Categorizing software vulnerabilities using overlapping self-organizing map

Sima Hassanvand[1], Mohammad Ghasemzadeh[2*]

[1,2]Department of Computer Engineering, Yazd University, Yazd, Iran
*Corresponding Author's Email: m.ghasemzadeh@yazd.ac.ir

Abstract— *Software has always been vulnerable to various vulnerably issues. Increasing the number of vulnerabilities and their complexity in the software area has made it more important to categorize them. In this research work, by selecting the MoSCoW prioritization method and by combining it with the SOM self-organizing mapping algorithm, we present a new categorization for the frequent software vulnerabilities. We implemented the proposed method in MATLAB using the relevant tool boxes. The experimental results were evaluated using in-class and out-of-class distance measurements. Classification of software vulnerabilities using OSOM algorithms gives us better results than conventional clustering methods. It can be inferred that the classification of software vulnerabilities is of particular importance in improving the security of a software application. The proposed algorithm can provide an appropriate categorization by taking advantage from the existing overlapping feature.*

Keywords— Overlapping, Overlapping Self-Organizing Map, Software Vulnerability, Vulnerability Categorization.

I. INTRODUCTION

In many cases, programmers' faults during programming, which could easily be prevented, create vulnerabilities, providing an opportunity for hackers to misuse it. A proper classification for vulnerability could be sufficient to understand vulnerabilities and propose a solution to prevent them. By collected information from vulnerabilities, suitable classification is achievable, and new vulnerabilities could be easily classified into appropriate classes. Vulnerability classification is a substantial task due to weak software that could be easily manipulated. In the present study, a proper approach for vulnerabilities via MoSCoW method to select reliable database coupled with self-organizing map, is introduced, and the classification results are compared with the self-organizing map. The algorithm is generated from combination of SOM and K-means clustering, and by considering overlapping, a suitable classification is introduced. Overlapped self-organizing map is applied on different databases and is present in acceptable results compared to previous methods, and it is planned to examine the algorithm on software vulnerabilities, leading to appropriate standard classification. This paper is organized as follows: Section 2 describes the few works related to the study. Section 3 presents the applied approach, which in this section; database and extracting Eigen vectors are evaluated. Section 4 is dedicated to experiments and results, and Section 5 discusses the study conclusions.

II. RELATED WORKS

Primary researches were conducted in1976 by the United States making calculation centers to conduct studies on software vulnerability, specifically on operating systems[1].Bishop and Krsul could be introduced as pioneers on presenting vulnerability classification methods. Bishop [2], by studying vulnerabilities, explained their various types; for example, when vulnerability was introduced? What would occur after misuse? On what issues are affected by vulnerability? What are the minimum necessary components to use vulnerability and it's recognizing resources? Bishop investigated 11 failures in Unix and provided six classifications. Afterward, Krsul[3]studied the same subject and presented a vulnerability classification based on the decision tree. Decision trees are in accordance with prior assumptions. Krsul's purpose was to assign a specific classification for each relative vulnerability. Venter introduced a category range, including 13 homogeneous vulnerability portions, providing a complete scale for known vulnerabilities [4].

The classification consisted of ahead items: shapes in password, system and network information collection, backdoors, trojans and remote control, unauthorized accessibility to junctions and remote services, privileges and user accessibilities, spoofing or masquerading, misconfiguration, service rejection, buffer overflow, viruses and worms, absolute hardware, absolute software and updates, and security method bug. The mentioned items were classified based on knowledge and personal or general experiences. After Venter's research, Cisco[5], a known security corporation, in 2003 categorized

vulnerabilities in 5 different sections as design errors, protocol feeble, software vulnerabilities, misconfiguration, and bad codes.

The major vulnerability scanners have their own classification such as SAINT, which in 2004, vulnerabilities were classified in 12 divisions [6]: web, Email, FTP, shell, print, RPC, DNS, database, network, windows, password, and miscellaneous. Microsoft [7] proposed a warning model in 2002, including six vulnerability categories, namely spoofing identity, data manipulation, repudiation, information disclosure, service rejection and privilege elevation. SF-Protect [8] introduced seven groups for classification: user reports, audit policy, system logon, file system, registration, services, and shares. Missouri Research and Education Network classified vulnerabilities in 25 groups in 2014[9]. Venter, Ellof and Li [10] presented few articles for general categorization of vulnerabilities in CVE database by means of SOM. Their approach is effective to eliminate tiredness and fatigue of human classification and to decrease error due to tiredness. SOM advantage is to ease explanation and interpretation. SOM is useful as a device to visualize large amount of data in 2D dimension. In many practical programs such as classification, due to several reasons, overlapping always exists between different sections [11].

Ideally, different groups are pure distinct, nevertheless, having overlapping is unavoidable. Whenever different classes have overlapping in between, generally ´´winner-take-all´´ will be used to choose groups, but it is not effective in most cases. Classification overlapping is not a new issue and Sibson and Jardine investigated it and recommended an overlapping categorizing model in which performance was similar to present-day taxonomy [12]. Diday[13]extended K-Ultra-metric model by Jardin, and introduced a new model, but in the model, each class could have overlapping only with two other classes. PoBoc's model was introduced by Cleuziou[14]. Henceforth, Banerjee presented overlapping clustering [15]. The model is known MOC and is a generalization of EM. Cleuziou (2007), by using Banerjee's clustering model, recommended K-means overlapping model. Two major differences exist between Banerjee's model and K-means; the difference in determining one or several classes for each sample, and the difference in updating process of center classes [16].

III. THE PROPOSED METHOD

3.1 Research Database

After identification of vulnerabilities, they could be stored in a database. Generally, vulnerabilities database can be classified into two categories: general vulnerabilities database and software developer vulnerability database. General vulnerability database such as NVD, CVE, OSVDB. Software developer vulnerability database is like MFSA (Mozilla Foundation Security Advisories). National vulnerability database is a standard source developed by the United States. It provides automatic vulnerabilities management and security measurement and coincident. It also utilizes SCAP automation protocol for management and automatic categorization of known security bugs governed by the National Institute of Standards and Technology (NIST)[17]. NVD consists of CVE, vulnerability effect, standard code of vulnerability intensity, vulnerability description in a natural language, vulnerability type, and name of vulnerability software, published date, update date, and available references for vulnerabilities. Selecting a suitable database is one of the basic steps in data analysis. In some cases, selecting an inappropriate database leads to unreliable results. The employed data should be valid and reliable. There are various databases registering software vulnerability reports. Known databases are OVAL, NVD, OSVDB, and CVE. In other words, in most studies, combination and integration of different databases are not required, and researchers only need to choose a proper database according to the research aims [18]. In addition, vulnerability description on CVE and CWE is required to extract features from vulnerability explanation. The database is selected based on MoSCoW prioritization technique. The term MoSCoW itself is an acronym derived from the first letter of each of the four prioritization categories (Must have, Should have, Could have, and won't have), with the interstitial o's added to make the word pronounceable.

M – Must have:

This point describes requirements that must be met in the final solution. These requirements are non-negotiable, and the project will fail without them.

S – Should have:

A high-priority feature that is not critical to launch, but it is considered to be important and of a high value to users. Such requirements occupy the second place in the priority list.

C – Could have:

A requirement that is desirable, but not necessary. According to the method, this point will be removed first from the scope if the project's timescales are at risk.

W – Won't have:

A requirement that will not be implemented in a current release, but may be included in a future stage of development. Such requirements usually do not affect the project success. For priorities assessment with MoSCoW technique, a score should be considered for each level and score of each database is calculated by means of the following expression.

$$(\text{Must})\times 4 + (\text{Should})\times 3 + (\text{Could})\times 2 + (\text{Want})\times 1$$

3.2 Feature vectors extraction

Term Frequency–Inverse Document Frequency (TF-IDF) is a document. The TFIDF gives weight to each word based on its frequency in document. In fact, TFIDF shows how a word in a document is important. This is highly practical in information detection. Weight of a word increased via repetition in the document, but it is controlled by words quantity, while if the document is long, some words normally will be more repeated than others, even they had less importance[19]. The study objective is to present a new classification according to document information of vulnerability explanation. Therefore, features vector is generated from document tools and evaluation method through vulnerability explanation. Feature vector creation methods in the research are:

- Words extraction from documents
- Delete pause words
- Root recognition of words
- TF-IDF calculation for each feature

In this step, WVT (Word Vector Tool) tool is used, which generates two files as output, including words list and created vectors list. Feature vector is constructed from NUMBER and TF-IDF VALUE. For non-existing words in the document, no. 1 is used, and for existing words in the document, its equivalent value from TF-IDF is employed. The feature vector reached 2997 for each vulnerability. Figure 1 illustrates feature extraction steps.

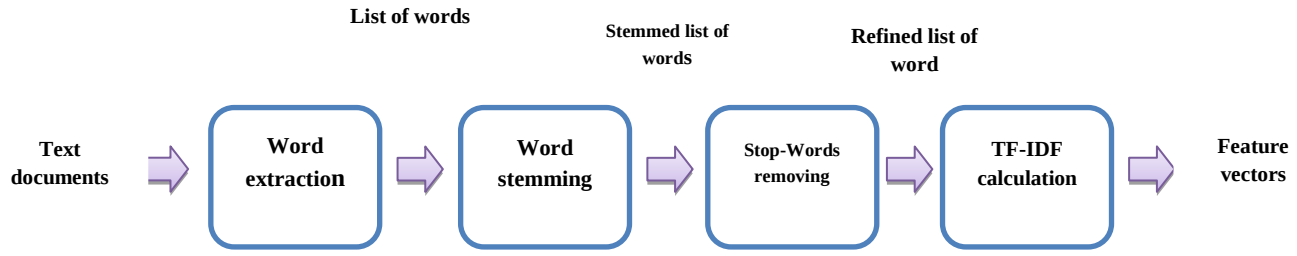

FIGURE 1: FEATURE EXTRACTION STEPS

3.3 Evaluation Method

Clustering evaluation is a boring process, which can be accomplished with two measurement approaches: external and internal measurement [20]. According to previous investigations, there is a general framework to apply on internal and external measurements. Inter-cluster interval is calculated by formula (1) and intra-cluster interval is computed by formula (2).

$$int-classif = \sum_{i \in 1:N} \sum_{j \in 1:N} d(x_i - x_j) \quad (1)$$

$$ext - classif = \sum_{i \in 1:N} d(x_i - c) \tag{2}$$

In the mentioned equations, N is quantity of data being placed in a clustering. To evaluate efficiency in the model, confusion matrix is used to determine correctness or imperfection amount of data recognition in each class. Moreover, precision and accuracy, sensitivity, and specificity could be achieved to have a proper estimation of algorithm quality. Other criteria for classification consist of inter-cluster and intra-cluster intervals [21].

TABLE 1
CONFUSION MATRIX

Actual \ Predicate	Positive	Negative
Positive	True Positives	False Negatives
Negative	False positives	True Negatives

Overlapped self-organizing map uses SOM, providing a structural algorithm with less sensitivity on quantity of clusters. Data are categorized on a matrix with specific number of neurons, and topology correctness is shown with a simple matrix method. Finally, it prepares a suitable clustering structure on data, causing to decrease complexity. Advantage of utilizing overlapping clusters is to have a datum belonging to different clusters. When a datum is sent to matrix, instead of a neuron, a set of neurons is needed for evaluation. Validity of overlapping topology could be evaluated by confining a set to a specific class ς in matrix [22]. OSOM algorithm is introduced by combination of K-means and type of SOM, which was proposed by Heskes [23]. A new model is demonstrated as (3) criteria.

$$E_{osom} = \frac{1}{N} \sum_{x_i \in X} \sum_{G_r \in W} h_{rg(W, x_i)} \| x_i - \bar{w}_r \|^2 \tag{3}$$

IV. EXPERIMENT AND RESULTS

To analyze clusters data, model recognition and model behavior prediction, first the data should be classified and then evaluated and predicted by the obtained model. For example, a classified model for vulnerability taxonomy may be categorized in two sections: low risk vulnerabilities sand high risk vulnerabilities, while in the recommended model, prediction is applied on vulnerability clusters based on vulnerability interpretation. Classification is a model detection along with cluster determination or data concepts, which could predict unknown clusters of other objects. Classification is a learning process assigning a cluster to a datum. Data are divided into two sections: training and test data. Training data are employed for system learning, and test data are used for the model accuracy evaluation. As it was discussed, after valid database selection of software vulnerabilities, proper fields are chosen as: CVE number, CWE and software vulnerabilities interpretation. Vulnerabilities interpretation is used to generate the feature vector. The field is textual and is employed for feature extraction by using WVT. The tool produces the feature involving TF-IDF pattern. Afterward, the vector is converted to a matrix, and if the pattern holds the feature amount in TFF-IDF, it is situated in matrix, otherwise, 1 is replaced instead. Up to now, there is a matrix with 2997 rows by 47295 columns filled by TF-IDF amounts and 1. Furthermore, the overlapped amount between the patterns is obtained 1.01. To process assurance, each pattern of CVE is correspondingly settled in addition to its CWE pattern. To develop the model, the selected algorithm is OSOM, being selected based on the experiment results from vulnerability issues. The obtained matrix from previous steps is brought forward to algorithm and output is achieved. There is no regular or specific process for number of clusters, but clustering correctness could be evaluated via assessment. In total, 80% of the utilized data is used in training step and 20% in test step, and the experiments are accomplished 10 times in average. Clustering by OSOM is made in three phases: competition phase, collaboration phase and accommodation phase. Overlapping self-organized map is utilized in competition phase that uses HESKES error criteria. In clusters centers selection step, it uses clusters average as clusters centers, and in collaboration phase, to select adjacency, it uses Hasdorf interval instead of Euclidean. Table 2 presents the obtained results from the proposed method via OSOM.

TABLE 2
RESULTS FROM THE PROPOSED METHOD VIA OSOM

		$Qext_{classif}$			$Qint_{classif}$	
	3x3	**5x5**	**10x10**	**3x3**	**5x5**	**10x10**
Scene	0.525	0.542	0.542	0.199	**0.146**	0.116
Emotion	**0.479**	**0.472**	0.484	**0.176**	0.150	**0.099**
CVE	0.561	0.496	**0.448**	0.597	0.412	0.284

As Table 2 illustrates, it is concluded that OSOM inter-cluster interval has greater improvement in smaller patterns. In contrary, software vulnerabilities are increased by pattern increment, and the improvement is increased, correspondingly. Classification evaluation is made for each cluster separately, and favorite criteria are obtained.

TABLE 3
SAMPLE CONFUSION MATRIX FOR A CLASS

	Positive	**Negative**	**Total**
Positive	860	496	1356
Negative	578	356	934
Total	1438	852	2290

Table 4 shows the results obtained from the proposed model and primary model as the prediction model on CVE. During the study, 10 times algorithm executing and 10x10 pattern with 100 times repeat were accomplished.

TABLE 4
ESTIMATION OF THE PREDICTOR MODEL

	Precision	**Sensitivity**	**F-measure**
SOM	0.33	**0.60**	0.42
OSOM	**0.37**	0.54	**0.43**

According to Table 4, OSOM accuracy is higher than the ordinary method, and its precision value is less, but eventually, F criterion of the proposed model is better than the ordinary method.

V. CONCLUSION

This article deals with software vulnerability clustering as an important and time-consuming issue for researchers. After selecting a valid database and required fields such as CVE, CWE and vulnerability interpretation, relevant features were extracted. By using WVT tool, TF-IDF of each pattern was obtained and 2997 features were achieved. According to the experiment results, high amount of vulnerability could belong to different patterns. Therefore, it was required to database be evaluated and value of vulnerability database overlapping is obtained 1.01. The proposed OSOM method was introduced as an extension from SOM algorithm by means of overlapped K-means. Classification accuracy increased through a new algorithm via combination of centers and a new definition for winner neuron. The accuracy was obtained from variations in the ordinary self-organized map. After applying experiments on vulnerabilities and evaluation on favorite criterions, it is concluded that overlapped self-organized map could be a suitable approach for software vulnerability classification. Performed activities on the research were only a step to present a standard classification on software vulnerabilities. To have an accurate model, complete date is required. In this research, the overlapped self-organized method has been utilized for prediction, and it is planned to use other unemployed methods for software vulnerabilities classification. Moreover, in future studies. We can use the ideas proposed in OSOM model, for clustering models in which the overlapping issue is not considered.

REFERENCES

[1] Ammala, DE. (2004). Derivation OF Metrics for Effective Evaluation of Vulnerability Assessment Technology. Mississippi State University.

[2] Bishop, M. (1995). A taxonomy of UNIX system and network vulnerabilities. Technical Report CSE-9510. Davis: Department of Computer Science, University of California.

[3] Krsul IV (1998) Software vulnerability analysis. Available at: http://www.krsul.org/ivan [Accessed 1 Sep. 2015].

[4] Venter, H. Elofe, J. (2002).Harmonizing Vulnerability Categories. Computer Society of South Africa South Africa, ISSN 1015-7999, pp. 24-31.

[5] Kujawski, P. (2003). Why Networks Must Be Secured. Cisco Systems, Incorporation.

[6] The SAINT Corporation. Available at: http://www.saintcorporation.com [Accessed 11 Feb. 2017].

[7] Microsoft Commerce Server (2002). The STRIDE threat model. Available at: http://msdn2.microsoft.com [Accessed 11 Feb. 2017].

[8] MOREnet, Missouri Research & Education Network Available at: http://www.more.net/services [Accessed 1 Sep. 2015].

[9] Venter, H. Eloff, J. Li, Y. Standardising vulnerability categories. Computers & Security.

[10] Tang, W. Mao, KZ. Mak, G. (2010). Classification for Overlapping Classes Using Optimized Overlapping Region Detection and Soft Decision. International Conference on Information Fusion.

[11] Jardine, N. Sibson, R. (1971). Mathematical Taxonomy. Statistical Data Analysis, pp. 405–416

[12] Diday, E. (1987). Orders and overlapping clusters by pyramids. Technical report, INRIA num.730, France.

[13] Cleuziou, G. Martin, L. Vrain, C. (2004). PoBOC: an Overlapping Clustering Algorithm. Application to Rule-Based Classication and Textual Data.

[14] R. L´opez de M´antaras and L. Saitta, IOS Press, editor, Proceedings of the 16th European Conf. on Articial Intelligence, pp. 440-444.

[15] Banerjee, A. Krumpelman, C. Ghosh, J. Basu, S. Mooney, R. (2005). Model-based overlapping clustering. Proceeding of the eleventh ACMSIGKDD, pp. 532–537.

[16] Cleuziou, G. (2007). OKM: une extension des k-moyennes pour la recherche de classes recouvrantes. EGC, 2.

[17] Last, D. (2011). Using Historical Software Vulnerability Data to Forecast Future Vulnerabilities. Air Force Research Laboratory information Directorate, RISB.

[18] Khazaei, A. Ghasemzadeh, M. (2016). Software Vulnerabilities Database Selection Using MoSCoW Prioritization Method. 3rd Int. Conference on Applied Research in Computer & Information, Tehran, pp. 1-7.

[19] The wikipedia (2017). Tfidf [online] Available at: https://en.wikipedia.org [Accessed 1 Sep. 2016].

[20] Jain, A. Dubes, R. (1988). Algorithms for Clustering Data. Prentice Hall Englewood Cliffs.

[21] T, Fawcett. (2006). An introduction to ROC analysis. Pattern Recognition, 27, 8, pp. 861–874.

[22] Cleuziou, G. (2013). A Method for building overlapping topological map. Pattern Recognition Letters, 34, 3, pp. 239–246.

[23] Heskes, T. (1999). Energy functions for self-organizing maps. University of Nijmegen Geert Grooteplein 21, 6252 EZ, Nijmegen, The Netherlands.