A Prediction Model Based Approach to Open Space Steganography Detection in HTML Webpages

✉Iman Sedeeq, Frans Coenen, and Alexei Lisitsa

Liverpool University, Liverpool L69 3BX, UK
{iman.sedeeq,coenen,lisitsa@liverpool.ac.uk}

Abstract. A mechanism for detecting Open Space Steganography (OSS) is described founded on the observation that the length of white space segments increases in the presence of OSS. The frequency of white space segments of different length is conceptualized in terms of an n-dimensional feature. This feature space is used to encode webpages (labelled as OSS or not OSS) so that each page is represented in terms of a feature vector. This representation was used to train a classifier which can subsequently be used to detect the presence, or otherwise, of OSS in unseen webpages. The proposed approach is evaluated using a number of different classifiers and with and without feature selection. Its operation is also compared with two existing OSS detection approaches. From the evaluation a best accuracy of 96.7% was obtained. The evaluation also demonstrated that the proposed method outperforms the two alternative techniques by a significant margin.

Keywords: Open space·Steganography·Classification

1 Introduction

Open Space Steganography (OSS) is a mechanism for hiding data in text "cover" files by utilizing white space characters. The idea was first proposed in the mid 1990s [1]. Although embedding white space characters in a text file will increase the file size, these methods offer the advantage that in any document white space characters appear frequently, more than any other character, therefore the existence of additional white space characters is unlikely to cause suspicion. In addition, and subject to how the OSS is applied and how the cover text file is viewed, in many cases the inclusion of additional white space characters will not result in any noticeable change in the look of the file from the viewer's perspective. Even where additional white space characters are visible, an observer is unlikely to pay significant attention to their presence and is unlikely consider these spaces to represent a hidden message.

WWW pages, written using HTML (Hyper-Text Markup Language) are well suited to OSS. Access to WWW pages on a daily basis will not raise any alarm. Features of HTML such as tag case-insensitivity and no bounds on attribute orderings do not change the way WWW browsers display webpages; similarly the

inclusion of redundant space characters also do not change the way that webpages are displayed. HTML thus provides an ideal opportunity for the steganographer. A number of HTML steganography mechanisms are available, such as: (i) attribute permutation whereby the ordering of HTML tag attributes is used to hide data [2–4] and (ii) switching HTML tag letters case as in [5–7]. An alternative is OSS, the mechanism of interest with respect to this paper. OSS uses one or more of the following to hide data: (i) inter-word spacing, spacing between two successive words; (ii) inter-sentence spacing, spacing between two successive sentences, (iii) additional spaces at the end of lines; and (iv) inter-paragraph spacing, spacing between two consecutive paragraphs. There are several steganography tools that implement data hiding using OSS methods, examples include: Spacemimic [8], wbStego4open [9], SNOW [10] and WhiteSteg [11]. Of these wbStego4open and SNOW are freely available for download and hence are used in the context of the evaluation presented later in this paper. SNOW utilizes end-of-line spacing to hide data, whilst wbStego4open makes use of both inter-word spacing and inter-sentence-spacing to embed data. Unlike SNOW, the wbStego4open tool also checks that the cover file is large enough to accommodate the desired hidden message.

In this paper an HTML OSS detection method is proposed based on the idea, first suggested in [12], that the embedding of a message using white space characters will affect the frequency distribution of continuous white space characters being used in sequence; an observation that holds regardless of the adopted OSS method used to hide data. The idea presented in this paper is to build a classifier that uses frequency distribution of continuous sequences (segments) of white space characters, of different lengths, to distinguish between normal webpages and "stego" -webpages. To the best knowledge of the authors this approach is novel. For evaluation purposes a data set of 150 well-known webpages from different domains (news, education, shopping and business) was compiled. In the context of this set of webpages the proposed approach provided a very promising steganography detection result, an average accuracy of 96.7% was obtained, significantly outperforming competitive approaches reported in [12] and [13].

The remainder of this paper is organized as follows. Firstly Section 2 presents related work, especially the work of [12] and [13]. Section 3 presents an analysis of the work of [12] and [13] and establishes the motivation for the classification-based approach presented in this paper. Section 4 then presents the proposed prediction model based approach. The evaluation of the proposed approach, and comparisons with existing approaches, is then reported on in Section 5. The paper is concluded in Section 6 with a summary of the main findings and some suggested areas for future work.

2 Related Work

The main challenge of OSS detection in WWW pages is the large number of white space characters that will normally exists in a webpage regardless of whether embedding has taken place or not. Although the proposed OSS detection approach

presented in this paper is unlike any other detection approaches, there has been some previous work directed at OSS detection in HTML pages. Of note with respect to the work presented in this paper is the work of [12] and [13].

In [12] a probabilistic model was used to detect HTML OSS. Two occurrence probability values were used for this purpose: (i) total white space character occurrence (p_{tsco}) and (ii) total white space character sequence occurrences (p_{scso}). The first is estimated using Equation 1 where w is a webpage while the second probability is estimated using Equation 2 . These probabilities were compared with predetermined thresholds to decide whether a given webpage was a normal webpage or a stego-webpage. The thresholds in this case was identified using Zipf's and Heaps' law, and a Finite-State Model [14]. They estimated that the probability of occurrence of a white space character (p_{tsco}) in a text file was approximately 0.2 plus or minus 0.1, a threshold of 0.3 was thus proposed. Also they estimated that total white space character sequence occurrences (p_{scso}) in a text file is 0.2.

$$p_{tsco}(w) = \frac{number\ of\ white\ space\ characters\ in\ w}{total\ number\ of\ all\ characters\ in\ w} \tag{1}$$

$$p_{scso}(w) = \frac{number\ of\ all\ white\ space\ character\ sequences\ in\ w}{number\ of\ whitespace\ characters\ in\ w} \tag{2}$$

The detection approach presented in [13] used what was termed the "embedding rate" (e_{rate}). This is the number of characters in a given WWW page with white space characters removed and the number of characters without such characters being removed (Equation 3). The authors defined the normal distribution of the embedding rate using the mean (μ) and standard deviation (δ) of the e_{rate} distribution to define a threshold with which to distinguish normal webpages from stego-webpages.

$$e_{rate}(w) = \frac{characters\ in\ w\ minus\ white\ space\ characters}{characters\ in\ w\ with\ white\ space\ characters} \tag{3}$$

In [12] the OSS mechanism which the author's adopted was not mentioned; however, in [13] the wbStego4open OSS tool for embedding hidden messages (as also used with respect to the evaluation presented later in this paper) was used. The approaches of [12] and [13] are both used with respect to the evaluation of the proposed OSS detection approach presented later in this paper.

3 Analysis of Existing Methods

As noted above there has been little work on OSS detection. The only work that the authors are aware of is that of [12] and [13]. To analyze these two approaches

a collection of 150 commonly visited WWW pages was assembled, covering a variety of domains (education, news, business, shopping). OSS was applied, using both SNOW and wbStego4open (because they were publicly available), to half of the pages (75 pages) using a 83 characters English language text. In this manner two data sets were generated, D_{ws} and D_{snow}, of 150 webpages each.

Equation 1, used with respect to [12], was then applied to obtain before and after total white space character occurrence probability values (p_{tsco_e} and $p_{tsco_{\neg e}}$) with respect to webpages to which OSS had been applied. Table 1 shows the before and after p_{tsco} values obtained with respect to ten of the sample webpages in D_{ws} and D_{snow}. From the table it can clearly be seen that there is a wide variation in the range of p values obtained. A full analysis with respect to both data sets confirmed this result. A summary of this analysis, with respect to the OSS seeded WWW pages, is given in Figure 1. The figure shows two "box plots" one for each set of OSS seeded WWW pages. Inspection of the figure confirms that the range of p values is substantial. Thus it would seem that using a p value static threshold, as proposed in [12], is unlikely to provide good OSS detection results because of this variability.

Table 1: White Space character frequency in selected webpages before and after embedding

Webpage	$p_{tsco_{\neg e}}$	D_{ws} p_{tsco_e}	D_{SNOW} p_{tsco_e}
www.bbc.co.uk	0.090	0.094	0.092
www.bbc.co.uk /weather	0.171	0.189	0.173
www.linkedin.com	0.032	0.078	0.046
www.cnn.com	0.046	0.053	0.048
www.microsoft.com	0.136	0.147	0.141
www.webmd.com	0.104	0.137	0.109
www.wikipedia.com	0.041	0.087	0.057
www.fda.gov	0.062	0.152	0.075

In the case of the approach proposed in [13] Equation 3 was applied to the data sets. A summary of the *er* values obtained is given in Figure 2, also using box plots. From the figure it can be seen that the application of a static *er* threshold for detecting OSS, as promoted in [13], is also not ideal.

4 Proposed HTML OSS Detection Method

This section presents the proposed OSS detection mechanism. The idea presented in this paper is that we use the frequency distribution of different lengths of sequences of white space characters. We use the term *segment* to describe

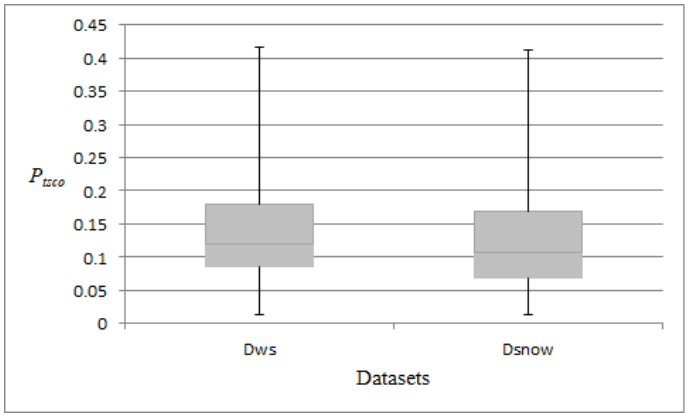

Fig. 1: Box plots for p_{tsco}

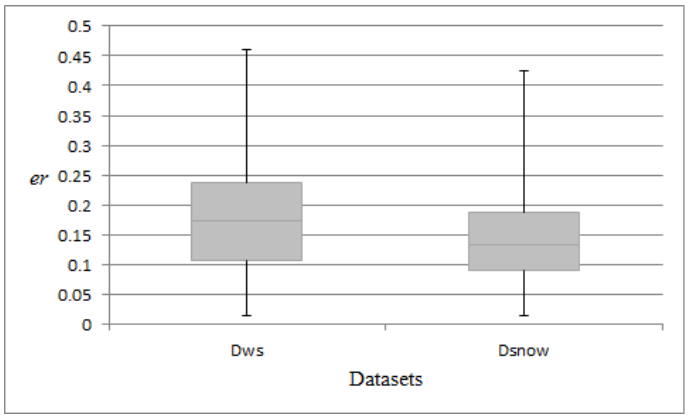

Fig. 2: Box plots for er

contiguous sequences either of white space characters, or non white space characters. Table 2 shows the length of the first ten white space character segments before and after embedding for a selection of eight webpages from the data collection. The · (dot) symbol used in the table indicates the presence of character segments other than white space segments. From the table it can clearly be seen, as expected, that the length of at least some of the white space segments increase in the presence of OSS. Although some webpages already have space character segments of length more than 1 before any embedding has taken place (the Khan Academy and Sony webpages in the table), the idea is that we utilize this feature to detect OSS.

The frequency of a white space character segment of length l can be calculated using Equation 4 where l is the segment length. Note that $\sum_{l=1}^{l=k} f_{cscs_l}(w) = 1.0$ (where k is the maximum size of the segments featured in w whereby w is a webpage).

$$f_{cscs_l}(w) = \frac{number\ of\ segments\ of\ length\ l\ in\ w}{total\ number\ of\ segments\ in\ w} \tag{4}$$

Table 2: First ten white space character segment lengths in selected webpages before and after embedding

Webpage	Before embedding	After embedding (wbStego4open)	After embedding (SNOW)
www.bbc.co.uk	1·1·1·1·1·1·1·1·1·1	1·1·1·1·1·1·1·8·1·1	2·6·5·3·2·3·1·1·1·1
www.dhl.com	1·1·1·1·1·1·1·1·1·1	1·3·1·1·1·1·1·5·1·1	1·1·1·1·1·1·1·7·6·7
www.ieee.org	1·1·1·1·1·1·1·1·1·1	1·1·1·1·1·1·5·1·1·1	1·1·1·1·7·6·1·1·1·1
www.google.com	1·1·1·1·1·1·1·1·1·1	2·1·1·1·1·1·7·1·1	6·1·3·3·5·1·1·1·1·1
www.sdwik.net	1·1·1·1·1·1·1·1·1·1	1·1·1·2·1·1·1·1·1·6	1·1·1·1·7·6·7·6·1·1
dailyroutines.typepad.com	1·1·1·1·1·1·1·1·1·1	2·1·2·1·1·1·1·5·1·1	1·1·1·1·6·1·1·1·1·3
www.khanacademy.com	1·1·1·1·1·1·1·8·1·1	4·1·1·1·1·1·4·1·1·1	7·6·7·6·5·3·1·1·1·1
www.sony.com	1·1·1·1·1·1·1·4·4·4	1·3·1·1·1·1·1·5·1·1	7·6·7·6·5·3·1·1·1·1

Table 3: f_{cscs_5} for some sample webpages before and after embedding

Webpage	f_{cscs_5} without embedding	f_{cscs_5} with embedding (wbStego4open)	f_{cscs_5} with embedding (SNOW)
www.cnn.com	0.00000	0.03590	0.0399
www.wilkipedia.com	0.00000	0.06870	0.1566
www.bbc.co.uk	0.02250	0.05580	0.0612
www.bbc.co.uk/weather	0.00630	0.03180	0.0256
www.adobe.com	0.00080	0.03360	0.0216
www.cisco.com	0.02310	0.05010	0.0432
www.stackoverflow.com	0.00066	0.02999	0.0088
www.sony.com	0.00170	0.03340	0.0176
www.ipod.com	0.00200	0.04330	0.0287
www.nbc.com	0.02300	0.03500	0.0266
www.amazon.com	0.00110	0.03500	0.0152
www.expedia.com	0.00600	0.03660	0.0240

Table 3 shows the frequency of continuous white space character segments of length 5 (f_{cscs_5}) before and after embedding with respect to some of the webpages in the collected set of webpages. From the table it can clearly be seen, as expected, that the length of at least some of the white space segments will increase in presence of OSS.

Given the above the basic idea presented in this paper is to use the frequency of white space segments, within webpages, of different length, with and without embedding, as an indicator of OSS. More specifically the idea was to build a binary prediction model (a classifier) generated using a n-dimensional feature space where n is the number of different potential white space segment lengths of interest that might be included in a webpage. The value for each dimension was then the frequency of the segment of that length occurred in a given webpage. In this manner a webpage would be defined in terms of a feature vector $V = \{v_1, v_2, \ldots, v_n\}$. The process for generating a set of feature vectors given a set of webpages $D = \{w_1, w_2, \ldots, w_m\}$ is given by the pseudo code presented in Algorithm 1. Using the algorithm, for each webpage w_i in D, a feature vector

V_i is created of length n where the elements are the frequency of white space sequences of lengths 1 to n. The process was used to generate sets of feature vectors from our evaluation data set. Note that for the experimental data set the maximum size of a space character segment was found to be 30, hence $n = 30$ was used.

Algorithm 1 Feature vector generation

```
Input: A set of webpages D = {w_1, w_2, ..., w_m}, and the maximum size of the
       white space segments to be considered n
Output: A set of feature vectors Φ = {V_1, V_2, ... V_m}
Φ = ∅
for all w_i ∈ D do
    V_i = ∅
    s = number of white space character segments of length >= 1 in w_i
    for j = 1 to j = n do
        t = number of segments in w_i of length j
        V_i[j] = t/s
    end for
    Φ = Φ ∪ V_i
end for
```

There are a variety of classification models that could have been adopted, however, with respect to the evaluation presented in the following section, three well known classifiers were chosen: (i) a Multi-Layer Perceptron (MLP) Neural Network, (ii) a Support Vector Machine (SVM) and (iii) a Naive Bayes (NB) classifier, as provided within the Waikato Environment for Knowledge Analysis (WEKA) machine learning workbench [15].

5 Evaluation

In this section an evaluation of the proposed OSS detection approach, described in Section 4 above, is presented. Two sets of experiments were conducted, using the feature vector represented data sets generated as described above. The objectives of the evaluation were as follows:

- To determine if the proposed detection approach can be effectively applied to HTML webpage files to detect OSS using the three different classifier generation algorithms identified above.
- To determine the effect of applying a feature selection strategy to the feature vector representation prior to a classifier generation.
- To provide a comparison between the proposed detection approach and the two existing approaches to OSS detection, [12] and [13], as identified in Section 2 above.

The first two of the above objectives are considered in Subsection 5.1, while the third is considered in Sub-section 5.2.

Table 4: Effectiveness of proposed OSS detection technique using D_{ws} without feature selection

TCV	MLP			SVM			NB		
	Prec.	Rec.	Acc.%	Prec.	Rec.	Acc.%	Prec.	Rec.	Acc.%
1	0.88	1.00	93.33	1.00	1.00	100.00	1.00	1.00	100.00
2	1.00	1.00	100.00	0.88	1.00	93.33	0.88	1.00	93.33
3	0.78	1.00	86.67	0.88	1.00	93.33	0.64	1.00	73.33
4	1.00	1.00	100.00	1.00	1.00	100.00	1.00	1.00	100.00
5	0.88	1.00	93.33	0.88	1.00	93.33	0.86	0.86	86.67
6	0.88	0.88	86.67	0.88	0.88	86.67	0.88	0.88	86.67
7	1.00	0.88	93.33	1.00	1.00	100.00	1.00	0.75	86.67
8	1.00	1.00	100.00	1.00	1.00	100.00	0.89	1.00	93.33
9	0.80	1.00	86.67	0.89	1.00	93.33	0.75	0.75	73.33
10	0.86	0.75	80.00	1.00	0.88	93.33	0.88	0.88	86.67
Average	0.91	0.95	92.00	**0.94**	**0.97**	**95.33**	0.88	0.91	88.00
SD	0.09	0.09	6.89	0.06	0.05	4.50	0.12	0.10	9.32

5.1 Effectiveness of the proposed detection approach

The effectiveness of the proposed OSS detection technique was measured in terms of precision (Prec.), recall (Rec.) and accuracy (Acc.). Precision denotes the proportion of true positives detected OSS webpages out of all webpages labeled as OSS (true and false positives), recall is a ratio of a number of true positives identified OSS webpages out of the number of all actually OSS webpages (true positives and false negatives), and accuracy is a proportion of correctly identified webpages (both OSS and non-OSS). Ten cross validation was used whereby the data set was first stratified and then divided into tenths and ten classifiers generated using different nine tenths and tested on the remaining tenth. For the feature selection the "CfsubsetEval" attribute evaluation algorithm, and best first search, as provided in WEKA, was used. The idea behind attribute evaluation is to select a subset of the dimensions (recall that each dimension represents an attribute), in the feature space, that are good discriminators of class. The search method used defines how we search the feature space to identify the best attributes. Note that, unlike other feature selection mechanisms, CfsubsetEval does not select the k best dimensions (attributes) but the best performing dimensions according to some threshold. There are many different techniques that can be used for both. The advantages that are typically offered by feature selection are: (i) better accuracy than if no feature selection was undertaken (the argument being that only good discriminators are retained), (ii) prevention of overfitting (where a classifier is so precisely matched to the training data that it does not work well with other data) and (iii) better classification generation time (fewer dimensions to consider) [16].

The results are presented in Tables 4 to 7 (best results highlighted in bold font). Tables 4 and 5 show the results obtained without feature section, whilst Tables 6 and 7 the results obtained with feature selection. A summary is presented in Table 8. From the summary table it is clear that better results were obtained when using feature selection than without feature selection. With respect to the D_{ws} data set SVM produced consistently the best results with respect to all

Table 5: Effectiveness of proposed OSS detection technique using D_{SNOW} without feature selection

TCV	MLP			SVM			NB		
	Prec.	Rec.	Acc.%	Prec.	Rec.	Acc.%	Prec.	Rec.	Acc.%
1	0.86	0.86	86.67	0.78	1.00	86.67	0.83	0.71	80.00
2	1.00	1.00	100.00	1.00	1.00	100.00	0.67	0.57	66.67
3	0.75	0.86	80.00	0.86	0.86	86.67	0.56	0.71	60.00
4	0.75	0.86	80.00	1.00	0.86	93.33	1.00	0.86	93.33
5	1.00	1.00	100.00	1.00	0.86	93.33	0.86	0.86	86.67
6	1.00	0.88	93.33	0.78	0.88	80.00	0.75	0.75	73.33
7	1.00	0.88	93.33	1.00	0.63	80.00	0.80	0.50	66.67
8	0.86	0.88	86.67	1.00	0.63	80.00	0.86	0.75	80.00
9	0.78	0.88	80.00	0.80	1.00	86.67	0.67	0.75	66.67
10	0.88	0.88	86.67	0.75	0.75	73.33	0.62	0.63	60.00
Average	0.89	**0.89**	**88.67**	**0.90**	0.84	86.00	0.76	0.71	73.33
SD	0.11	0.06	7.73	0.11	0.14	7.98	0.13	0.11	11.33

Table 6: Effectiveness of proposed OSS detection technique using D_{ws} with feature selection

TCV	MLP			SVM			NB		
	Prec.	Rec.	Acc.%	Prec.	Rec.	Acc.%	Prec.	Rec.	Acc.%
1	0.88	1.00	93.33	1.00	1.00	100.00	1.00	1.00	100.00
2	1.00	1.00	100.00	1.00	1.00	100.00	0.88	1.00	93.33
3	0.88	1.00	93.33	0.88	1.00	93.33	0.88	1.00	93.33
4	1.00	1.00	100.00	1.00	1.00	100.00	1.00	1.00	100.00
5	0.88	1.00	93.33	0.88	1.00	93.33	1.00	1.00	100.00
6	0.88	0.88	86.67	1.00	0.88	93.33	1.00	0.88	93.33
7	1.00	0.88	93.33	1.00	1.00	100.00	1.00	0.88	93.33
8	1.00	1.00	100.00	1.00	1.00	100.00	1.00	1.00	100.00
9	0.89	1.00	93.33	0.89	1.00	93.33	0.89	1.00	93.33
10	1.00	0.75	86.67	1.00	0.88	93.33	1.00	0.75	86.67
Average	0.94	0.95	94.00	**0.96**	**0.97**	**96.67**	0.96	0.95	95.33
SD	0.06	0.09	4.92	0.06	0.05	3.51	0.06	0.09	4.5

three metrics considered. In the case of the D_{SNOW} data set best recall and accuracy were produced using MLP, whilst best precision was produced using SVM without feature selection, and MLP with feature selection.

With respect to the feature section it is interesting to note that the selected values of l with respect to D_{ws} were $\{1, 2, 3, 4, 5, 6, 7, 8, 21\}$, the value of 21 seems odd and might simply be an "outlier". In the case of D_{SNOW} the selected values for l were $\{1, 2, 5, 7\}$.

5.2 Comparison with other detection approaches

With respect to the second objective, comparison with existing techniques, the comparison was conducted with respect to the performance of the approaches proposed by [12] and [13] as described in Section 2. So that a fair comparison could be arrived at the evaluation was again conducted using TCV, although it should be noted that the approaches proposed by [12] and [13] are both statistical in nature and do not require any training. The intuition here was that the

Table 7: Effectiveness of proposed OSS detection technique using D_{SNOW} with feature selection

TVC	MLP			SVM			NB		
	Prec.	Rec.	Acc.%	Prec.	Rec.	Acc.%	Prec.	Rec.	Acc.%
1	0.88	1.00	93.33	0.86	0.86	86.67	0.64	1.00	73.33
2	1.00	1.00	100.00	1.00	0.71	86.67	0.64	1.00	73.33
3	0.86	0.86	86.67	0.83	0.71	80.00	0.55	0.86	69.00
4	1.00	1.00	100.00	1.00	0.86	93.33	0.88	1.00	93.33
5	1.00	1.00	100.00	1.00	1.00	100.00	1.00	1.00	100.00
6	1.00	1.00	100.00	1.00	1.00	100.00	0.72	1.00	80.00
7	0.80	1.00	86.67	1.00	0.50	73.33	0.67	1.00	73.33
8	1.00	1.00	100.00	1.00	0.63	80.00	0.67	1.00	73.33
9	0.89	1.00	93.33	0.88	0.88	86.67	0.80	1.00	86.67
10	1.00	0.88	93.33	1.00	0.88	93.33	0.64	0.88	66.67
Average	**0.94**	**0.97**	**95.33**	0.96	0.80	88.00	0.72	0.97	78.89
SD	0.08	0.06	5.49	0.07	0.16	8.78	0.14	0.06	12.19

Table 8: Summary of results presented in Tables 4 to 7 (best results highlighted in bold font)

Data Set	Feature Selection	MLP			SVM			NB		
		Prec.	Rec.	Acc.%	Prec.	Rec.	Acc.%	Prec.	Rec.	Acc.%
D_{ws}	No	0.91	0.95	92.00	0.94	0.97	95.33	0.88	0.91	88.00
	Yes	0.94	0.95	94.00	**0.96**	**0.97**	**96.67**	0.96	0.95	95.33
D_{SNOW}	No	0.89	0.89	88.67	0.90	0.84	86.00	0.76	0.71	73.33
	Yes	**0.94**	**0.97**	**95.33**	0.96	0.80	88.00	0.72	0.97	78.89

evaluation would provide an unfair advantage to the proposed system if we trained on the entire data set and then tested on the same data set (it might also result in "overfitting"). For comparison both SVM and MLP classification with feature selection were used with respect to the proposed method, as this had been shown to produce the best performance. For the [13] and [12] methods the reported threshold values were adopted. The evaluation metrics used were again precision, recall and overall accuracy.

The results are presented in Tables 9 and 10. Note that the precision, recall and accuracy values for the proposed approach have been reproduced from Tables 6 and 7 respectively. From the table it can clearly be seen that the proposed approach outperformed the previously proposed approaches by a significant margin. The proposed approach was good at identifying OSS webpages while at the the same time not miss-classifying many non-OSS webpages. Using the approach of [12] both the precision and recall were poor for both data sets, whilst using the approach proposed in [13] produced good recall values but was not very precise. The later was because the approach was classifying most of the webpages as OSS webpages, hence it was correctly classifying most of the OSS webpages but also wrongly classifying most of the many webpages that did not feature OSS as OSS webpages, hence the accuracy was poor. Given that the OSS and non-OSS classes were equally distributed within the data set the accuracies obtained using [12] and [13] and the D_{ws} data, 54.69% and 50.01% respectively,

Table 9: Comparison of proposed OSS detection with that proposed in [12] and [13] using D_{ws} (best results highlighted on bold font)

TCV	Proposed Approach			[12]			[13]		
	Prec.	Rec.	Acc.%	Prec.	Rec.	Acc.%	Prec.	Rec.	Acc.%
1	1.00	1.00	100.00	0.50	0.57	53.00	0.50	1.00	53.33
2	1.00	1.00	100.00	0.67	0.29	60.00	0.54	1.00	60.00
3	0.88	1.00	93.33	0.50	0.29	53.33	0.38	0.71	33.33
4	1.00	1.00	100.00	1.00	0.14	60.00	0.50	1.00	53.33
5	0.88	1.00	93.33	0.25	0.14	40.00	0.47	1.00	47.00
6	1.00	0.88	93.00	0.44	0.57	47.00	0.64	1.00	73.00
7	1.00	1.00	100.00	0.33	0.14	47.00	0.38	0.71	33.33
8	1.00	1.00	100.00	0.60	0.43	60.00	0.50	1.00	53.33
9	0.89	1.00	93.33	0.50	0.43	53.33	0.47	1.00	47.00
10	1.00	0.88	100.00	0.71	0.71	73.33	0.47	1.00	47.00
Average	**0.96**	**0.97**	**96.67**	0.55	0.37	54.69	0.49	0.94	50.01
SD	0.06	0.05	3.51	0.20	0.19	8.79	0.07	0.12	11.16

Table 10: Comparison of proposed OSS detection with that proposed in [12] and [13] using D_{SNOW} (best results highlighted on bold font)

TCV	Proposed Approach			[12]			[13]		
	Prec.	Rec.	Acc.%	Prec.	Rec.	Acc.%	Prec.	Rec.	Acc.%
1	0.88	1.00	93.33	0.43	0.43	46.67	0.46	0.86	46.67
2	1.00	1.00	100.00	0.50	0.14	53.33	0.53	1.00	60.00
3	0.86	0.86	86.67	0.50	0.29	53.33	0.38	0.71	33.33
4	1.00	1.00	100.00	1.00	0.14	60.00	0.50	1.00	53.33
5	1.00	1.00	100.00	0.25	0.14	40.00	0.43	0.86	40.00
6	1.00	1.00	100.00	0.38	0.43	40.00	0.64	1.00	73.33
7	0.80	1.00	86.67	0.33	0.14	46.67	0.33	0.57	26.67
8	1.00	1.00	100.00	0.60	0.43	60.00	0.50	1.00	53.33
9	0.89	1.00	93.33	0.40	0.29	46.67	0.43	0.86	40.00
10	1.00	0.88	93.33	0.67	0.57	66.67	0.47	1.00	46.67
Average	**0.94**	**0.97**	**95.33**	0.51	0.30	51.00	0.47	0.89	47.00
SD	0.08	0.06	5.49	0.20	0.15	0.08	0.08	0.14	0.13

are little better than a guess. In the case of D_{SNOW} accuraccies of 51.00% and 46.00% were obtained, the last worse than a guess.

6 Conclusion

This paper has proposed an OSS detection approach based on the observation that the length of white space segments in webpages increases in the presence of OSS and that this information can be captured in a feature vector format which can subsequently be used to build an OSS prediction (classification) model provided we have pre-labelled training data available. To evaluate the proposed mechanism a collection of 150 commonly viewed webpages was collated. This was split into two, half seeded with an embedded messages and half not. In this manner two 150 webpage test data sets were created, one using the wbStego4open OSS tool (D_{ws}) and one using the SNOW OSS tool (D_{SNOW}). The proposed approach was tested using three well known classifiers and with and

without feature selection. A best average accuracy of 96.7% was obtained indicating the effectiveness of the proposed approach. The proposed approach was also evaluated by comparing its operation with the mechanisms presented in [12] and [13]. In this comparison, the results obtained indicated that the proposed OSS detection mechanism outperformed the two existing approaches considered by a significant margin, [12] and [13] achieved best accuracy of 54.7% and 50.0% for D_{ws} and 51.0% and 47.0% for D_{SNOW} respectively.While the results reported in this paper indicate viability of our approach, more work needs to be done. For future work the authors intend to examine the performance of the proposed OSS detection approach with respect larger data sets and alternative OSS tools. We plan to investigate the proposed method efficiency with respect to different embedding rates. Finally we intend to examine the approach functionality with other than white space character embeddings.

References

1. W. Bender, D. Gruh, N. Morimoto, and A. Lu. Techniques for data hiding. *IBM SYSTEMS*, 35:313–336, (1996).
2. S. Forrest. Introduction to deogol. http://www.wandership.ca/projects/deogol, (2006).
3. Huajun Huang, Shaohong Zhong, and Xingming Sun. An algorithm of webpage information hiding based on attributes permutation. In *4th International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP*, pages 257–260, (2008).
4. Dongsheng Shen and Hong Zhao. A novel scheme of webpage information hiding based on attributes. In *Information Theory and Information Security (ICITIS), 2010 IEEE International Conference on*, pages 1147–1150, (2010).
5. Xin-Guang Sui and Hui Luo. A new steganography method based on hypertext. In *in Procedings of Radio Science Conference*, pages 181–184, (2004).
6. Shen Y. A scheme of information hiding based on html document. *Journal of Wuhan University*, 50:217–220, (2004).
7. Qijun Zhao and Hongtao Lu. A pca-based web page watermarking. *Pattern Recogn.*, 40:1334–1341, 2007.
8. D. McKellar. Space mimic : http://www.spammimic.com/encodespace.shtml, (2000).
9. wbStego4open. http://www.wbstego.wbailer.com/, (2004).
10. Mattew Kwan. The snow home page: http://www.darkside.com.au/snow/, (2006).
11. L.Y.Por, T.F.Ang, and B.Delina. Whitesteg: A new scheme in information hiding using text steganography. *WSEAS Transactions on Computers*, 7:735–745, (2008).
12. Xin-Guang Sui and Hui Luo. A steganalysis method based on the distribution of space characters. In *in Procedings of Communications, Circuits and Systems International conference Guilin Guangzi, China*, pages 54–56, (2006).
13. Huajun Huang, Xingming Sun, Zinshuai Li, and Guang Sun. Detection of hidden information in webpage. *Fourth International Conference of Fuzzy Systems and Knowledge Discovery FSKD*, (2007).
14. R.Baeza-Yates and G.Navarro. Modeling text databases. *"Recent Advences in Applied Probablity"*, pages 1–25, 2006.

15. Eibe Frank, Mark A. Hall, and Ian H. Witten. The weka workbench. online appendix for data mining: Practical machine learning tools and techniques, morgan kaufmann, fourth edition, (2016).
16. N.B. Omar, F.B. Jusoh, M.S. Bin Othman, and R.B. Ibrahim. Review of feature selection for solving classification problems. *Journal of Research and Innovation in Information Systems*, pages 54–60, (2013).