

DETECTING MALICIOUS USE WITH UNLABELLED DATA USING CLUSTERING AND OUTLIER ANALYSIS

G.S. Knight

Dept of Electrical and Computer Engineering, Royal Military College of Canada

L. Carosielli

Department of National Defence, Canada

Abstract: Most commercial intrusion detection systems (IDSs) presently available are signature-based network IDSs. Organisations using these IDSs are still experiencing difficulties in detecting intrusive activity on their networks since novel new attacks are consistently being encountered, and analysts can miss legitimate alarms when reviewing large alarm logs that contain a high number of false positives. There has been research investigating the use of data mining techniques to effectively detect malicious activity in an enterprise network. The results of many of these projects have demonstrated that these techniques can be effective when trained/calibrated using labelled datasets. Labelled datasets identify and characterize normal and malicious traffic for use in training/calibrating the detection sensor. However, the creation of labelled datasets is resource intensive. It requires a significant effort by security analysts to create a data set that characterises the traffic in a specific enterprise network environment. This research simulates and analyses malicious activity on an enterprise network to explore the detection of malicious activity with data mining techniques using unlabelled datasets. Semi-discrete decomposition (SDD) is used as a clustering and outlier analysis technique to characterize network traffic as either normal or anomalous.

1. INTRODUCTION

Signature-based network IDSs currently make up the greater part of deployed IDS sensors. Organisations using these IDSs are still experiencing difficulties in detecting intrusive activity on their networks since novel new attacks are consistently being encountered, and analysts can miss legitimate alarms when reviewing large alarm

logs that contain a high number of false positives. Additionally, the original network attack may not have been prosecuted over the network and been visible at the IDS sensor. However, network traffic characterizing the malicious use of the compromised machine may be visible on the network as the attacker uses the machine for his/her own purposes. Often there is more traffic associated with this malicious use than there is associated with the original attack.

Anomaly-based intrusion detection techniques characterize normal network traffic and use this model of normality to identify abnormal, malicious traffic. This can allow an anomaly-based sensor to identify new types of malicious activity, which is difficult for signature-based IDSs.

Previous results [Blo01, Lee00] have demonstrated that data mining techniques can be effective when trained/calibrated using labelled datasets. Labelled datasets identify and characterize normal and malicious traffic for use in training/calibrating the detection sensor. However, the creation of labelled datasets is resource intensive. It requires a significant effort by security analysts to create a dataset that characterizes the traffic in a specific enterprise network environment. Moreover, the effort may have to be repeated at regular intervals as the network traffic patterns change with the changing use patterns of the users of the network and the applications that they use. This research simulates and analyses malicious activity on an enterprise network to explore the detection of malicious activity with data mining techniques using unlabelled datasets. In this project network data collected in normal daily usage is combined with known malicious traffic to create training and validation datasets. Semi-discrete decomposition (SDD) is used as a clustering and outlier analysis technique to characterize network traffic as either normal or anomalous.

Sections 2 and 3 of the paper provide some background information on clustering, outlier analysis, and SDD. Section 4 describes how to use SDD-based anomaly detection to detect malicious traffic in a network environment. The results of the work are presented in section 5 and section 6 provides discussion and conclusion.

2. CLUSTERING AND OUTLIER ANALYSIS

Cluster analysis is a data mining technique that groups objects into classes in an effort to maximize the intraclass similarity and minimize

the interclass similarity. Similarities or dissimilarities are assessed based on attribute values describing the objects. The similarity or dissimilarity can be assessed using distance measures, partitioning methods, hierarchical methods, density methods, grid-based methods, or model-based methods [Han01]. Cluster analysis has been used to build customer profiles as well as in electronic fraud detection.

Outlier analysis tries to identify which data objects do not comply with the general behaviour of the dataset. Hawkins defines an outlier as an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism [Haw80]. Outliers can be detected using statistical, distance-based, or deviation-based tests. For example, a statistical model might define outliers to be observations that lie three or more standard deviations from the mean. Deviation-based tests identify outliers by examining differences in the main characteristics of objects in a group. Objects that are significantly different (i.e. exceed the specified threshold) from the expected values are considered outliers [Han01]. Outlier analysis has been extensively used to detect the fraudulent use of credit cards and cellular telephone service.

3. SEMI-DISCRETE DECOMPOSITION

Semi-discrete decomposition (SDD) [Ole83, Zyt00] is a useful dimension-reduction and clustering technique. SDD has proved itself useful in applications involving storage compression, data filtering, and feature extractions. It has been used for image compression [Ole83, Zyt00], chromosome classification [Kol99], and latent semantic indexing of documents [Kol97]. The use of SDD as a deviation-based outlier detector was first suggested by McConnell and Skillicorn [Con01] when they were able to obtain outlier clusters that other outlier techniques were not able to find in a complex and highly correlated geochemical dataset.

3.1 SDD Description.

Given an $m \times n$ matrix A , the semi-discrete decomposition of A of dimension k is:

$$A_k = X_k D_k Y_k^T$$

Where X_k is an $m \times k$ matrix, D_k is an $k \times k$ matrix and Y_k is an $k \times n$ matrix. The entries of X_k and Y_k are from $\{-1, 0, 1\}$ and D_k is a diagonal matrix of elements of \mathbf{R} . The $m \times n$ decomposition matrix A_k is an approximation of the matrix A . This decomposition does not reproduce A exactly even if $k = n$, but uses very little storage with respect to the observed accuracy of the approximation [Kol97]. The matrix A represents a dataset with m rows representing records, and n columns representing attributes. Each record can be considered as a point in the n -dimensional feature space represented by the attributes. The SDD transforms the n -dimensional space into a new k -dimensional space. This in essence generates a faithful representation of the original data in a lower dimensional space. The rows of X_k are the coordinates of the corresponding rows of A in the space defined by the k new axes described by the rows of Y_k^T . The axes are ordered Y_k^T such that the variation in the original data is concentrated along the earlier axes [Con01]. That is, the axes associated with rows at the top of the matrix define dimensions along which there is a greater variance in data; there is less variance in the original data along the axes associated with rows lower in the matrix.

The columns of X_k are approximately ordered by decreasing variation among the records. Therefore, if we divide the rows of A according to whether they have a -1 , 0 , or 1 in the first column of the corresponding row of X_k , we have separated them into three classes which McConnell and Skillicorn expect captures the most important differences between them. The intuition here is that the 1 and -1 cases identify records/rows that have attributes significantly different from the normal data represented by the 0 case. The second and subsequent columns can be used to further subdivide the rows of A . In each case 1 and -1 values in the row of X_k for that column isolate classes of records that deviate from the norm. This produces a ternary decision tree. The further down the tree (i.e. the longer the sequence of leading zeros in a record in X_k) the more “normal” the records become.

The SDD approximation is formed iteratively. An efficient algorithm for the decomposition can be found in [Con01]. Matlab and C code for this SDD algorithm is available on O’Leary’s website [OLEARY].

The SDD algorithm tries to select new axes along which the data exhibits considerable variance. The algorithm begins by selecting the values of the left most column of X_k , the first axis vector of Y_k^T and the multiplier in D that give the least amount of error between the current approximation, A_x , and the original matrix A . Column values, axes, and multipliers are iteratively selected for successive columns of

X_k . Therefore at each iteration the error between the approximation, A_x , and the original matrix A is reduced.

3.2 SDD as an Outlier Detector.

The general effect of SDD is to find regions of the matrix in which variation in the magnitude of the values is relatively large. During the approximation iterations, large variations are removed first followed by smaller variations. Since SDD tends to emphasize the most unusual patterns in a dataset it can be considered an outlier detector.

The algorithm used to calculate the SDD is sensitive to the initial parameters used in the iterative algorithm. Therefore, the algorithm does not always choose at each step a new axis that will remove the largest amount of error between the decomposition approximation and the actual matrix A . Hence later steps can find a more significant axis that was missed on previous steps. It is therefore possible that the weight the axis has in reducing the amount of error between A_x and A does not always decrease monotonically with the order of the axes in the matrix.

McConnell and Skillicorn [Con01] suggest a reordering scheme for the columns of the SDD matrices. This modification ensures that the strongest outliers appear closest to the top of the decision tree since it reorders the axes so that those with the largest weight appear first.

4. ANOMOLY DETECTION

For the proposed SDD technique to effectively detect malicious use of compromised systems, there are several assumptions that this method relies upon:

- a. the number of normal elements in the dataset is significantly larger than the number of anomalous elements (typically 98% and 2 % respectively),
- b. data instances of the same type will be close to each other in feature space, and
- c. normal data of different sub-types will cluster together, while sub-types of malicious data will not be near the normal region of feature space.

In para a. above, the anomalous traffic is required in the training dataset so that the SDD analysis can differentiate between what normal and anomalous traffic attribute values are. The amount of

anomalous traffic to interleave, 2%, was chosen to maintain consistency with the malicious to normal traffic ratios used in research by Portnoy et al. [Por01]. In this analysis anomalous traffic is considered to be malicious traffic. The assumptions of paras b. and c. are required if the data is to be suitable for clustering and outlier analysis.

4.1 Detection Procedures

In order to verify the proposed malicious use detection theory, the following procedures will be followed:

- a. create the training model,
- b. analyse independent network traffic, and
- c. conduct network diversity validation.

To create the training model an example of normal network traffic is collected on the live target network. Malicious traffic is also collected in a closed lab network by using actual malware tools. This normal usage and malicious usage traffic is interleaved to form the training and validation datasets.

The raw network traffic is then processed using a connection analyser to produce network connection records containing the basic attributes of the network connections.

In the next step in the process, these attributes are normalized by calculating the Z-score of each attribute value [Han01]. The normalized connection data is then analysed using SDD. The code used to perform this analysis is a version of the SDD code provided by O’Leary [OLEARY], which has been modified by the researchers to implement the modifications suggested by McConnell and Skillicorn [Con01]. The resulting SDD X matrix is then used to create a ternary decision tree. The detection threshold is chosen such that the records that have a leading -1 or 1 in the first 50% of the k columns are labelled as anomalous. The value of 50 % of k was chosen empirically as a result of a series of experimental trials.

The training model is then used to analyse previously unseen network traffic. The validation datasets are constructed by interleaving new normal usage data, the malicious traffic data used to create the training model, and malicious traffic data that was not previously used to create the training model. The validation dataset includes malicious traffic data that was not used to create the training model in order to verify how well the proposed technique identifies new malicious use examples. The following procedures are followed to classify the connections in the validation dataset as either normal or anomalous:

- a. convert/normalize the network traffic based on the statistical information of the training set,
- b. compute the validation dataset's SDD values by appending additional records to the training model's SDD X matrix, and
- c. use the newly appended rows of the SDD's X matrix to create a ternary decision tree where records that have a leading -1 or 1 in the first 50% of the k columns are labelled as anomalous.

5. RESULTS

The KDD Cup 1999 (KDD99) dataset is commonly used to verify the effectiveness of IDSs as it contains a wide variety of intrusions simulated in a military network environment [KDD99]. The KDD99 dataset consists of approximately 4,900,000 records. Early in the project two subsets of the KDD99 dataset were analysed to determine if it was worthwhile to continue investigating the feasibility of using the SDD technique. The promising results obtained by analysing the KDD99 dataset were used to further develop the malicious use detection methodology.

In further proof-of-concept trials of the malicious use detection methodology, representative data of daily usage network traffic was required to produce training and validation datasets. Examples of intranet and Internet network traffic were collected on RMC's LAN in order to verify how the proposed methodology performs with varying degrees of data complexity. A more complete presentation of the results is presented in [Car02].

5.1 KDD Results

Due to the large size of the KDD99 dataset, two randomly selected subsets of the overall KDD99 dataset were created for analysis. The KDD 37084 and KDD 98770 datasets respectively contain 37084 and 98770 connection records. 41 attributes were used in the SDD analysis of KDD99.

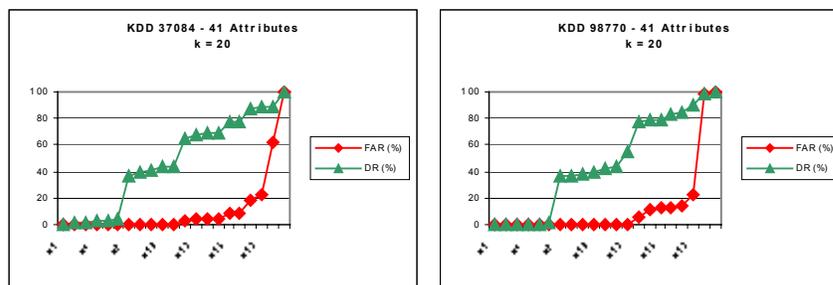


Figure 1: KDD99 Analysis Results

The X-axes of the graphs in Figure 1 represent the columns of the SDD X matrix (i.e X1 = first column, X2 = second column, etc.). These X-axes values are thresholds that correspond to the graph branches in the related ternary decision tree. The detection rates (DRs) represent how many malicious records are classified by the scheme to be abnormal when a given X-axes value, i , is used as a detection threshold. Recall that a -1 or a 1 any leading column (column 0 to i) classifies a record as abnormal. . The false alarm rates represent how many normal records are misclassified by the scheme to be abnormal when a given X-axes value is used as the detection threshold. From Figure 1, we can see that $k = 20$ returned the best results with a 43% DR and 0.1% FAR for KDD 37084 and 39% DR and 0.12% FAR for KDD 98770 if the first 10 columns are selected (X10) in the resulting SDD X matrix.

5.2 RMC Intranet Results

The normal usage traffic was collected from a live LAN segment. The machines and applications run on this network segment are fairly homogeneous. Malicious traffic was collected in a closed lab network by using actual malware tools. This normal usage and malicious usage traffic was interleaved to form the training and validation datasets.

This raw network traffic is then processed using a connection analyser called Argus [ARGUS] to produce network connection records containing the basic features of network connections. Of the 14 basic features provided by Argus, the 11 features listed in Table 1 are used for further analysis to determine if the connection is normal or anomalous. Note that SrcRatio and DstRatio are derived from SrcBytes/SrcPkt and DstBytes/DstPkt respectively. These 11 features were selected for this research since they represent a combination of the intrinsic features used for the KDD99 dataset and the basic connection features used by Iguchi and Goto [Igu99].

Table 1: Eleven Basic Connection Features

| Feature | Description |
|----------|--|
| Duration | Total duration of a connection |
| Type | Type of protocol, e.g. TCP, UDP, etc. |
| Sport | Port number connection is coming from |
| Dport | Port number connection is going to |
| SrcPkt | Total numbers of outbound packets sent through a connection |
| DstPkt | Total numbers of inbound packets received through a connection |
| SrcBytes | Total numbers of outbound bytes sent through a connection |
| DstBytes | Total numbers of inbound bytes received through a connection |
| SrcRatio | Overall bytes per packet rate outbound through a connection |
| DstRatio | Overall bytes per packet rate inbound through a connection |
| State | Basic state of a transaction and how it is terminated |

The malware tools selected for this research were installed on a stand-alone experimental network. Linux and Windows-based malware tools were used. For every experiment, there were typically two or three hosts involved with one collecting the network data using a packet-sniffing tool. Table 2 provides an example of the malware tools used.

The intranet training dataset analysed contained 115,925 daily usage and 3003 malware usage records. Figure 2 provides the results from one of the SDD analysis decompositions of the training dataset. If the threshold for malicious use detection is set at X3 then the expected false alarm rate is 0.697% and the detection rate is 33.5%.

Table 2: Malware Tools Used to Create Datasets

| Backdoors | Scanning | Denial of Service | Covert Tunnels |
|-----------|----------|-------------------|----------------|
| BO2K | Netcat | TFN2K | Covert-tcp |
| Cd00r | Nmap | Winnuke | Httpunnel |
| Cgiback | Nessus | | Rwwwshell |
| Netcat | | | Itunnel |
| Bindshell | | | |
| NetBus | | | |
| Pingback | | | |
| Subseven | | | |

The SDD technique was validated for the intranet by using the matrices that were produced by the training decomposition to analyse another dataset. The validation dataset was prepared in the same way as the training dataset. The new dataset was prepared from different daily traffic logs from the target network, and included a different set of malware traffic data.

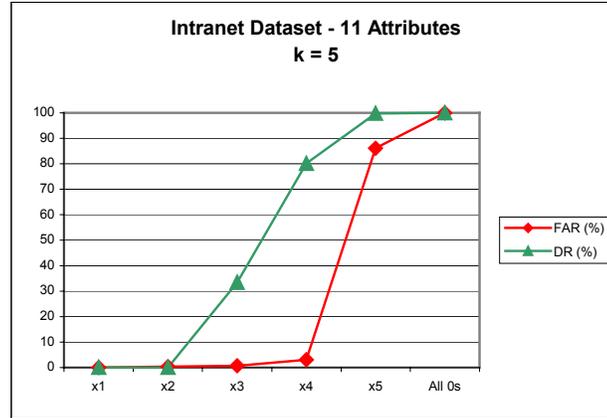


Figure 2: Intranet Training Dataset Analysis Results

The intranet validation dataset analysed contained 179,554 daily usage and 4888 malware usage records. The results obtained by computing the decomposition with $k = 5$, shown in Figure 3, demonstrate that the SDD methodology has a classification DR of 39.4% and a FAR of 0.70% at the X3 threshold. These results are comparable to the 33.3% DR and 0.7% FAR obtained for the intranet training dataset. The SDD analysis of the intranet validation dataset correctly classified 1926 malware records. This includes those that were also present in the training dataset as well as 960 previously unseen records. This demonstrates that the SDD methodology can detect certain types of previously unseen malware.

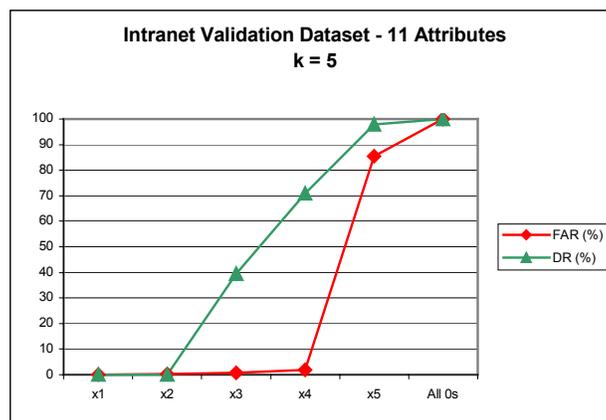


Figure 3: Intranet Validation Dataset Analysis Results

5.3 RMC Internet Results

In other experiments to explore the efficacy of the SDD technique, training and validation datasets were prepared for a sensor located at the RMC Internet firewall. This sensor monitors the much richer, heterogeneous network communications associated with the entire university campus network. This traffic also includes a significant volume of peer-to-peer traffic associated with the university dormitory. The training datasets and the validation datasets were prepared in the same way as for the intranet experiments. The results of the analysis technique when applied to this broad-spectrum network traffic were poor. The technique did not provide useful results in this mode. However, if the data at this sensor is filtered into several data streams in order to isolate specific kinds of network traffic, e.g. selections based on port number, interesting results are observed. Four datasets selected to filter traffic for ports 21, 23, 53, and 1000-6000 produced results similar to those for the intranet and KDD99 experiments. These results demonstrated that the analysis method would be useful in the reduction of a dataset for a specific protocol for further protocol analysis. The majority of the normal connections can be removed while keeping a large number of the malicious connections in the reduced dataset. This simplifies further protocol analysis by enabling the analysts to concentrate on the most interesting connections. The analysis method also detected an actual malicious scan of the RMC network that was coincidentally taking place at the time that the data for the validation dataset was being collected.

6. DISCUSSION & CONCLUSION

The SDD methodology's effectiveness is comparable to the effectiveness of existing methodologies. Labelled dataset techniques [Blo01, Lee00] generally obtained better DRs and FARs, but these improvements have to be weighed against the cost of producing the labelled datasets. The results from KDD 98770 and intranet test datasets had lower DRs than those obtained by Portnoy et al.'s clustering technique [Por01] but the FARs were better. Increasing the detection threshold to higher values of X could increase the DRs to

77.29% and 80.15% respectively for the KDD 98770 and the intranet test datasets, but their respective FARs would increase to 6.09% and 3.02%.

The effectiveness of the SDD methodology for the intranet dataset is further supported by the fact that the intranet dataset contained only 11 intrinsic TCP/IP connection attributes. The KDD99 dataset contained 41 attributes, of which 9 were intrinsic TCP/IP network connection attributes, 13 were content or packet payload based connection attributes, and 19 were time based connection attributes. These 41 attributes would be extremely resource intensive to produce since the analyser has to review each packet's payload and keep the state of numerous connections to produce the required attributes.

An interesting area of future work is the identification of additional attributes or features of network connections that will improve the performance of SDD based anomaly detection. The 11 basic features used in this work may be supplemented with measurements such as: timing latencies on either end of the network connection, recurring use of addresses/ports, or the overlap and interleaving of connection events.

This research has shown that an anomaly detection sensor trained using unlabelled data can be used to detect the malicious use of compromised systems. The SDD methodology obtained acceptable FARs and DRs when analysing intranet datasets. Systems can become compromised in many ways, some not yet known. An unknown compromise is difficult to detect using a signature-based system. This technique is therefore a useful part of a layered security suite. The SDD methodology can also be used to reduce Internet datasets for further specific protocol analysis. This would enable the analysts to concentrate on the most interesting connections.

REFERENCES

- [ARGUS] Argus 2.0.2, <http://www.qosient.com/argus/>
- [Blo01] E. Bloedorn, L. Talbot, C. Skorupka, A. Christiansen, B. Hill, J. Tivel, "Data mining Applied to Intrusion Detection: MITRE Experiences", MITRE Technical Report MTR01W0000103, September 2001.
- [Car02] L.Carosielli, "Detecting Malicious Use With Unlabelled Data Using Clustering And Outlier Analysis", Masters Thesis, Department of Electrical and Computer Engineering, Royal Military College Of Canada, Kingston, Canada, April 2002.
- [Con01] S. McConnell, D.B. Skillicorn, "Outlier Detection Using Semi-Discrete Decomposition", External Technical Report ISSN-0836-0227-2001-452,

- Department of Computing and Information Science Queen's University, Kingston, Canada, November 2001.
- [Han01] J. Han, M. Kamber, "Data Mining: Concepts and Techniques", Morgan Kaufmann, San Francisco, 2001.
- [Haw80] D. Hawkins, "Identification of Outliers", Chapman and Hall, London, 1980.
- [Igu99] M. Iguchi, S. Goto, "Detecting Malicious Activities through Port Profiling", IEICE Transactions on Information and Systems, E82-D(4), April 1999, 784-792.
- [KDD99] KDD Cup 1999 Data, <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>
- [Kol99] T.G. Kolda, D.P. O'Leary, "Computation and Uses of the Semidiscrete Matrix Decomposition", ACM Transactions of Information Processing, 1999.
- [Kol97] T.G. Kolda, D.P. O'Leary, "A Semi-Discrete Matrix Decomposition for Latent Semantic Indexing in Information Retrieval", ACM Transactions on Information Systems, 16, 1997, 322-346.
- [Lee00] W. Lee, S. Stolfo, "A Framework for Constructing Features and Models for Intrusion Detection Systems", ACM Transactions on Information and System Security, 3(4), 2000, 227-261.
- [Ole83] D.P. O'Leary, S. Peleg, "Digital Image Compression by Outer Product Expansion", IEEE Transactions on Communications, 31, 1983, 441-444.
- [OLEARY] SDDPACK, Software for the Semi-Discrete Decomposition, <http://www.cs.umd.edu/users/oleary/SDDPACK/>
- [Por01] L. Portnoy, E. Eskin, S.J. Stolfo, "Intrusion Detection With Unlabeled Data Using Clustering", To Appear in Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001), Philadelphia, USA, 5-8 November, 2001.
- [Zyt00] S. Zyto, A. Grama, W. Szpankowski, "Semi-Discrete Matrix Transforms (SDD) for Image and Video Compression", Technical Report, Department of Computer Science, Purdue University, 2000.