Proceeding of the IEEE
International Conference on Information and Automation
Yinchuan, China, August 2013

Automatic Classification for Vulnerability Based on Machine Learning*

Bo Shuai
School of Electronic Science and Engineering
National University of Defense Technology
Changsha, Hunan, P. R. China, 410073
shuaibo85@163.com

Haifeng Li, Mengjun Li, Quan Zhang, Chaojing Tang
School of Electronic Science and Engineering
National University of Defense Technology
Changsha Hunan , China
quanzhang@nudt.edu.cn

***Abstract* - In order to solve the problems of traditional machine learning methods for automatic classification of vulnerability, this paper presents a novel machine learning method based on LDA model and SVM. Firstly, word location information is introduced into LDA model called WL-LDA (Weighted Location LDA), which could acquire better effect through generating vector space on themes other than on words. Secondly, a multi-class classifier called HT-SVM (Huffman Tree SVM) is constructed, which could make a faster and more stable classification by making good use of the prior knowledge about distribution of the number of vulnerabilities. Experiments show that the method could obtain higher classification accuracy and efficiency.**

Index Terms* - *Vulnerability classification, Words location, Vulnerability distribution, LDA model, SVM

I. INTRODUCTION

Vulnerability is considered to be intentional or unintentional defects exist in information technology, products and systems. Once vulnerabilities are exploited by malicious body, the security of information systems will under a great risk. An increasing number of vulnerabilities results in great threats to reliability of information systems in recent years. For example, Aurora vulnerability of Microsoft causes a serious attack against Google in 2010. Website vulnerabilities of CSDN lead to a widely information leakage involving 6 million users in 2011. In 2012 the famous flame virus outbreak in the Middle East which utilized the digital signature spoofing vulnerability of Microsoft to disguise itself not to be found. Thus, the vulnerability analysis and research has become more and more important to protect network security.

Classification of vulnerabilities is the first and fundamental step in vulnerability analysis. The purpose of vulnerability classification is to makes it possible to enhance the understanding of vulnerabilities through constructing a hierarchical classification form. A reasonable vulnerability classification can help in analysing, identifying and eliminating vulnerabilities even in discovering unknown vulnerabilities.

Fortunately, many computer administrators and cyber security researchers have focused on vulnerability analysis and classification. In the early 70s there has been artificial construct vulnerability classification system RISOS, the project focuses on operating system vulnerabilities, and delimiting operating system security vulnerabilities into seven different types [1]. PA project is committed to discovery common vulnerabilities through a fixed pattern which consists of four categories and four small classes [2]. Aslam from the Prudure University makes a complete vulnerability classification model and vulnerability database for Unix operating system [3]. However, with the rapid growth in the number and types of new discovered vulnerabilities, the traditional manual method in classifying vulnerabilities exposed more and more disadvantages such as low accuracy and efficiency. Therefore, researchers pay more attention to automatic classification of vulnerabilities.

Automatic classification of vulnerability problem can essentially be seen as an application of pattern recognition, which could be solved in machine learning methods. Support Vector Machine (SVM) proved to be an effective machine learning approach has been applied to automatic vulnerability classification. Chen constructs a CVE classifier with SVM, which serves as a classification framework that can be applied to query and classify vulnerability [4]. LI makes an unsupervised vulnerability classification through SOM clustering method [5]. However, most existing methods are based on words Vector Space Model (VSM) model [6], which is unable to deal with the high dimensional and sparse issues. Meanwhile, these methods ignore the vulnerability specific feature information resulting in a decrease in classification accuracy.

In order to verify the effectiveness of automatic vulnerability classification method, we should choose a more mature vulnerability database research. National Vulnerability Database (NVD) includes a large number of vulnerabilities resources with comprehensive and detailed descriptions. Furthermore, NVD is famous for its standardization, normalization and authoritative, which is consistent with serials of standards such as Common Vulnerabilities & Exposures (CVE), Common Vulnerability Scoring System (CVSS), Common Platform Enumeration (CPE), Common Weakness Enumeration (CWE). Therefore, we choose NVD vulnerability database as the data source for automatic vulnerability classification.

In this paper, an intelligent approach is proposed based on Latent Dirichlet Allocation (LDA) model and SVM. LDA model is introduced into feature selection and integrate the location information into the feature words. The Weighted

* This work is partially supported by the National Natural Science Foundation of China Grant #61101073.

978-1-4799-1334-3/13/$31.00 ©2013 IEEE

Location LDA (WL-LDA) model could establish the generative model bases on weighted location themes which is better than on feature words. Then the multiclass SVM classifier is constructed using Huffman Tree arithmetic which could make good use of the prior knowledge of distribution of the number of vulnerabilities. The Huffman Tree SVM (HT-SVM) is trained based on the implicit topics–words matrix, which is obtained from the feature words probabilistic distributions of WL-LDA model. The approach combines the useful features of vulnerabilities, the outstanding feature dimensionality reduction and text representation capabilities of LDA with the powerful classification ability of SVM to improve the vulnerabilities classification performance.

The remainder of this paper is organized as follows. Section II points the two main shortcomings of the traditional methods in vulnerability classification. In section III, we briefly give the frameworks of our machine learning methods. Section IV illustrates the topic modeling process based on weighted location words. Section V elaborates the multiclass SVM constructing algorithm based on vulnerabilities distribution. Section VI presents the test results using NVD vulnerability database and compares with other methods. Section VII outlines conclusion and future network.

II. Issues Analysis

The traditional classification algorithm based on word vector space has achieved good results in many fields. However, when dealing with the vulnerability data which has characteristics of large-scale and short text, it still exhibits obvious shortcomings, mainly in the following two aspects.

1) Insufficient capacity in processing high dimensional and sparse data. Different with other data mining problems, the description of vulnerability is relatively short, usually no more than 250 words. On this basis, using the method based on word vector space will result not only a considerable high dimension of the vector but also serious sparse vectors, which brings great difficulties in vulnerability classification.

2) Inadequate use of vulnerability database characteristics. Since the vulnerability database is produced by related security experts and especially the description of the vulnerability usually has a fixed format, for example, some words in specific location commonly provide more useful information about how the vulnerability is caused, which is very useful for vulnerability classification. Furthermore, due to objective reasons, the distribution of the different types of vulnerabilities is not balance, which will also affect the performance of the classifier. However, such factors are not considered in the traditional methods.

III. Frameworks

The proposed method in the paper gives full consideration to the above two issues through combining WL-LDA model based on weighted location words with the HT-SVM based on vulnerabilities distribution. Thus, the overall framework of the method mainly consists of two parts: WL-LDA model and HT-SVM. The detailed procedures are shown in Figure 1.

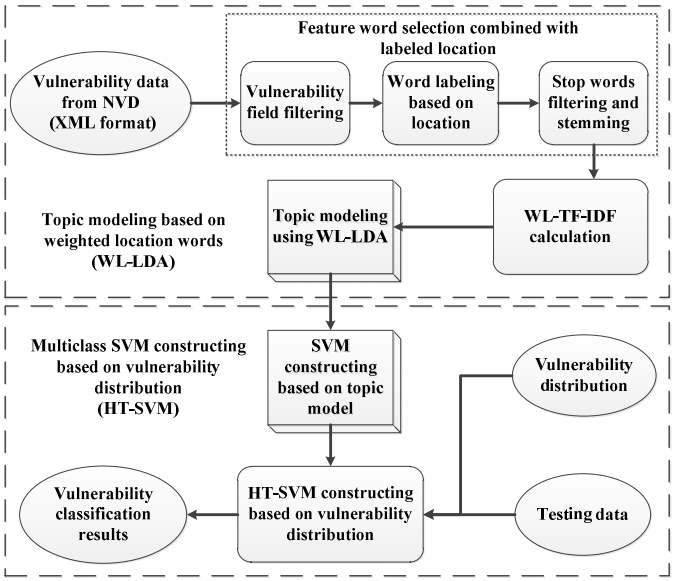

Fig. 1 Framework of the method

A. Topic modeling based on weighted location words

Firstly, NVD vulnerability data file in XML format will be transformed to a database file such as MDB format for convenient analysis. Then, the related vulnerability fields will be extracted according to need. Since different words in different location have different relations to classification, one description will be divided into several locations which will be labeled as different levels. Through eliminating the stop words and stemming, a labeled vulnerability vocabulary has been established. Secondly, an improvement is made to the classical TF-IDF formula through considering the effect of different level labeled above, which is called the Weighted Location TF-IDF (WL-TF-IDF) algorithm. Thirdly, with the help of topic model based on result calculated by WL-TF-IDF, the vulnerability information is mapped to the low dimensional topic vector space from the high dimensional word vector space successfully.

B. Multiclass SVM constructing based on vulnerability distribution

Firstly, the two-class SVM will be constructed based on topic model established above, using the weighted location topic vector matrix. Detailed procedures include training data, selecting kernel function, estimating parameters and testing data. Secondly, according to imbalance of the vulnerability distribution, a HT-SVM will be constructed through Huffman tree algorithm which makes the larger partition of sample categories access the root of the tree as close as possible. At this point, the classification of vulnerability has been completed.

IV. Topic Modeling Based on Weighted Location Words

A. Feature word selection combined with labeled location

1) Vulnerability field filtering

NVD database [7] provides comprehensive information of vulnerability such as CVE number, CWE number, summary , published date, CVSS SCORE, exploit location, exploit

degree of difficulty, harm on CAIA (confidentiality, authentication, integrity, availability) and other information. Up to March 2013, NVD database contained total of 54,992 vulnerability records, the average daily increment has reached more than a dozen vulnerabilities. Annual vulnerability increment of NVD (1996-2012) is shown in Figure 2.

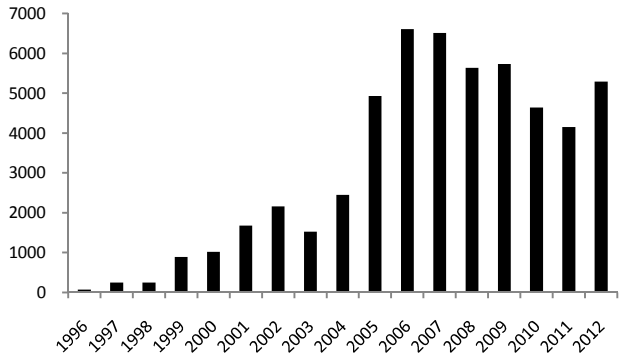

Fig. 2 Annual vulnerability increment of NVD (1996-2012)

According to the need for automatic classification of vulnerabilities, irrelevant fields and incomplete data will be removed, leaving only three related fields, such as CVE-id, summary, CWE-id, details are depicted in Table I. Summary field is the most important, which includes the cause of vulnerability or object exploited, vulnerable products, permission to exploit, consequence, exploit methods and other information.

TABLE I
LIST OF VULNERABILITY FIELD RESERVED

No.	Database field	Field Meaning
1	CVE-id	Vulnerability name accords with CVE standard
2	summary	Detailed description of vulnerability
3	CWE-id	Vulnerability type accords with CWE standard

2) Word labeling based on location

After examining the vulnerability records, we find out that the summary field of NVD has a fixed representation format. The purpose may be to ensure the readability and normative of NVD. For example, the vulnerability named "CVE-2013-0085", whose CWE-id field is "CWE-119" indicates the vulnerability type is "Buffer Errors". This is the summary fields: Buffer overflow in Microsoft SharePoint Server 2010 SP1 and SharePoint Foundation 2010 SP1 allows remote attackers to cause a denial of service (W3WP process crash and site outage) via a crafted URL, aka "Buffer Overflow Vulnerability."

As can be seen from the above information, the summary field is divided into five fixed portion through some fixed words such as "in, allows, to, via". We believe that different portion affects different influence on the classification. the related specific details are depicted in Table II. According to Table II, we can easily conclude that the "CVE-2013-0085" vulnerability may belong to "Buffer Errors" type, since "Buffer overflow" exists in the L5 location which plays the most important role for classification.

TABLE II
LOCATIONS AND THEIR SIGNIFICANCE IN SUMMARY FIELD

Label	Location Description		Meaning	Important Degree
	Begin	End		
L1	The first "in/on"	"allows"	Vulnerable products	1
L2	"via / by" after "Allows"	Field End	Exploit methods	2
L3	"allows"	the first one "to" after "Allows"	Permission to exploit	3
L4	the first one "to" after "Allows"	"via/by"	Consequence	4
L5	Field begins	The first "in/on"	Cause of vulnerability or object exploited	5

3) Stop words filtering and stemming

Stop words filtering and stemming are common steps in text pre-process. The common stop words list includes the typical article, auxiliary, conjunctions, pronouns, auxiliary verbs, prepositions, etc. Considering the feature of vulnerability database, some common words such as "allow, via, by, through, vulnerability" should be appended either. Stemming is another important pre-process in English which means to reduce the duplicate information. In English one word may have several forms such as changes in singular, plural nouns and so on. Thus the step of retaining only one copy is called "stemming". For example, words like "computer", "compute", "computing", "computational" all could be treated as "compute" when using the Porter Stemming algorithm[8] for stemming.

B. WL-TF-IDF calculation

TF-IDF [9] is calculated from three aspects to consider the contribution of the word in the document, namely: the word frequency TF, anti-document frequency IDF and the normalization factor, formula is shown below:

$$\mathrm{TF\_IDF}(\mathrm{W}_i, D_j) = \frac{\mathrm{TF}(\mathrm{W}_i, D_j) \times \mathrm{IDF}(\mathrm{W}_i)}{\left\|\mathrm{TF}(\mathrm{W}_i, D_j) \times \mathrm{IDF}(\mathrm{W}_i)\right\|} = \frac{\mathrm{TF}(\mathrm{W}_i, D_j) \times \log(\mathrm{N}/\mathrm{n}_i + 0.01)}{\sqrt{\sum_{i=1}^{n}\left[\mathrm{TF}(\mathrm{W}_i, D_j) \times \log(\mathrm{N}/\mathrm{n}_i + 0.01)\right]^2}} \quad (1)$$

$$\text{Where, } \mathrm{IDF}(\mathrm{W}_i) = \log\left(\frac{\mathrm{N}}{\mathrm{n}_i} + 0.01\right)$$

$\mathrm{TF}(\mathrm{W}_i, \mathrm{D}_j)$ indicates the frequency of word W_i exists in a document D_j, N represents the total number of documents, n_i indicate the number of documents which contain the word W_i. Equation indicates that if a feature word exists more frequently in a certain document and less frequently in others, then its TF-IDF value should be larger.

The TF-IDF algorithm does not reflect the location information of feature words, whereas, as mentioned above, words in different locations will make different contribution to

vulnerability classification. Therefore, we propose the location factor into TF-IDF algorithm for considering the affection of location in vulnerability classification, which is called WL-TF-IDF.

$$\mathrm{WL\_TF\_IDF}(\mathrm{W}_i, D_j) = \frac{\mathrm{WL\_TF}(\mathrm{W}_i, D_j) \times \mathrm{IDF}(\mathrm{W}_i)}{\left\|\mathrm{PTF}(\mathrm{W}_i, D_j) \times \mathrm{IDF}(\mathrm{W}_i)\right\|}$$

Where,

$$\begin{cases} \mathrm{WL\_TF}(\mathrm{W}_i, D_j) = \mathrm{TF}(\mathrm{W}_i, D_j) \times \sum_{k=1}^{5} \lambda_k \\ \lambda_k = \frac{1+k}{2} \end{cases} \quad (2)$$

k is the key factor which denotes the importance of location corresponding Label in Table II. When a word lays in the Li(i=1,2,3,4,5) location ,its WL_TF_IDF value will be calculated as k equals to the corresponding i. As can be seen in (2), the improved WL-TF-IDF formula is calculated based on the cumulative location factors, which means that if a word exists in a more important location, then its WL_TF_IDF value will be higher.

C. Topic modeling using WL-LDA

1) The LDA model

LDA [10] is a completely generating probabilistic model on discrete data set such as documents. Assuming that there are K independent implicit topics in the text set which include M documents, each topic is the polynomial probabilistic distribution of words, and each document is randomly generated by the K implicit topics.

We can understand the "generation behavior" by the following steps:

(1) Choose the parameter θ, it obey the distribution: θ～P(α);

(2) To consider the generation of word $w_{j,i}$ in document d;

a) First randomly choose a topic according to the distribution: z_j～P(z|θ);

b) Then to a certain topic z_j, randomly choose a word $w_{j,i}$ according the distribution: □$_Z$～P(w|z).

α is a 1×k random row vector and its specific function form is Dirichlet distribution. Here, θ is non-negative and normalized. z_j is a discrete random variable, P(z|θ) is the distribution of z when θ is given, it can directly regard θ as probability value. □$_Z$ is also a discrete random variable, P(w|z) is the distribution of w when z_j is given, we can take it as a k×|V| matrix β, V indicates a vocabulary table with size |V| . Through the above generative process, we can get the likelihood value of each document:

$$\mathrm{P}(\mathrm{w}, \mathrm{z}, \theta, \phi | \alpha, \beta) = \prod_{n=1}^{N} \prod P(w_i|\phi_{zi}) P(z_j|\theta)\, P(\theta|\alpha) P(\phi_z|\beta) \quad (3)$$

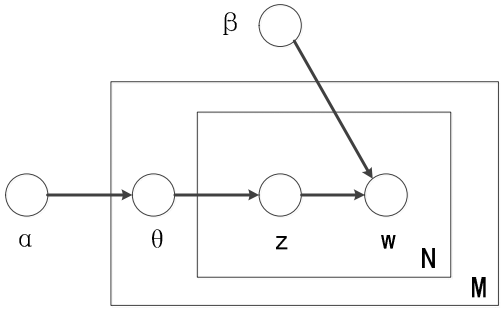

Fig. 3 Latent Dirichlet Allocation model

2) Gibbs sampling

After the discussion of LDA generative model above, we can conclude that the task is to obtain the specific form of LDA by studying the training sample,that is, to make sure the control parameter of model. There are two main methods for LDA parameters estimation: one is Varitional Inference in original LDA article; another is Gibbs sampling based on MCMC [11] (Markov Chain Monte Carlo) theory. MCMC provides the approximate iteration method of extracting sample value from the complex probabilistic distribution. The Gibbs sampling of MCMC is a straightforward realization method, and it aims to structure the Markov chain which converges to a target probabilistic distribution and extracts the specimen with probabilistic approximation. Consequently, composing the objective probabilistic distribution function is the key point of using the Gibbs sampling algorithm. In order to obtain the probabilistic distribution of the word layer, we consider the posteriori probability p(w/z) and take advantage of Gibbs sampling to gain the value of the posterior parameters □ and ψ indirectly. In LDA model, it only needs to sample the topic variables z_j.

3) Process of topic modeling

For the particularity of vulnerability classification, the weighted location information is combined with LDA model. The main steps of generating the topic model with WL-LDA are illustrated as follows:

(1) Assuming that the c category vulnerability sets include n vulnerabilities, the known fixed parameters K, M and N are imported.

(2) Converting vulnerability data into a bag of words format, which is a two-dimensional matrix. The longitudinal dimension is the vulnerability name, transversal dimension is word indices in vocabulary, the values of each element in the matrix is calculated through WL-TF-IDF algorithm. After the format conversion, location information is introduced into LDA model.

(3) Confirming the optimal topics K with the standard method of Bayesian statistical theory. The optimal topics K will be confirmed when the value best fit the effective information of corpora.

(4) The model parameters will be estimated with the Gibbs sampling algorithms for the purpose of confirming the value of parameters α and β after enough iterative times.

(5) Obtaining the probability distribution of feature words, namely, the implicit topics-documents matrix.

V. MULTICLASS SVM CONSTRUCTING BASED ON VULNERABILITY DISTRIBUTION

A. SVM constructing based on topic model

SVM [12] (Support Vector Machine) is a supervised classification algorithm which was first presented by Cortes and Vapnik. SVM classifications may be more accurate than the widely used alternatives such as classification by maximum likelihood, decision tree, and neural network-based approaches. An SVM aims to fit an optimal separating hyper-plane (OSH) between classes by focusing on the training samples that lie at the edge of the class distributions, the support vectors. The OSH is oriented such that it is placed at the maximum distance between the sets of support vectors. It is because of this orientation that SVM is expected to generalize more accurately on unseen cases relative to classifiers that aim to minimize the training error such as neural networks. Thus, with SVM classification only some of the training samples that lie at the edge of the class distributions in feature space called support vectors are needed in the establishment of the decision surface unlike statistical classifiers such as the widely used maximum-likelihood classifiers in which all training cases are used to characterize the classes. This potential for accurate classification based on small training sets means that the adoption of SVM classification can provide the analyst with considerable savings in training data acquisition.

One single SVM classifier can only separate two classes: a positive class (L1: y = 1) and a negative class (L2: y = -1). After receiving training examples in the form: $\{x_i, y_i\}$, $x_i \in D_{train}$, $y_i \in C$, a OSH which is located between positive and negative examples is determined by SVM algorithm. The OSH may be defined in a linear or non-linear equation by setting y = 0.

B. HT-SVM constructing based on vulnerability distribution

1) Multiclass SVM Methods

Multiclass SVM proposed can be broadly divided into two categories: a single step method and the decomposition and reconstruction method.

A single step method was first proposed by Weston in 1998, by reducing the classification to a single optimization problem, it is supposed to achieve the classification to all the classes occurs in a single step [13]. However, due to the high computational complexity, especially when the number of categories is very high, the classification accuracy and the train speed are too low to satisfy the realistic application.

On the other hand, experimental results show that decomposition and reconstruction method is more suitable in the practical application, such as 1-A-R(One-Against-REST) [14], 1-A-1 (One-Against-One) [15] and DAG-SVM (Directed Acyclic Graph SVM) [16] methods. The earliest used implementation for SVM multiclass classification is probably the 1-A-R method. It constructs k SVM models where k is the number of classes. The ith SVM is trained with all of the examples in the ith class with positive labels, and all other examples with negative labels. Another major method is called the 1-A-1 method. This method constructs k(k-1)/2 classifiers where each one is trained on data from two classes. The third method is DAG-SVM, whose training phase is the same as the 1-A-1 method by solving k(k-1)/2 binary SVMs. However, in the testing phase, it uses a rooted binary directed acyclic graph which has k(k-1)/2 internal nodes and k leaves. Each node is a binary SVM of ith and jth classes. Given a test sample, starting at the root node, the binary decision function is evaluated. Then it moves to either left or right depending on the output value. Therefore, we go through a path before reaching a leaf node which indicates the predicted class.

2) HT-SVM Model constructing

Though there are several multiclass SVM methods discussed above, the existing methods are not completely suitable for the vulnerability classification. First, the above multiclass SVM methods all have some deficiencies, such as too many SVMs are needed in the training step for 1-A-1 SVM and DAG-SVM, blind spots may not be classified for 1-A-1 and 1-A-R SVM, the structure of DAG-SVM is not stable and so on. Second, none of the above methods has utilized the prior knowledge of the vulnerability database, which is very useful in the multiclass issue.

Based on the conventional method and the prior knowledge of the vulnerability database, a novel method is proposed drawing on Huffman tree algorithm [17] for constructing multiclass SVM classifier called HT-SVM (Huffman tree SVM) is proposed. The uneven distribution information of the vulnerability database is introduced into the HT-SVM constructing procedure to improve the classification accuracy and efficiency. Drawing on the principles of Huffman tree algorithm, the larger partition sample categories are obliged to accessing the root of the tree as close as possible, which makes the structure of the HT-SVM is more stable and only K-1 SVMs are needed for k-class classification issue.

The constructing algorithm of HT-SVM is listed as follows:

(1) Making a statistic analysis of the vulnerability distribution, and record the proportions as data set $N(K)=\{n_1, n_2, \ldots\ldots, n_k\}$. n_i denotes the proportion of the ith category in the distribution.

For example, the vulnerability distribution of different category in NVD from 2002 to 2012 is shown Figure 4. We can conclude that the vulnerability distribution is quite lopsided. The proportion of the eighth maximum amount vulnerabilities exceeds over 80%. The amount of the XSS vulnerability which is the largest is over 50 times than that of Command Injections which is the least.

(2) Reordering the data set N(K) in an ascending order to form a new set $N(K')=\{n_1', n_2', \ldots\ldots, n_k'\}$ ($n_1' \leqslant n_2' \leqslant \cdots\cdots \leqslant n_k'$)

3) Utilizing the first two minimum element in the data set N(K') as the right and left child node to construct a new binary tree node n_j , and setting the weight of the parent n_j the sum of its two child node.

(4) Remove the two child nodes in the data set N(K'), while adding the new parent node n_j into the set N(K').

(5) Repeating step (2), (3) and (4) until the data set N(K') contains only one node element.

(6) Algorithm ending.

At this point, the HT-SVM tree has been established.

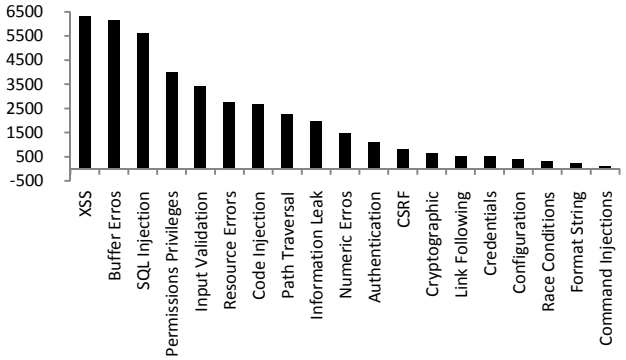

Fig. 4 Vulnerability Distribution of different category (2002-2012)

From the above algorithm, we can also conclude the following characteristics of HT-SVM tree:

(1) The HT-SVM tree Contains total 2K-1 nodes, which has k-1 internal nodes and k leaves.

(2) The weight of the root node equals to sum of all the rest internal nodes and leaves. The weight of each parent node equals to the sum of its child nodes.

(3) The k-1 internal nodes denote the K-1 SVMs, while k leaves denote k categories.

3) Data training

For HT-SVM method, the training samples needed for K-1 classifier are not the same. Given a sample ith SVM, all the categories belong to its left child are supposed to be positive samples, while the negative samples contain the different categories belong to its right child.

VI. Experiments and Results

A. Environment and data set

The experiment condition contains a PC with Intel(R) Core(TM) i5-3450 3.10 GHZ cpu and 4.0GB memory, using Windows XP as OS. All experiments have been implemented based on open source software LIBSVM [18] and GibbsLDA.

In order to validate the efficiency of the proposed approach, we use the NVD as the experimental data to test the algorithm. The data set contain a total of 41,314 effective vulnerability records of 19 categories from January 2002 to December 2012 in NVD, 19 categories. The data is divided into train set and test set according to the proportion of one to one.

B. Evaluation criteria

True Positives (TP_i), True Negatives (TN_i), False Positives (FP_i) and False Negatives (FN_i) are four common measures for evaluating how successful a classifier is. TP_i are the relevant cases that belong to category C_i and are correctly identified as yes. TN_i are the irrelevant cases that do not belong to category C_i and are correctly identified as no. FP_i are the irrelevant cases that do not belong to category C_i but are incorrectly identified as yes. FN_i are the relevant cases that belong to category C_i but are incorrectly identified as no.

Given above four measures, another three common measures can be defined. The precision P_i estimates the probability that a document really belongs to the category C_i which the classifier assigns to it. However, the recall R_i indicates the probability that a document is correctly assigned to the category C_i which it actually belongs to. Moreover, precision and recall rate are generally combined into one single metric called F1 [19]. F1 is defined to be a harmonic mean of precision and recall. The higher F1 of classifier indicates better performances. These three measures are separately defined as:

$$\begin{cases} P_i = \dfrac{TP_i}{TP_i + FP_i} \\ R_i = \dfrac{TP_i}{TP_i + FN_i} \\ F1 = \dfrac{2 \times P_i \times R_i}{P_i + R_i} \end{cases} \tag{4}$$

In order to measure the global performance of the classifier taking into account of all predefined categories, we introduced two averaged measure called macro F1 value and micro F1 value. Macro F1 focuses more on frequent categories while micro F1 value gives more emphasis on rare ones. They are defined by:

$$\begin{cases} \text{Macro } F1 = \dfrac{2 \times \frac{1}{n}\sum P_i \times \frac{1}{n}\sum R_i}{\frac{1}{n}\sum P_i + \frac{1}{n}\sum R_i} \\ \text{Micro F1} = \dfrac{\sum TP_i}{\sum TP_i + \sum FP_i} \end{cases} \tag{5}$$

C. Results and Comparisons

WL-LDA and HT-SVM model has been established base on the analysis of word location information and vulnerability distribution. In the WL-LDA model, we select the experience value α=50/k, β=0.01, iterations equals to 1000, the number of topics T to 120. In the HT-SVM model, RBF kernel function is selected.

1) Comparisons among vector space models

In order to estimate the improvement of the WL-LDA model, we compare it with the other two methods respectively are VSM model and the classical LDA model. HT-SVM model is used as classification algorithm in all three experiments. Figure 6 shows the results.

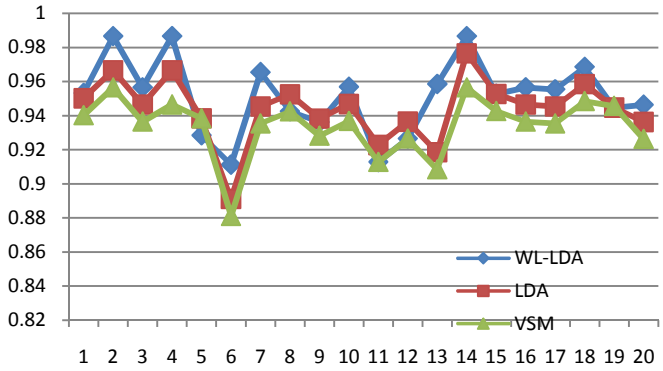

Fig. 5 The value of F1 of different vector space models

As can be seen in Figure 5, WL-LDA model obtains the highest accuracy for vulnerability classification, for feature dimensionality reduction and word location information are both introduced into the model. The value of macro F1 and micro F1 of WL-LDA model is 0.952 and 0.953, which denotes the approach achieve a great accuracy.

2) Comparisons among classification methods

In order to estimate the improvement of the HT-SVM method, we compare it with the other three methods respectively are 1-A-R, 1-A-1 and DAG-SVM method based on Weka platform [20]. WL-LDA model is used as vector space algorithm in all four experiments. Table III shows the results.

TABLE III
RESULTS OF DIFFERENT CLASSIFICATION METHODS

METHOD	MACRO F1	MICRO F1	TRAINING TIME (SEC)	TESTING TIME (SEC)
1-A-R	0.8923	0.9016	591.02	58.53
1-A-1	0.8857	0.8905	157.41	46.26
DAG-SVM	0.8984	0.9106	156.38	22.58
HT-SVM	0.9149	0.9198	168.52	20.31

For the values of macro F1 and micro F1, HT-SVM method obtains the highest accuracy. For the training time, 1-A-R method is the worst, as all the data need to be involved in the training step. 1-A-1 and DAG-SVM methods are the best. In fact the two methods have the same training procedure. Though we have to train as many as classifiers, as each step only data from two classes are needed, the total training time is still less. The training time of HT-SVM method is very approximate to 1-A-1 and DAG-SVM. For the training time, HT-SVM method is the best, for exploiting the prior knowledge of vulnerability distribution.

VII. CONCLUSION

In this paper, we have blended into the word location information based on the existing LDA model feature selection algorithm for the purpose of discovery the difference of underlying topics among the disparate vulnerability data, and combined the vulnerability distribution information with the SVM classifier. LDA model is an emergent probability model and it has the incomparable modeling strengths, SVM classification algorithm has the unique excellent properties on categorization. Prior knowledge of vulnerability data such as word location and amount distribution is both very important for the classification. We have combined the three factors above all together. The experimental results confirm the effectiveness and superiority of this method. However, the experiment is based on the hypothesis cases of single labeled vulnerability data set, but there exists multi-labeled data in some occasions. So the theme modeling and the classification algorithm for multi-labeled vulnerability data set become the further step unfolding study.

ACKNOWLEDGMENT

The authors would like to thank Wang Jian, Liu Jian, Zhang Lei for many helpful comments. Thanks also to Associate Professor Chih-Jen Lin for providing the open source software LIBSVM.

REFERENCES

[1] Abbott R, Chin J, Donnelley J, et al, "Security Analysis and Enhancements of Computer Operating Systems," Washington DC, USA: US Department of Commerce, 1976.
[2] Bisbey II R, Hollingworth D, "Protection Analysis: Final Report," Marina Del Rey, USA: University of Southern California, 1978.
[3] Adam T, Krsul I, Spafford E, "Use of a Taxonomy of Security Faults," West Lafayette, USA: Pordure University, 1996.
[4] CHEN Zhongqiang, ZHANG Yuan, CHEN Zhongrong, "A categorization framework for common computer vulnerabilities and exposures," The Computer Journal, vol. 53, no. 5, pp. 551-580, 2010.
[5] Li Y I, "An Approach towards Standardising Vulnerability Categories," Pretoria, South Africa: University of Pretoria, 2007.
[6] G Salton, A Wong, CS Yang, "A vector space model for automatic indexing," Communications of the ACM, vol. 18, no. 11, pp. 613-620, 1975.
[7] National Vulnerability Database. http://nvd.nist.gov/
[8] Martin Braschler, Bärbel Ripplinger, "How Effective is Stemming and Decompounding for German Text Retrieval," Information Retrieval, 2004.
[9] G Salton, B.Buekley, "Term weighting approaches in automatic text Retrieval," Information Proeessing and Management, vol. 24, no. 5, pp. 513-523, 1988.
[10] Blei D, Ng A, Jordan M, "Latent dirichlet allocation," Journal of Machine Learning Research, vol. 3, no. 1, pp. 993-1022, 2003.
[11] Grilfiths T, "Gibbs Sampling in the Generative Model of Latent Dirichlet Allocation," Stanford, USA: Stanford University, 2004.
[12] C Cortes, V Vapnik, "Support-Vector Networks," Machine learning, no. 20, pp. 273-297, 1995.
[13] WestonJ, Watkins C, "Modeling Multi-class support vector machines," Technical Report CSD-TR-98-04, Dept. of Computer Science, University of London, 1998.
[14] Krebel U, "Pairwise classification and support vector machines," MA: MITPress, pp. 255-268, 1999.
[15] Bottou L, Cortes C, Denker J, et al, "Comparisonof classifier methods:a case study inhandwriting digit recognition," Int Conf Pattern Recognition, Jerusalem, Israel, pp.77-87, 1994.
[16] Platt J C, Cristianini N, Shawe Taylor J, "Large margin DAGs for multiclass classification," Advances in Neural Information Processing Systems, vol. 12, no. 3, pp. 547-553, 2000.
[17] J van Leeuwen, "On the construction of Huffman trees," Proceedings of the 3rd International Colloquium, 1976.
[18] Chih-Chung Chang, Chih-Jen Lin, "LIBSVM: A library for support vector machines," ACM Transactions on Intelligent Systems and Technology, vol. 2, no. 3, April 2011.
[19] K.Van Rijsbergen, "Information Retrieval," Butterworths, London, 1979.
[20] Xiu-yu Zhong, "The Research And Application of Web Log Mining Based on the Platform Weka," Procedia Engineering, 2011.
[21] Zhu, T. and K. Li, "The Similarity Measure Based on LDA for Automatic Summarization," Procedia Engineering, vol. 29, 2012.
[22] Xu, Z., P. Li and Y. Wang, "Text Classifier Based on an Improved SVM Decision Tree," Physics Procedia, vol. 33, 2012.
[23] Kuo, T. and Y. Yajima, "Ranking and selecting terms for text categorization via SVM discriminate boundary," Int. J. Intell. Syst., vol. 25, 2010.
[24] Hassan, S., M. Rafi and M.S. Shaikh, "Comparing SVM and Naive Bayes classifiers for text categorization with Wikitology as knowledge enrichment," CoRR, 2012.
[25] Lu, G., J. Zou and Y. Wang, "Incremental complete LDA for face recognition," Pattern Recognition, vol. 45, 2012.
[26] La Lei, et al., "LDA boost classification: boosting by topics," EURASIP Journal on Advances in Signal Processing, 2012.
[27] Rezarta, I.D. and Y. Lana, "Topics in machine learning for biomedical literature analysis and text retrieval," Journal of biomedical semantics, vol. 3, 2012.
[28] A, V.K. and G. Aghila, "A Survey of Naïve Bayes Machine Learning approach in Text Document Classification," International Journal of Computer Science and Information Security, vol. 7, no. 2, 2010.