Research on Steganalysis for Text Steganography Based on Font Format

Lingyun Xiang, Xingming Sun, Gang Luo, Can Gan
School of Computer & Communication, Hunan University, Changsha, Hunan P.R.China, 410082
Suhong210@yahoo.com.cn

Abstract

In the research area of text steganography, algorithms based on font format have advantages of great capacity, good imperceptibility and wide application range. However, little work on steganalysis for such algorithms has been reported in the literature. Based on the fact that the statistic features of font format will be changed after using font-format-based steganographic algorithms, we present a novel Support Vector Machine-based steganalysis algorithm to detect whether hidden information exists or not. This algorithm can not only effectively detect the existence of hidden information, but also estimate the hidden information length according to variations of font attribute value. As shown by experimental results, the detection accuracy of our algorithm reaches as high as 99.3% when the hidden information length is at least 16 bits.

Keywords: *Steganalysis; Text; Font format; Hidden information length; Support vector machine*

1. Introduction

With the tremendous development of Internet, large bulk of text data are transmitted and exchanged daily. The text steganography has become a hotspot in the research area of information security. All existing methods of text steganography mainly fall into three categories, which are based on invisible characters, format, and natural language respectively. The steganographic methods based on format are used most widely in practice because of their great capacity and good imperceptibility. They mainly include methods based on line-shift coding, word-shift coding, character coding[1], and methods based on font format which we will focus on in this paper.

Many font-format-based methods, such as changing the type, size, color or underline color of the font, have been proposed. These methods, having the same basic principle, mainly make imperceptible modification of font format to hide information. For example, the method based on font color embeds a bit string of the hidden information into the lowest bits of RGB value. Such slight modifications of the font color can't be perceived by human vision, which make the hidden information invisible under normal circumstances.

Even though text steganography has attracted great interest, there has been little research reported on text steganalysis[2,3,4,5] in the literature. The steganalysis algorithms usually use statistical analysis to detect the existence of hidden information. For instance, the reference [5] has designed a steganalysis method on semantic steganography, which is based on the distribution of first letters of words according to the differences of the statistical features between stego-texts and normal texts. As far as we know, only one font-format-based steganalysis algorithm has been proposed in the literature, which is based on text noise[4]. However, the algorithm has disadvantage of detecting less hidden information in larger text and is not good at eliminating the influence of noise in text, and as a result, the detection accuracy is low.

In order to improve the accuracy of detecting hidden information based on font format, and try to estimate the hidden information length, this paper proposes a novel steganalysis algorithm summarized as follows: First, get the characteristic vector according to the distance of font attributes between every two adjacent characters; Second, take the characteristic vector as input of Support Vector Machine (SVM) and train the classifier; Finally, use the trained classifier to detect the existence of hidden information in test document. If exists, its length has been estimated according to variations of font attribute value.

The organization of the paper is as follows: Sect.2 includes a brief introduction to framework of the detection model. Sect. 3 describes how to obtain the input vector of SVM. In Sect. 4, we explain the principle of estimating the hidden information length. The experimental results and analysis are shown in Sect. 5, followed by a conclusion in Sect.6.

0-7695-2876-7/07 $25.00 © 2007 IEEE
DOI 10.1109/IAS.2007.48

2. Detection model

Detecting the existence of hidden information in text can be regarded as a classification problem, that is how to identify the text as the normal text, or the stego-text with hidden information. We can use support vector machine, which has very good performance of classification[6,7], as the classifier. SVM has been extensively used and it has delivered state-of-the-art performance in steganalysis of image and video[8,9]. It has been shown to be superior to traditional pattern classifiers in classification problems.

The theoretical foundation of SVM comes from the statistic learning theory proposed by Vapnik et al[6]. The basic training principle behind SVM is to find the optimal linear hyperplane such that the expected classification error for unseen test samples is minimized. According to the structural risk minimization principle and VC dimension minimization principle, a linear SVM uses a systematic approach to find a linear function with the lowest capacity[7]. For linearly nonseparable data, SVMs can nonlinearly map the input to a high-dimensional feature space where a linear hyperplane can be found.

In this study, we employ SVM to classify the characteristic vector of each font attribute belonging to two classes – normal text and stego-text. The process can be described as follows: first, train a classifier M_j for font attribute j ($0 \le j < m$, m is the total number of attributes), then classify the characteristic vector of each font attribute extracted from input text using classifier M_j. The training process of classifier M_j is illustrated in Figure 1.

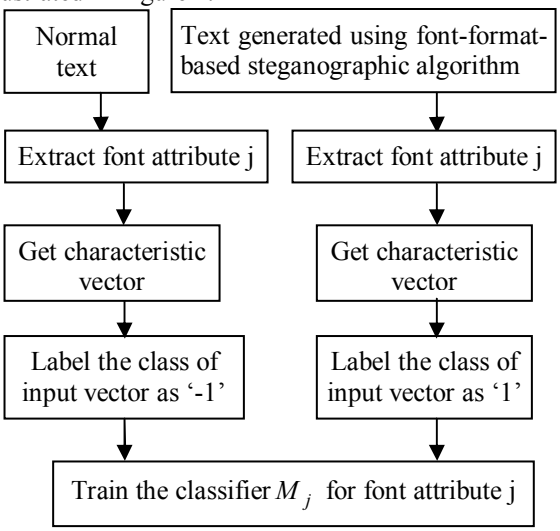

Figure 1. **The training process of classifier** M_j

Usually, there are several font attributes in a formatted text, for instance, the number of font attributes that may be used to hide information is as many as 35[4] in a Word document. Therefore, a algorithm described following is designed for detecting all the font attributes.

Algorithm 1: General steganalysis algorithm for text steganography based on font format.

Input: Font attributes and the corresponding classifiers.

Output: The names of attributes which include information and the total hidden information length.

Step1: Initialize the hidden information length $c_j = 0$ and marker variable T_j=False($0 \le j < m$);

Step2: Traverse every font attribute of all characters in the text, and extract values of each m font attributes;

Step3: For each attribute, generate the characteristic vector according to the values;

Step4: For each non-empty characteristic vector, use the trained classifier M_j ($0 \le j < m$) to identify whether there is embedded information or not. If hidden information was found, set T_j =True and estimate the hidden information length c_j for attribute j;

Step5: If T_j=True, output the name of attribute j and the value of c_j, and compute the total hidden information length.

Since the key of algorithm above is to choose appropriate characteristic vector as the input vector of SVM, we will introduce how to generate the characteristic vector in detail in next section.

3. Selection of characteristic vector

3.1. The analysis of characteristics of font attribute

Generally speaking, the font attributes of stego-text were frequently and insignificantly changed by using font-format-based steganographic algorithm, and the modifications can not be perceived by the human vision. Normal operations of font adjustment can also modify font attributes, but the modifications are usually obvious. For instance, the red font are usually used for some important sentences in text and other words are normally black. Since the principles of embedding information based on different font attributes are similar, we just analyze characteristic of one certain font attribute. In this paper, the term of font attribute refers to a particular kind of font attributes. We map each character's font attribute to an integer, and get a sequence $\{P(t_i)\}$. $P(t_i)$ can be described as follows:

$$P(t_i) = p(t_i) + \gamma(t_i) + \delta(t_i),\ t_i \in \textit{characters of text},$$

where $p(t_i)$ represents a normal font attribute value and is a constant. $\gamma(t_i)$, which can be either positive or negative, represents the variation of font attribute value due to hidden information. $\delta(t_i)$, defined as noise signal, represents the variation of font attribute value, which is caused by normal operations of font adjustment. So the attribute distance between character t_i and the adjacent character t_{i+1}, denoted by D_i, can be expressed as follows:

$$D_i = |P(t_i) - P(t_{i+1})|$$
$$= |p(t_i) - p(t_{i+1}) + \gamma(t_i) - \gamma(t_{i+1}) + \delta(t_i) - \delta(t_{i+1})|.$$

Because $p(t_i)$ is a stationary signal, $p(t_i) - p(t_{i+1}) = 0$. The nonzero values of $|\delta(t_i) - \delta(t_{i+1})|$, which are caused by changing font attribute value of character t_{i+1} from the value of adjacent character t_i into another value when adjusting font format for visual effect, are usually large. But the nonzero values of $|\gamma(t_i) - \gamma(t_{i+1})|$ are usually small, since steganographic algorithms request good imperceptibility characteristic. So $|\delta(t_i) - \delta(t_{i+1})| >> |\gamma(t_i) - \gamma(t_{i+1})|$, when $|\delta(t_i) - \delta(t_{i+1})|$ is nonzero.

Considering the limitations of human vision, we suppose the distinguishable limit of the attribute distance is λ, When $D_i > \lambda$, we set $D_i = 0$. In this way, part of strong noise can be eliminated (to avoid missing detecting hidden information that doesn't have good imperceptibility, we usually set λ a little bigger number).

3.2. Getting the characteristic vector

As the analysis of each part of D_i in previous subsection, whether character t_i or t_{i+1} contains hidden information can be well presented by D_i. Nevertheless, there are too many elements in sequence $\{D_i\}$, so we compute attribute distance frequency of $\{D_i\}$ as the characteristic vector. Attribute distance frequency is defined as follows:

Definition 1: Attribute distance frequency is the number of times that a given distance value appears in $\{D_i\}$, and it is denoted by S.

According to definition 1, we can compute the attribute distance frequency of D_i whose value is k as $S_k = \sum_{i=0}^{n-2} (D_i = k)$.

Given the attribute with a large λ, there may be too many S_ks and the noise space of attribute is large. If we employ $\{S_k\}$ as the characteristic vector, then the feature of the characteristic vector is unconspicuous. It will degrade the classification performance of SVM. Therefore, in order to use a characteristic vector with more obvious feature, in other words, to effectively eliminate the influence of noise, especially for the larger text with less hidden information, we propose to divide $\{D_i\}$ into certain subareas according to λ (the larger λ is, the more subareas are), and then compute the total distance frequency in each area as characteristic vector of SVM. As the attribute values are always expressed in binary, the way to divide $\{D_i\}$ into subareas is shown below:

$$\begin{cases} (1, 2^g - 1) & g = 1 \text{ or } 2 \\ (1, 2^2 - 1), (2^2, 2^g - 1) & g = 3 \text{ or } 4 \\ (1, 2^2 - 1), (2^2, 2^4 - 1), (2^4, 2^5 - 1) \ldots (2^{g-1}, 2^g - 1) & g > 4 \end{cases},$$

where g satisfies inequality: $2^{g-1} \le \lambda < 2^g$ （$g \ge 1$）.

Therefore, the D_is caused by hidden information cluster in low subareas, and the D_is caused by noise discontinuously distribute in each subarea, especially cluster in high subareas. The characteristic vector obtained from subareas represents the feature of $\{D_i\}$ in a better way, thus improving the classification performance of SVM. The final characteristic vector is calculated as follows:

$$\begin{cases} V = \{v_0\} \\ V = \{v_0, v_1\} \\ V = \{v_0, v_1, \ldots v_{g-3}\} \end{cases}, \text{ where}$$

$$\begin{cases} v_0 = \sum_{i=1}^{2^g - 1} S_i & g = 1 \text{ or } 2 \\ v_0 = \sum_{i=1}^{3} S_i, v_1 = \sum_{i=4}^{2^g - 1} S_i & g = 3 \text{ or } 4 \\ v_0 = \sum_{i=1}^{2^2 - 1} S_i, v_1 = \sum_{i=2^2}^{2^4 - 1} S_i, v_k = \sum_{i=2^{k+2}}^{2^{k+3} - 1} S_i & g > 4, 2 \le k \le g - 3 \end{cases}.$$

4. Estimating the hidden information length

4.1. Principle of estimation

In practice, we are not only interested in whether or not the hidden information is present, but also in extracting the hidden information, and even more. However, because of the absence of knowledge about

steganographic algorithms, the encrypt algorithms and the key el., it's difficult for us to extract accurate hidden information[10]. Therefore, additional information, such as the hidden information length in the text, is sometime available and appears to be very usefull. In the following, the principle of estimating the length of hidden information using steganographic algorithms based on font format is introduced in detail.

The key of estimating hidden information length is the parameter β which is defined as bits of information embedded in each character. After eliminating strong noise, suppose the number of different values of $\{P(t_i)\}$ is n, if $\exists\delta(t_i)\neq 0$, we need to classify the n values of $P(t_i)$ according to different $\delta(t_i)$ (the distance between two arbitrary different $\delta(t_i)$ is larger than λ). After that, the number m of the elements in the largest class can be computed, and $m\le 2^{\beta}+1$ ($2^{\beta}+1$ is the number of possible values of $\gamma(t_i)$, including null); if $\delta(t_i)=0$ for each t_i, then $m=n\le 2^{\beta}+1$. Therefore, the minimal β is $\log(m-1)$. Because the steganography always modifies the statistic features of carrier object while embedding hidden information, the more information is embedded, the more statistic features are modified[10]. So m approximates to $2^{\beta}+1$, we set $\beta=\lceil\log(m-1)\rceil$ in experiment.

For stego-text, nonzero points of $|\gamma(t_i)-\gamma(t_{i+1})|$ are usually brought by different β bits information embedded into characters t_i and t_{i+1}. Therefore, C, the number of nonzero elements in $\{D_i\}$, is approximately equal to the times of that each adjacent β bits in the bit string converted from hidden information is unequal. Considering a given bit string serially divided into substrings with β bits and setting α as the ratio of the total bits of substrings, which are unequal to their next adjacent substrings, to the total bits of the given bit string, we can estimate the hidden information length as: $L=C\times\beta\times\frac{1}{\alpha}$ (bit).

4.2. Estimation of α

First, we have 1000 text files partly downloaded from Internet and else from automatic generation, then we convert the texts into the corresponding bit strings and compute the α of each bit string for different β values. The mean, standard deviation, variance of α are shown in table 1. As can be seen from table 1, the variance of α is relatively steady, so α can be used to approximately estimate the hidden information length.

Table 1. **The mean, standard deviation and variance of α**

β	Mean of α	Standard deviation of α	Variance of α
1	0.5073	0.009581	9.1804e-005
2	0.8023	0.007477	5.5898e-005
4	0.9669	0.008035	6.4557e-005
8	0.9786	0.01175	1.3798e-004

5. Experimental results and analysis

Table 2. **Steganographic methods used in experiments**

Methods	Principles	Min
Method one	Choose two arbitrary values from the four lowest bits values (0-15) of blue (B) component of font color to represent '0' and '1' respectively.	8 bits
Method two	Choose four arbitrary values from the four lowest bits values(0-15) of blue (B) component of font color to represent '00', '01','10','11' respectively.	16 bits
Method three	Choose two arbitrary values from the four lowest bits values (0-15) of blue (B) component and synchronously two values of green (G) component of font color, which are combined to represent '0000' to '1111'.	16 bits
Method four	Software:Bytershelter[11], the values of R, G and B components of font color have been changed at the same time, the hidden information embedded in each font-color attribute is approximately 8 bits, and some appendant markers have been embedded into the document.	8 bits

The steganalysis methods for each font attribute are similar and the color attribute is one of the attributes with largest embedding space, so experiments are only done to the attribute of font color. We collected 3,000 Word documents from Internet, which are divided into two groups – the training sample set and the test sample set. Each group includes 1,000 normal documents and 500 documents in which random hidden information has been embedded using the

methods listed in table 2, and at least 10 samples with minimum hidden information length have been made. In table 2, Min means minimum hidden information length.

5.1. Experimental results of detection accuracy

In our program, when eliminating the strong noise in the document, we set the limit value λ=127 for the R, G, B components of font color. SVMs with different kernel functions have different classification abilities. If the classification ability exceeds the complexity of the problem, the classification performance will be degraded. When training SVM classifier, we contrast the experimental results of SVM with different kernel function and finally adopt the nonlinear SVM whose kernel function is a polynomial $K(u,v)=(u\cdot v+1)^3$ and the software used is LIBSVM2.82[12]. Using the trained SVM classifier, we classified the documents in test sample set. At the same time, we detected the texts in test sample set using the steganalysis algorithm based on text noise proposed by [4]. The experimental results in table 3 show that the detection accuracy of algorithm proposed in this paper is greatly improved over the previous. Here, missed detection ratio is defined as (the number of missed detection)/(the number of detected stego-texts). Similarly, false detection ratio is defined as (the number of false detection)/(the number of detected normal texts). Then the detection accuracy is defined as (1-missed detection ratio- false detection ratio)×100%.

Table 3. **Experimental results of algorithms based on SVM classification and based on text noise**

Steganalysis algorithms	Missed detection ratio	False detection ratio	Detection accuracy
This paper's algorithm	2/500	3/1000	99.3%
The reference [4]'s algorithm	68/500	35/1000	82.9%

5.2. Experimental results of estimating the hidden information length

We made 40 sample stego-texts for each steganographic method listed in table 2, 20 of which with 160 bits information embedded and others with 800 bits embedded. Then we used the principle presented in 4.2 to estimate the hidden information length in texts. The results of estimation are shown in figure3-4, the average relative error is listed in table 4. Because in method four not only hidden information but also appendant markers are embedded in a document, its average relative error is larger. But the average relative error of other three methods are relatively small, so the estimation results for this four methods are comparatively ideal. Therefore we can use the estimation principle proposed in this paper in practical application.

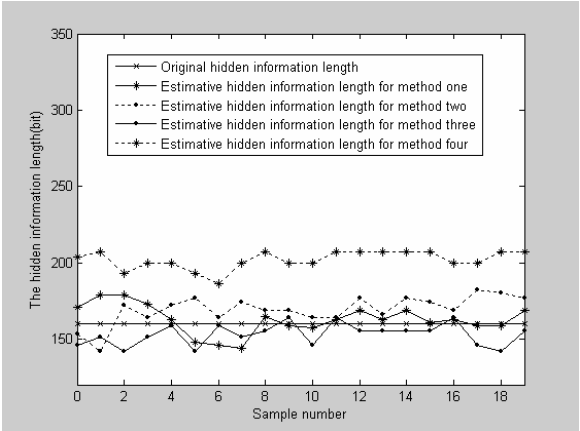

Figure 2. **Results of estimating the hidden information length (original=160bits)**

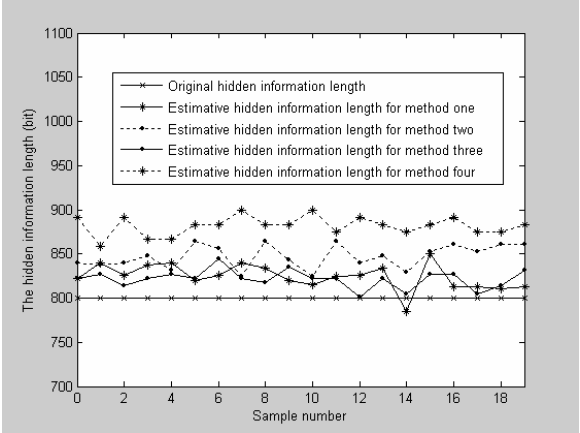

Figure 3. **Results of estimating the hidden information length(original=800 bits)**

Table 4. **The average relative error of estimating the hidden information length**

Steganographic method	The average relative error	
	160 bits of original information	800 bits of original information
Method one	0.0483	0.0325
Method two	0.0737	0.0589
Method three	0.0522	0.0268
Method four	0.2622	0.1022

6. Conclusion

A novel text steganalysis algorithm for font-format-based steganography has been proposed in this paper. It makes use of SVM to classify the characteristic vector extracted from font attributes to find the existence of hidden information, and then estimates the hidden information length according to variations of font attribute value. Experimental results have shown that the algorithm obtains a high detection accuracy of identifying whether hidden information based on font format steganography exists in formatted documents or not. In contrast with other algorithms, this algorithm has advantages of detecting documents with less hidden information and effectively eliminating the influence of noise in texts, achieving lower missed detection ratio and false detection ratio.

Acknowledgement

This work is supported by National Basic Research Program of China (No.2006CB303000), National Natural Science Foundation of China (No.60573045), Specialized Research Fund for the Doctoral Program of Higher Education (No.20050532007).

References

[1] J. Brassil, S. Low, and N.F. Maxemchuk, "Copyright Protection for the Electronic Distribution of Text Documents", Proceedings of the IEEE, 1999, vol. 87(7), pp. 1181-1196.

[2] Zhou Jijun, Yang Zhu, Niu Xinxin, and Yang Yixian, "Research on the Detecting Algorithm of Text Document Information Hiding", Journal on Communications, China, 2004, vol. 25(12), pp. 97-101.

[3] C.Taskiran, U.Topkara, M.Topkara, and E.J.Delp, "Attacks on Lexical Natural Language Steganography Systems", Proceedings of SPIE, Security, Steganography, and Watermarking of Multimedia Contents VIII, San Jose, CA, USA, 2006, vol. 6072, pp. 97-105.

[4] Luo Gang, Sun Xingming, and Liu Yuling, "Research on Steganalysis of Stegotext Based on Noise Detecting", Journal of Hunan University(Natural Sciences), China, 2005, vol. 32(6), pp. 181-184.

[5] Sui Xin-guang, Luo Hui, Zhu Zhong-liang, "A Steganalysis Method Based on the Distribution of First Letters of Words", Proceeding of the 2006 International Conference on Intelligent Information Hiding and Multimedia Signal Processing(IIH-MSP'2006), Pasadena, California, USA, 2006, pp.369-372.

[6] V. Vapnik, The Nature of Statistical Learning Theory[M], Spring Verlag, 1995.

[7] C. Cortes, and V. Vapnik, "Support Vector Networks", MachineLearning, Springer Netherlands, 1995, vol. 20(3), pp. 273-297.

[8] Siwei Lyu, and Hany Farid, "Detecting Hidden Messages Using Higher-order Statistics and Support Vector Machines", Lecture Notes in Computer Science, Springer, 2003, vol. 2578, pp. 340-354.

[9] H. Ozer, I. Avcibacs, B. Sankur, et al., "Steganalysis of Audio Based on Audio Quality Metrics", Proceedings of SPIE, Security and Watermarking of Multimedia Contents V, Santa Clara, CA, USA, 2003, vol. 5020, pp. 55-66.

[10] J. Fridrich, R. Goljan, D. Hogea, et al., "Quantitative Steganalysis of Digital Images: Estimating the Secret Message Length", Multimedia Systems Journal, Springer Berlin / Heidelberg, 2003, vol. 9(3), pp. 288-302.

[11] MazZoft NDA, Bytershelter[EB/OL], 2006.
URL: http://www.mazzoft.com/bs1.zip.

[12] Chih-Chung Chang, and Chih-Jen Lin, LIBSVM: A Library for Support Vector Machines[EB/OL], 2006.
URL: http://www.csie.ntu. edu.tw/~cjlin/libsvm.