A Novel Automatic Severity Vulnerability Assessment Framework

Tao Wen[1], Yuqing Zhang[2,1], Ying Dong[2], and Gang Yang[2]
[1] State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China;
[2] National Computer Network Intrusion Protection Center, University of Chinese Academy of Sciences, Beijing 101408, China
Email: wentao_beijing@126.com; {zhangyq, dongy, yang}@nipc.org.cn

Abstract —Security vulnerabilities play an important role in network security. With the development of the network and the increasing number of vulnerabilities, many Quantitative Vulnerability Assessment Standards (QVAS) was proposed in order to enable professionals to prioritize the most important vulnerabilities with limited energy. However, it is difficult to apply QVAS manually due to the large number of vulnerabilities and lack of information. In order to address these problems, an Automatic Security Vulnerability Assessment Framework (ASVA) is proposed, which can automatically apply any QVAS to special Vulnerability Databases. ASVA obtain values of the metrics of a QVAS with new features of Text Mining; assign these values to a formula of QVAS and finally compute the severity values of the vulnerabilities. New features proposed in ASVA are special combinations of metrics of QVAS, so that consider the influence of metrics each other and improve the accuracy of Text Mining. Based on ASVA, CVSS as a QVAS is applied to a representative Vulnerability Database. The results show that ASVA reduces the cost and period of the application of QVAS and promotes the standardization of security vulnerability management.

Index Terms—Vulnerability assessment, vulnerability database, vulnerability, information security, ASVA, text mining

I. Introduction

Security vulnerability plays a central role in network security. If vulnerabilities are exploited by attackers, they could destroy the confidentiality, the integrity and the availability of the system [1]. Therefore, when vulnerabilities appear in the system, they need to be patched so that they cannot be exploited by attackers. Unfortunately, risks are still in the process of the vulnerability fixing, such as the close of some important function or the crash of the system. However, the vulnerabilities that we patched under these risks are always scarcely utilized by attackers or cannot do harm to the system when utilized by attackers. Since the vulnerability patching is time-consuming, so administrators and patch management systems need to balance the risk which comes from the patching of vulnerabilities and the risk which comes from the utilizing of vulnerabilities [2]-[4]. Therefore, with the development of the network and the increasing number of vulnerabilities, vulnerability severity assessment becomes more and more important [5]-[7].

In the past ten years, lots of IT vendors assessed the severity of vulnerabilities of their products, for example, Microsoft [8], Oracle [9] and RedHat [10] and so on. However, each Vulnerability Assessment Standard (VAS) cannot be unified. At the same time, some of security organizations, such as Secunia [11], Symantec [12], and OSVDB [13] also developed their own Vulnerability Databases and VASs. However, the results of these security organization VASs cannot be shared and even mutually contradicted sometimes. In fact, we do need a unified VAS on vulnerabilities not only from different IT vendors, but also from different Vulnerability Databases [14].

In 2003, National Infrastructure Advisory Council (NIAC) proposed the Common Vulnerability Scoring System (CVSS) [15] which is a Quantitative Vulnerability Assessment Standard (QVAS). Compared with other VASs, CVSS is objective, authoritative and transparent, so CVSS is approved of by IT managers, security organizations, IT vendors and researchers. CVSS promotes the standardization of the vulnerability assessment to a great extent [16].

In the process of CVSS, values of several given metrics are needed manually, and then assign these values to a given formula and compute the severity value of the vulnerability. In the past years, CVSS is further improved [17]-[18]. Currently, the formula which is given by CVSS is effective and the metrics given by CVSS can reflect the severity of vulnerabilities accurately, so if we can get the objective values of these metrics, then we will obtain an objective severity value of the vulnerability [19].

A. *Problems of the Existing Categorizations*

Although CVSS has so many advantages, it is only used by National Vulnerability Database (NVD) [20], since it is difficult to apply CVSS to any Vulnerability Database. The reasons lie in following challenges:

- The number of published vulnerabilities is huge and growing fast. So it is a huge amount of work if we assess these vulnerabilities again.
- Before CVSS is proposed, we never know what the metrics of CVSS are. Therefore, even if the security

Manuscript received March 4, 2015; revised May 13, 2015.
This work was supported by The Natural Science Foundation of China under Grant No.61272481.
Corresponding author email: wentao_beijing@126.com.
doi:10.12720/jcm.10.5.320-329

organizations want to spend the effort to assess thousands of vulnerabilities they have collected in the past ten years, unfortunately, the key metrics which are used to compute CVSS severity values cannot be obtained since the founders of vulnerabilities cannot be contacted.

- Even if the person who determines the key metrics of vulnerabilities is the founder of the vulnerabilities, he may not give the correct values of the key metrics. For example, a metric of CVSS is the complexity of exploiting a vulnerability. Complexity of exploiting is low for a founder; however, it may be high in comparison with exploit other vulnerabilities. The root of the problem lies in that the founder does not know the complexity of the other tens of thousands of vulnerabilities so he cannot give objective assessment.
- There are only simple descriptions and affected products on newly public vulnerabilities and the metric values which are needed by CVSS are always not given [21]. After a new vulnerability is published, the information about it will be completed within several days or months. During this period, the severity of vulnerabilities cannot be assessed since the information about them is insufficient; however, the severity of the vulnerabilities should be knowable urgently in this period.

Because of the challenges mentioned above, not only CVSS cannot be applied in practice easily, but all the QVASs at present will meet these challenges either. In order to address these problems and give improvement of existing works, considering Text Mining has the powerful function in automatically finding the laws of the historical information and foreseeing the unknown information, we propose a new automatic framework based on Text Mining.

B. Contribution

The idea of this paper is choosing a QVAS first, and then obtaining the metric values using Text Mining, finally computing the severity values and ranks of vulnerabilities using these metric values. The framework we proposed in this paper is not a QVAS but an application process of existing QVAS. Essentially, our new framework is suitable for every QVAS. The goals of this framework are to reduce the cost and period of the application of a QVAS, and promote the standardization of security vulnerability management. The contributions of this paper are as follows:

- We propose a new framework termed Automatic Security Vulnerability Assessment Framework (ASVA) based on Text Mining. With ASVA, we can compute the severity values and ranks of vulnerabilities using any QVAS, in the condition of insufficient information, such as we do not know the conditions of authentication and confidentiality impact. The steps in the assessment are automatic, and millions of vulnerabilities can be assessed within several days. What's more, as the assessment process based on statistics on a large number of vulnerabilities, ASVA avoid the manual subjectivity.
- In order to obtain more effective features of Text Mining and improve the accuracy, we propose three modes, i.e., Direct Mode, Original Mode and Combined Mode. With these modes, we can take into account both the original metrics and the associations among original metrics of QVAS. Meanwhile, in order to further improve the accuracy, we propose the rule of mode mixture, i.e., Mixed Mode, which can combine the assessment results of the three modes.
- In order to further obtain more effective features of Text Mining, we propose the rule of metric combination for Combined Mode, which optimizes the selection strategy of metric combination. With this rule, we can not only improve the accuracy, but also analyze the associations among the metrics of QVAS. Based on this rule, we analyze the associations among the metrics of CVSS from a new angle of view, and explain the reasons for these associations.
- We collect and collate two representative Vulnerability Databases, namely NVD [22] and OSVDB [13], [23], all of which contain about 160 thousand of vulnerabilities totally. Then, we apply CVSS as a QVAS (NVD adopted only) to other non-NVD Databases. We adopt three dimensions (accuracy, coverage rate and dispersity) to analyze the results, results show that the accuracy of severity rank is 90.1% and the dispersity is perfect. Finally, we explain the reason of errors in vulnerability assessment.

C. Organization

The rest of this paper are as follows. In Section 2, we review the related works in vulnerability assessment. In Section 3, we introduce the idea and the process of ASVA. Then, in Section 4, we give the detailed process of ASVA. In Section 5, we give the experiment results. Finally, in Section 6 we provide the conclusion and future work.

II. Related Work

A. Vulnerability Assessment Standard (VAS)

IT Vendors have established Vulnerability Database which is used to public vulnerabilities for their own products and assessed vulnerabilities [24], such as Microsoft and Cisco [25] and so on. Because each vendor only records their own products so the users cannot make a comparison of severity among different vendors. In this situation, researchers and the security organizations proposed Vulnerability Databases and VASs, such as NVD [20] and Symantec and so on. Each Vulnerability Database collects vulnerabilities of the products from different vendors. VASs are usually divided into three types.

- Experience Type. The professionals assess vulnerabilities, according to their experience [26]-[27].

The advantage of this type is easy operation and strong feasibility so it is the most widely used type.

- Voting Type. The assessment of vulnerabilities is determined both by users and professionals. According to certain weights, professionals from Vulnerability Database institutes and users of Vulnerability Database vote the severity ranks of vulnerabilities together. The Vulnerability Databases of Mozilla [28] and Wooyun [29] take advantage of this type. However, the result will be affected by the number of voters and their professional levels. Therefore, this type is uncontrollable and indeterminate.
- Quantification Type. The value of quantitative VAS is computed referring to several key metrics of the standard. At first, the QVAS fixes several key metrics of the vulnerability, then assigns these values to a given formula, finally the severity ranks of the vulnerability is computed [30]-[32]. Compared with other types, this type greatly reduces the subjectivity in the vulnerability assessment (although the subjectivity is not eliminated totally). Usually, this type needs to show: (a) the key metric values of QVAS to vulnerability; (b) the formula according to which the severity value is computed using these metric values; (c) how many ranks that the vulnerabilities are divided into; (d) the thresholds which map values to ranks.

B. Common Vulnerability Scoring System (CVSS)

Quantification type, i.e. QVAS, is a high-grade type. However, the difficulty lies in how to choose the key metrics and define the formula [33]. Lots of researchers turn to this topic in recent years [32][34]. The most successful QVAS is Common Vulnerability Scoring System (CVSS). However, in the past few years, the CVSS Standard is further improved according to the new case of vulnerabilities and millions of vulnerabilities in history [35]-[36].

Six metrics need to be obtained first in the computation process of the CVSS value, see Table 1, the metric column represents the six metrics. The metric value column represents the possible situations of the metrics and the corresponding values. For example, Access Vector (Av) has three kinds of values, (local, adjacent network and network). When Av is local, the value of Av is 0.395, and so on. The meaning of the six metrics is shown in the documentation of CVSS [15]. After obtaining the values of these six metrics, assign them to formula (1)-(4), then the CVSS severity value is computed, which is the final value of CVSS.

TABLE I: THE METRIC VALUES OF THE CVSS

Metric	Metric Value
Access Vector (Av)	*Local (L) = 0.395, Network (N) = 1.0, Adjacent Network (A) = 0.646*
Access Complexity (Ac)	*Medium (M) = 0.61, Low (L) = 0.71, High (H) = 0.35*
Authentication (Au)	*Multiple (M) = 0.45, Single (S) = 0.56, No (N) = 0.704*
Confidentiality Impact (C)	*None (N) = 0.0, Partial (P) = 0.275, Complete (C) = 0.660*
Integrity Impact (I)	*None (N) = 0.0, Partial (P) = 0.275, Complete (C) = 0.660*
Availability Impact (A)	*None (N) = 0.0, Partial (P) = 0.275, Complete (C) = 0.660*

$$CVSS\ Severity\ Value = \big((0.6 * Impact) + (0.4 * Exploitability) – 1.5\big) * f(Impact) \tag{1}$$

$$Impact = 10.41 * (1 - (1 - C) * (1 - I) * (1 - A)) \tag{2}$$

$$Exploitability = 20 * \ AV * AC * Au \tag{3}$$

$$f(impact) = \ 0\ if\ Impact = 0, 1.176\ otherwise \tag{4}$$

C. Automatic Security Vulnerability Assessment Framework

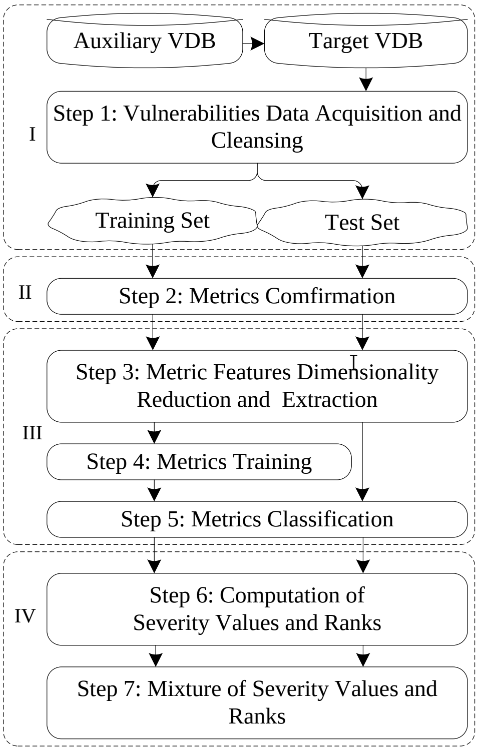
The main problem of CVSS is that it is difficult to apply it. Because the number of vulnerabilities is on a sharp increase, determining the values of metrics manually is time-consuming and subjective [37]. Furthermore, the metric values may not be obtained due to the lack of information about vulnerabilities sometimes. Therefore, some researchers [38]-[39] have proposed automatic vulnerability assessment frameworks respectively which can automatically assess the severity of vulnerabilities with specified QVAS. However, since these frameworks are not based on Text Ming, they must assess the vulnerabilities one by one, and cannot in batch, so the efficiency is as low as manual assessment. Chani *et al.* [32] proposed an automatic vulnerability assessment framework based on the scheme of Linear Discriminant Analysis (LDA) in the situation of lacking the information about vulnerabilities. However, more than half values of metrics of CVSS were needed.

In order to address these problems and give improvement of existing works, considering Text Mining has the powerful function in automatically finding the laws of the historical information and foreseeing the unknown information, we propose a new automatic vulnerability assessment scheme termed ASVA based on the supervised learning theory. As a comparison to the works of Chani [32], ASVA do not need to know any values of metrics.

III. Idea and Process of ASVA

In this section we introduce the general ideal, realization process and main algorithms of Automatic Security Vulnerability Assessment Framework (ASVA).

A. *The Process of ASVA*



Auxiliary VDB
Target VDB
I
Step 1: Vulnerabilities Data Acquisition and Cleansing
Training Set
Test Set
II
Step 2: Metrics Comfirmation
III
Step 3: Metric Features Dimensionality Reduction and Extraction
Step 4: Metrics Training
Step 5: Metrics Classification
IV
Step 6: Computation of Severity Values and Ranks
Step 7: Mixture of Severity Values and Ranks

Fig. 1. The process of ASVA

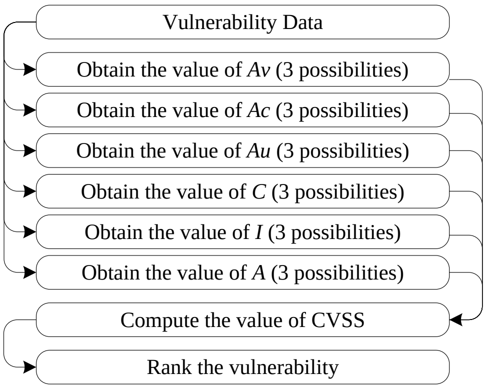
The realization process of ASVA framework consists of four stages, see Fig. 1. However, before the process, we should select the QVAS, adopted by Auxiliary Vulnerability Database (Auxiliary VDB), and the Target Vulnerability Database (Target VDB) which will be assessed.

- Stage I includes Step 1. The task of Stage I is acquiring and cleansing data. We will obtain the one-to-one corresponding vulnerabilities between Auxiliary VDB and Target VDB as Training Set, on the condition of that the Target VDB do not contains enough marked items which had been ranked.
- Stage II includes Step 2. The task of Stage II is determining the classification metrics. ASVA contains three modes, all of them have different metrics.
- Stage III includes Step 3, Step 4 and Step 5. The task of Stage III is classifying metrics with Text Mining and obtaining the values of metrics.
- Stage IV includes Step 6 and Step 7. The task of Stage IV is assigning values of metrics to formulas of the QVAS and computing the severity values and severity ranks of vulnerabilities.

B. *Three Modes in ASVA*

The framework proposed in this paper has three modes. When assessing vulnerabilities, each mode can be used. Modes are selected in Step 2 to determine metrics. The three modes are one of our contributions which brings higher accuracy of the vulnerability assessment. We will take CVSS as an example to show these modes: Direct Mode, Original Mode and Combined Mode.

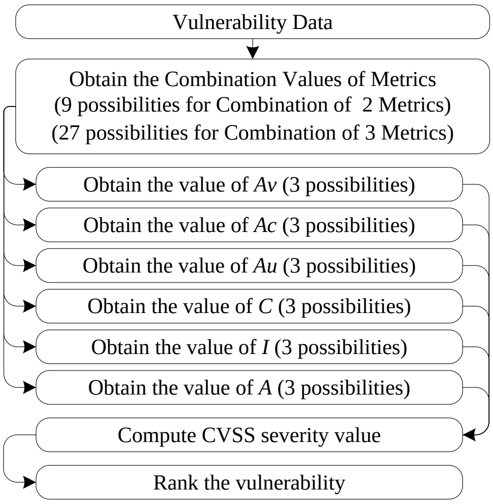
Mode 1: Direct Mode. Divide vulnerabilities using the Text Mining into Low, Medium and High directly. Because Mode1 classifies the ranks directly so we entitle Mode 1 the Direct Mode.



Vulnerability Data
Obtain the value of *Av* (3 possibilities)
Obtain the value of *Ac* (3 possibilities)
Obtain the value of *Au* (3 possibilities)
Obtain the value of *C* (3 possibilities)
Obtain the value of *I* (3 possibilities)
Obtain the value of *A* (3 possibilities)
Compute the value of CVSS
Rank the vulnerability

Fig. 2. The mode 2 of CVSS

Mode 2: Original Mode. Classify every original metric using the Text Mining (CVSS has six metrics) first. Because there are three possible values in each metric, so we divide each metric into three categories. Then convert these categories to corresponding values and compute a severity value using each metric value. Finally, divide the vulnerabilities into Low, Medium and High according to the severity value. Fig. 2 shows the Mode 2 of ASVA. Because Mode 2 classifies each original metric of CVSS separately, so we entitle Mode 2 Original Mode.



Vulnerability Data
Obtain the Combination Values of Metrics
(9 possibilities for Combination of 2 Metrics)
(27 possibilities for Combination of 3 Metrics)
Obtain the value of *Av* (3 possibilities)
Obtain the value of *Ac* (3 possibilities)
Obtain the value of *Au* (3 possibilities)
Obtain the value of *C* (3 possibilities)
Obtain the value of *I* (3 possibilities)
Obtain the value of *A* (3 possibilities)
Compute CVSS severity value
Rank the vulnerability

Fig. 3. The mode 3 of CVSS

Mode 3: Combined Mode. Because the metrics of Mode 3 are a combination of metrics of QVAS so we entitle Mode 3 Combined Mode. Metrics are not entirely independent, but impacted by each other. Effect of Mode 3 is to consider the relevance among the metrics.

Mode 3 also needs to obtain the metric values first. Then compute the severity value using each metric value. The difference between Mode 2 and Mode 3 is that Mode

3 does not classify original metrics, but combines the metrics first then classify the combined metrics using Text Mining. Then, according to the results of classification, we can extract the values of all six original metrics. Fig. 3 shows the Mode 3 of CVSS and the detailed process see Step 2.

C. *Mixed Mode in ASVA*

When the three modes obtain severity values and ranks, we compare the results of the three modes and extract the same result. For example, we assess vulnerability X using three modes. There may be a situation that all of the ranks of the three modes are High; there may be the metric values from two modes are High and the other one is Medium; or there may be totally different results from Mode 2 and Mode 3 in the six metrics of CVSS. The Mixed Mode will discuss all kinds of possibilities and concludes the better practical possibilities. In this paper, we discuss five situations of Mixed Mode, see Table II

TABLE II: FIVE POSSIBLE METHODS OF MIXED MODE

Name	Mode Definition
X1	Rank(Mode 1) = Rank(Mode 2); Rank of Mode 1 is the same as that of Mode 2.
X2	Rank(Mode 1) = Rank(Mode 3); Rank of Mode 1 is the same as that of Mode 3.
X3	Rank(Mode 2) = Rank(Mode 3); Rank of Mode 2 is the same as that of Mode 3.
X4	Rank(Mode 1) = Rank(Mode 2) = Rank(Mode 3); Rank of Mode 1, Mode 2 and Mode 3 are the same.
X5	Metric(Mode 2) = Metric(Mode 3) = 6; All of the key Metrics of Mode 2 are the same as that of Mode 3

IV. IMPLEMENTATION OF ASVA

In this section, we introduce each step in Fig. 1 in detail with OSVDB as Target VDB, NVD as Auxiliary VDB and CVSS as QVAS.

A. *Step 1: Vulnerability Data Acquisition and Cleansing Vulnerabilities Data Acquisition*

This step aims at obtaining the data of vulnerabilities, which is used for assessment. In ASVA, we aim at assessing the vulnerabilities using the CVSS, so we need to obtain the data of the vulnerabilities in Target VDB first. We take the vulnerabilities which have one-to-one relationship between Auxiliary VDB and Target VDB as Training Set since CVSS is adopted by NVD. It means we need to know the real metric values of vulnerabilities in the Training Set. In practice, another example, if we want to apply a QVAS adopted by Vulnerability Database Secunia to Vulnerability Database EDB, we need to find the vulnerabilities which have one-to-one relationship between Secunia and EDB and take these vulnerabilities as Training Set.

1) *Vulnerability data processing*

The vulnerability data which is obtained from the Target VDB needs to be pre-treated since it cannot be used in Text Mining directly. The purpose of data cleansing is to get the data in regular forms. The steps of data cleansing are as follows:

- Segment the words. The classification features mainly come from the text of the fields of Description, Title and Affected Vendors. These fields are separated and stored in the form of single word vectors.
- Remove special symbols, such as the commas, periods, brackets and line breaks and so on. Worth mentioned, not all of the special symbols are removed, for instance, the brackets which represent the function call need to be reserved, such as “save()” which is the name of a function in a source code, if we remove “()”, then the original meaning of it will be changed.
- Remove the words without effective information. In order to increase the accuracy, we need to remove some words which do not have effective information related to categorization, for example, the words whose length is 1; the pure digital words, such as 1234; the words which represent the versions, such as x1.8 and 2.0.
- Remove stop words, such as “are” and “what” and so on, since these words have no real effect on the categorization of vulnerabilities.
- Deal with the tense and grammar. Change the passive voice and plural into the original form. After being cleansed, the data will be the vector of strings which are stored in the form of a single word.

B. *Step 2: Metrics Confirmation*

In this paper, we take advantage of CVSS with six key metrics (*Av*, *Ac*, *Au*, *C*, *I* and *A*, each metric also has three possible values) and three ranks (*High*, *Medium* and *Low*).

- Metric confirmation of Mode 1. Mode 1 assesses vulnerabilities directly. So only one metric needs to be computed, i.e., rank of severity. It means the Text Mining classifies vulnerabilities into three categories directly, i.e. High, Medium and Low.
- Metric confirmation of Mode 2. CVSS includes six original metrics, and each metric contains three cases. For example, C has three cases: None, Partial and Complete, so we use Text Mining to classify each metric into three categories.
- Metric confirmation of Mode 3. In Mode 3, we combine the six original metrics of CVSS. There are 15 combinations when we put two original metrics into one group, see Table III, and there are 20 combinations when put three into one group, see Table IV.

Each combination of the metrics is used as the new classification metric. Because each original metric has three states, for example, *Av* has three states: L_{Av}, N_{Av} and A_{Av}; and *Ac* has three states: M_{Ac}, L_{Ac} and H_{Ac}, see Table I. *AvAc*, as the combination of *Av* and *Ac*, has 9 states: $L_{Av}M_{Ac}$, $L_{Av}L_{Ac}$, $L_{Av}H_{Ac}$, $N_{Av}M_{Ac}$, $N_{Av}L_{Ac}$, $N_{Av}H_{Ac}$, $A_{Av}M_{Ac}$, $A_{Av}L_{Ac}$, $A_{Av}H_{Ac}$. Therefore, *AvAc* should be divided into 9 categories. The combination of three metrics should be divided into 27 categories.

TABLE III: THE COMBINATIONS OF TWO METRICS

AvAc	*AvAu*	*AvC*	*AvI*	*AvA*
AcAu	*AcC*	*AcI*	*AcA*	*AuC*
AuI	*AuA*	*CI*	*CA*	*IA*

TABLE IV: THE COMBINATIONS OF THREE METRICS

AvAcAu	*AvAcC*	*AvAcI*	*AvAcA*
AvAuC	*AvAuI*	*AvAuA*	*AvCI*
AvCA	*AvIA*	*AcAuC*	*AcAuI*
AcAuA	*AcCI*	*AcCA*	*AcIA*
AuCI	*AuCA*	*AuIA*	*CIA*

Mode 3 should achieve the following purposes in other steps, a. determining values of each combination by Data Mining; b. obtaining the values of original metrics, for instance, the value of *AvAc* is $L_{Av}M_{Ac}$, so the value of *Av* is *L* and the value of *Ac* is *M*; c. determining accuracy of original metrics from each combination, for instance, the accuracy of *Av* in *AvAc* or the accuracy of *Av* in *AvAu*; d. determining the highest accuracy of combination original metric is in, for instance, the accuracy of *Av* in *AvAc* is higher than other combination *Av* is in, the accuracy of *AvAc* is highest to *Av*.

C. Step 3: Metric Feature Dimensionality Reduction and Acquisition

1) Metric features dimensionality reduction

In Text Mining, a single word is usually used as a dimension. The frequency of a word is the value of the dimension. For example, we assume the word sequence of the vulnerability *O1* is: *word*, *SQL*, *computer*, and the word sequence of the vulnerability *O2* is: *SQL*, *computer*, *function*. Let *O1* and *O2* be a set, then features include four dimensions: *word*, *SQL*, *computer* and *function*, where the value of *O1* is *{1,1,1,0}*, value of *O2* is *{0,1,1,1}*.

Description and Affected Vendors of vulnerabilities consist of a huge number of words. So dimensionality curse will occur when we classify the text if we do not deal with the words. Such a large dimension is not realistic. Therefore, dimensionality reduction is needed. The frequently-used dimensionality reduction algorithm is Document Frequency (DF) [40], Information Gain (IG) [41], Mutual Information (MI) [42] and Chi-square (CHI) [43] at present. Among them, IG and CHI are better [44]. However, the problem of IG is that it can only investigate the features which contribute to the whole system, i.e., the global feature. It cannot choose the features against individual categories [45]. So IG is only suitable in the case that the number of each category is close to each other. However, the vulnerability numbers of each category are significantly different respectively in this paper, so if we use IG algorithm the result will not be perfect. Therefore, we use the CHI algorithm as the dimensionality reduction algorithm. The specific calculation formula is shown in equation (1).

$$\chi^2 = \frac{N(AD-BC)^2}{(A+C)(A+B)(B+D)(B+C)} \qquad (5)$$

In the formula, *N* denotes the total number of documents in the statistical sample set, *A* denotes the frequency of occurrence of some word's positive document, *B* denotes the frequency of occurrence of some words' negative document, *C* denotes the frequency of non-occurrence of some words' positive document, *D* denotes the frequency of non-occurrence of some words' negative document. Every unique word (i.e., a feature dimensionality) needs to compute a value χ^2 against a category, for example, if there are 1000 dimensionalities, then 3000 values are computed. These values are in descending order. A certain number of these values are chosen as the features of this category. However, if the features we chose are too many, then the computation complexity will be increased and if the features we chose are too few, then the accuracy and the coverage rate will be decreased.

2) Metric feature extraction

The taxonomic features we use are the fields of Description, Title and Vendor of the vulnerability entries in Target VDB. We find nearly all of the Vulnerability Databases have the fields of Description, Title and Vendor. It means the universality of these three fields is the best. So we extract features Set from them. Then apply the CHI dimensionality reduction algorithm, extract the first n feature words of each category from the mixed word sequence.

Through Experiments, we draw the conclusions that accuracy, coverage rate and time-consumption increase with the increase of the number of features; however, accuracy and coverage rate do not increase after the feature number is greater than 200. So, we select 200 as the feature number.

D. Step 4&5: Training and Classification of Metrics

Compared with the Bayes algorithm [46] and the Random Forests algorithm [47], SVM algorithm [48] used in this paper has high accuracy but is time-consuming. Considering the requirement of time we need is lower than that of accuracy in this paper, so we use SVM algorithm as the classification algorithm.

SVM algorithm maps the sample space to feature space with high dimension by nonlinear mapping. The mapping transfers the nonlinear and separable problem in the original sample space transformed into the linear and separable problem in the feature space. Then different kernel functions generate different SVM. Frequently-used kernel functions are: (1) Liner; (2) Polynomial; (3) Radial; (4) Sigmoid.

The kernel function we use is Radial. After data cleansing and feature selection, the Training Set gets the feature vectors as the input of SVM. Through training, the Training Set, we get the optimal weight and mode. Then we can test the Test Set and compute categorization of metrics in the Test Set.

E. Step 6&7: Computation and Mixture of Severity Values and Ranks

In this step, we assign values of metrics to formulas of VAS and compute the severity values of vulnerabilities. After that, we map severity values to given ranks of severity according to the thresholds.

Take CVSS for example, assign the six metric values into formulas from (1) to (4) and compute CVSS value. The range of CVSS value is from 0.0 to 10.0. The rank of the vulnerability is Low if CVSS value is less than or equal to 4, it is Medium if CVSS value is greater than 4 and less than or equal to 7 and it is High if CVSS value is greater than 7.

The steps mentioned above are directed against to Mode 2 and Mode 3. There is no CVSS value in Mode 1 since Mode assesses the rank of vulnerabilities directly.

Mixed Mode is the mixture of Mode 1, Mode 2 and Mode 3 according to the rules mentioned in C. Mixed Mode needs to make overall consideration of Mode 1, Mode 2 and Mode 3.

V. The Experiment Result and Analysis

In the implementation of our system, the operating system is Windows 7, the database system is Microsoft SQL Server and algorithm implementation is the package e1071 of R language. The memory of needs to be larger than 8G.

A. Vulnerability Data Acquisition

TABLE V: THE NUMBER OF VULNERABILITIES PROCESSED

Vulnerability Database	The number of vulnerabilities obtained	Training Set	Test Set
OSVDB	98252	5400	600
NVD	65200	--	--
Total	163452	70600	600

We should obtain Vulnerabilities which have corresponding ones in NVD. The aim of our experiment is to apply the CVSS which is used by NVD to Target VDB, so we select NVD as Auxiliary Database. We select OSVDB as Target VDB. We can easily know the rank of parts of vulnerabilities in Target VDB according to NVD. In practice, all of equivalent vulnerabilities in NVD are classified into Training Set. In order to verify the accuracy of the classification algorithm, we only choose vulnerabilities, which are in one-to-one correspondence between Auxiliary VDB and Target VDB as Training Data. The purpose is to get the accuracy objectively. In this paper, We collect and collate the four Vulnerability Databases, which contain 300 thousand of vulnerabilities totally, the number of vulnerabilities we obtained are shown in Table V. All of the data obtained up to 2013 November.

B. Analysis of Metric Combinations of Mode 3

Table III and Table IV list all of the combined metrics, we compute the values of all these combinations with Text Mining. Then, we determine the accuracy of each original metric in each combination, and determine the highest accuracy of combination.

TABLE VI: THE HIGHEST ACCURACY OF COMBINATION

Original Metric	Combination	The Highest Accuracy
Av	*AvA*	94.12%
Ac	*AcI*	75.63%
Au	*AcAu*	89.58%
C	*CIA*	82.35%
I	*CIA*	84.71%
A	*CIA*	81.51%

Table VI shows the highest accuracy of combination for each original metric, for instance, the highest accuracy of combination is 94.12% for *Av*, the combination is *AvA*. According to Table VI, we can draw the conclusion that:

(a) The metrics of correlation cannot correspond to each other. For example, the highest accuracy of *Av* appears in *AvA*, this suggests that *A* has the most significant influence on the value of *Av*. But the opposite is not true, since the highest accuracy of *A* appears in *CIA*, instead of *AvA*. (b) The affected relationships among original metric values are the reverse of affected relationships among actual metrics. For instance, the metrics of *Av*, *Ac* and *Au* can affect *C*, *I* and *A* in practice; however, the values of *Av*, *Ac* and *Au* are affected by *C*, *I* and *A*.

The relationships among six metrics of CVSS are as follows, see Fig. 4:

- The value of Av is most affected by A. This is because Av reflects how the vulnerability is exploited, and A measures the impact on the availability of a successfully exploited vulnerability. The denial of service attack may appear in the situation that the value of Av is Network (N), at this time, the value of A is Complete (C).
- The value of Ac is most affected by I. This is because Ac measures the complexity of the attack required to exploit the vulnerability once an attacker has gained access to the target system, and I measures the impact on the integrity of a successfully exploited vulnerability. Complex attacks can always damage the integrity of the system.

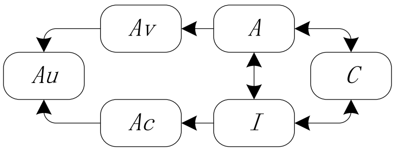


Fig. 4. The relationship among six metrics of CVSS

- The value of Au is most affected by A. This is because Au measures the number of times that an attacker must be authenticated in order to exploit a vulnerability. The more times of authentication, the more complexity of an attack.

C. Accuracy and Coverage Rate of Modes

In this subsection, we discuss the accuracy and coverage rate of Rank in each mode, conclude 3 modes and 5 possible Mixed Mode aforementioned, see Table II.

From Fig. 5 and Fig. 6 we can see, in comparison with Mode 1, Mode 2 and Mode 3, the accuracy of Mode 1 is the highest which is 84.2% and the coverage rates of the three modes are all 100%. In all of the modes, the highest accuracy appears in X5, which is 90.45%; however, the coverage rate of it is only 70.0%. X3 is a compromised choice whose accuracy is 88.0%, which is higher than that of Mode 2. Meanwhile, the coverage rate of X3 is higher than that of X5. The accuracy and the coverage rate of X3 is higher than that of X1 and X2. The accuracy and the coverage rate of X5 is higher than that of X4. So X3 and X5 are optimal in Mixed Mode. Therefore, we conclude three modes which have better effect on the comparison among the modes, i.e., Mode 3, X3 and X5.

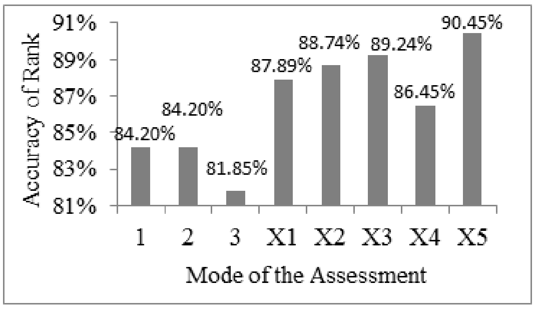


Fig. 5. Rank accuracy comparison among different modes

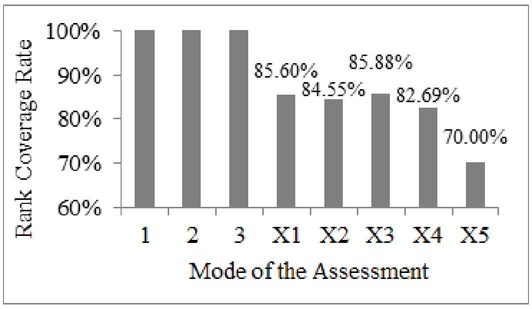


Fig. 6. Rank coverage rate comparison among different modes

D. Value Dispersity of Modes

In this subsection, we discuss the dispersity of CVSS severity values. Since Mode 1 has no CVSS severity value and Mixed Mode is attached to Mode 1, Mode 2 and Mode 3, therefore we only need to compare between Mode 2 and Mode 3.

The comparison among different modes is shown in Table VII. In this paper, the higher dispersity of CVSS severity value is, the better the assessment will be.

- Dispersity of Variance. Variance denotes the dispersity of a set of values. From Table VII we can see, dispersity of Mode 3 is the higher which means the CVSS severity values are evenly dispersed. The computation formula of Variance is shown in equation (6), where E represents the mean value of CVSS severity values.

$$Variance = \frac{1}{n}\sum(CVSS - E)^2 \tag{6}$$

$$E = \frac{1}{n}\sum CVSS \tag{7}$$

- Dispersity of Value Number. This is another indicator which reflects dispersity whose value denotes the number of included CVSS severity values. For example, the range of CVSS values is from 0.0 to 10.0 and it includes 101 numbers of values such as 0.0, 0.1 and 1.1. There are only 33 CVSS values which are computed from Mode 2. From Table VII we can see, there are 45 CVSS values which are in the range of CVSS in Mode 3, which means the dispersity of CVSS severity values is better in Mode 3.

TABLE VII: DISPERSITY OF CVSS SEVERITY VALUES

Mode	Variance	Value Number
2	3.85	33
3	4.14	45

Considering Variance and Value Number, we can draw the conclusion that dispersity of Mode 3 is better than Mode 2.

E. Explanation of Error Ranks

In this subsection, we will discuss the reason why there are errors in QVAS.

Fig. 7 shows the distribution of the number of vulnerabilities on the CVSS severity values. The x-coordinate represents CVSS severity values, for example, “0” denotes the vulnerabilities whose CVSS values are lower than 1, “1” denotes the vulnerabilities whose CVSS values are from 1.0 to 1.9 and “10” denotes the vulnerabilities whose CVSS values are 10. The y-coordinate represents the number of vulnerabilities on the corresponding values. The y-coordinate contains the real number of vulnerabilities and the number of vulnerabilities from Mode 2 and Mode 3 on every interval of the value.

We can see the number of vulnerabilities is the largest if the CVSS severity values are close to 7 and there are also many vulnerabilities when the CVSS values are close to 4. The values of 4 and 7 are key values in our assessment. When there are three ranks, *High*, *Medium* and *Low*, the severity of a vulnerability is *High* if its CVSS severity value is greater than 7 and it is Low if its CVSS severity value is lower than 4. Therefore, it is easy to make the wrong assessment of rank when the CVSS Value of the vulnerability is close to 4 and 7. When the CVSS value is 4, the number of wrongly assessed vulnerabilities is large too. So a question arises that CVSS may have some defects which could lead to the result of the assessment be not objective and could CVSS be improved? So we can reduce the number of vulnerabilities in the key point.

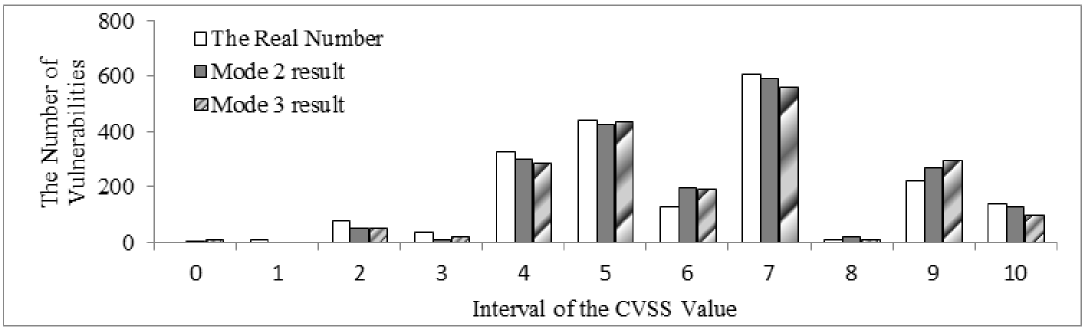


Fig. 7. Distribution of the number of vulnerabilities on CVSS severity values

VI. Conclusion and Future Work

The security vulnerability assessment standard cannot be spread manually since the number of vulnerabilities is huge and the information about vulnerabilities is lacking.

In this paper, we propose a new framework termed Automatic Security Vulnerability Assessment Framework (ASVA) based on Text Mining. With ASVA, we can apply any Quantitative Vulnerability Assessment Standard to a Vulnerability Database automatically.

Based on ASVA Framework, we propose three modes (Direct Mode, Original Mode and Combined Mode) and two import rules (the rule of mode mixture, and the rule of metric combination of Combined Mode) to improve the accuracy of ASVA.

We use ASVA and CVSS to assess a representative Vulnerability Databases (OSVDB). When the coverage rate is 100%, the rank accuracy is 82.5%; however, when the coverage rate is 70.0%, the rank accuracy is 90.45%. Meanwhile, the dispersity of Combined Mode is perfect.

Our future work will be research on finding new mode to further improve the accuracy of rank.

Acknowledgments

The authors wish to thank The Natural Science Foundation of China (61272481). The authors also wish to thank the anonymous reviewers for providing constructive suggestions.

References

[1] M. D. Aime and F. Guasconi, “Enhanced vulnerability ontology or information risk assessment and dependability management,” in *Proc. Third International Conference*, Venice, 2010, pp. 92-97.

[2] C. H. Lin, C. H. Chen, and C. S. Laih, “A study and implementation of vulnerability assessment and misconfiguration detection,” in *Proc. Asia-Pacific Services Computing Conference*, Yilan, 2008, pp. 1252-1257.

[3] K. Kim, J. Kang, and D. Lee, “PCChecker: Hardening windows security configurations,” in *Proc. International Conference on Convergence and Hybrid Information Technology*, Busan, 2008, pp. 1252–1257.

[4] U. J. Premaratne, T. Samarabandu, and Sidhu, *et al.*, “Application of security metrics in auditing computer network security: A case study,” in *Proc. International Conference on Information and Automation for Sustainability*, Colombo, 2008, pp. 448–453.

[5] M. A. Rahman and E. Al-Shaer, “A formal approach for network security management based on qualitative risk analysis,” in *Proc. International Symposium Integrated Network Management*, May 2013, pp. 244-251.

[6] M. Barrere, G. Hurel, R. Badonnel, and O. Festor, “Ovaldroid: An OVAL-based vulnerability assessment framework for Android,” in *Proc. International Symposium Integrated Network Management*, Ghent, 2013, pp. 1074-1075.

[7] J. A. Wang and M. Z. Guo, “Security Data Mining in an Ontology for Vulnerability Management,” in *Proc. International Joint Conference on Bioinformatics, Systems Biology and Intelligent Computing*, Shanghai, 2009, pp. 597-603.

[8] Microsoft. Microsoft TechNet: Resources for IT Professionals. [Online]. Available: http://technet.microsoft.com/en-ca/

[9] ORACLE. ORACLE Help Center. [Online]. Available: https://docs.oracle.com/en/

[10] RedHat. Red Hat Technologies. [Online]. Available: http://www.redhat.com/en/resources/securityclassification

[11] Secunia. Secunia Stay Security. [Online]. Available: http://secunia.com/advisories/historic/

[12] Symantec. Symantec Security Response - Severity Assessment. [Online]. Available: http://www.symantec.com/security_response/severityassessment.jsp

[13] OSVDB. Open Source Vulnerability Database. [Online]. Available: http://osvdb.org/

[14] C. Zheng, Y. Q. Zhang, Y. F. Sun, and Q. X. Liu, “IVDA: International vulnerability database alliance,” in *Proc. Cybersecurity Summit*, London, 2011, pp. 1-6.

[15] CVSS. Common Vulnerability Scoring System. [Online]. Available: http://nvd.nist.gov/cvss.cfm

[16] K. Scarfone and P. Mell, “An analysis of CVSS version 2 vulnerability scoring,” in *Proc. International Symposium on Empirical Software Engineering and Measurement*, Lake Buena Vista, 2009, pp. 516-525.

[17] P. Mell, K. Scarfone, and S. Romanosky, “Common vulnerability scoring system,” *Security & Privacy*, vol. 4, no. 6, pp. 85-89, November 2006.

[18] P. Mell and K. Scarfone, “Improving the common vulnerability scoring system,” *IET Information Security*, vol. 1, no. 3, pp. 119-127, September 2007.

[19] R. Y. Wang, L. Gao, and D. H. Sun, “An improved CVSS-based vulnerability scoring mechanism,” in *Proc. International Conference on Multimedia Information Networking and Security*, Shanghai, 2011, pp. 352-355.

[20] NVD. National Vulnerability Database. [Online]. Available: http://web.nvd.nist.gov/

[21] L. Y. Wang, S. Jajodia, A. Singhal, and P. S. Cheng, “k-Zero day safety: A network security metric for measuring the risk of unknown vulnerabilities,” *IEEE Transactions on Dependable and Secure Computing*, vol. 11, no. 1, pp. 30-44, June 2013.

[22] CVE. Common Vulnerabilities and Exposures. [Online]. Available: http://cve.mitre.org/

[23] SCAP. [Online]. Available: http://www.scap.org.cn/index.html

[24] J. L. Salmeron and I. Herrero, “An AHP-based methodology to rank critical success factors of executive information systems,”

Computer Standards and Interfaces, vol. 28, pp. 1-12, 2005.

[25] Cisco. Cisco Security. [Online]. Available: http://tools.cisco.com/security/center/home.x

[26] H. Holm, M. Ekstedt, and D. Andersson, "Empirical analysis of system-level vulnerability metrics through actual attacks," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 6, pp. 825-837, January 2012.

[27] T. L. Saaty, "A scaling method for priorities in hierarchical structures," *Journal of Mathematical Psychology*, vol. 15, pp. 234-281, 1977.

[28] Mozilla. Mozilla Foundation Security Advisories. [Online]. Available: https://www.mozilla.org/security/advisories/

[29] WooYun. WooYun.org. [Online]. Available: http://www.wooyun.org/bugs/

[30] G. Vache, "Vulnerability analysis for a quantitative security evaluation," in *Proc. International Symposium on Empirical Software Engineering and Measurement*, Lake Buena Vista, 2009, pp. 526-534.

[31] M. Keramati, A. Akbari, and M. Keramati, "CVSS-based security metrics for quantitative analysis of attack graphs," in *Proc. 3th International Conference Computer and Knowledge Engineering*, Mashhad, 2013, pp. 178-183.

[32] H. Ghani, J. Luna, and N. Suri, "Quantitative assessment of software vulnerabilities based on economic-driven security metrics," in *Proc. International Conference on Risks and Security of Internet and Systems*, La Rochelle, 2013, pp. 1-8.

[33] S. H. Houmb, V. N. L. Franqueira, and E. A. Engum, "Quantifying security risk level from CVSS estimates of frequency and impact," *Journal of Systems and Software*, vol. 2, no. 83, pp. 1622-1634, February 2010.

[34] J. Zheng, X. S. Zhang, and X. H. Pan, "A host deployed vulnerability assessment system based on OVAL," in *Proc.* International Conference *on Computer Application and System Modeling*, Taiyuan, 2010, pp. V2-123-V2-126.

[35] Q. X. Liu and Y. Q. Zhang, "VRSS: A new system for rating and scoring vulnerabilities," *Computer Communications*, vol. 34, pp. 246-273, 2011.

[36] Q. X. Liu, Y. Q. Zhang, and Y. Kong, "Improving VRSS-based vulnerability prioritization using analytic hierarchy process," *The Journal of Systems and Software*, vol. 85, pp. 1699-1078, August 2012.

[37] C. Fruhwirth and T. Mannisto, "Improving CVSS-based vulnerability prioritization and response with context information," in *Proc. International Symposium on Empirical Software Engineering and Measurement*, Lake Buena Vista, 2009, pp. 535-544.

[38] F. L. Guo, Y. Yu, and T. Chiueh, "Automated and safe vulnerability assessment," in *Proc. Computer Security Applications Conference*, Tucson, AZ, 2005, pp. 149-159.

[39] Y. Jun and S. Wontae, "Implementation of the automated network vulnerability assessment framework," in Proc. *4th International Conference on Innovations in Information Technology*, Dubai, 2007, pp. 153-157.

[40] D. Karakos, M. Dredze, K. Church, and A. Jansen, "Estimating document frequencies in a speech corpus," in *Proc. IEEE Workshop Automatic Speech Recognition and Understanding*, Waikoloa, HI, 2011, pp. 407-412.

[41] Z. Gao, Y. J. Xu, F. Y. Meng, and F. Qi, "Improved information gain-based feature selection for text categorization," in *Proc. International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems*, Aalborg, 2014, pp. 1-5.

[42] J. Y. Liang, X. P. Liu, K. N. Huang, and X. Li, "Automatic registration of multisensor images using an integrated spatial and mutual information (SMI) metric," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 1, pp. 603-615, March 2013.

[43] X. H. Niu, F. Shi, J. B. Xia, and X. H. Hu, "Comparisons among the novel measurements based on chi square criterion for sequence dissimilarity and their applications to phylogeny," in *Proc.* International Conference *on Computational and Information Sciences*, Shiyang, 2013, pp. 470-473.

[44] S. L. Liu, X. Chen, W. S. Liu, and J. Q. Chen, "FECAR: A feature selection framework for software defect prediction It cannot choose features against individual categories," in *Proc. Computer Software and Applications Confer*ence, Dallas, 2014, pp. 426-435.

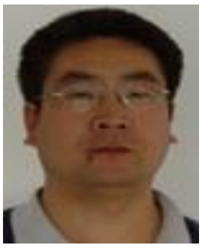
[45] C. Z. Yang, C. C. Hou, W. C. Kao, and I. X. Chen, "An empirical study on improving severity prediction of defect reports using feature selection," in *Proc. Software Engineering Conference*, Hong Kong, 2012, pp. 240-249.

[46] Y. G. Ji, S. N. Yu, and Y. F. Zhang, "A novel naive bayes model: Packaged hidden naive bayes," in *Proc. Information Technology and Artificial Intelligence Conference*, Chongqing, 2011, pp. 484-487.

[47] J. Lv and Z. G. Yan, "Estimating leaf chlorophyll concentration in soybean using random forests and field imaging spectroscopy," in *Proc. International Conference on Agro-geo informatics (Agro-geo Informatics)*, Beijing, 2014, pp. 1-4.

[48] X. M. Liu and J. S. Tang, "Mass classification in mammograms using selected geometry and texture features, and a new SVM-based feature selection method," *Systems Journal*, vol. 8, no. 3, pp. 910-920, November 2013.

Tao Wen is a postgraduate student of Xidian University, China. He received his B.Sc. in Fudan University, in 2007. He received his M.Sc. in Geosciences University, in 2011. Currently, he is studying security information and security vulnerability.



Yuqing Zhang is a professor and supervisor of Ph.D. students of Graduate University of Chinese Academy of Sciences. He received his B.Sc. and M.Sc. in Computer Science from Xidian University, China, in 1987 and 1990 respectively. He received his Ph.D. degree in Cryptography from Xidian in 2000. His research interests include cryptography, wireless security and trust management.

Ying Dong is a postgraduate student of Information Science and Engineering of Graduate University of Chinese Academy of Sciences, Beijing, China. She received her B.Sc. in Information Security from LanZhou University of China, in 2011. Currently, she is studying web application security.

Gang Yang is a postgraduate student of Information Science and Engineering of Graduate University of Chinese Academy of Sciences, Beijing, China. He received his B.Sc. in Wuhan University, in 2014. Currently, he is studying security vulnerability.