Aiding Prediction Algorithms in Detecting High-Dimensional Malicious Applications Using a Randomized Projection Technique

Travis Atkison
Department of Computer Science
Louisiana Tech University
Ruston, LA 71270
(318) 257-2940
atkison@latech.edu

ABSTRACT
This research paper describes an on-going effort to design, develop and improve upon malicious application detection algorithms. This work looks specifically at improving a cosine similarity, information retrieval technique to enhance detection of known and variances of known malicious applications by applying the feature extraction technique known as randomized projection. Document similarity techniques, such as cosine similarity, have been used with great success in several document retrieval applications. By following a standard information retrieval methodology, software, in machine readable format, can be regarded as documents in the corpus. These "documents" may or may not have a known malicious functionality. The query is software, again in machine readable format, which contains a certain type of malicious software. This methodology provides an ability to search the corpus with a query and retrieve/identify potentially malicious software as well as other instances of the same type of vulnerability. Retrieval is based on the similarity of the query to a given document in the corpus. There have been several efforts to overcome what is known as 'the curse of dimensionality' that can occur with the use of this type of information retrieval technique including mutual information and randomized projections. Randomized projections are used to create a low-order embedding of the high dimensional data. Results from experimentation have shown promise over previously published efforts.

Categories and Subject Descriptors
D.2.0 **[Software Engineering]**: General – Protection mechanisms

General Terms
Security

ACMSE '10, April 15-17, 2010, Oxford, MS, USA

Keywords
Malicious software detection, information retrieval, *n*-gram analysis, cosine similarity, randomized projections

1. INTRODUCTION
With today's market globalization of software development and the proliferation of malicious attackers, it is becoming almost impossible to have any trust in the software that is loaded on to our machines. To combat these infiltrations, consumers, as well as corporations, are turning to anti-virus software products which contain virus detection engines. "A large percentage of the security software industry is built on the practice of looking for the digital patterns (signatures) that identify known threats." [21] According to the 2007 CSI Computer Crime and Security Survey, anti-virus software, which is a signature based solution, accounted for 54.3 percent of the total budget for industry software security in 2005. [21] Though good at what they do, virus detection engines rely on a database of signatures to detect known malicious applications. Signature based systems inherently limit the detection of new and previously unknown types of malicious attacks. To that end there have been several research attempts to overcome these limitations. One notable avenue of exploration has been to develop tools and techniques based off the field of information retrieval. Previous attempts [1, 8, 12, 20] to use methodologies and techniques from information retrieval and data mining have had some success, but all have been subjected to the 'curse of dimensionality'. The 'curse of dimensionality', first referred to by Bellman [5], generally describes the computational issues related to performing mathematical operations within an extremely high dimensional space. A capability to reduce the number of features to a more manageable number is very useful in countering this problem. Making any decisions based on this high-dimensional data will require the construction of a low-dimensional embedding that preserves the underlying "structure" hidden in the data. This research will use a technique called randomized projection [15, 17] to create the low-dimensional embeddings. This research paper is a follow-on effort of preliminary work [3] the author has done in the area of using randomized projections as a feature extraction methodology when attacking malicious application detection. The goal of this research effort is to show that this is a viable alternate mechanism for reducing high-dimensional data to a more manageable lower dimension. Information retrieval techniques can, then, be applied with better results in both speed to solution and accuracy of

prediction creating a more robust malicious application detection capability.

The following section provides a short background description of information retrieval and randomized projection and discusses malicious software vulnerabilities. In Section 3, the experimental design of this work is discussed including the software and data used. In Section 4, results achieved are described. Finally, in Section 5 the conclusion and future directions are presented.

2. BACKGROUND

Evaluating the effectiveness of a potential solution to the malicious software detection problem, in which a low-dimensional embedding is used to reduce the dimensions of an information retrieval technique, is an important direction in host security research. Below, a description of the information retrieval technique, randomized projection as the dimensionality reduction method and malicious software vulnerabilities used in these experiments are described.

2.1. Information Retrieval

Information retrieval, traditionally, is the "part of computer science which studies the retrieval of information (not data) from a collection of written documents." [4] These retrieved documents' aim is to "satisfy a user's information need." [4] The process can be thought of as combing through a set of documents, called the corpus, to find a certain piece of information that has a relationship to a given entity, called the query. That piece of information can either be an entire document, set of documents or a subset of a document. Within the information retrieval community several methods exist for finding these pieces of relevant information. These methods include vector space models, latent semantic indexing models and statistical confidence models as well as others. "Vector space models are the first approach to represent a document as a set of terms." [16] As their name implies vector space models represent their data as a vector with each dimension being defined as a term which may or may not have a weight associated with it. [23] One of the most common vector space models is cosine similarity. Cosine similarity determines the similarity between two data vectors by measuring the angular distance between them. "Cosine has the nice property that it is 1.0 for identical vectors and 0.0 for orthogonal vectors." [25] The following is the formula used in this work for computing cosine similarity:

$$\textit{Cosine Similarity (Q, D)} = \frac{\sum_i w_{Q,i} w_{D,i}}{\sqrt{\sum_i w_{Q,i}^2} \sqrt{\sum_i w_{D,i}^2}} \qquad 1$$

This formula computes the similarity between a query Q and a document D by summing the individual components of the two entities represented in the formula as w. The individual components for this research, w, are defined as n-grams. An n-gram is "any substring of length n." [4] Here the gram (which will be the composite of the substring) is a byte in hexadecimal form. Therefore, $w_{Q,i}$ is the weight of the *ith* n-gram in the query and $w_{D,i}$ is the weight of the *ith* n-gram in the document.

There have been other efforts [1, 2, 8, 11, 18, 20] to use the information retrieval concept of n-grams as a potential for features. Henchiri et al. [8] and Abou-Assaleh et al. [1, 2] both use the Common N-Gram (CNG) analysis method, which uses the most frequent n-grams to represent a class, to detect malicious applications. Henchiri further limits the number of features by imposing a "hierarchical feature selection process". [8] Marceau [18] puts an interesting twist on the problem of using n-grams as features by having "multiple-length" grams instead of the tradition single n-length gram. Marceau does this by first creating and then compacting a suffix tree to a DAG. [18] Reddy et al. [20] develop their own unique n-gram feature selection measure called, 'class-wise document frequency'

2.2 Randomized Projections

Malicious application detection, following the genre of information retrieval, suffers from the problem that the data, once processed, is encoded in extremely high dimensions. This high-dimensional data limits the kind and amount of analysis that can be preformed. One method for dealing with the reduction of this type of high-dimensional data is known as feature extraction. Feature extraction transforms, either linearly or non-linearly, the original feature set into a reduced set that retains the most important predictive information. Examples of this type include principal component analysis, latent semantic analysis and randomized projection. In randomized projection, "the original high-dimensional data is projected onto a lower-dimensional subspace using a random matrix whose columns have unit lengths." [6] This type of projection attempts to retain the maximum amount of information embedded in the original feature set while substantially reducing the number of features required. By reducing the number of features, greater amounts of analysis can be performed. The core concept has been developed out of the Johnson-Lindenstrauss lemma [9] which states that any set of n points in a Euclidean space can be mapped to $\Re^t$ where $t = O(\frac{\log n}{\varepsilon^2})$ with distortion $\leq 1 + \varepsilon$ in the distances. Such a mapping may be found in random polynomial time. A proof of this lemma can be found in [7].

There have been some efforts [6, 17, 19] that look at using randomized projection techniques for dimensionality reduction. "Randomized projection refers to the technique of projecting a set of points from a high-dimensional space to a randomly chosen low-dimensional subspace or embedding." [27] Minnila et al. [17] are using random projection techniques to map sequences of events and find similarities between them. Their specific application is in the telecommunication field looking at how to better handle network alarms. Their goal is to "show the human analyst previous situations that resemble the current one" [17] so that a more informed decision about the current situation can be made. Though their proposed solution is not perfect, it does show the promise of using randomized projections in a similarity based application. Bingham et al. [6] applies randomized projections to an image and text retrieval problem. In comparison to this research problem, their dimensions are not as large (2500 for images and 5000 for text), but the results are still significant. The purpose of their work was to show that compared to other more traditional dimensionality reduction techniques, such as principal component analysis or singular value decomposition, randomized projections offered a greater detail of accuracy. The authors were also able to show a significant computation saving by using randomized projections over other feature extraction techniques, such as principal component analysis.

In another text retrieval application, Kaski [10] successfully applied randomized projections in his text retrieval application that used WEBSOM, a graphical self-organizing map. Again Kaski turned to randomized projection as a method to overcome the computation expense that made other dimensionality reduction techniques infeasible when handling high-dimensional data sets. After incorporating randomized projection into their tool, the authors gained an additional 5% increase in classification and topic separation than in previous methods used. [10] The following efforts [13, 14, 19] use randomized projection in conjunction with latent semantic indexing. Papadimitriou et al. [19], looking at another information retrieval technique, shows positive results in using randomized projections as a preprocessor to the computationally expensive Latent Semantic Indexing. By simply applying randomized projection to their data before computing the Latent Semantic Indexing, their asymptotic running time for the overall system improved from O(*mnc*) to O($m(\log^2 n + c \log n)$), where m and n are the matrix size, c is the average number of terms per document. [19]

2.3. Malicious Software Vulnerabilities

Today there are a tremendous number of different variations of malicious software vulnerabilities floating around, from buffer overflows to injection attacks to information leakage attacks. This research concentrates on information leakage vulnerability attacks. Information leakage can be defined as when "non-public" information is released (or leaked) without the information owner's knowledge or consent. An information leakage vulnerability can be introduced within an application at design time through malice or through poor programming practices (intentional versus accidental). It can also be introduced by a malicious attacker after deployment by being bundled with, or concealed within, a seemingly non-threatening application. Symantec reported in their bi-annual threat report for the first half of 2005, that "six of the top ten spyware (information leakage) programs were delivered to their victim by being bundled with some other program." [26]

This research effort concentrates on detecting malicious applications before execution, while still packaged in their transporter. This transporter is often called a Trojan horse and the malicious application is referred to as a Trojan. A Trojan horse, similar to the myth, may provide a useful service (for example, a calculator or Notepad) but once executed performs harmful actions. One specific kind of Trojan horse is known as a *binder* or *dropper*. Binders are applications that have the ability to combine (or bind) two or more applications together, yet allow them to run autonomously when executed. This autonomous nature allows the attacker to place a non-threatening, useful service together with one or more malicious applications. The unsuspecting user then executes the application expecting only the useful part; however, unbeknown to them, the malicious application(s) are also executed.

3. EXPERIMENT

The following provides a description of the components of the experimental methodology that was used. All of the experiments were run on commodity hardware running the Fedora Linux operating system. It is very significant that these experiments were able to be completed on commodity hardware. It shows that large specialized machines are not needed to perform malicious application detection and that this work can be broadly applied across almost any level of architecture that researchers/developers may have and still gain the significantly positive results that were obtained and discussed below. In addition, this software and the methods that it supports can easily take advantage of commodity cluster hardware for substantial gains in performance. Details of the software application developed as well as a description of the data set used in the experiments are also described below. This section concludes with an overall experimental design description that provides a description of how the experiments were conducted.

3.1 Similarity Software

The software created for this experiment provides functionality to ingest Windows formatted, binary executables and creates an m-dimensional data space containing vectors that represent those applications. In these experiments, m is the number of total possible n-grams that can be extracted from the ingested applications, one dimension for each possible n-gram. The information stored in each of the dimensions can take on one of several possible values: the absolute total number of occurrences of the particular n-gram in the application, the normalized value of the total number of occurrences of the particular n-gram in the application, or finally, the binary values of a 1 if the application contains the particular n-gram or a 0 if it does not. Once the m-dimensional vectors have been created, the randomized projection matrix algorithm is applied. The random matrix used for the dimensionality reduction can be populated in several ways. Of these the similarity software uses one of two methods: 1) by selecting vectors that are normally distributed, random variables with a mean of 0 and a standard deviation of 1 or 2) by selecting vectors that take on the values of 0, +1 or -1 following a probability distribution of 2/3, 1/6 and 1/6 respectively [6]. The result is a low-dimensional embedding of the original high-dimensional features. For this set of experiments, the cosine similarity algorithm, shown in Eq. 1, was then applied to the query application's vector and the corpus applications' vectors for each set of reductions. This application of the algorithm produced the prediction results.

3.2 Data set

The data set that was compiled together for the experiments described in this section consisted of 1544 Windows formatted binary executable files. None of the files in the data set were larger than 950KB. Of these files 303 were extracted from a fresh installation of the Windows XP operating system. Another 406 were extracted from a fresh installation of the Windows Vista operating system. Both of these sets were obtained by installing the respective operating system in a virtual environment that was installed on a commodity PC. These virtual environments were not connected to the Internet and therefore provided a safe location. This ensured that it would allow for application extraction without the worry of malicious infiltration during the gathering phase of the research effort. This process provided a total of 709 files that were in the data set and that were considered benign. The remaining 835 files for the data set were malicious Trojan horse applications that were downloaded from various websites on the Internet including http://www.trojanfrance.com and http://vx.netlux.org.

3.3 Design

This section describes the overall design of this experiment. The size of the *n*-grams was varied from a 3-byte, 5-byte and a 7-byte window. Only the binary value weighting scheme described above was used for this effort. For the dimensionality reduction portion, a random matrix was projected upon the original high-dimensional data set to produce three separate new low-dimensional embeddings that contained 500, 1000 and 1500 features. The random matrix for the projection was created by randomly selecting values to populate the vectors of the matrix. These values were normally distributed, random variables with a mean of 0 and a standard deviation of 1. The cosine similarity algorithm was then applied to these reduced dimensional data sets to produce a prediction value. The results of these experiments are presented below.

4. RESULTS

The following is a subset of the results generated throughout these experiments, shown here as evidence that applying the randomized projection, feature extraction technique has improved the cosine similarity algorithm when applied to the malicious software detection problem.

4.1 Validation

As with any new method, technique or technology that is introduced, a system for determining its accuracy or validity must also be presented. Validation is a key component to providing feasible confidence that any new method is effective at reaching a viable solution, in this case a viable solution to the malicious application detection problem. Validation is not only comparing the results to what the expected result should be, but it is also comparing the results to other published methods.

To that end several performance values were used to measure and compare the performance of the experiments conducted in this research effort These values include true positive rate (TPR), false positive rate (FPR), accuracy and precision. TPR, also known as recall, "is the proportion of relevant applications retrieved, measured by the ratio of the number of relevant retrieved applications to the total number of relevant applications in the data set." [22] In other words TPR is the ratio of actual positive instances that were correctly identified. FPR is the ratio of negative instances that were incorrectly identified. Accuracy is the ratio of the number of positive instances, either true positive or false positive, that were correct. "Precision is the proportion of retrieved applications that are relevant, measured by the ratio of the number of relevant retrieved applications to the total number of retrieved applications," [22] or the ratio of predicted true positive instances that were identified correctly. All of these values are derived from information provided from the truth table. A truth table, also known as a confusion matrix, provides the actual and predicted classifications from the predictor. The following are the mathematical definitions of the performance formulas as well as the truth table (Table 1)

Table 1. Definition of Truth Table

		Actual	
		Positive	**Negative**
Predicted	**Positive**	a	b
	Negative	c	d

where, *a* (true positive) is the number of malicious applications in the data set that were classified as malicious applications, *b* (false positive) is the number of benign applications in the data set that were classified as malicious applications, *c* (false negative) is the number of malicious applications in the data set that were classified as benign applications, and *d* (true negative) is the number of benign applications in the data set that were classified as benign applications. [24] Below are the formulas for the four performance calculations that were used in this research effort for validation of the predicted results.

$$TPR = \frac{a}{a+c} \quad (2)$$

$$FPR = \frac{b}{b+d} \quad (3)$$

$$Accuracy = \frac{a+d}{a+b+c+d} \quad (4)$$

$$Precision = \frac{a}{a+b} \quad (5)$$

4.2 Experimental Performance

Using the calculated performance values described above, this work can be validated and shown that the proposed randomized projection method added a performance increase to the malicious detection algorithm presented. The performance increase is defined in terms of absolute comparison of the validation methods. Note that the results presented in this paper are just samples of the entire breadth of experiments that were preformed on this data set.

Tables 2, 3 and 4 depict the performance values for the entire data set after sample dimensionality reductions of 500, 1000 and 1500 features respectively and using sample *n*-gram values of 3, 5 and 7 respectively. It must be noted that the non-dimensionality reduced, original data set had upwards of 10^9 features; thus the reductions presented here are significant reductions. These results show that for all dimensionality reduction sizes and various *n*-gram feature sizes the results are extremely positive. Each result has high accuracy, precision and TPR values, approaching 1, while maintaining a low FPR. This means that the applications that are presented to the analyst are, with a high confidence, applications that contain malicious functionality. Furthermore, because of the low FPR and high TPR an analyst will be presented for examination much fewer applications before the malicious applications are scrutinized.

It must be noted that the results for the entire data set without randomized projection applied acquired similar, though consistently lower, accuracies, in the range of an average of over 10% lower. Experimental results found that there was also a substantially lower TPR, up to 30% lower, lower precision and higher FPR. This accentuates the validity that by applying the randomized projection algorithm the malicious software detection algorithm has improved performance.

Table 2. Performance Values for *n*-gram size of 3 and Dimension Reduction of 500

Performance Metric	Threshold Values		
	0.20	**0.25**	**0.30**
TRP	0.91	0.99	0.99
FPR	0.007	0.01	0.02
Accuracy	0.95	0.99	0.98
Precision	0.93	0.91	0.86

Table 3. Performance Values for *n*-gram size of 5 and Dimension Reduction of 1000

Performance Metric	Threshold Values		
	0.15	**0.20**	**0.25**
TRP	0.99	1	1
FPR	0.01	0.03	0.1
Accuracy	0.98	0.98	0.95
Precision	0.98	0.97	0.92

Table 4. Performance Values for *n*-gram size of 7 and Dimension Reduction of 1500

Performance Metric	Threshold Values		
	0.10	**0.15**	**0.20**
TRP	0.95	0.99	1
FPR	0.02	0.02	0.08
Accuracy	0.96	0.99	0.96
Precision	0.98	0.98	0.93

This can be attributed to the 'curse of dimensionality' complicating the prediction method. Significant gains were also made from a computational performance standpoint. The addition of computing the matrix multiplication to acquire the reduced dimensional data set was minimal and can be improved with further refinements and taking advantage of advances in fast matrix multiplication. Furthermore, obtaining a prediction result for an individual application saw over a 100-time increase. Over a small number of predictions, the minimal time to compute the matrix was absorbed. The data space required to contain the non-reduced feature vectors was a factor of 3 greater than that required to hold the reduced data set.

These results of applying the randomized matrix projection algorithm are significant suggesting that a high precision can be maintained without sacrificing accuracy or TPR. It is important to note that most methods used in previous research, report only accuracy value ratings. However, a high accuracy rate may not tell the entire story. For example, assume that the data set contains a high number of true negatives but a low number of true positives predicted. If the value of negative instances is much greater than the number of positive instances then using the formula for Accuracy (Eq. 4) above would produce a high value and using the formula for TPR (Eq. 2) above potentially would produce a low value. The accuracy values reported in the literature reviewed above range from 93% to 98%, so the results presented with this effort are very comparable. Looking at just the accuracy values of Tables 2, 3 and 4, one can conclude that this method is successful. More importantly, it can be concluded that the results presented here are successful when comparing them with the results from the non-reduced feature vectors.

5. CONCLUSION

The results presented here along with the entire set of results gained from the experiments support the hypothesis that applying the technique of random matrix projection as a dimensionality reduction method for the cosine similarity metric has merit in determining if an application may contain a malicious application. This conclusion has been validated using traditional validation measures as well as through performance gains in both size and time derived through experimentation.

There is no claim that this is a complete solution, rather a tool designed to fit into the security administrator's toolbox as a data point or first pass to help reduce the number of applications needing review. This potential reduction in number of applications to sort through can provide an administrator or analyst with valuable time savings by not having to analyze applications that clearly do not contain malicious software. With more and more applications not being developed "in-house," this is a positive result for those responsible for providing secure solutions.

Future efforts for this research are to expand it with the addition of prediction algorithms from the data mining realm, for example decision trees. Also the author plans to investigate additional dimensionality reduction methods and techniques in order to further expand and enhance the analysis capability. Additional research is also planned into determining the threshold values for the similarity algorithm. Determining the key factors in choosing an optimal threshold value is crucial, as can be seen above, to gaining high confidence and to the success rate of the algorithm.

6. ACKNOWLEDGEMENTS

The author would like to thank Dr. Ray Vaughn for his insight and thoughtful review. The author would also like to thank Rebekah Atkison for reviewing this document for its grammatical content. This work was partially supported by the National Science Foundation under grant SCI0430354 04090852.

7. REFERENCES

[1] T. Abou-Assaleh, N. Cercone, V. Keselj, and R. Sweidan, "Detection of New Malicious Code Using N-grams Signatures," *Proceedings of the 2nd Annual Conference on Privacy, Security and Trust*, New Brunswick, Canada, 2004, pp. 193 - 196.

[2] T. Abou-Assaleh, N. Cercone, V. Keselj, and R. Sweidan, "N-gram-based Detection of New Malicious Code," *Proceedings of the 28th Annual International Computer Software and Applications Conference, COMPSAC*. vol. 2, 2004.

[3] T. Atkison, "Applying Randomized Projection to aid Prediction Algorithms in Detecting High-Dimensional Rogue Applications," *Proceedings of the 47th ACM Southeast Conference*, Clemson, SC, 2009.

[4] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*, Harlow, England, Addison Wesley, 1999.

[5] R. Bellman, *Adaptive Control Processes: A Guided Tour*, Princeton University Press, 1961.

[6] E. Bingham and H. Mannila, "Random projection in dimensionality reduction: applications to image and text data," *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* 2001, pp. 245-250.

[7] S. Dasgupta and A. Gupta, *An elementary proof of the Johnson-Lindenstrauss Lemma*, Technical Report TR-99-006, International Computer Science Institute, Berkley, California, USA 1999.

[8] O. Henchiri and N. Japkowicz, "A Feature Selection and Evaluation Scheme for Computer Virus Detection," *6th International Conference on Data Mining, ICDM'06,* 2006, pp. 891-895.

[9] W. B. Johnson and J. Lindenstrauss, "Extensions of Lipschitz mappings into a Hilbert space," *Contemporary Mathematics,* vol. 26, 1984, pp. 189-206.

[10] S. Kaski, "Dimensionality Reduction by Random Mapping: Fast Similarity Computation for Clustering," *The 1998 IEEE International Joint Conference on Neural Networks. IEEE World Congress on Computational Intelligence,* vol. 1, 1998,

[11] J. O. Kephart, G. B. Sorkin, W. C. Arnold, D. M. Chess, G. J. Tesauro, and S. R. White, "Biologically inspired defenses against computer viruses," *Proceedings of the 14th International Joint Conference on Artificial Intellgence*, San Francisco, CA, 1995, pp. 985 - 996.

[12] J. Z. Kolter and M. A. Maloof, "Learning to Detect and Classify Malicious Executables in the Wild," *The Journal of Machine Learning Research,* vol. 7, 2006, pp. 2721-2744.

[13] M. Kurimo, "Indexing Audio Documents by using Latent Semantic Analysis and SOM," *Kohonen Maps,* 1999, pp. 363-374.

[14] J. Lin and D. Gunopulos, "Dimensionality reduction by random projection and latent semantic indexing," *Proceedings of the Text Mining Workshop at the 3rd SIAM International Conference on Data Mining*, 2003.

[15] N. Linial, E. London, and Y. Rabinovich, "The geometry of graphs and some of its algorithmic applications," *Combinatorica,* vol. 15, no. 2, 1995, pp. 215-245.

[16] N. Liu, B. Zhang, J. Yan, Q. Yang, S. Yan, Z. Chen, F. Bai, and W.-Y. Ma, "Learning Similarity Measures in Non-Orthogonal Space," *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management* Washington, D.C., USA: ACM Press, 2004, pp. 334 - 341.

[17] H. Mannila and J. K. Seppänen, "Finding similar situations in sequences of events," *1st SIAM International Conference on Data Mining,* 2001,

[18] C. Marceau, "Characterizing the Behavior of a Program Using Multiple-Length N-grams," *Proceedings of the 2000 Workshop on New Security Paradigms,* Ballycotton, County Cork, Ireland: ACM, 2000.

[19] C. H. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala, "Latent Semantic Indexing: A Probabilistic Analysis," *Journal of Computer and System Sciences,* vol. 61, no. 2, 2000, pp. 217-235.

[20] D. K. S. Reddy and A. K. Pujari, "N-gram analysis for computer virus detection," *Journal in Computer Virology,* vol. 2, no. 3, 2006, pp. 231 - 239.

[21] R. Richardson, *2007 CSI Computer Crime and Security Survey*, Computer Security Institute 2007.

[22] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing and Management: an International Journal,* vol. 24, no. 5, 1988, pp. 513-523.

[23] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Communications of the ACM,* vol. 18, no. 11, 1975, pp. 613-620.

[24] M. Schultz, E. Eskin, E. Zadok, and S. Stolfo, "Data Mining Methods for Detection of New Malicious Executables," *Proceedings of the IEEE Symposium on Security and Privacy,* Oakland, California: IEEE, 2001, pp. 38 - 49.

[25] A. Singhal, "Modern Information Retrieval: A Brief Overview," *Bulletin of the Technical Committee on Data Engineering,* vol. 24, no. 4, 2001, pp. 35 - 43.

[26] Symantec, *Symantec Internet Security Threat Report: Trends for January 05 - June 05*, September 2005.

[27] S. S. Vempala, *The Random Projection Method*, American Mathematical Society, 2004.