Determining provenance in phishing websites using automated conceptual analysis

Robert Layton
Internet Commerce Security Laboratory
University of Ballarat
Ballarat, Australia
Email: r.layton@icsl.ballarat.edu.au

Paul Watters
Internet Commerce Security Laboratory
University of Ballarat
Ballarat, Australia
Email: p.watters@icsl.ballarat.edu.au

***Abstract*—Phishing is a form of online fraud with drastic consequences for the victims and institutions being defrauded. A phishing attack tries to create a believable environment for the intended victim to enter their confidential data such that the attacker can use or sell this information later. In order to apprehend phishers, law enforcement agencies need automated systems capable of tracking the size and scope of phishing attacks, in order to more wisely use their resources shutting down the major players, rather then wasting resources stopping smaller operations. In order to develop these systems, phishing attacks need to be clustered by provenance in a way that adequately profiles these evolving attackers. The research presented in this paper looks at the viability of using automated conceptual analysis through cluster analysis techniques on phishing websites, with the aim of determining provenance of these phishing attacks. Conceptual analysis is performed on the source code of the websites, rather than the final text that is displayed to the user, eliminating problems with rendering obfuscation and increasing the distinctiveness brought about by differences in coding styles of the phishers. By using cluster analysis algorithms, distinguishing factors between groups of phishing websites can be obtained. The results indicate that it is difficult to separate websites by provenance without also separating by intent, by looking at the phishing websites alone. Instead, the methods discussed in this paper should form part of a larger system that uses more information about the phishing attacks.**

I. Introduction

A phishing attack is an attempt by an attacker, the phisher, to obtain confidential information from a victim, such as passwords or bank account details. Phishing attacks generally target users of online financial systems, although recent information has indicated that phishing attacks are now used to get passwords to critical systems such as government or industrial machines.

Estimates on the damage of phishing widely vary, and while recent work in [10] indicate that while previous estimates of billions of losses [5] due to phishing are probably overestimates there are hidden costs such as damage to branding of attacked companies (which can be incorrectly blamed for phishing attacks against them) and more generally, losses that occur from a general lack of trust in conducting business and banking online. These losses are hard to estimate, but indicate that phishing is a problem that delves deeper then initial monetary losses, which were estimated in [10] to be approximately $US61 million per year.

A phishing attack often has two major elements, an email and a website. The aim of the email is to convince the victim to navigate to the website, while the website's role is to have the victim enter their confidential information, at which point the information is saved. Much work has been done previously on the categorization and filtering of both the websites and the emails [21], [2]. This research is critical to the guarding of intended victim's data, however new methods of detecting phishing emails results in new methods of circumventing those detection systems, creating an arms race between phishers and anti-phishers. Another addition to the anti-phishing problem is that many users see security as a secondary goal [26], which lessens their awareness of security issues such as phishing. Many users are also unaware of the technical information surrounding the Internet based systems that they use, allowing phishers to exploit weaknesses in knowledge such as creating websites that appear legitimate in order to instill false trust in their victims [6], [12]. One example is the use of a picture with the SSL 'lock' logo, which can be easily faked when placed in the content of the webpage as a simple image rather than in the browser as an indicator.

While each element of a phishing attack can be directly linked to an IP address, phishers often employ means so that these IP addresses cannot be traced back directly to them. Botnets are often used [17], which distribute the attack across a wide range of computers (and therefore, a wide range of IP addresses) and control of these botnets is often sold, allowing the phishers responsible for the attacks to remain hidden behind the wall of protection that a botnet provides. Botnets also use evasive measures, such as continually changing the DNS location of the source of an attack, a technique known as fast flux [11]. Therefore a deeper analysis of the information is required in order to determine provenance. Research discussed in [16] used features such as the structure of the phishing site and the fields of the phishing emails to discover that in their dataset, just three identifiable groups were responsible for 86% of attacks. This research provided a great insight into phishing attacks, and hopefully this process can be automated so that

978-1-4244-4626-1/09/$25.00 2009 Crown

these features can be automatically extracted.

Other related work in forensic analysis of email messages was discussed in [25], where data mining was used to find links in spam messages. These results interested LEAs, as some of the clusters found had high levels of confidence, linking spam messages that were previously not linked by human investigators. These results used a manually selected set of features, such as the length of the body of the email or the sending IP address. This provides a problem in tracking an evolving enemy, as if the spammers change their habits to overcome these known discriminating features, than new features need to be selected as old features become ineffective. One of the challenges of the research here is to provide a method for automatically finding these features, so that profiling systems can evolve as the phishers do.

Despite the great deal of effort by researchers and governments, the apprehension rate of phishers remains low. The reasons for this vary, but are generally related to the anonymity granted by the Internet, in which it can be difficult for a knowledgeable target to be tracked online, and also by the borderless Internet, where the laws and law enforcement agencies (LEAs) of the victim's country are powerless in the attacker's country. To help LEAs better identify and apprehend offenders, automated systems need to be set up that can take regular information and provide starting points for investigations to begin. LEAs are unlikely to spend resources chasing a small number of phishing attacks [20], as the expended resources would far outweigh the gains from chasing down the phisher.

The methods used in this research are not just important in dealing with phishing attacks but could also be used to deal with the more general problem of email spam, such as finding automated methods to replicate the work in [25]. The processes in this paper could also be extended to deal with authorship issues such as finding and dealing with 'blog spam' where stories are copied from credible websites and put on another blog to generate ad revenue and could also be used to help determine authorship in legal disputes. This would help individuals and corporations protect their intellectual property against an increasing problem of copyright violations.

To date, there is no research the authors are aware of that focuses on clustering the websites used in phishing attacks to determine provenance, with research on phishing focusing on either the detection or prevention of phishing attacks against end users. This investigation hopes to determine whether this lack of research is justified, or whether a significant gap exists that could be exploited to help group phishing attacks together. Therefore, the main purpose of this research is to investigate the potential application of conceptual analysis on phishing websites in order to cluster phishing attacks by provenance.

II. Related Research

A. Text based clustering

Cluster analysis techniques are often used to identify themes and patterns in text based documents, which includes websites. There has been a wide range of research on performing cluster analysis on text based documents including clustering blog entries to find 'hot stories' [19], finding clusters to aid web searching [27] and finding new genres appearing on the Internet [24]. One classic example of clustering text based documents is in [15] in which a thesaurus as developed by clustering words based on corpus data.

The most prevalent method of text based clustering starts with the bag-of-words model ([13]), where the order of words in a document is ignored and instead replaced with a vector of word frequency counts. While simple, this method has been shown to perform very well against more complicated document models. One common method of improving on the standard bag-of-words model is to use term frequency-inverse document frequency (TF-IDF, discussed in [23]), which weights rare words appearing in documents more heavily then common words appearing in many documents.

One drawback of the bag-of-words model, and of many models of text based documents, is the high dimensionality of the resulting space. It is not uncommon for the bag-of-words model to have a dimensionality in the thousands, causing practical problems for many clustering algorithms that have complexities based exponentially on the number of dimensions of the instance space. Another problem is due to a finding in [3] that as the dimensionality of a problem increases, the distances between points becomes very similar, which causes problems when clustering algorithms are trying to clusters instances based on their similarity to each other.

One common method to overcome the problem of dimensionality is to use dimension reduction techniques such as Principal Component Analysis (PCA) which finds the top 'feature vectors', which are the set of orthogonal vectors that account for the majority of the variance in the data. It is common for these techniques to reduce the vector space by orders of magnitude while maintaining a similar accuracy to the original model. The practical benefits of being able to document large corpus's of data in a smaller time often outweighs the small gains by using the full instance space. PCA was used in [14] to speed up the training of support vector machines (SVM) and found that SVM was invariant under the PCA transformation, and was able to use the dimensionality reduction through PCA to speed up the evaluation of SVM by an order of magnitude while maintaining comparable accuracy, highlighting the effectiveness of this technique.

In some instances, one drawback of PCA and of reducing the number of attributes in a dataset by feature creation is that distances between objects are often preserved, the issue of ambiguous distances, albeit in a lower dimensional space [18]. Subspace clustering algorithms overcome this problem by finding clustering is specific subspaces of the full set of attributes. Often clusters exist in these subspaces that are not considered clusters in higher dimensional spaces due to the fact that unnecessary attributes or noisy attributes skew the overall density in the regions where these clusters would otherwise exist. In this research both dimensional reduction via PCA and also subspace clustering is used to examine if either method is more suited for this task.

B. Phishing Profiling

Profiling was identified in [4] as 'a technique whereby a set of characteristics of a particular class of person is inferred from past experience, and data-holdings are then searched for individuals for close fit to that set of characteristics'. In this research, this definition is used to infer that if one website portrays traits very similar to another website, and those traits are not universal, then the two are likely to be linked by the same class of author, either because they have the same author or they were developed in the same way, suggesting provenance.

To date, there has been little research into profiling the organizations behind phishing. As a result, criminal investigations have been reluctant to track down phishers because of the unknown size of many of these operation. It is difficult to justify the costs of tracking down phishers for law enforcement agencies (LEAs) if the size of the impact is not known prior to the outlay of resources. However through using forensic and data mining techniques, it was found in [16] that there was a 'level of organization in phishing attacks' and that three groups were responsible for around 86% of all offenses for one major financial institution. Research such as this can provide information to LEAs that they can use to justify the resource spending, to provide a better return on investment for their investigations.

One other benefit to profiling phishers is to develop methods to discourage more people from becoming phishers, such as developing advertising campaigns targeted against demographics more likely to become phishers. If phishing does generally have poor returns for the individual phisher, then there will be many people leaving the crime for other forms of income [10]. However if phishing has an appearance of a crime with high rewards and low risks (as pointed out in [7]), then there will be plenty of new phishers to take the place of those leaving. Discouragement against those new entrants would result in fewer phishers operating, which would be easier to manage globally by LEAs.

C. Clustering algorithms

Three very different clustering algorithms are used in the research presented here, in order to present a varied view of the data.

The first algorithm is an iterative version of the k-means algorithm [9] with random starting values. For this algorithm, k begins at a small value (usually 2), and the k-means algorithm is run a number of times, each time with random starting clusters. The k-means algorithm is run to train those starting clusters, to find the best locally optimal clustering for each of the iterations for each given k value. The value for k is then incremented and the procedure is repeated until k hits a predefined upper limit. The best values for k is then decided using a procedure described in the Methodology.

The second algorithm used in this research is the DBSCAN method [8], which looks for points which are in dense areas, i.e. those with 'many, close' neighbours. These points are then assumed to be in a cluster, so other points that are in the same area of high density are added to the same cluster. DBSCAN is an automatic method of finding the number of clusters, which is something that many other clustering algorithms have trouble with. By focusing on density, rather than number of clusters, the problem is changed from identifying the number of clusters, to identifying the concept of density.

The final algorithm used in this research is a subspace clustering algorithm, CLIQUE ([1]). This algorithm begins by finding dense intervals in each of the single dimensions, before iteratively joining these intervals to create dense units in progressively higher subspaces until no further dense units can be found. The set of dense units is then partitioned into clusters, where each unit in a cluster is connected to one another. In this context, connected is defined as sharing a common face (the same range in $(k-1)$ dimensions) or both being connected to a third unit. This algorithm overcomes many of the problems of high dimensionality, which were discussed earlier, but comes at a cost of speed, as many subspaces must be searched to find clusters.

Each of these algorithms suffer from the problem of free parameters. DBSCAN has two free parameters, namely the radius of the neighbourhood and the required density of this neighbourhood (the number of 'close' neighbours). The iterative k-means algorithm require both a starting and ending value for the search, namely the number of clusters to look for. As well as the specified parameters, both k-means and DBSCAN are non-deterministic, meaning different results can occur for different randomizations of the process, such as the data access order for DBSCAN or the initial centroids for k-means. CLIQUE requires two parameters, the first being the number of intervals to split each dimension into and the second being the required density of units to be considered as 'dense' units. One benefit CLIQUE has is that it is a deterministic algorithm, and the same set of parameters and dataset will always result in the same model. These values can not be known in advance, and need to be discovered through searching the parameter space using either a simple search or performing some heuristic on the data to find 'good' values. Methods exists for searching such parameter spaces, however a manual method will be used in this research, with automating this sub-process differed for further research.

These algorithms were chosen to represent a variety of the different classes of clustering algorithms While some better algorithms have been developed since k-means and DBSCAN were developed for cluster analysis, each of these algorithms are still widely used for their speed and accuracy, even in high dimensional data. One final justification for the choice of algorithms presented is the use of the results as part of an automated system. Having faster algorithms is certainly a benefit in online automated systems, as hypothesis testing needs to be performed quickly, and if these algorithms perform the task required, there is little reason to use more involved algorithms. In this case, CLIQUE can provide problems, as it is known to be slower than both k-means and DBSCAN. However if CLIQUE can identify attributes that are critical to finding clusters, this information could be used to select

attributes later on.

III. Data

The data used for the research presented in this paper is a collection of the source code of websites obtained from tracking addresses known to contain phishing websites targeting a major Australian financial institution during 2007. This means that while many of the websites are phishing websites, there are a majority that are not. There are 24403 instances in the dataset, which is noisy, containing both phishing websites and non-phishing websites. There is little sense in cleaning the dataset to include only phishing websites, as any method that aims to be automated (as this research is) should be able to handle dirty data as input. Another benefit is that the non-phishing websites are often placeholder websites, i.e. temporary websites put up in place of a phishing website while the phishing attack is not active. This information could be used in later research to add information outside of the conceptual analysis presented here.

The dataset used in this research was derived from the original data by taking a bag-of-words model on the data and then applying PCA to that model for feature reduction. The bag-of-words model had 653 dimensions, which was reduced to just 17 dimensions after PCA was applied. These 17 dimensions represent the smallest set of dimensions needed to account for over 90% of the total variance of the initial dataset.

IV. Methodology

The reduced dataset from the previous section was clustered using the previously described DBSCAN and Iterative k-means algorithms. The parameters for these algorithms is searched for by running the algorithm with different parameters and investigating the results from each set of parameters, with the aim of finding parameter values that result in good models of the data. CLIQUE is run on the bag-of-words model (the non-reduced dataset of 653 attributes), in order to find sets of attributes that correspond both to clusters and also to original words or symbols in the code.

In order to determine the validity of the results, the silhouette coefficient [22] was used. This method measures the ratio of the intra-cluster distance compared to the inter-cluster distance. The silhouette coefficient has a range of -1.0 to 1.0, and higher values indicate most distinct clusterings, where the clusters found and dense and well separated. Negative values for the silhouette coefficient, indicate a poor clustering where the clusters are poorly separated and overlapping.

The silhouette coefficient punishes models which underfit or overfit the data, as the ratio of the distance to an instance's cluster is compared to the next closest cluster. If the model underfits the data, the intra-cluster distance becomes large, and ultimately the silhouette coefficient drops. Likewise, if the model overfits the data, the inter-cluster distance drops and also results in a drop in the silhouette coefficient (although this does not always happen in practice). Sudden increases between two similar sets of parameters (such as comparing a value for k to $k+1$ in k-means) indicate that significant improvement has been observed in the model, such as selecting a good starting parameter. As the silhouette value becomes much higher, as the model no longer underfits or overfits the data. One caveat of this procedure is that the silhouette coefficient can have many local peaks and troughs. This is caused by clusterings of different types, such as those found by hierarchical clusterings. As an example, clustering a variety webpages using this model might find a few different clusters that are separated by different language, then by language and intent of the webpage and finally by provenance.

To verify that the results do separate based on provenance, a small sample of phishing websites that have been labeled by an expert in the area will be used as a class attribute for some of the instances. These labels represent empirical knowledge that is difficult to quantify and for this reason, it is difficult to gain larger samples without a prohibitive cost in terms of time. The sample is of 30 phishing websites from the original dataset, and have been labeled with an integer class depending on the assumed phishing group behind the website. The clusterings have been evaluated using the F-Measure against this sample, with the goal to achieve scores close to 1.0, indicating that whenever the empirical knowledge says that the two websites belong to the phishing group, the websites also appear in the same cluster. It should be noted that this information was not used in any way during the training process, only as a supervised method of evaluation on the trained models.

In this experiment, the goal is to find a model that assigns provenance of phishing websites rather than complete separation of different attacks, so a simpler model with a comparable silhouette coefficient is preferred to an overly complex but better fitting model. In order to determine this, we search for a plateau of silhouette values where altering the parameters of the algorithm (eps and neighbourhood size for DBSCAN, k for Iterative k-means or the density and interval numbers for CLIQUE) does not drastically affect the silhouette coefficient, which usually occurs when the model fits the data well. In these cases, the bounds of the interval represent the changes in parameter values that give the greatest improvement, which indicates that these values result in good models.

V. Results

The results for iterative k-means are given in table I, for k values between 2 and 40, with 500 iterations of the k-means algorithm run for each value. The higher values for the silhouette coefficient were obtained for the lowest k values, and generally becomes lower as k becomes higher. There are, however, significant increases in the silhouette coefficient (probability less than 0.05) at $k = 5, 7$ or 26. Looking at the median of the scores, it can be seen that for values of $k > 7$, the median silhouette coefficient is just below 0.06, before increasing quickly for smaller k values. This indicates that while the data separates very well into two clusters (as this is the highest silhouette coefficient), $5 \leq k \leq 7$ clusters are also apparent in the dataset. A further good separation of the data occurs with $k = 26$. At this value, the mean, median and

highest silhouette coefficient is abnormally high, and this is also indicated by the t-test, testing this value against the value obtained for $k = 27$

k	Mean	Median	Max	t-test	F
2	0.13	0.19	0.7	0.02	0.44
3	0.16	0.15	0.7	0.0	0.44
4	0.12	0.09	0.6	0.47	0.44
5	0.11	0.09	0.5	0.01	0.60
6	0.09	0.07	0.52	0.88	0.60
7	0.09	0.07	0.49	0.0	0.73
8	0.07	0.05	0.42	0.38	0.71
9	0.07	0.06	0.45	0.42	0.53
10	0.08	0.07	0.43	0.24	0.70
11	0.07	0.06	0.39	0.55	0.73
12	0.06	0.05	0.39	0.55	0.87
13	0.07	0.06	0.47	0.88	0.88
14	0.07	0.05	0.44	0.76	0.70
15	0.07	0.07	0.42	0.25	0.81
16	0.06	0.05	0.4	0.11	0.70
17	0.07	0.06	0.4	0.8	0.82
18	0.07	0.06	0.44	0.38	0.88
19	0.07	0.06	0.36	0.38	0.85
20	0.07	0.06	0.35	0.39	0.82
21	0.07	0.05	0.42	0.77	0.84
22	0.06	0.04	0.41	0.89	0.82
23	0.06	0.05	0.35	0.23	0.74
24	0.07	0.06	0.34	0.54	0.86
25	0.07	0.05	0.36	0.22	0.85
26	0.07	0.06	0.44	0.0	0.86
27	0.06	0.04	0.36	0.12	0.81
28	0.06	0.05	0.38	0.1	0.89
29	0.07	0.07	0.4	0.71	0.81

TABLE I
SILHOUETTE COEFFICIENT VALUES AND F-MEASURE SCORES FOR ITERATIVE k-MEANS

The t-tests in table I are against the null hypothesis that 'this value for k for k-means produces the same silhouette coefficient scores as $k + 1$ does'. Significant results (for a probability of 0.05) occur at $k = 2, 3, 5, 7$ and 26. At these values, strong results are expected, as the model fits the data better than the next highest k-value, indicating that the data splits naturally into these numbers of clusters.

The critical information for the DBSCAN parameter search is presented in table II. There is a significant trade-off between the silhouette coefficient and the number of instances actually clustered, as DBSCAN discards any instance without a significant neighbourhood as noise. If we place the caveat that at least 50% of the instances must be clustered, the best silhouette coefficient obtained is 0.89, for a neighbourhood diameter of 0.015 and minimum points value of 20 and the number of clusters found for that model is 11. From these results, it is clear that DBSCAN can model a portion of the data more accurately than k-means, but has difficulty in modeling a significant portion of the dataset. The number of clusters found was typically around 9, with the size of the neighbourhood (N_{size}) value increasing the size of the clusters, rather than increasing the number of clusters. This strongly suggests that there is a natural separation of around half of the data into between 9 and 11 distinct clusters.

CLIQUE found the highest silhouette score out of the three

N_{size}	s	k	Noise
0.005	0.95	7.8	17879
0.006	0.95	8.0	17918
0.007	0.97	8.8	17512
0.008	0.94	7.8	18041
0.009	0.95	8.4	16877
0.010	0.95	8.8	16712
0.011	0.91	9.8	15541
0.012	0.92	8.8	15467
0.013	0.90	9.6	15136
0.014	0.86	8.8	14756
0.015	0.88	9.2	13825
0.016	0.84	9.0	12938
0.017	0.85	11.0	11805
0.018	0.80	9.8	11683
0.019	0.81	10.4	10962
0.020	0.77	10.4	10525
0.021	0.77	10.4	10154
0.022	0.78	11.2	9759
0.023	0.79	11.0	9561
0.024	0.77	10.6	9463
0.025	0.78	11.8	8742
0.026	0.78	12.2	8338
0.027	0.76	11.6	8698
0.028	0.76	11.0	8759
0.029	0.77	11.8	8201

TABLE II
SILHOUETTE COEFFICIENT VALUES FOR DBSCAN FOR DIFFERENT NEIGHBOURHOOD SIZES (N_{size}). s, k AND NOISE ARE THE MEAN SILHOUETTE, NUMBER OF CLUSTERS AND NUMBER OF NOISE INSTANCES RESPECTIVELY.

algorithms used, as shown in table III. A score of 0.99 was achieved with a density of 0.02 and 30 intervals per dimension. CLIQUE is a deterministic algorithm, meaning that these parameters with this database will always achieve the same model, however 'similar' parameters did not achieve as high scores, with two exceptions (values of 0.91 and 0.88 were also recorded). This indicates that CLIQUE is very sensitive to its input parameters and as it takes a long time to run on large datasets, it may be difficult to use this algorithm against an evolving dataset. The parameters might lead to a good model for this dataset, but a different set of parameters may be necessary for a slightly different dataset.

Tables I and IV show the F-Measure values using the empirical knowledge as class labels for sets of parameters for iterative k-means and DBSCAN respectively. It can be seen that both models perform well, with DBSCAN having slightly higher scores for this set of parameters. F-Measure scores are the harmonic mean of the precision and recall, and correlate closely to the accuracy of a model. These scores are quite high for such simple models, indicating that the model fits the data well (although not perfectly). These results give a strong indication that the models correlate to real-world data, although more work needs to be done to increase this value. Table V shows the same scores for a select number of CLIQUE parameters. CLIQUE performed badly for most parameter values, with the key problem being a lack of ability to cluster the instances. For the parameter models where the largest number of instances from the labelled dataset were clustered (21), the performance was poor (0.45), indicating

I	d	s	k
10	0.02	0.43	9
10	0.04	0.28	7
10	0.06	0.21	3
10	0.08	0.18	2
10	0.1	0.52	5
10	0.12	0.04	5
10	0.14	0.29	2
10	0.16	0.32	2
10	0.18	0.13	5
10	0.2	0.05	5
20	0.02	0.91	10
20	0.04	0.47	11
20	0.06	0.51	4
20	0.08	0.26	2
20	0.1	0.37	5
20	0.12	0.22	2
20	0.14	0.45	2
20	0.16	0.12	4
20	0.18	0.19	4
20	0.2	0.14	4
30	0.02	0.99	10
30	0.04	0.5	10
30	0.06	0.88	7
30	0.08	0.2	2
30	0.1	0.35	2
30	0.12	0.02	4
30	0.14	0.07	4
30	0.16	0.03	4
30	0.18	0.05	4
30	0.2	0.15	2

TABLE III
SILHOUETTE COEFFICIENT VALUES FOR CLIQUE FOR DIFFERENT INTERVAL SEPARATIONS (I) AND REQUIRED DENSITIES (d).

	Points in neighbourhood				
eps	10	20	30	40	50
0.010	0.783	0.783	0.783	1.000	1.000
0.020	0.725	0.803	0.803	0.857	0.857
0.030	0.786	0.817	0.786	0.817	0.789
0.040	0.817	0.817	0.817	0.817	0.817
0.050	0.852	0.793	0.793	0.793	0.783
0.060	0.875	0.793	0.800	0.793	0.793
0.070	0.811	0.794	0.794	0.794	0.794
0.080	0.871	0.871	0.794	0.794	0.794
0.090	0.871	0.794	0.794	0.794	0.794

TABLE IV
F-MEASURE SCORES FOR DBSCAN AGAINST EMPIRICAL KNOWLEDGE. RESULTS ARE CONSISTENT FOR VALUES LESS THAN 0.3.

that the model was not sufficient for this data.

VI. CONCLUSIONS

High scores for the evaluation methods were achieved overall, indicating that when the parameters were correct, the models fit the data well. The empirical tests in particular fared very well, indicating that the models are getting close to being able to identify existing known clusters in the datasets. The issue of verifying the other clusters found (to see whether they correspond to new or different phishing groups), or verifying the purity of the clusters, remains to be tested, and cannot be tested using the websites dataset used alone.

	I		
d	20	30	40
0.020	N/A (0)	N/A (0)	1.0 (2)
0.040	0.75 (3)	N/A (0)	1.0 (1)
0.060	0.67 (5)	1.0 (1)	0.67 (2)
0.080	0.5 (4)	0.75 (3)	0.63 (4)
0.100	0.5 (4)	0.75 (3)	0.75 (3)
0.120	0.67 (2)	0.75 (3)	0.45 (21)
0.140	0.67 (2)	0.75 (3)	0.45 (21)
0.160	0.68 (5)	0.78 (4)	0.45 (21)
0.180	0.67 (4)	0.78 (4)	0.45 (21)
0.200	0.45 (21)	0.67 (4)	0.45 (21)
0.220	0.45 (21)	0.67 (4)	0.45 (21)
0.240	0.45 (21)	0.67 (4)	0.45 (21)
0.260	0.45 (21)	0.45 (21)	0.45 (21)
0.280	0.45 (21)	0.45 (21)	0.45 (21)
0.300	0.45 (21)	0.45 (21)	0.45 (21)

TABLE V
F-MEASURE SCORES FOR CLIQUE AGAINST EMPIRICAL KNOWLEDGE. NUMBERS IN BRACKETS ARE THE NUMBER OF INSTANCES FROM THE LABELLED DATA THAT WERE IDENTIFIED BY THE ALGORITHM

One of the difficulties in this research, as in most unsupervised data mining, is the problem of initial parameter values. CLIQUE achieved the highest silhouette value, but its lack of stability with other parameter values and poor performance against empirical knowledge indicates that it may be problematic to rely on this algorithm without some other way of selecting good parameters for the algorithm. DBSCAN and iterative k-means both performed well, and their stability with varying-but-similar parameters indicates that these simpler algorithms can perform better in an automated system due to their reliability under small change, which is a key issue with automated systems. While accurate, these models are not perfect, suggesting that verification is an integral part of using these algorithms in an automated system, to ensure that no false assumptions are made about the data.

For these reasons, future work can use the methods shown here could form a part of a larger system, where the clusters are verified against independently derived clusters using other methods such as structural features or forensic analysis, as was performed by [16]. The ability to cross verify the results, as well as using cluster ensemble techniques, could prove to be a powerful way to generate verifiable results in an automated system. This can be used to verify parameter selection, even as the datasets evolve over time, and can also be used in a boosting scenario where the whole is greater than the sum of the parts.

VII. ACKNOWLEDGMENTS

This research was funded by the State Government of Victoria, IBM, Westpac, the Australian Federal Police and the University of Ballarat.

REFERENCES

[1] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings ACM SIGMOD International Conference on Management of Data*, pages 94–105, Seattle, WA, June 1998. ACM Press.

[2] R. Basnet, S. Mukkamala, and A. Sung. Detection of phishing attacks: A machine learning approach. *Soft Computing Applications in Industry*, 2008.

[3] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is 'nearest neighbor' meaningful? *LECTURE NOTES IN COMPUTER SCIENCE*, pages 217–235, 1999.

[4] R. Clarke. Profiling: A hidden challenge to the regulation of data surveillance. *Journal of Law and Information Science*, 4:403, 1993.

[5] F. T. Commission. Identity theft survey report. Technical report, www.ftc.gov/os/2007/11/SynovateFinalReportIDTheft2006.pdf., 2007.

[6] R. Dhamija, J. D. Tygar, and M. Hearst. Why phishing works. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 581–590, New York, NY, USA, 2006. ACM.

[7] H. S. A. H. Dinna N. M. N., Leau Y. B. and Y. A. S. Managing legal, consumers and commerce risks in phishing. In *WORLD ACADEMY OF SCIENCE, ENGINEERING AND TECHNOLOGY*, volume 26, 2007.

[8] M. Ester, H. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining, Portland, OR, AAAI Press*, pages 226–231, 1996.

[9] J. A. Hartigan and M. A. Wong. A K-means clustering algorithm. *Applied Statistics*, 28:100–108, 1979.

[10] C. Herley and D. Florencio. A profitless endeavor: Phishing as tragedy of the commons. *New Security Paradigms Workshop*, 2008.

[11] T. Holz, C. Gorecki, K. Rieck, and F. Freiling. Measuring and detecting fast-flux service networks. In *Proceedings of the Network & Distributed System Security Symposium*, 2008.

[12] M. Jakobsson. The human factor in phishing. *Privacy & Security of Consumer Information*, 7:1–19, 2007.

[13] R. Kosala and H. Blockeel. Web mining research: a survey. *SIGKDD Explor. Newsl.*, 2(1):1–15, 2000.

[14] H. Lei and V. Govindaraju. Speeding up multi-class svm evaluation by pca and feature selection. *Feature Selection for Data Mining*, 2005.

[15] H. Li and N. Abe. Clustering words with the mdl principle. *Proceedings of the 16th conference on Computational linguistics-Volume 1*, pages 4–9, 1996.

[16] S. McCombie, P. Watters, A. Ng, and B. Watson. Forensic characteristics of phishing - petty theft or organized crime? In J. Cordeiro, J. Filipe, and S. Hammoudi, editors, *WEBIST (1)*, pages 149–157. INSTICC Press, 2008.

[17] J. Milletary and C. Center. Technical trends in phishing attacks. *US-CERT, Tech. Rep*, 2005.

[18] L. Parsons, E. Haque, and H. Liu. Subspace clustering for high dimensional data: a review. *ACM SIGKDD Explorations Newsletter*, 6(1):90–105, 2004.

[19] A. Qamra, B. Tseng, and E. Y. Chang. Mining blog stories using community-based and temporal clustering. In *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 58–67, New York, NY, USA, 2006. ACM.

[20] J. S. Quarterman. Phishscope: Tracking phish server clusters. *Journal of Digital Forensic Practice*, 1(2):103–114, July 2006.

[21] T. Ronda, S. Saroiu, and A. Wolman. Itrustpage: a user-assisted anti-phishing tool. In *Proceedings of the 3rd ACM SIGOPS/EuroSys European Conference on Computer Systems 2008*, pages 261–272. ACM New York, NY, USA, 2008.

[22] P. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20(1):53–65, 1987.

[23] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. In *Information Processing and Management*, pages 513–523, 1988.

[24] M. Santini. Genres in formation? an exploratory study of web pages using cluster analysis. *Proc. CLUK*, 5, 2005.

[25] C. Wei, A. Sprague, G. Warner, and A. Skjellum. Mining spam email to identify common origins for forensic application. In *SAC '08: Proceedings of the 2008 ACM symposium on Applied computing*, pages 1433–1437, New York, NY, USA, 2008. ACM.

[26] A. Whitten and J. Tygar. Why johnny can't encrypt. *USENIX Security*, 1999:1, 1999.

[27] H.-J. Zeng, Q.-C. He, Z. Chen, W.-Y. Ma, and J. Ma. Learning to cluster web search results. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 210–217, New York, NY, USA, 2004. ACM.