cve" : {
  "data_type" : "CVE",
  "data_format" : "MITRE",
  "data_version" : "4.0",
  "CVE_data_meta" : {
    "ID" : "CVE-2018-6153",
    "ASSIGNER" : "cve@mitre.org"
  },
  "problemtype" : {
    "problemtype_data" : [ {
      "description" : [ {
        "lang" : "en",
        "value" : "CWE-787"
      } ]
    } ]
  },

  "description" : {
    "description_data" : [ {
      "lang" : "en",
      "value" : "A precision error in Skia in Google Chrome prior to 68.0.3440.75 allowed a
      remote attacker who had compromised the renderer process to perform an out of bounds
      memory write via a crafted HTML page."
    } ]
  }
},
```

Fig. 3. CVE-2018-6153 Vulnerability Description

We classify the vulnerabilities according to different vendors according to the products in which the vulnerabilities exist in the vulnerability text description. For example, the CVE-2018-6135 vulnerability shown in Figure 3, its text description describes that this vulnerability exists in Google Chrome, so we believe that this vulnerability belongs to Google.

Through the above methods, we classify all the vulnerabilities with exploit information, and select the three vendors with the most exploit information as the experimental objects of this paper. They are 1,597 pieces of data from Apple, 1,208 pieces of data from Google, and 1,034 pieces of data from Adobe. This paper regards these three companies as three participants, and performs federated learning on the vulnerability data of these three participants.

TABLE III
VENDOR VULNERABILITY STATISTICS

Vendor	Sum of Samples
Apple	1597
Adobe	1208
Google	1034

B. Evaluation Indicators

We evaluate the performance of our model in four aspects: accuracy, precision, recall and F1 score. The accuracy measures the accuracy of the test data set, which is the proportion of all the correctly predicted samples in the test set among the total samples. Precision reflects the proportion of vulnerabilities judged to be exploitable in the true exploitable sample. The recall rate reflects the proportion of vulnerabilities judged to be exploitable to the total exploitable vulnerabilities. F1-score is the balanced F-score, which is the harmonic mean of precision and recall. The F1 score is used to judge, which can not only ensure the accuracy of whether the detected vulnerabilities are exploitable, but also ensure that the exploitable vulnerabilities can be detected to the maximum extent.

C. Experimental Results

We have conducted multiple comparative experiments, considered text feature extraction algorithms and prediction algorithms in various vulnerability exploitability prediction models, and compared the performance of models with and without federated learning. Table 4 presents our experimental results.

First, we make a horizontal comparison to compare the effects of different vulnerability exploitability prediction models in the environments with and without federated learning. In

TABLE IV
Three Vendors Evaluation

Method		Accuracy	Precision	Recall	F1-Score
Federal Learning	Prediction Model				
YES	FastText+LR	0.782	0.598	0.748	0.655
YES	FastText+XGBoost	0.851	0.731	0.589	0.652
YES	**FastText+DNN**	**0.899**	**0.673**	**0.843**	**0.731**
NO	FastText+LR	0.884	0.841	0.700	0.761
NO	FastText+XGBoost	0.910	0.791	0.811	0.808
NO	**FastText+DNN**	**0.911**	**0.812**	**0.834**	**0.846**
YES	BERT+LR	0.777	0.119	0.867	0.227
YES	BERT+XGBoost	0.753	0.563	0.712	0.629
YES	BERT+DNN	0.806	0.567	0.771	0.591
NO	BERT+LR	0.804	0.666	0.348	0.456
NO	BERT+XGBoost	0.777	0.603	0.784	0.681
NO	BERT+DNN	0.829	0.432	0.757	0.558

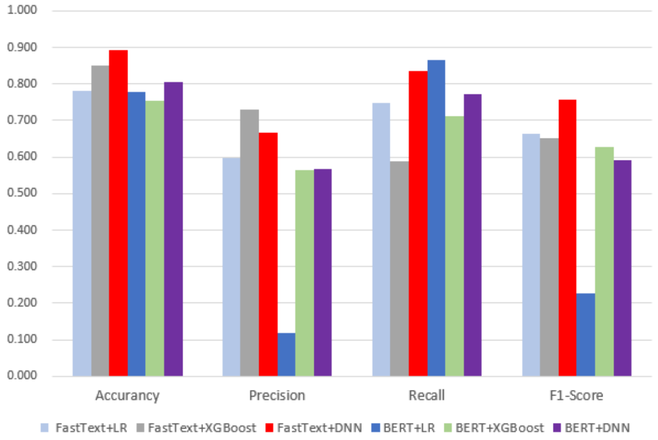


Fig. 4. Use Federated Learning Model Evaluation

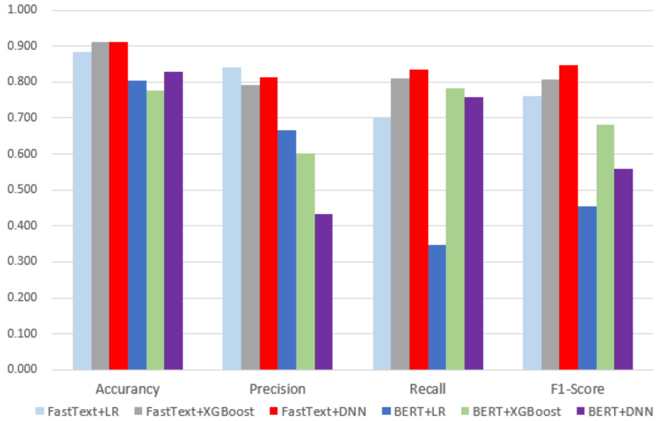


Fig. 5. Not Use Federated Learning Model Evaluation

Figure4 and Figure5, we specifically show the performance evaluation of each model using federated learning and not using federated learning. In general, FastText + DNN is more suitable for the construction of vulnerability exploitability prediction model. In the case of using federated learning, the accuracy rate of FastText + DNN model is 89.90%, which is 4.78% higher than the accuracy rate of FastText + XGBoost (85.12%). Although the accuracy is lower than that of FastText + XGBoost, the recall rate and F1 score are higher than that of FastText + XGBoost, and the comprehensive performance of FastText + DNN is better. For FastText + LR, the performance of FastText + DNN is better than that of FastText + LR, Therefore, this paper believes that FastText + DNN is better when federated learning is used. Without federal learning, the accuracy rate of FastText + DNN model is 91.14%, which is 0.14% higher than that of FastText + XGBoost (91.00%), and about 3% higher than that of FastText + LR (88.36%). Although the accuracy is lower than that of FastText + LR, the recall rate and F1 score are higher than that of FastText + LR, and the comprehensive performance of FastText + DNN is better. For FastText + LR, the performance of FastText + DNN is better than that of FastText + XGBoost. So we believe that FastText + DNN works better without federated learning. Therefore, we use the FastText + DNN model to build the vulnerability exploitability prediction model of the participants.

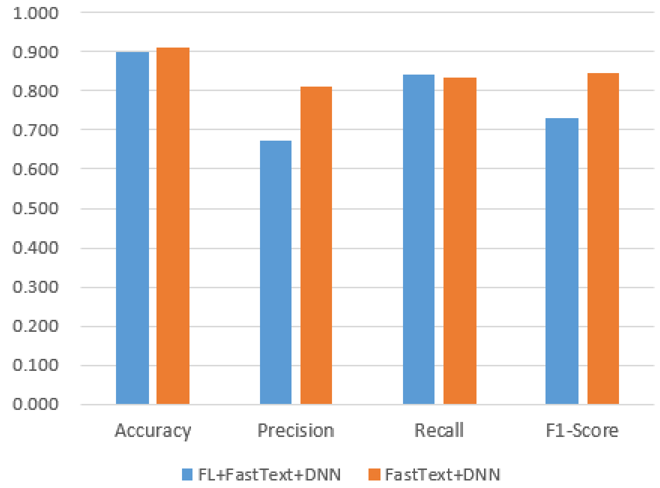


Fig. 6. FastText+DNN Evaluation

Then, we specifically analyze the effects of the Fast-

Text+DNN model proposed in this paper with and without federated learning. Figure 6 shows our model evaluation of FastText+DNN. We find that in the case of federated learning, the accuracy of FastText+DNN model is about 1.2% lower, the accuracy decreases by 0.14, the recall rate is similar, and the effect of F1-Score is about 0.1 lower. Overall, using federated learning has little impact on the FastText+DNN model.

Because of the use of federated learning, the three participants (Adobe, Apple, and Google) need to calculate their own model gradients, and then perform safe aggregation through the federated learning mechanism. The performance of aggregation will definitely be poorer than directly combining the data of the three participants. By continuously optimizing the parameters of federated learning, we can improve the effect of the model as much as possible, so that the model that uses federated learning to predict vulnerability exploitability can improve the effect and the accuracy of the prediction as much as possible. In general, the use of federated learning has little effect on the FastText+DNN model. Therefore, we believe that the Federated Learning+FastText+DNN model can be used for exploitability prediction. We believe that the use of federated learning+FastText+DNN to achieve vulnerability exploitability prediction has less impact than not using federated learning. Using federated learning can protect the privacy and security of data, solve the current problem of data island, and is more suitable for vulnerability exploitability prediction.

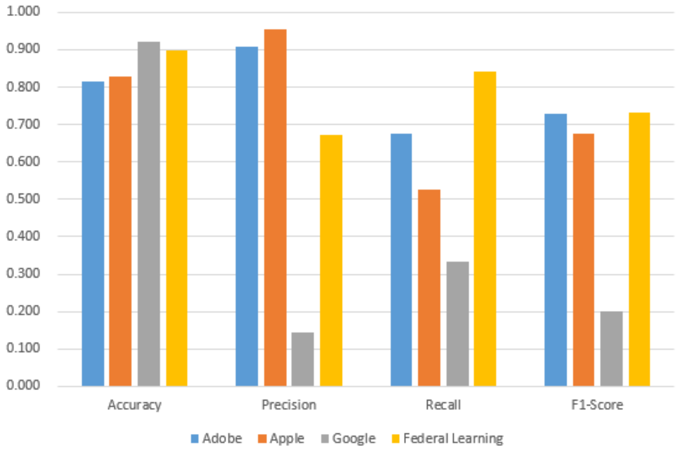


Fig. 7. Three Vendors Evaluation

TABLE V
THREE VENDORS EVALUATION

	Accuracy	Precision	Recall	F1-Score
Adobe	0.814	0.907	0.677	0.729
Apple	0.829	0.955	0.525	0.677
Google	0.921	0.143	0.333	0.200
Federal Learning	0.899	0.673	0.843	0.731

At the same time, Table 5 and Figure 7 show the prediction performance of the three vendors that we compared without using federated learning through the vulnerability exploitability prediction model and the three vendors using federated learning. Without federated learning, Adobe's prediction accuracy rate is 81.4%, precision is 0.907, recall rate is 0.677, and F1-score is 0.729. Compared with the method using federated learning, the accuracy rate is about 8.5% lower, although the accuracy It is higher than using federated learning, but both recall and F1-score are lower than using federated learning, and the overall performance of the model is worse than using federated learning. Similar for Apple, the accuracy rate is 7% lower than using federated learning, although the accuracy is higher, but the overall performance of the model is lower. For Google, although the accuracy is more than 2% higher than using federated learning, the precision, recall and F1-Score of the model are much lower, and the overall performance of the model is poor.

We find that, overall, using the federated learning model performs better than the three vendors without federated learning. For Google, the accuracy rate without federated learning is higher than that with federated learning, but the precision, recall and F1-Score are far lower than those with federated learning. Therefore, for Google, federated learning is more suitable. For Adobe and Apple, the accuracy without federated learning is much lower than the accuracy, recall and F1-Score of federated learning, and the accuracy is higher. Overall, the effect is not as good as federated learning. Therefore, for these two vendors, this experiment proves that federated learning is also more suitable.

V. CONCLUSION

In this paper, we proposed a federated learning-based vulnerability exploitability prediction method for the first time, which realizes vulnerability exploitability prediction while protecting the security of vulnerability data. Specifically, we collected multiple open-source datasets and classified vulnerability data by vendor to simulate federated learning training in a real environment. We trained the model through federated learning, which protects the security of vulnerability data. In addition, we evaluated a variety of existing vulnerability exploitability prediction models, and proposed FastText+DNN to build the model. Experiments show that on our dataset, the accuracy of the Federated Learning+FastText+DNN model reaches 89.90%, the F1-Score reached 0.731. Compared with the existing models, our proposed FastText+DNN model improves the prediction effect and is more suitable for the federated learning environment.

Our future work is to further expand the vulnerability dataset, and at the same time, we will also further improve the efficiency of the vulnerability exploitability prediction model based on federated learning.

REFERENCES

[1] Z. Han, X. Li, Z. Xing, H. Liu, and Z. Feng, "Learning to predict severity of software vulnerability using only vulnerability description," in *2017 IEEE International conference on software maintenance and evolution (ICSME)*. IEEE, 2017, pp. 125–136.

[2] Y. Fang, Y. Liu, C. Huang, and L. Liu, "Fastembed: Predicting vulnerability exploitation possibility based on ensemble machine learning algorithm," *Plos one*, vol. 15, no. 2, p. e0228439, 2020.

[3] M. S. Hoque, N. Jamil, N. Amin, and K.-Y. Lam, "An improved vulnerability exploitation prediction model with novel cost function and custom trained word vector embedding," *Sensors*, vol. 21, no. 12, p. 4220, 2021.

[4] J. Lyu, Y. Bai, Z. Xing, X. Li, and W. Ge, "A character-level convolutional neural network for predicting exploitability of vulnerability," in *2021 International Symposium on Theoretical Aspects of Software Engineering (TASE)*. IEEE, 2021, pp. 119–126.

[5] O. Suciu, C. Nelson, Z. Lyu, T. Bao, and T. Dumitras, "Expected exploitability: Predicting the development of functional vulnerability exploits," *arXiv preprint arXiv:2102.07869*, 2021.

[6] J. Jacobs, S. Romanosky, I. Adjerid, and W. Baker, "Improving vulnerability remediation through better exploit prediction," *Journal of Cybersecurity*, vol. 6, no. 1, p. tyaa015, 2020.

[7] B. L. Bullough, A. K. Yanchenko, C. L. Smith, and J. R. Zipkin, "Predicting exploitation of disclosed software vulnerabilities using open-source data," in *Proceedings of the 3rd ACM on International Workshop on Security and Privacy Analytics*, 2017, pp. 45–53.

[8] C. Sabottke, O. Suciu, and T. Dumitraș, "Vulnerability disclosure in the age of social media: Exploiting twitter for predicting real-world exploits," in *24th USENIX Security Symposium (USENIX Security 15)*, 2015, pp. 1041–1056.

[9] Y. Liu, T. Fan, T. Chen, Q. Xu, and Q. Yang, "FATE: An industrial grade platform for collaborative learning with data protection." *J. Mach. Learn. Res.*, vol. 22, no. 226, pp. 1–6, 2021.

[10] L. Jian, S. Yunfeng, L. Yi, and W. Jun, "Overview of federal learning and its application in telecom industry," *Information and Communications Technology and Policy*, vol. 46, no. 9, p. 35, 2020.

[11] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Future Generation Computer Systems*, vol. 115, pp. 619–640, 2021.

[12] K. M. Ahmed, A. Imteaj, and M. H. Amini, "Federated deep learning for heterogeneous edge computing," in *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2021, pp. 1146–1152.

[13] Q. Tang, R. Xie, F. R. Yu, T. Chen, R. Zhang, T. Huang, and Y. Liu, "Collective deep reinforcement learning for intelligence sharing in the internet of intelligence-empowered edge computing," *IEEE Transactions on Mobile Computing*, pp. 1–16, 2022.

[14] B. Ghimire and D. B. Rawat, "Recent advances on federated learning for cybersecurity and cybersecurity for federated learning for internet of things," *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 8229–8249, 2022.

[15] C. Elbaz, L. Rilling, and C. Morin, "Fighting N-day vulnerabilities with automated cvss vector prediction at disclosure," in *Proceedings of the 15th International Conference on Availability, Reliability and Security*, 2020, pp. 1–10.

[16] L. A. B. Sanguino and R. Uetz, "Software vulnerability analysis using CPE and CVE," *arXiv preprint arXiv:1705.05347*, 2017.

[17] B. Martin, "Common vulnerabilities enumeration (cve), common weakness enumeration (cwe), and common quality enumeration (cqe) attempting to systematically catalog the safety and security challenges for modern, networked, software-intensive systems," *ACM SIGAda Ada Letters*, vol. 38, no. 2, pp. 9–42, 2019.

[18] Y. Kwon, H. J. Lee, and G. Lee, "A vulnerability assessment tool based on oval in linux system," in *IFIP International Conference on Network and Parallel Computing*. Springer, 2004, pp. 653–660.