OCFSII: A New Feature Selection Based on Orthogonal Centroid Both Inter-class and Intra-class for Vulnerability Classification

Jialiang Song[1], Jihong Han[1], Xirui Zhang[1], Lulu Shao[1], and Yang Wang[2]

(Corresponding author: Jialiang Song)

Zhengzhou Information Science and Technology Institute[1]
Zhengzhou 450001, P.R. China
Xi'an Surveying and Mapping Technological Center[2]
(Email: sjl1032011026@163.com, 496443004@qq.com)

(Received Sept. 20, 2017; Revised and Accepted Mar. 27, 2018; First Online Dec. 10, 2018)

Abstract

With the rapid development of information technology, vulnerability has become a major threat to network security management. Vulnerability classification plays a vital role in the whole process of vulnerability management. It is the key point to select proper features to represent categories. Due to the low efficiency and accuracy of some common feature selection algorithms, in this paper, we proposed a new method called OCFSII, which measures the importance of the feature terms both in inter-class and intra-class based on orthogonal centroid. We evaluated the method on the vulnerability database, using two classifiers, namely, KNN and SVM. The experimental results show that the proposed method OCFSII outperforms Information Gain (IG), Document Frequency (DF), Orthogonal Centroid (OC), and is comparable with Improved Gini index (IGI) when KNN used while OCFSII is superior to the four algorithms. In addition, OCFSII is more advanced than OC.

Keywords: Classifier; Feature Selection; Information Security; OCFSII; Vulnerability Classification

1 Introduction

Nowadays, with the development of network technology, people can get information in different channels. To meet the demands of various users, the relevant products, such as different operation systems and application software, are developed, which greatly promotes transmission and sharing of information. However, because of the defects of operation and application software on design or on own disadvantage of programming languages, these products have various disadvantages on design and realization. In terms of information security, the most significant defect is inevitable security vulnerabilities [12]. With the improvement of information society, the coverage rate of Internet devices is improving while the number of security vulnerabilities increases in exponential type. Therefore, it is significant to manage the numerous vulnerabilities [10].

As an important link of vulnerability management, the key point in vulnerability classification is to describe and distinguish different vulnerabilities accurately. Accurate classification of security vulnerability is the basis to continue to analyze and manage vulnerability. It can also greatly help vulnerability researcher know profoundly generation cause and attack influence of the same kind of vulnerability, and it provides key reference information for security administrator to assess severity of vulnerability correctly. Detailed data about vulnerability is indispensable core data for computer security tool and vulnerability classification, while these data are based on text information.

It is the key for vulnerability classification to establish relationship between feature and category and moreover, it is the key point for research of this paper about how to select proper vulnerability features to represent vulnerability category. Whether vulnerability features are proper or not will greatly influence the accuracy of vulnerability classification. In recent years, research popularity for text features selection still increases. Venter *et al.* [13] have put forward a kind of automatic classification scheme based on Self Organization Maps (SOM), which is a kind of data cluster algorithm. The main contribution of this method lies in a type of experimental vulnerability classification model, which does not need to define the vulnerability category manually in advance. It can collect vulnerability samples with similar features into different categories automatically by SOM algorithm. But, this method has low accuracy and efficiency. Mingoti *et al.* [9] has improved vulnerability classification model based on SOM cluster algorithm in [13] using N-Gram replacement word, which has advanced accuracy of cluster. Wang *et al.* [14] has proposed a kind of automatic classification model for vulnerability based on Bayesian network, which trains Bayesian network by vulnerability information obtained from NVD database and then divides vulnerability into categories defined by CWEs. Chen *et al.* [2] have presented a kind of automatic classification model based on SVM, which trains the SVM classifier with vulnerability information obtained from the CVE list and divides vulnerability automatically into predefined vulnerability features categories. Zhang *et al.* [17] have put forward the research on vulnerability classification method based on fuzzy entropy features selection algorithm. This method can classifies different vulnerabilities combining the advantages of fuzzy entropy theory and SVM classification method and give the evidence for vulnerability features selection to calculate fuzzy entropy. In addition, many scholars have put forward various feature selection algorithms to select more reasonable vulnerability features and improve accuracy and learning ability of vulnerability classification.

However, these methods have some disadvantages. Here are the following points to be improved:

1) Factors for assessing these algorithms are too simple and the situation that distinguishes categories via features is usually considered from one perspective. For example, in [16], document frequency only measures the significance of a feature term in the intra-class while in [1, 15], orthogonal centroid feature selection algorithm and DIA association only calculate the score of a feature in the inter-class. Namely, these algorithms do not take into account importance of features both in the inter-class and intra-class.

2) During the results and analysis of these algorithms, experiments are carried out only by utilizing one same kind of classifier. And the influence in different types of classifiers on accuracy of vulnerability features is not compared. For example, in [4], only naïve Bayes is taken as an experiment tool of vulnerability classification while in [5, 6], only SVM is taken as an experiment tool of vulnerability classification.

3) These algorithms need to obtain vulnerability text resource from vulnerability database, while different vulnerability database has different text factors.

It is necessary to formulate the unified vulnerability text factors to enhance the applicability

To solve the above problems, this paper compares factors in different vulnerability databases, and proposes the standard vulnerability text factors. Moreover, we put forward a new features selection algorithm, called Orthogonal Centroid Features Selection algorithm both in Inter-class and Intra-class (OCFSII). To confirm this method, we use two classifiers including SVM and KNN in vulnerability data, and compare it with four feature selection algorithms including in Information Gain, Improved Gini index, Document Frequency and Orthogonal Centroid. The experiment results show that the proposed method OCFSII outperforms IG, DF OC, and is comparable with IGI when KNN used while OCFSII is superior to IG, DF, OC and IGI when SVM used.

The main contributions of our paper are as follows.

1) This paper gives the standard and unified vulnerability text factors from different vulnerability database.

2) The proposed method measures the significance of a feature term both in inter-class and intra-class.

The remainder of the paper is organized as follows. In Section 2, vulnerability classification principle, feature selection and feature term -classification matrix are briefly reviewed. After that, the proposed method algorithm is presented in Section 3. Experimental setup and Results are included in Section 4 and Section 5 respectively. Finally, the concluding remarks are drawn in Section 6.

2 Related Work

2.1 Vulnerability Classification Principle

Since vulnerabilities from regular vulnerability databases and open vulnerability resource mainly are presented in text form, this paper classifies vulnerabilities via referring to relevant technologies of text classification. Text classification is a process that divides the given text to one or more predefined text categories according to contents [7]. Similarly, Vulnerability classification is a process that classifies the unknown vulnerabilities into predefined vulnerability categories. From the mathematical perspective, vulnerability classification is a special mapping process actually.

This paper gives formalized description for vulnerability classification: Giving a vulnerability text set $D = \{d_1, d_2, \cdots, d_{|D|}\}$ and a vulnerability category set $C = \{c_1, c_2, \cdots, c_{|C|}\}$, where, $|D|$ and $|C|$ represent the number of vulnerability text and vulnerability categories. There is an unknown ideal mapping Φ between vulnerability text set and vulnerability category set:

$$\Phi : D \to C. \tag{1}$$

The purpose of classification learning is to find a mapping model φ that is the most similar to ideal mapping Φ and based on the given assessment function f, the aim of learning is to make Φ and φ fulfilling the following formula

$$Min\left(\sum_{i=1}^{|D|} f(\Phi(d_i) - \varphi(d_i))\right) \tag{2}$$

Generally, the process of vulnerability classification is shown as Figure 1, which includes learning stage and classification stage. Learning stage consists of training process and test process. In order to find the proper parameters for classifying, the feedback mechanism is introduced, which could improve the training results. Classification stage classifies unmarked vulnerabilities by utilizing classifiers ultimately generated in learning stage and vulnerability classification results are output.

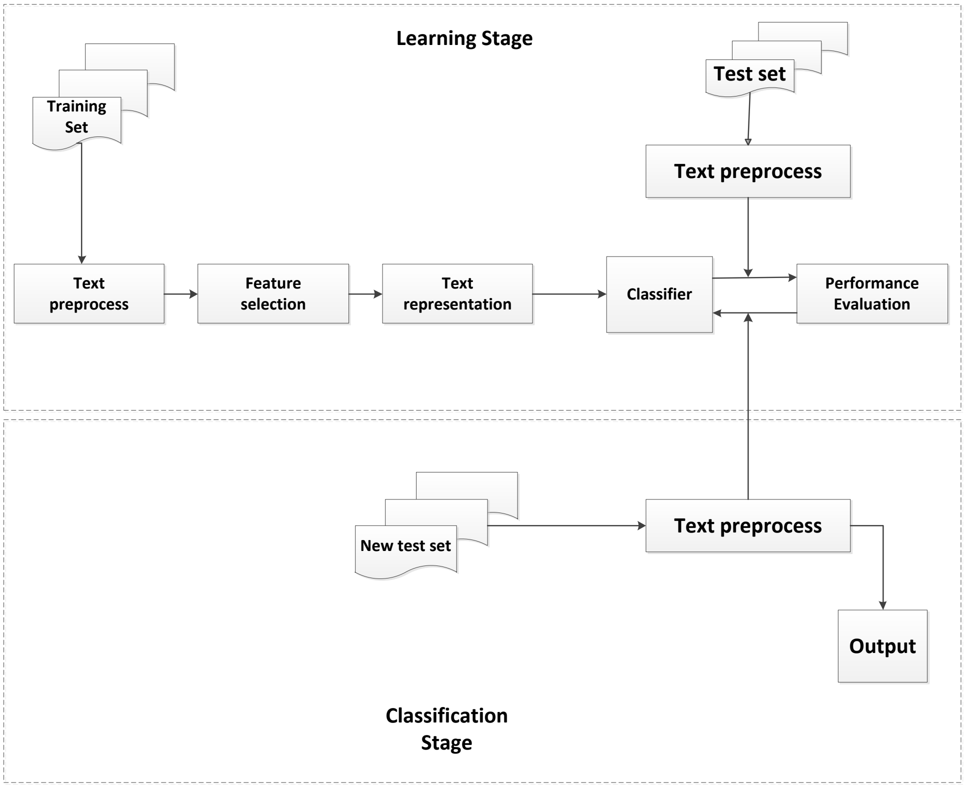

Figure 1: Vulnerability classification process

2.2 Feature Selection

Feature selection is a method which we use proper evaluation criteria to select the optimal features subset from the original feature set. The aim is to select the smallest features subset according to some criteria, so that some tasks, such as classification and regression, achieve better results. Through feature selection, some irrelevant and redundant features are removed, so the simplified data sets often get more accurate models and are easier to understand. In this paper, we give a general framework of feature selection, as shown in Figure 2.

A feature selection algorithm is mainly composed of four parts: generation strategy, evaluation criteria, stop condition and conclusion. The generation strategy refers to generate some feature subsets from the original feature set, while the evaluation criteria means to evaluate the rationality and relevance of feature subsets. Moreover, the stop condition is to determine whether the feature subsets in accordance with initial requirements while conclusion means the validity of feature subsets.

We give the presentation of some popular feature selections including Information Gain, Improved Gini index, Document Frequency and Orthogonal Centroid.

1) Information gain: Information gain is a widely used algorithm in the field of machine learning. The Information Gain of a given feature t_k with respect to the class c_i is the reduction in uncertainty about the value of c_i when the value of t_k is known. The larger Information Gain of a feature is, the more useful the feature is for classification. Information Gain of a feature t_k toward a classification c_i can be defined as

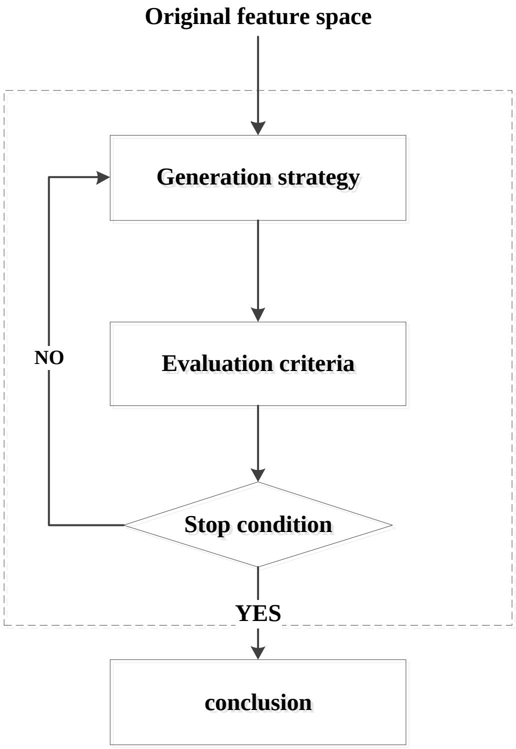

Figure 2: Framework of feature selection

follows:

$$IG(t_k, c_i) = \sum_c \sum_t P(t,c) log \frac{P(t,c)}{P(t)P(c)} \tag{3}$$

where $P(c)$ is the fraction of the documents in category c over the total number of documents. is the fraction of documents in the category c that contain the word t over the total number of documents. $P(t)$ is the fraction of the documents containing the term t over the total number of documents.

2) Improved gini index: Improved Gini index measures the purity of feature t_k toward a classification c_i. The larger the value of purity is, the better the feature is. The formula of the improved Gini index can be calculated as follows:

$$IGI(t_k) = \sum_i P(t_k|c_i)^2 P(c_i|t_k)^2. \tag{4}$$

Where, $P(t_k|c_i)$ is the probability that the feature t_k occurs in category ci. $P(c_i|t_k)$ refers to the conditional probability that the feature t_k belongs to category c_i when the feature t_k occurs.

3) Document frequency: Document frequency is a simple and effective feature selection algorithm that computes the number of documents that contain a feature. The main idea of this algorithm is that if a feature appears in a small number of texts, it is not useful for classification and may even reduce the classification performance. Therefore, the features which possess high document frequency need to be preserved. The formula of Document Frequency can be calculated as follows:

$$DF(t_k, c_i) = P(t_k|c_i). \tag{5}$$

4) Orthogonal centroid: The orthogonal centroid firstly computes the centroid of each category and the whole training set. Then the score of feature is calculated according to the centroid of each class and entire training set. The larger the score of the feature is, the more classification information the feature contains. The formula of orthogonal centroid can be described as follows:

$$OC(t_k) = \sum_{i=0}^{|C|} \frac{n_i}{n} (m_i^k - m^k)^2. \tag{6}$$

Where n_j is the number of documents in the category c_j, is the total number of documents in the training set, m_j^k is the kth element of the centroid vector m_j of category c_j, m^k is the kth element of the centroid vector m of entire training set, $|C|$ refers to the total number of categories in the corpus.

2.3 Feature Term - Classification Matrix

At present, common features selection algorithm is based on Vector Space Model (VSM) and taken into account the property of a features term in a classification, which is called as feature term—classification matrix. In this matrix, row represents feature term in vector space and column represents classification. The property, such as frequency of a feature term in certain classification can be represented by corresponding element value in the matrix. Table 1 shows a feature term—classification matrix, where the value expresses the frequency.

In the table, for example, the frequency of Home in C5 is 111 and other feature terms have low frequency in C5, so Home can represent the C5.

Table 1: Feature term - Classification matrix

Feature term	**C1**	**C2**	**C3**	**C4**	**C5**
Home	80	27	11	0	11
Products	5	155	21	0	98
Plan	7	7	0	79	36
Projects	2	0	145	19	1
Design	3	0	6	65	0

3 Unified Description of Vulnerability Factors

Nowadays, different business and institutions possess their own vulnerability database, and the same vulnerability in different database may have different factors. It is not convenient for us to determine which database to choose and which factor to opt. Therefore, it is necessary to formulate the unified vulnerability text factors to enhance the applicability.

3.1 Common Vulnerability Database

CVE is a well-known, widely recognized vulnerability database [3]. Every vulnerability gets a standard name, so it is easy to share data in all kinds of vulnerability database and vulnerability assessment tools. Therefore, we can find the security vulnerabilities of software products more quickly and effectively and give the solution to avoid the threat.

X-FORCE, belonging to ISS, has the most complete the vulnerability items [11]. However, it cannot publish the vulnerability free. ISS offers the online search service.

US-CERT is a middle-class vulnerability database from the Computer Emergency Response Team, and it is built in the Carnegie Mellon University [8]. Also, it can provide online search service.

3.2 Select Unified Factor

The selection of the vulnerability unified factor is the foundation of vulnerability Classification. According to some factors from different database, three institutions selecting the factors for vulnerability classification are shown in Table 2.

Therefore, we select four factors for the unified standards, where the number of the factors selected is three times. It shows that these factors are recognized as the representative attributes in the world. Actually, the factor Date Public is just the time and it has no use for us to vulnerability classification.

Ultimately, we use three factors, CVE name, severity rank and description to express vulnerability. We give an example in Table 3.

4 Algorithm Design

4.1 Algorithm Idea

Orthogonal Centroid Algorithm firstly calculates the centroid of all features in each class and the training set and then calculates the score. We can find that orthogonal centroid algorithm focuses on inter-class, namely calculating the most important feature term compared with other feature terms in one classification. Document frequency is a simple and effective feature selection algorithm. However, Document Frequency method only measures the significance of a feature term in the intra-class. Thus the Document Frequency method concentrates on the column of the feature term-classification matrix while Orthogonal Centroid Algorithm focuses on the row.

Both Document Frequency method and Orthogonal Centroid Algorithm just focus on one respect of the matrix. Therefore, this paper puts forward a kind of new feature selection algorithm, Orthogonal Centroid Features Selection algorithm both in Inter-class and Intra-class (OCFSII), which can make up deficiency of Orthogonal Centroid Algorithm and Document Frequency method and measure comprehensively the importance of a feature term to classification.

4.2 Algorithm Flow

As is shown in Figure 3, we give the flow chart of OCFSII algorithm, which mainly includes two parts, including the construction of feature term-classification matrix and selection of text feature. Text feature selection needs to calculate the centroid of training set. Moreover, we calculate the offset of feature terms both in inter-class and intra-class respectively. Finally, we can obtain the total offset of feature terms and then make a rank for those.

Here, feature term-classification matrix is $V_{T\times C}$, which consists of Tfeatures and Cclasses, matrix element v_{ij} represents the frequency of the ith feature in the jth class, the vector $D = \{d_1, d_2 \cdots d_i\}, 1 \leq i \leq C$, where d_i represents text number of the ith class. There are some calculation formulas as following:

1) Feature term centroid in training set $M = \{m^1, m^2, \cdots m^i\}$

$$m^i = \sum_{j=1}^{C} v_{ij} / \sum_{j=1}^{C} d_j \qquad (7)$$

Where m^irepresents the ith feature term centroid;

2) Feature term centroid in inter-class $M_j = \{m_j^1, m_j^2 \cdots m_j^i\}$

$$m_j^i = \frac{v_{ij}}{d_j} \qquad (8)$$

Where, m_j^i represents the centroid of the ith feature term in the jth class;

3) Feature term centroid in intra-class

$$\bar{m} = \frac{\sum_{j=1}^{C} v_{ij}}{C} \qquad (9)$$

4.3 Algorithm Description

Algorithm 1 OCFSII algorithm

```
Input feature term - classification matrix $V_{T\times C}$ and matrix element $v_{ij}$represents frequency of the $i$th feature in the $j$th class; text number vector of class vulnerability$D = \{d_1, d_2 \cdots d_i\}, 1 \leq i \leq C$; feature number$K$
Output feature subset $V_S$
$m_i = F_1(v_{ij}, d_i)$ // calculate feature term centroid in training set
$m_j^i = \frac{v_{ij}}{d_j}$ // calculate feature term centroid in inter-class
$\bar{m} = F_2(V_{ij}, C)$ // calculate feature term centroid in intra-class
for $i = 1$ to T
    for $j = 1$ to C
        $a_{ij} = v_{ij} - \bar{m}$ // calculate offset in intra-class
        $b_{ij} = m_j^i - m^i$ // calculate offset in inter-class
    end for
    $OCFSII_{ij} = a_{ij} * b_{ij}$
end for
$V_S = OCFSII_{TOPK}$
```

Moreover, the function F1 and F2 are defined as follows:

OCFSII algorithm measures the importance of features both in inter-class and intra-class. And it is so simple to implement. The time complexity is O(T*C) - namely the product of the number of rows and columns in Feature Term-classification Matrix.

Table 2: Institution selecting the factors for vulnerability classification

ID	Factor	CVE	X-FORCE	US-CERT	Times
1	CVE name	✓	✓	✓	3
2	Data public	✓	✓	✓	3
3	Date-up	×	×	✓	1
4	Severity rank	✓	✓	✓	3
5	Credit	×	×	✓	1
6	Solution	×	×	✓	1
7	Description	✓	✓	✓	3

Table 3: CVE-2015-1611 information

CVE name	description	Severity rank
CVE-2015-1611	OpenFlow plugin for Daylight before Helium SR3 allows remote attackers to spoof the SDN topology and affect the flow of data, related to fake LLDP injection.	Middle (CVSS score: 5.0)

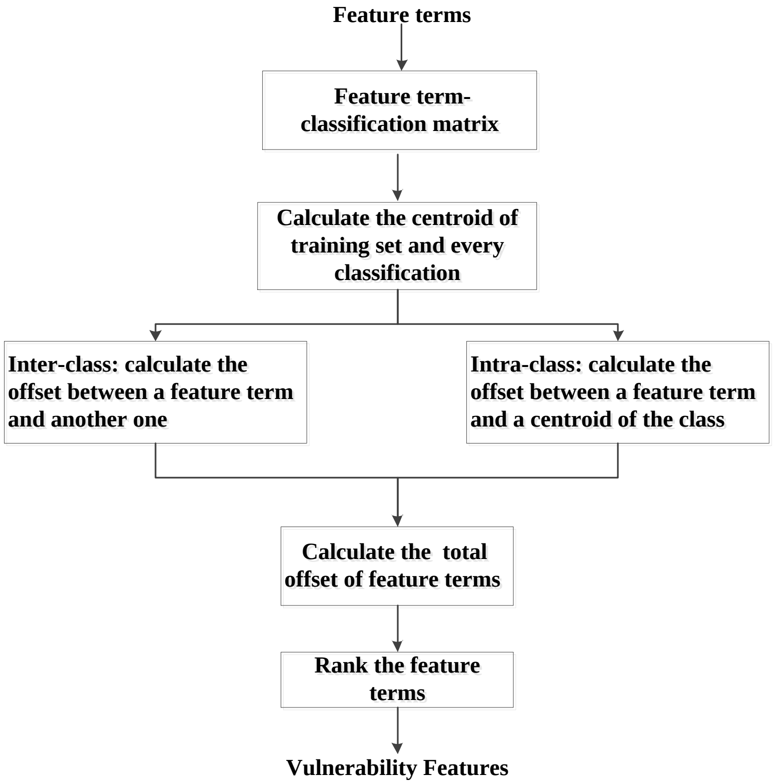
Feature terms
Feature term-classification matrix
Calculate the centroid of training set and every classification
Inter-class: calculate the offset between a feature term and another one
Intra-class: calculate the offset between a feature term and a centroid of the class
Calculate the total offset of feature terms
Rank the feature terms
Vulnerability Features

Figure 3: Flow chart of OCFSII algorithm

5 Experiment Setup

5.1 Experimental Classifier

In this section, K-Nearest Neighbor (KNN) and Support Vector Machine (SVM) are described briefly. Both of them are supervised learning method.

1) KNN classifier: KNN classification is a kind of learning algorithm based on sample, which is considered as an inert method. This algorithm shows wonderful performance in many applications. The key point of this method is to find a proper similarity measure to determine the degree of similarity between sample and training set. Therefore, we can get the nearest training set from the unmarked samples.

2) SVM classifier: SVM is a kind of machine learning algorithm, which is widely used in machine learning.

Algorithm 2 $F_1(v_{ij}, d_i)$

```
$v_{ij} = 0; d_j = 0$
for $i = 1$ to T
    for $j = 1$ to C
        $v_{ij} = v_{ij} + 1$
        $d_j = d_j + 1$
        $m_i = \frac{v_{ij}}{d_{ij}}$
    end for
end for
return $(v_{ij}, d_j)$
```

Algorithm 3 $F_2(v_{ij})$

```
$v_{ij} = 0$
for $i = 1$ to T
    for $j = 1$ to C
        $v_{ij} = v_{ij} + 1$
        $\bar{m} = \frac{v_{ij}}{C}$
    end for
end for
return $(v_{ij})$
```

Moreover, SVM is a high efficient classifier in classification. In our study, we choose liner kernel SVM.

5.2 Experimental Data

The purpose of this experiment is to select the feature of vulnerability, so as to verify the accuracy and efficiency of the vulnerability classification. In addition, we do not give a profound study on the selection of the categories. The vulnerabilities are divided into the most common six categories, authentication, buffer errors, cross-site scripting, code injection, information leak and input validation respectively. The sample set of vulnerabilities is shown in Table 4.

We select 3500 vulnerabilities from Security Content Automation Protocol (SCAP), from which 3000 vulnerabilities belong to training sample and 500 vulnerabilities belong to test training. As is seen from the Table 4, 3000 samples will train the classifier alter the feature selection and, the 500 samples are utilized to test the accuracy of OSFCII.

5.3 Experimental Steps

In this section, we give the concrete the steps of vulnerability classification experiment.

1) Obtain the original vulnerability features via preprocessing the vulnerabilities text from the database;

2) Construct the feature term- class matrix;

3) Get the feature terms of each category by utilizing the proposed feature selection OCFSII;

4) Use VSM to quantify the vulnerability feature terms;

5) Utilize one-to-many method to structure the vulnerability classifier;

6) Calculate the F1 and accuracy and give the experiment results.

5.4 Performance Measures

In our experiment, we utilize the F1 and Accuracy to measure the performance of the vulnerability classification.

1) Precision and micro-precision: Precision is the ratio of the number of vulnerability texts which are correctly classified as the positive class to the total number of those which are classified as the positive class. The formula of the precision for class c_i is defined as:

$$P_i = \frac{TP_i}{TP_i + FP_i} \tag{10}$$

Where TP_i is the number of vulnerability texts which are correctly classified as class $c_i$and FP_i means the number of vulnerability texts which are misclassified as class.

Similarly, in order to evaluate the performance average across the classes and micro-precision is used in this paper. The formula of the micro-precision can be calculated:

$$P_{micro} = \frac{TP}{TP + FP} = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} (TP_i + FP_i)} \tag{11}$$

Where $|C|$ is the number of the classes.

2) Recall and micro-recall: Recall is the ratio of the number of vulnerability texts which are correctly classified as the positive class to the total number of those which are actually belong to the positive class. The formula of the precision for class c_i is defined as:

$$R_i = \frac{TP_i}{TP_i + FN_i} \tag{12}$$

Where FN_i means the number of vulnerability texts belonging to class c_i are misclassified to other classes.

Similarly, in order to evaluate the performance average across the classes and micro-precision is used in this paper. The formula of the micro-precision can be calculated:

$$R_{micro} = \frac{TP}{TP + FN} = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} (TP_i + FN_i)} \tag{13}$$

3) F1 and accuracy: When we obtain the micro-precision and micro-recall, the formula of the F1 and Accuracy can be calculated:

$$F_1 = \frac{2P_{micro}R_{micro}}{P_{micro} + R_{micro}} \tag{14}$$

Table 4: Experimental vulnerability sample

Category	Number		
	Training sample	Testing sample	The total
authentication	221	30	251
buffer errors	303	80	383
cross-site scripting	945	200	1145
code injection	830	110	940
information leak	250	30	280
input validation	451	50	501

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (15)$$

Where TN means the number of vulnerability texts which are correctly classified to other classes excluding the positive class.

6 Results

6.1 Experimental Results when KNN Classifier

Table 5 shows the F1 measure results when KNN classifier is used. In this chart, we can see that OCFSII actually has the best performance when the number of features is 300, 500, 1300 and 1500. Moreover, the IGI has the similar performance compared with OCFSII but the latter is superior to the former. All of the algorithms have the positive correlation when the number of the features between 300 and 1300 and from 1300 on, the F1 measure begins to decrease. Therefore, the number of the features is 1300 for the database and the experiments can get the excellent results.

Similarity, Figure 4 shows the accuracy measure results when KNN classifier is used. In this graph, OCFSII and IGI have the better results compared with other methods. And all the curves rise from 300 to 1300 and decline later. It proves that when the number of features is 1300, and we can get the good results. OCFSII has the greatly improved when we consider the importance of features both in inter-class and intra-class compared with OC.

6.2 Experimental Results when SVM Classifier

Table 6 shows the F1 measure results when SVM classifier is used. In this chart, it can be seen that F1 measure results when utilized OCFSII outperforms any other methods. Although IGI has the similar performance compared with OCFSII, the latter precedes the former a little. Similarly, all of the algorithms have the positive correlation when the number of the features between 300 and 1300 and from 1300 on, the F1 measure begins to decrease. So, we can select 1300 features for the database approximately to obtain the good results. Compared with KNN classifier used in the experiment, SVM classifier performs better when we use the same methods.

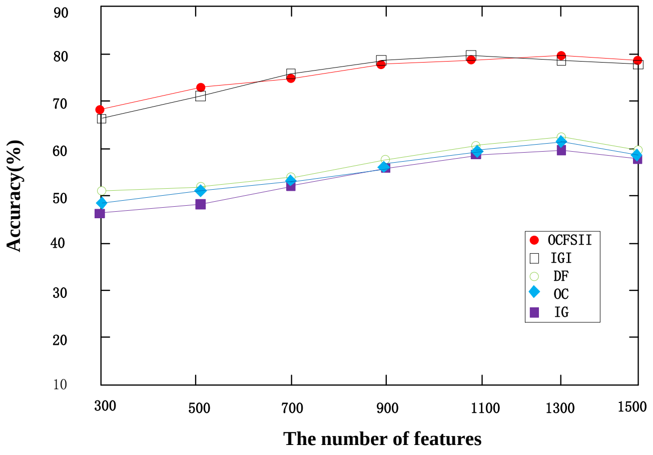

Figure 4: Accuracy measure curve using KNN classifier (%)

Similarity, Figure 5 shows the accuracy measure results when SVM classifier is used. In this graph, OCFSII has the better results compared with other methods. All of the curves ascend gradually with the increasing of the number of features, and they reach the highest point when the number is 1300. It tells us that we can get the excellent performance when we select 1300 features approximately. Obviously, OCFSII has the greatly improved compared with OC because both inter-class and intra-class are taken into consideration. Compared with KNN classifier used in the experiment, SVM classifier outperforms when we use the same methods.

7 Conclusion

In order to protect the information and network from the numerous numbers of the vulnerabilities, it is significant to manage the vulnerabilities. Classification, as a key link of vulnerability management, plays a major role in this whole process. Due to some feature selection method just consider the importance of the feature term from one aspect, we proposed a new feature selection algorithm called

Table 5: F1 measure results using KNN classifier (%)

The number of feature	300	500	700	900	1100	1300	1500
OCFSII	69.22	72.54	74.65	76.70	77.76	78.55	77.21
IG	47.88	49.43	52.12	55.23	57.12	58.21	57.33
DF	50.32	52.88	54.76	57.45	59.23	60.52	59.21
IGI	68.54	71.21	74.87	76.92	77. 99	78.32	77.10
OC	49.83	51.43	53.43	56.32	58.22	59.43	58.45

Table 6: F1 measure results using SVM classifier (%)

The number of feature	300	500	700	900	1100	1300	1500
OCFSII	71.54	73.85	75.81	78.60	79.76	80.21	79.21
IG	50.39	51.66	53.94	56.43	57.99	58.32	57.10
DF	52.47	54.18	56.85	58.21	60.23	61.76	60.53
IGI	70.35	72.31	75.32	78.10	78.54	79.21	78.29
OC	51.43	53.57	55.22	57.47	59.22	60.25	59.64

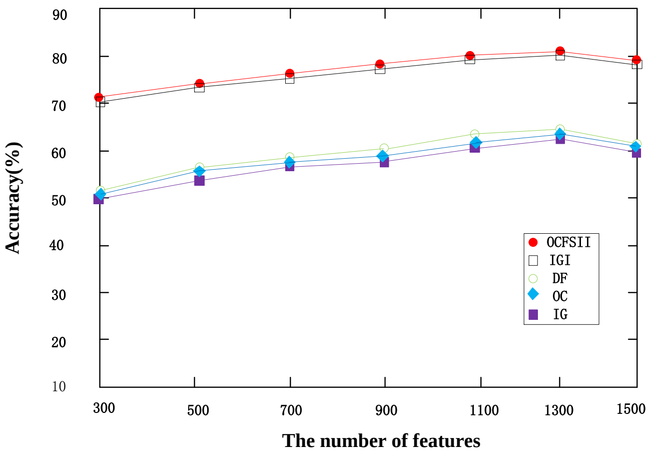

Figure 5: Accuracy measure curve using SVM classifier (%)

OCFSII, considering the importance of the feature term both in inter-class and intra-class. To confirm the validity of this method, we use two classifiers including SVM and KNN in our experiment, and compare it with four feature selection algorithms including Information Gain, IGI, Document Frequency and Orthogonal Centroid. The experiment results show that the proposed method OCFSII outperforms IG, DF OC, and is comparable with IGI when KNN used while OCFSII is superior to IG, DF, OC and IGI when SVM used. As part of our future research, we plan to design the better method to improve the accuracy and efficiency to enhance the understanding of vulnerability essence. [8]

References

[1] B. Bigi, “Using kullback-leibler distance for text categorization,” *Lecture Notes in Computer Science*, vol. 2633, pp. 305–319, 2016.

[2] Z. Chen, Y. Zhang, and Z. Chen, “A categorization framework for common computer vulnerabilities and exposures,” *The Computer Journal*, vol 53, no. 5, pp. 551-580, 2010.

[3] S. Christey and R. A. Martin, “Vulnerability type distributions in CVE,” *The MITRE Corporation*, 2007. (`http://cwe.mitre.org/documents/vuln-trends/index.html`)

[4] A. K. Gupta and N. Sardana, “Naive bayes approach for predicting missing links in ego networks,” in *IEEE International Symposium on Nanoelectronic and Information Systems*, pp. 161–165, 2017.

[5] K. M. A. Hasan, M. S. Sabuj, and Z. Afrin, “Opinion mining using naive bayes,” in *IEEE International Wie Conference on Electrical and Computer Engineering*, pp. 511–514, 2016.

[6] H. J. Kim, J. Kim, and J. Kim, “Semantic text classification with tensor space model-based naive bayes,” in *IEEE International Conference on Systems, Man, and Cybernetics*, pp. 004206–004210, 2017.

[7] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins, “Text classification using string kernels,” *Journal of Machine Learning Research*, vol. 2, no. 3, pp. 419–444, 2002.

[8] P. Minarik and T. Dymacek, “Netflow data visualization based on graphs,” in *Proceedings of Visualization for Computer Security, International Workshop*, pp. 144–151, 2008.

[9] S. A. Mingoti and J. O. Lima, “Comparing som neural network with fuzzy -means, -means and traditional hierarchical clustering algorithms,” *European*

Journal of Operational Research, vol. 174, no. 3, pp. 1742–1759, 2006.

[10] E. U. Opara, O. A. Soluade, "Straddling the next cyber frontier: The empirical analysis on network security, exploits, and vulnerabilities," *International Journal of Electronics and Information Engineering*, vol. 3, no. 1, pp. 10–18, 2015.

[11] F. H. Schmitz, "Reduction of blade-vortex interaction (BVI) noise through x-force control," *Journal of the American Helicopter Society*, vol. 43, no. 1, pp. 14–24(11), 1995.

[12] J. Song, J. Han, D. Zhang, L. Yuan, and L. Shao, "Evaluation of security vulnerability severity based on cmahp," in *IEEE International Conference on Computer and Communications*, pp. 1056–1060, 2017.

[13] H. S. Venter, J. H. P. Eloff, and Y. L. Li, "Standardising vulnerability categories," *Computers & Security*, vol. 27, no. 3-4, pp. 71–83, 2008.

[14] J. A. Wang and M. Guo, "Vulnerability categorization using bayesian networks," in *Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research (CSIIRW'10)*, pp. 1–4, 2010.

[15] J. Yan, N. Liu, B. Zhang, S. Yan, Z. Chen, Q. Cheng, W. Fan, and W. Y. Ma, "Ocfs: Optimal orthogonal centroid feature selection for text categorization," in *International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 122–129, 2005.

[16] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *Fourteenth International Conference on Machine Learning*, pp. 412–420, 1997.

[17] P. Zhang and X. Y. Xie, "Research on vulnerability classification based on svm with fuzzy entropy feature selection algorithm (in Chinese)," *Application Research of Computers*, vol. 32, no. 4, pp. 1145-1148, 2015.

Biography

Jialiang Song received the B.S. degree in electronic science and technology from Zhengzhou information science and technology institute, Henan, China, in 2015. He is pursuing the M.S. degree in information security at Zhengzhou information science and technology institute. His research interests include vulnerability management and information security.

Jihong Han received the B.S., M.S. and Ph.D degrees in information security from Zhengzhou information science and technology institute, Henan, China, in 1983, 1990, and 2008, respectively. She is now a professor at the Department of Information Security, Zhengzhou information science and technology institute. Her research interests include information hiding, watermarking and software reliability. She has published 60 research articles and 3 books in these areas.

Xirui Zhang received the B.S. degree in management from Zhengzhou information science and technology institute, Henan, China, in 2016. He is pursuing the M.S. degree in information security at Zhengzhou information science and technology institute. His research interests include information security and data mining.

Lulu Shao received the B.S., M.S in information security from Chongqing communications college, Chongqing, China, in 2004 and 2011. She is pursuing the Ph.D degree in information security at Zhengzhou information science and technology institute. Her research interests include wireless communication security.

Yang Wang received the B.S. degree in computer science and technology from Henan University, China, in 2014. He is pursuing the M.S. degree in computer science and technology at Zhengzhou information science and technology institute. His research interests include software reverse analysis and network security.