PREDICT SOFTWARE VULNERABILITY

Dr.S.Tamil selvan AP/CSE, R.Divya, S.Jayapriya, N.K.Nivitha

Department of Computer Science and Engineering,
Erode Sengunthar Engineering College,Erode-638057, Tamil Nadu.

Abstract

Changes made in para- Security concerns are primarily brought about by software defects. If a hostile attack exploits a weakness, the system's safety will be gravely compromised. Even catastrophic losses could result. Automatic classification techniques are thus advantageous for managing software vulnerabilities effectively and enhancing system security. It will lessen the possibility of system compromise and attack. Model for automatically classifying vulnerabilities (IGTF-DNN) In this study, a novel model based on term frequency-deep neural networks dubbed Information Gain has been put forth. Deep neural networks (DNN) and information gain (IG), based on frequency- inverse document frequency, are used to build the model (TF-IDF). The frequency and weight of phrases extracted from vulnerability descriptions using TF-IDF are determined using Information Gain, and the optimal set of feature words is assembled using Choose features. The automatic vulnerability classifier is then built utilising a deep neural network model to categorise vulnerabilities effectively. The effectiveness ofthe suggested model has been evaluated using theUS National Vulnerability Database. The TFI- DNN model performs better on assessment indices like precision and recall measures when compared to KNN. It may not only improve the effectiveness of the vulnerability recovery and management, but also reduce the danger of systems being attacked and collapsing, which is critical for systems' security capabilities, assuming the vulnerability can be classified and handled with effectiveness. A growing number of studies on vulnerability classification are being undertaken by qualified security researchers as software security vulnerabilities play is a significant part in cyber-security assaults.

Keywords: Software Engineering, Software Vulnerability, Deep Neural Network, Inverse Document Frequency, Information Gain.

I. INTRODUCTION

Information security issues are becoming more and more significant as sectors' levels of digitalization advance. Software/hardware flaws and issues with a system being illegally exploitable produced by those who are not allowed are vulnerabilities. The security of the information system will be seriously jeopardized the moment a suspicious attack takes advantage of a vulnerability. It might even have incalculable effects. 2017 saw the exploitation of Windows system vulnerabilities Hackers will make Bitcoin ransomware available to 100,000 organizations worldwide. Microsoft made 372 Office vulnerability patches available overall that year. Hackers make use of office vulnerabilities to conduct Advanced Persistent Threat (APT) attacks, spread ransomware, botnets and so on. Nowadays, the count and variety of vulnerabilities are gradually increasing, so that the analysis and management of software vulnerabilities are becoming more important. It may not only improve the effectiveness of vulnerability recovery and management, but also reduce the danger of systems being attacked and collapsing, if the vulnerability can be effectively identified and treated, this is crucial for the security performance of systems. While software security flaws are a significant contributor to cyber-security attacks, relevant security experts are increasingly doing studies on vulnerability classification. The earlier RISOS [1]vulnerability classification method, which is directed at the computer's operating system, primarily divides OS vulnerabilities into seven categories from the attack perspective and explains how to exploit vulnerabilities rather thansetting up scenarios that will lead to vulnerabilities. The PA vulnerability classification approach in [2] classifies vulnerabilities in addition to researching operating system flaws incorporated into the application previously. A vulnerability classification system with often categories in accordance with the diverse analytical demands of vulnerability was introduced by Andy Gray's vulnerability classification approach [3]. As vulnerability complexity increases, the limitations of traditional artificial vulnerability classification algorithms become increasingly obvious. Scientists provide the automatic Therefore, the automatic classification of vulnerabilities is of more interest to researchers. Several machine learning algorithms have recently been disclosed [4] in the context of text classification. Text classification is the classification of data according to the description of a vulnerability. The SVM based upon topic model makes full use of number of distributed vulnerabilities for classification. The experiment results indicated that SVM has attained good results in vulnerability grouping. Wijayasekara et al. [6] evaluated Naïve Bayes classificationmethod by using textual information from error descriptions. That analysis illustrated the feasibility of Naïve Bayes classifier for classifying textual information based on vulnerability

description. Sarang et al. [7] introduced a classification method to classify CVE entries which could not give Using the Naive Bayes classifier, sufficient information is put into vulnerability groups. For the purpose of vulnerability categorization, Gawronetal [8] compared the Naive Bayes approach and the simplified artificial neural network (ANN) algorithm using the same data set. The experimental findings proved that the artificial neural network method classified vulnerabilities more accurately than the Naive Bayes approach. In order to efficiently categorise vulnerabilities, the automatic vulnerability classifier is then created using a deep neural network model.

II. LITERATURE REVIEW

Even though these machine learning algorithms have achieved hopeful results in many fields, because of the huge amount of vulnerability data with short description, generated word vector space handed the characteristics of high dimension and sparse. These machine learning algorithms are not very much effective dealt with high as well as sparse problems. Meanwhile, theypay no attention to particular vulnerability information and so the classification accuracy is not elevated. In recent years, deep learning found application in variety of fields and has achieved triumph, such as the speech and image recognition field [9], [10], and there, the error rate in speech recognition is lowered by 20 – 30 percent [11]. The error rate in ImageNet evaluation task is lowered by 26 – 15 percent [12]. Deep learning also has a important impact in the natural language filed [13], [14].Jo et al. [15]studied the classification problems in the natural language field, and they applied convolutional neural networks (CNN) andrecurrent neural networks (RNN) for large-scale text classification and achieved success.Aziguli et al. [16] introduced a new text classifier using DNN model to progress the computational performance of processing large text data along

with mixed outliers. Therefore, to better deal with the high and sparse word vector space and thereby take benefits of automatic feature extraction by deep learning, this paper introduces an automatic vulnerability classification model IGTF-DNN based on term frequency-reverse information gain (IG), document frequency (TF- IDF), and deep neural network (DNN).

The model first employs the IGTF technique to capture the description text feature and lower the resultant high-dimensional word vector space dimension. Next a deep learning-based DNN neural network model is built. Using vulnerability information from the National Vulnerability Database, the model was trainedand tested (NVD). The test results demonstrated how successfully the automatic vulnerability classification approach in this study enhances vulnerability classification performance.

The remaining of this paper is organized as follows. This section discusses below the definition of relevant algorithms. Section 3, described the implementation details of the model. Section 4 discussed the experiment dataset and results with comparative analysis. Section 5 outlines experimental procedure and Section 6 outlines the conclusions and possible future research.

The automatic classification model of vulnerability (IGTF-DNN) based on TF-IDF is constructed in this paper. The relevant definitionsare as follows.

A. TF-IDF :

(Term Frequency/Inverse Document Frequency) is a common weighted technology which is found out based on statistical methods [17]. For example, consider there are a set of documents and each document contains a number of terms/words. It is defined that the word I's importance in document j as follows.

$$Tf_{ij} = n_{i,j} / \sum_k n_{k,j} \quad (1)$$

where both I and j are positive integers, $n_{i,j}$denotes the term I's frequency in document j.

The IDF formula is as follows,

$$idf_i = \log (|F| / |\{ j : t_i \square d_j \}|) \quad (2)$$

where |F| is the total number of documents in corpus, f_j is the jth document, and $|\{j:t_i \in f_j|$ is the number of documents containing the term ti.

The TF-IDF formula is as follows.$TF{-}IDF = tf_{ij} * idf_i$ (3)

TF-IDF is used to measure the terms' importance word to a document in the document set o rin a corpus. The terms' importance increases proportionally with number of times it appears in the document, but also decreases inversely with frequency it appears in corpus.

B.INFORMATION GAIN (IG):

Refers to that, if a feature X in class Y is known already, information uncertainty of class Y decrease, and so reduced uncertainty degree will reflect importance of feature X to class Y. Set the training data set to D, |D| shows the count of samples in D. Suppose there are K classes Ck, k

= 1, 2, … , K $|C_K|$ is the count of samples fit in to class Ck. $\sum_{Ck=1}^{K} |C_K| = |D|$. If feature A has n different values {a1,

$a_2, \ldots, a_n\}$, D is segmented into 'n' sub groups according to feature A values,represented as D = (D1, D2, … , Dn), where $|D_i|$ is the samples count in Di, $\sum_{i=1}^{n} |D_i| = |D|$. The samples set fit into class C_k in D_i is D_{ik}, $D_{ik} = D_i \cap D_k$, $|D_{ik}|$ is the samples count of D_{ik}.

The empirical entropy H (D) of data set Dis calculated as follows.

(4)

The empirical conditional entropy H(D|A)of featureA for dataset D $H(D) = -\sum_{k=1}^{K} \frac{|C_k|}{|D|}$ws

(5)

The information gain calculation formula foreach feature is as fol $H(D|A) = -\sum_{i=1}^{n} \frac{|D_i|}{|D|} \sum_{k=1}^{K} \frac{|D_{ik}|}{|D_i|} \log_2 \frac{|D_{ik}|}{|D_i|}$

(6) $g(D, A) = H(D) - H(D|A)$

Based on feature selection method of informationgain criterion, each feature's information gain is

measured, and the features with larger IG value are selected.

III. THE IGTF-DNN ALGORITHM

The vulnerability automatic classification model IGTF-DNN is composed of IG, TF and DNN. The original vulnerability data is first preprocessed, and then TFI is used to grab features of vulnerability description text and lower the dimensionality of generated higher- dimensional word vector space, and later the DNN is build to comprehend automatic training and classification of vulnerability.

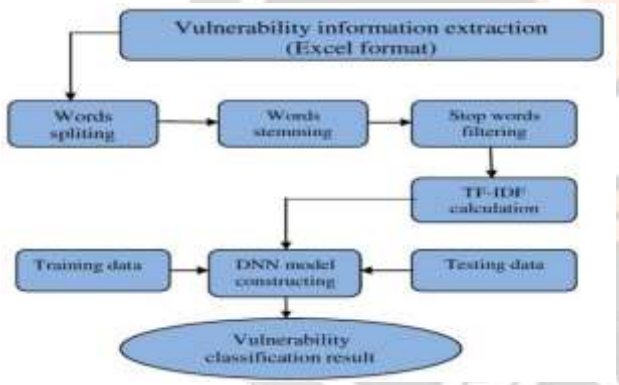


Figure 1. TFI-DNN algorithm

A. FEATURE SELECTION USING TFI:

TFI is used to grab feature word set. The steps are given in Algorithm 1.

B. OPTIMIZATIONS USING DNN: DNN includes single input layer, one hidden layer (can be made to two if required) and single output layer, whose input is the instance's feature vector and output is instances' category. It includes one propagation process, forward propagation and back propagation. The propagation process is in Algorithm 2.

IV.EXPERIMENTAL ENVIRONMENTAND DATA SET

A.ENVIRONMENT:

The experiment was conducted on PC with Intel(R) Core(TM) i3 processor, 2.4 GHz and
8.00 GB memory, running Windows 10 operatingsystem. Programming uses R 3.4 on R Studio version 0.99.

B.DATA SET OF EXPERIMENT:

In order to test effectiveness, internationally recognized National Vulnerability Database (NVD) [19] is used as experimental data. The source file of this dataset is a series of records taken into excel worksheet, which contains information about vulnerability, such as 'CVE number', 'vulnerability release date', 'CVSS_version', 'type' and 'vulnerability text description' of Vulnerability type and description are taken for analysis.. The annual vulnerability amount (2015-2019) NVD vulnerability database is taken and the record sample count is 600.The required vulnerability information is extracted from the records using program codes written inR including vulnerability text description, and vulnerability type (category).Some text information of vulnerabilities from 2015 to 2019 is collected for statistics, 500 of them were used for training data set and 100 for test data set.

Algorithm 1Input:
Word list(word_list) formed by term documentmatrix and stop word filtering.
Output:
Feature word set (feature_words).

1) 1)Traversing each word in the word_list.
2) Word frequency statistics for word_list,storedin the doc_frequency list.
3) Traversing the word frequency listdoc_frequency.
4) Calculate the TF value of each word accordingto (1) and store it in the word_tfdictionary.

5)Calculate the IDF value of each word according to (2) and store it in the word_idf dictionary. 6) Calculate the TF-IDF value of eachword according to (3) and store it in the word_tf_idf dictionary.

6)The word set is sorted in descending accordingto the TF-IDF value.

7) Select the first n words as an important featureset.

8) Save important words in the feature list(features_ vocabSet).

9) Traverse features_vocabSet, divide features_vocabSet and store the subset into thesubDataSet.

10) Calculate probability of subDataSet.

11) Calculate the empirical conditional entropy of each word according to (4) and (5) and store itin newEntropy.

12) Calculate the IG value of each wordaccording to (6).

13) Save each word and the corresponding IGvalue in the dictionary.

14) The word set is sorted descending by IGvalue.

15) Select the first m words as features and storethem in the feature_words.

16) Return feature_words.

IV. EXPERIMENTAL PROCEDURE

A. DATA PREPROCESSING

Excel work sheet records are taken from 5 excel work sheets for year 2015, 2016, 2017, 2018 and 2019. They are stored in data frame objects. Then category array is filled with unique values of categories from Vulnerability type column. Separate data frames are formed from subsets which store each category. Term Document

Matrix is found and all the terms are stored in array. Stop word removal and punctuation removal is carried out using tm package's tm_map method. Out of 2237 words/terms extracted 113 terms are found as unique and they are taken for further analysis. These numbersmay vary if more vulnerability description samples are taken as dataset.

B. MATRIX FORMATION

TF Matrix, IDF Matrix and TF-IDF array is found out using the equation given above (1 to 3) and the terms are sorted based on TF-IDF values.

C. INFORMATION GAIN

Then entropy H (D) of data set D is calculated and later Information Gain values are calculated for each feature and then sorted descending basedon IG values.

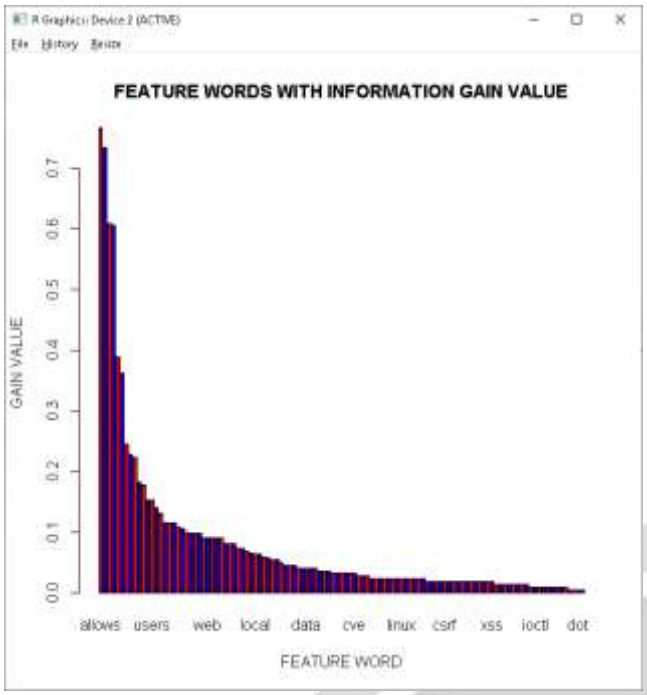
Sample TF-IDF features:tfidf_df_sorted:

	Word	Freq
1	xss	0.005312172
2	service	0.005032584
3	via	0.004962687
4	denial	0.004892790
5	web	0.004683099
6	attackers	0.004543305
7	remote	0.004473408
8	cause	0.004473408
9	scripting	0.004403511
10	crosssite	0.004333614

OutputDF:

Feature with GainValue:

	Feature	GainValue
1	xss	0.018107652
2	service	0.106077030
3	Via	0.735451987
4	Denial	0.081597715
5	Web	0.090538261
6	Attackers	0.610320229
7	remote	0.606898196
8	cause	0.130556345
9	scripting	0.018107652



category belongs to first (out of 30 categories), 0.334 to 0.666 belongs to second category and so on.

1).Fetch one description and convert to 113 neurons, 10 hidden layer one input weights are set and network is made to run.
2).The output is generated and noted.
3). For all descriptions, the above process is made and output is noted.
4).The weights and biases for hidden and output layer are recalculated for given number of epochs (ten iterations).
5).The forward propagation of the input layer and hidden layer uses 'tanh' as the activation function, while the output layer uses 'softmax' as the activation function.
6).The final weights and bias values are taken for further test description samples.

Figure 5.1 Informtion Gain Value for Features(Sorted)

D. OPTIMIZATIONS USING DNN:

DNN consists of one input layer, multiple hidden layers and one output layer, whose input is the feature vector of instance and output is the category of instance. It mainly includes two propagation processes, forward propagation and back propagation. The propagation process is in Algorithm 2.

ALGORITHM 2INPUT:

VECTOR ENCODED DESCRIPTION (If the feature word present in the given description 1 is set otherwise 0 is set. So the vector encoded value contains 113 values i.e., 1 and 0 combinations for 113 terms (feature words). So 113 neurons

OUTPUT:

1 Output neuron with value 0 to 1. The category count is set to 30 so for any input neurons with 113 neuron values, if value between 0 and 0.333 is prepared as output in output neuron, then the

V. SCOPE AND PURPOSE OF STUDY

A system is first scanned by an attacker to determine whether it contains a software vulnerability. The scan can provide information tothe attacker about the kinds of software installed on the system, whether they are current, and whether any software packages are vulnerable. Theattacker will have a better understanding of the kinds of attacks to launch against the system once they learn that. If the attack is successful, the attacker will have access to the target system and will be able to execute malicious commands. A deep neural network model is employed to produce an automatic vulnerability classifier that correctly categorises vulnerabilities. We can find assaultsusing this DNN with less manual labour.

VII.CONCLUSION

In order to better analyze and manage vulnerabilities according to their belonging classes, enhance the system's security capabilities,lower the chance that the system will be compromised, and attacked and damaged, this paper applied deep

neural network to vulnerability classification. The analysis of the method and construction process of TFI and DNN are discussed in detail. The comparison is made with the For the NVD dataset, compare the vulnerability classification model TFI-DNN against TFI-SVM, TFI-Naive Bayes, and TFI- KNN. The findings demonstrate that the suggested TFI-DNN model outperforms in creating weights and biases. And it is superior to general TF-IDF on comprehensive evaluation indexes. The work in this paper shows the effectiveness of TFI-DNN in vulnerability classification, and provides a basis for our future research using the benchmark vulnerability dataset.

REFERENCES

[1] Mozilla achine Learning for Language Toolkit," http://mallet.cs.umass.edu, 2002.

[13] Z. Harris, "Distributional Structure," Word, vol. 10, pp. 146-162, 1954.[1] R. P. Abbott, J. S. Chin, J. E. Donnelley, W. L. Konigsford, S. Tokubo, and D. A. Webb, Security Analysis and Enhancements of Computer Operating Systems. Washington, DC, USA: US Department of Commerce, 1976.

[2] I. R. Bisbey and D. Hollingworth, Protection Analysis: Final Report. Marina Del Rey, CA, USA: Univ. of Southern California, 1978.

[3] A. Gray, "An historical perspective of software vulnerability management," Inf. Secur. Tech. Rep., vol. 8, no. 4, pp. 34–44, 2003.

[4] P. J. Kim, ''An analytical study on automatic classification of domestic journal articles based on machine learning,'' J. Korean Soc. Inf.Manage., vol. 35, no. 2, pp. 37–62, 2018.

[5] B. Shua, H. Li, M. Li, Q. Zhang, and C. Tang,"Automatic classification for vulnerability based on machine learning," in Proc.IEEEInt.Conf.Inf. Automat. (ICIA), Aug. 2013, pp. 312–318.

[6] D. Wijayasekara, M. Manic, and M. McQueen, "Vulnerability identification and classification via text mining bug databases," in Proc.40th Annu. Conf. IEEE Ind. Electron. Soc., Nov. 2014, pp. 3612–3618.

[7] S. Na, T. Kim, and H. Kim, "A study on the classification of common vulnerabilities and exposures using Naïve Bayes," in Proc. Int. Conf.Broadband Wireless Comput., Commun. Appl.Cham, Switzerland: Springer, 2016, pp. 657–662.

[8] M. Gawron, F. Cheng, and C. Meinel, "Automatic vulnerability classification using machine learning," Proc. Int. Conf. Risks Secur. Internet Syst. Cham, Springer, 2017, pp. 3–17.

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2009, pp. 248–255.

[10] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," Int. J. Comput. Vis. vol. 115, no. 3, pp. 211–252, 2015.

[11] W. Xiong et al., "Toward human parity in conversational speech recognition," IEEE/ACM Trans. Audio Speech Lang. Process., vol. 25, no. 12. PP. 2410–2423, Dec. 2017.

[12] A. Krizhevsky, I. Sutskever, G. E. Hinton, ''ImageNet classification with deep convolutional neural networks," in Proc. Adv. Neural Inf. Proc. Syst., 2012, pp. 1097–1105.

[13] D. Silveretal., "Mastering the game of Go with deep neural networks and tree search," Nature vol. 529, pp. 484–489, Jan. 2016.

[14] M. Iyyer, V. Manjunatha, J. Boyd-Graber, and H. Daum,"Deep unordered compositionrivals syntactic methods for text classification," in Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process., 2015, pp. 1681–1691.

[15] H. Jo, J.-H. Kim, K.-M. Kim, J.-Ho Chang, J.-H. Eom, and B-T. Zhang, ''Large-scale text classification with deep neural networks,"Comput. Congnit., vol. 23, no. 5, pp. 322–327, 2016. [16] W. Aziguli et al., ''A robust text classifier based on denoising deep neural networkin the analysis of big data,'' Sci. Program., vol. 2017, 2017, pp. 1–10.