Network Anomaly Detection using Co-clustering

Evangelos E. Papalexakis[†], Alex Beutel[†], Peter Steenkiste[*†]
[*]Department of Electrical & Computer Engineering
[†]School of Computer Science
Carnegie Mellon University,
Pittsburgh, PA, USA
Email: {epapalex, abeutel, prs}@cs.cmu.edu

***Abstract*—Early Internet architecture design goals did not put security as a high priority. However, today Internet security is a quickly growing concern. The prevalence of Internet attacks has increased significantly, but still the challenge of detecting such attacks generally falls on the end hosts and service providers, requiring system administrators to detect and block attacks on their own. In particular, as social networks have become central hubs of information and communication, they are increasingly the target of attention and attacks. This creates a challenge of carefully distinguishing malicious connections from normal ones.**

Previous work has shown that for a variety of Internet attacks, there is a small subset of connection measurements that are good indicators of whether a connection is part of an attack or not. In this paper we look at the effectiveness of using two different co-clustering algorithms to both cluster connections as well as mark which connection measurements are strong indicators of what makes any given cluster anomalous relative to the total data set. We run experiments with these co-clustering algorithms on the KDD 1999 Cup data set. In our experiments we find that soft co-clustering, running on samples of data, finds consistent parameters that are strong indicators of anomalous detections and creates clusters, that are highly pure. When running hard co-clustering on the full data set (over 100 runs), we on average have one cluster with 92.44% attack connections and the other with 75.84% normal connections. These results are on par with the KDD 1999 Cup winning entry, showing that co-clustering is a strong, unsupervised method for separating normal connections from anomalous ones.

Finally, we believe that the ideas presented in this work may inspire research for anomaly detection in social networks, such as identifying spammers and fraudsters.

I. Introduction

Security is an increasingly large problem in today's Internet. However, the initial Internet architecture did not consider security to be a high priority, leaving the problem of managing security concerns to end hosts. This is therefore a growing problem for system administrators having to continually fend off a variety of attacks and intrusion attempts by both individuals and large botnets. An early study from 2001 suggested that there are roughly 2000-3000 denial-of-service (DoS) attacks per week, with many attacks having a lower bound of 100,000 packets-per-second [19]. With the rise of botnets, we can only expect for the prevalence of distributed denial-of-service (DDoS) attacks to have risen over the last decade.

In particular, as social networks have become central to communication internationally, there has been an increase in scrutiny and attacks on these services. For example, Twitter was taken down for multiple hours in 2009 due to a continued DDOS attack [6], and Google, along with numerous other companies, reported being attacked in 2010 [2].

This security issue creates the challenge for system administrators of how to distinguish normal connections of legitimate users from malicious connections. An administrator, in the face of an attack, wants to block such malicious connections without blocking legitimate users. We can never know for sure if any given connection is legitimate or malicious, but we can attempt to isolate a set of connections that stand out from normal user behavior, so as to help system administrators detect attacks. We call this problem network intrusion detection.

In this paper we propose to apply a set of emerging data mining and machine learning techniques to the problem of network intrusion detection. The abstract task description is as follows: We are given a set of connection measurements, each of which consists of a set of connection parameters; we are seeking to categorize each connection as 'normal' or 'anomalous' based solely on this set of parameters (i.e. without using any prior knowledge about the connection, or training data). An 'anomalous' connection may be loosely defined as any connection or set of connections that in some way deviates from common behavior. Given such a set of connections, we look to find methods to easily categorize and display why a set of connections are anomalous, with the hope that with human review we can identify malicious behavior.

In [16], the authors show that certain types of attacks are strongly correlated with only *subsets* of a connection's parameters. In other words, it is possible to determine the type of an attack just by looking at a specific subset of parameters that best characterizes this attack. The only problem is that these 'best' subsets of parameters are not known a-priori. Previous attempts, such as [16] have employed *feature selection* techniques in order to single out the most important parameters overall, and then apply traditional clustering algorithms, on the reduced set of parameters. This particular work essentially does Principal Component Analysis on the data, and subsequently select the most important features via thresholding.

We, however, propose another approach where we don't need to single out the most important parameters for the whole set of attacks. Instead, we seek to find directly the (possibly overlapping) subsets of parameters associated with a set of connections that make such a set abnormal in the larger set of normal connections. There is a class of data

mining algorithms by the name of *Co-clustering* [8], [10], [21] that may effectively tackle this problem. In rough terms, Co-clustering may be described as follows: Given a data matrix, we seek to find subsets of rows that are correlated (usually in the Euclidean sense) with only a subset of its columns (as opposed to the traditional clustering approach, where a subset of the rows has to be similar across all columns/dimensions). This description readily applies to the problem at hand, if we model the set of connection measurements as a matrix, where the rows correspond to each distinct connection, and the columns correspond to the connection's parameters.

The advantage of the proposed approach over the prior art is the fact that our analysis will not rely on any particular labelling of the dataset that might yield biased results according to the labeller's experience of what is normal or abnormal and what an attack might look like. This way, on the one hand, we avoid fine tuning our anomaly detection mechanism so that it works perfectly for just a specific dataset and recognizes only a specific set of attacks that pertains to a certain labelling of a certain dataset (whereas it might fail on another dataset). On the other hand, this allows us to possibly discover attacks and intrusion attempts that go unnoticed or have not been noticed before. By looking at anomalous connections more generally, with human review of the data, we are not dependent on just catching past attacks, but instead reviewing all anomalies that don't match normal behavior.

We use the KDD 1999 Cup Intrusion Detection Dataset [4]. This dataset is in a connection-by-parameter matrix form, and also contains labels for each connection; these labels will be used at the evaluation step of the algorithm, when we will attempt to evaluate the homogeneity of the resulting co-clusters (e.g. connections of the same type end up in the same co-cluster). It is worth pointing out that the labels will not be involved in the actual co-clustering process, for the reasons we explained earlier on.

The outline of the paper is as follows. In Section II we describe our initial motivation for pursuing co-clustering as our method of data analysis. In Section III we describe the methodology for our data mining, explaining the behavior and our usage of the different co-clustering algorithms. In Section IV we give the results of running numerous experiments with two different co-clustering algorithms. In Section V we provide a brief literature survey on network intrusion detection, with particular focus on applications using clustering. Finally, in Section VI we describe our conclusions about the usefulness of co-clustering and how the two algorithms could be used together to detect and analyze anomalous connections.

NOTATION PRELIMINARIES

A scalar is denoted by a lowercase, italic letter, e.g. x. A column vector is denoted by a lowercase, boldface letter, e.g. $\mathbf{x}$. An $I \times J$ matrix is denoted by an uppercase, boldface letter, e.g. $\mathbf{X}$. The Frobenius norm of a matrix $\mathbf{X}$ is denoted by $\|\mathbf{X}\|_F$ and is the square root of the sum of squared elements of the matrix $\mathbf{X}$.

II. MOTIVATION

To justify the use of co-clustering for network anomaly detection, we begin by demonstrating the shortcomings and difficulties of simpler methods, such as Principal Component Analysis, which essentially boils down to regular clustering, considering the entire list of connection parameters.

For this purpose, we use a labelled portion of the dataset, consisting of 494020 connections; 97277 out of them are 'normal' and 396743 are 'attacks.' There are 22 different types of attacks in the labelled data set. For each connection there are 37 recorded measurements, listed in Table I. We form a 494020×37 matrix $\mathbf{X}$ where each row corresponds to a connection and each column corresponds to a parameter. As a preliminary step, we choose to normalize each column of $\mathbf{X}$, dividing by the maximum element of each column; this scaling brings all parameters to the same range and prevents inherently high valued parameters from dominating the matrix. This step is essential because we will use ℓ_2 norms/Euclidean distances. In one of the two algorithms we use, we also take the natural logarithm of the non-zero values of $\mathbf{X}$.

Each connection is now a vector in 37-dimensional space, which makes it hard to view in two or even three dimensions. A standard dimensionality reduction technique, which enables us to visualize high dimensional data, comes by the name of Principal Component Analysis (PCA), which basically boils down to the Singular Value Decomposition of $\mathbf{X}$. Very briefly, $\mathbf{X}$ may be decomposed as

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$$

where $\mathbf{U}, \mathbf{V}$ are the left and right singular matrices (which are orthonormal), and $\mathbf{\Sigma}$ is a diagonal matrix that contains the singular values of $\mathbf{X}$. Let $\mathbf{u}_1, \mathbf{u}_2$ be the first two columns of $\mathbf{U}$. Then, if we plot those two vectors against each other, we get an optimal (in the least squares sense) projection of the data in the 2-dimensional space. In Figure 1 we demonstrate such a projection for $\mathbf{X}$. This procedure was done, after considering the whole set of 37 parameters. We observe that normal and anomalous connections cannot be clearly differentiated without the labels. To corroborate this observation, we also did K-means clustering to the rows of $\mathbf{X}$ [3] with $K = 2$, and measured the ratio of normal connections to the total number of connections in each cluster. For cluster 1 the ratio was $6 \cdot 10^{-4}$, which indicates very few normal connections, and for cluster 2 we got 0.1963, which is also really low. This indicates that even though most of the normal connections ended up in the second cluster, we were not able to tell them apart from the attacks.

The method we propose to use, co-clustering, selects a subset of the rows and a subset of the columns (i.e parameters) of $\mathbf{X}$. A naive and suboptimal way to emulate that behaviour, without actually using co-clustering, is the following. First, we do K-means clustering on the *columns* of $\mathbf{X}$. This way, hopefully, parameters that are similar should end up in the same cluster. Consider now $\mathbf{X}_k$ for $k = 1 \dots K$, being the resulting matrix when keeping only the columns that belong

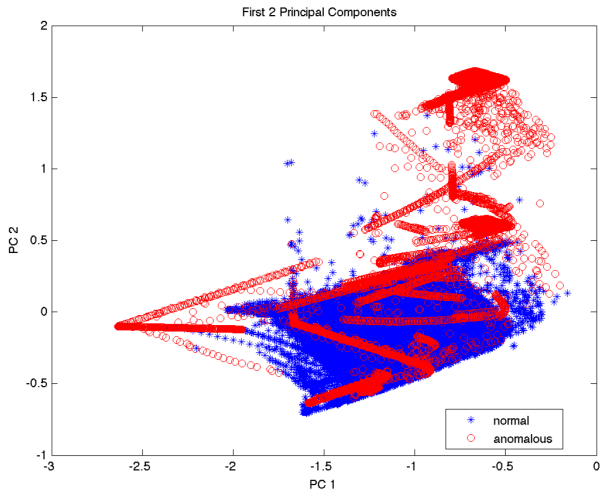


Fig. 1. PCA on $\mathbf{X}$ with points colored by given labels, where blue points were labelled as normal, and red points were labelled as an attack.

Measurement Name	Measurement Type
duration	continuous
protocol_type	symbolic
service	symbolic
flag	symbolic
src_bytes	continuous
dst_bytes	continuous
land	symbolic
wrong_fragment	continuous
urgent	continuous
hot	continuous
num_failed_logins	continuous
logged_in	symbolic
num_compromised	continuous
root_shell	continuous
su_attempted	continuous
num_root	continuous
num_file_creations	continuous
num_shells	continuous
num_access_files	continuous
num_outbound_cmds	continuous
is_host_login	symbolic
is_guest_login	symbolic
count	continuous
srv_count	continuous
serror_rate	continuous
srv_serror_rate	continuous
rerror_rate	continuous
srv_rerror_rate	continuous
same_srv_rate	continuous
diff_srv_rate	continuous
srv_diff_host_rate	continuous
dst_host_count	continuous
dst_host_srv_count	continuous
dst_host_same_srv_rate	continuous
dst_host_diff_srv_rate	continuous
dst_host_same_src_port_rate	continuous
dst_host_srv_diff_host_rate	continuous
dst_host_serror_rate	continuous
dst_host_srv_serror_rate	continuous
dst_host_rerror_rate	continuous
dst_host_srv_rerror_rate	continuous

TABLE I
LIST OF CONNECTION MEASUREMENTS IN THE DATA SET.

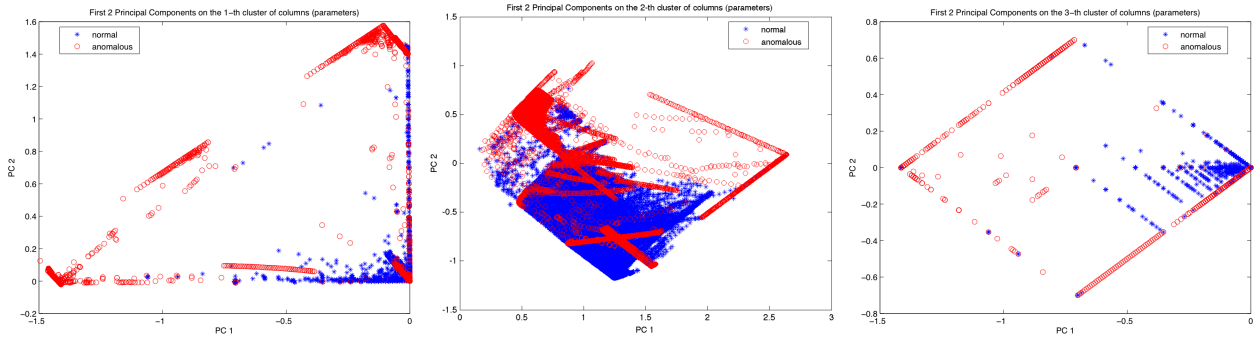
to the k-th cluster. In Fig. 2 we demonstrate this for $K = 3$. We can see that, at least, for $k = 1$ and $k = 3$, the red points (i.e. anomalous connections) are better separated from blue points, in contrast to Fig. 1. This observation provides intuition and motivation, as to why we need to consider only subsets of the parameters, in order to achieve a better result. We proceed to show that by using co-clustering, we can yield even better results.

III. METHOD DESCRIPTION

A. Introduction to Co-clustering

In a nutshell, co-clustering seeks to find subsets of rows and columns in the data matrix that are similar, usually in the least squares sense. Hard co-clustering attempts to *partition* the rows and the columns of the matrix, therefore 1) no overlap is allowed, and 2) all rows and columns have to be assigned to a row/column subset. This problem is NP-hard, and the algorithm we use is an approximation thereof. On the other hand, soft co-clustering relaxes the 'disjoint subsets' constraint, allowing for overlapping between subsets. Because we focus on the application of co-clustering rather than the algorithms themselves, we give only a brief overview of the two co-clustering algorithms below. In the following description, we may use the terms 'cluster' and 'co-cluster' interchangeably.

1) Hard co-clustering: Information Theoretic co-clustering: In [8], Banerjee et al. introduce a class of co-clustering algorithms that employ *Bregman divergences*, unified in an abstract framework. This approach seeks to minimize the loss of information incurred by the approximation of the data matrix $\mathbf{X}$, in terms of a predefined Bregman divergence function. Bregman co-clustering is what we called earlier a hard co-clustering technique, in the sense that it seeks to locate a non-overlapping "checkerboard" structure in the data. Furthermore, it provides the liberty to pick different number of rows and columns that belong to each co-cluster (denoted



(a) $\mathbf{X}_1$ (b) $\mathbf{X}_2$ (c) $\mathbf{X}_3$

Fig. 2. PCA on $\mathbf{X}_k$

by K and L respectively). Let ρ denote a mapping of the I row indices to the K row cocluster partitions, and γ denote a mapping of the J column indices to the L column partitions. Since Bregman co-clustering is hard co-clustering, ρ and γ define disjoint sets of row and column indices, respectively. Therefore, a co-clustering of $\mathbf{X}$ may be described by (ρ, γ).

Furthermore, given a co-clustering (ρ, γ), and a *matrix approximation scheme* $\mathbf{C}$ (we refer the interested reader to the original paper [8] for a detailed presentation of those schemes), a class of random variables $\mathcal{M}(\rho, \gamma, \mathcal{C})$ that preserve the original characteristics of matrix $\mathbf{X}$ is defined. It is shown that the best approximation $\hat{\mathbf{X}}$ of the data matrix that satisfies a given co-clustering (ρ, γ) and a matrix approximation scheme is the one that minimizes the *Bregman Information* of $\hat{\mathbf{X}}$, where the Bregman Information of a random variable X is defined as

$$I_\phi(X) = E\left[\log\left(\frac{X}{E[X]}\right)\right]$$

where $E[\cdot]$ denotes the expected value.

Finally, the optimal co-clustering $(\rho^\star, \gamma^\star)$, given a Bregman divergence d_ϕ, a data matrix $\mathbf{X}$ and a matrix approximation scheme $\mathcal{C}$ is given by

$$(\rho^\star, \gamma^\star) = \arg\min_{(\rho,\gamma)} E\left[d_\phi\left(\mathbf{X}, \hat{\mathbf{X}}\right)\right]$$

where $\hat{\mathbf{X}}$ is the minimizer of the Bregman Information, given (ρ, γ) and $\mathcal{C}$.

In place of d_ϕ, there are two distance or divergence functions that can be employed:

1) **I-Divergence**, which is defined as $d_\phi(x_1, x_2) = x_1 \log(\frac{x_1}{x_2}) - (x_1 - x_2)$, for any numbers $x_1, x_2 \geq 0$.
2) **Squared Euclidean Distance**, which is defined as $d_\phi(x_1, x_2) = (x_1 - x_2)^2$, for any real numbers x_1, x_2.

The above functions may be generalized (elementwise) for matrices, instead of scalar values.

The above Bregman Co-clustering framework includes previously introduced methods such as Information Theoretic Co-clustering [11] (by using the I-Divergence metric), and the Minimum Sum Residuals Co-clustering [9] (by using the Squared Euclidean Distance metric). Bregman Co-clustering is shown to be NP-hard as well, and therefore existing algorithms can only guarantee local minimum solutions. We shall review the issues that arise from this fact in the next section.

In our work, we choose to use the Information Theoretic Co-clustering variant as our hard co-clustering algorithm. We ran the implementation provided by [1] on our data set. For this algorithm, we do not normalize each column of $\mathbf{X}$, before running the algorithm, because we are not optimizing in the least squares sense, and therefore there is no need for such normalization.

2) Soft co-clustering: Sparse Matrix Regression: In [21], co-clustering is formulated as a constrained outer product decomposition of the data matrix, with sparsity on the latent factors of the data matrix $\mathbf{X}$.

The intuition behind SMR is the following. Consider a decomposition of $\mathbf{X}$ in K components, as follows:

$$\mathbf{X} \approx \mathbf{a}_1\mathbf{b}_1^T + \ldots \mathbf{a}_K\mathbf{b}_K^T$$

The above decomposition is also called *bilinear decomposition*, because it breaks down the data matrix into a sum of outer products of vector pairs. Each outer product $\mathbf{a}_k\mathbf{b}_k^T$ is a *rank one* component, which corresponds to the k-th co-cluster. We ideally want the so called *latent factors* $\mathbf{a}_k$ and $\mathbf{b}_k$ to be very *sparse* (i.e. to have many zeros and few non-zero values). This latent sparsity is key: The non-zero elements in $\mathbf{a}_k$ will *select* the rows that participate in the k-th co-cluster, and respectively, the non-zero elements in $\mathbf{b}_k$ will select the columns.

We may compactly organize vectors $\mathbf{a}_k$ and $\mathbf{b}_k$ for $k = 1 \ldots K$ as the columns of matrices $\mathbf{A}, \mathbf{B}$ (of sizes $I \times K$ and $J \times K$ in respect). Thus, SMR co-clustering may be formally stated as the minimization of the following loss function:

$$\|\mathbf{X} - \mathbf{A}\mathbf{B}^T\|_F^2 + \lambda \sum_{i,k} |\mathbf{A}(i,k)| + \lambda \sum_{j,k} |\mathbf{B}(j,k)| \quad (1)$$

where K is the number of extracted coclusters, and λ is a sparsity penalty parameter.

In [22], a coordinate descent algorithm is proposed, in order to solve the above optimization problem. More specifically, one may solve this problem in an alternating fashion, where each subproblem is basically a Lasso [27] problem, common

in statistics and machine learning literature. The algorithm is implemented in Matlab, and therefore has some scalability issues, pertaining to the way that Matlab treats *for*-loops and memory allocation.

The SMR co-clustering algorithm may be characterized as a lossy, soft co-clustering algorithm, in the sense that some rows and columns may not be assigned to any co-cluster, and that overlapping co-clusters can be extracted.

In this work, we choose SMR with non-negativity constraints as our soft co-clustering algorithm. Code for this algorithm may be downloaded from [5]. As a standard preprocessing step, on top of the column normalization that we described above, we choose to scale the data as follows: For each non-zero entry of matrix $\mathbf{X}$, say $\mathbf{X}(i,j)$ we take $\log(\mathbf{X}(i,j)) + 1$. This technique is empirically shown to *compress* the values of $\mathbf{X}$ in such a way that large entries don't dominate smaller ones, especially when dealing with loss functions such as the one shown here.

IV. Analysis of the Results

In using the above methodology, we ran multiple tests using both implementations and used results from SMR co-clustering to spur analysis in information theoretic co-clustering.

A. SMR Co-clustering

Unfortunately, due to the current implementation of the SMR co-clustering algorithm and its quadratic nature, as discussed in the previous section, it was too slow to run over the full data set. As a result, we ran the algorithm on random samples of the data matrix containing $\frac{1}{50}$ of the connections. We ran the algorithm with $K = 2$ over 10 runs. For each run we analyzed the quality of the results by looking at the percentage of each cluster that was labelled as normal or an attack. The CDF of the results of each cluster is shown in Figure 3. As shown in Figure 3(b), SMR co-clustering produces at least one cluster that is consistently almost entirely attacks. The average purity of this cluster is 99.36%; while not as strong of a result , Figure 3(a) shows that the other cluster is still usually a majority normal connections, 73.21% on average. Thus, by merely looking for connections that appear anomalous, the algorithm does a good job of separating the normal connections from the attacks.

For each of the 10 runs we also record the split in parameters. Interestingly, despite being run on different portions of the data set, the algorithm consistently separates the clusters along the same parameters, as shown in Figure 4. Here, we look at the cluster in each run that, based on the labels, contains a majority of attacks. We then record the parameters that distinguish that cluster and record it. We graphed how often each parameter was an indicator of this anomalous cluster. As shown, the same seven parameters always distinguished the anomalous cluster. Not listed on the graph, the seven parameters are `count`, `srv_count`, `same_srv_rate`, `dst_host_count`, `dst_host_srv_count`, `dst_host_same_srv_rate`, and `dst_host_same_src_port_rate`.

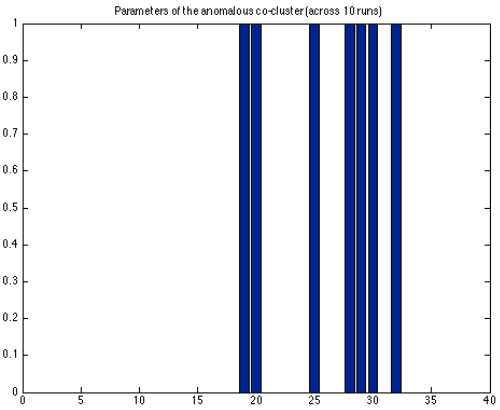


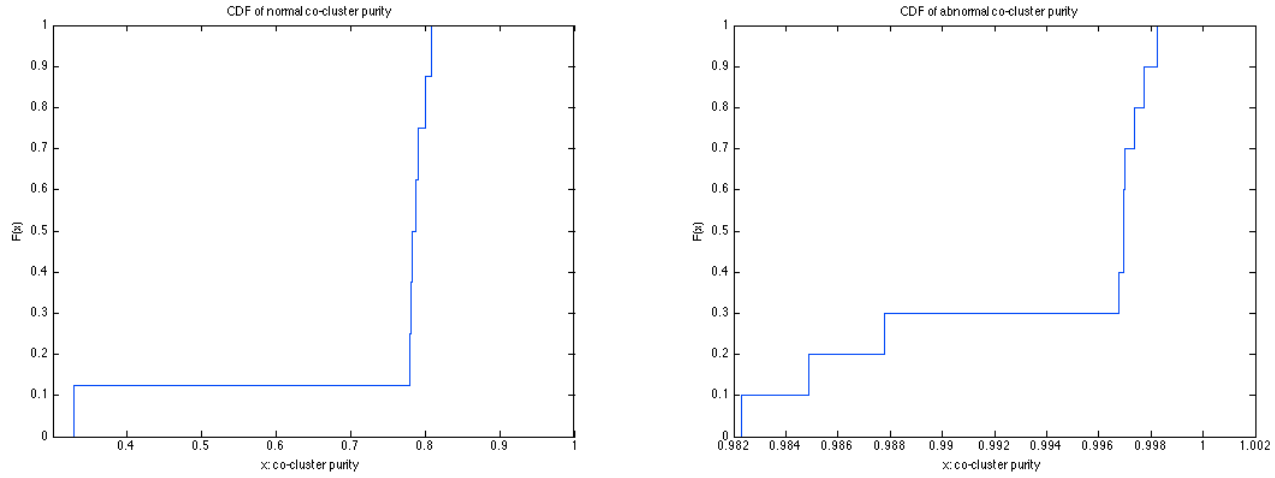
Fig. 4. Parameters associated with the 'anomalous' cluster.

B. Information Theoretic Co-clustering

The information theoretic co-clustering algorithm ran considerably faster than the SMR co-clustering algorithm, enabling us to run it over the entire data set. For all of our runs we set $K = L$ such that the number of column clusters was the same as the number of row clusters. For our initial tests we set $K = L = 2$. We ran the algorithm on the full data set for 100 runs, and as with the previous algorithm, analyzed the purity of each cluster based on the labels given. Again, as shown in Figure 5 we find that at least one cluster has a very high percentage of attack connections, on average 92.44% malicious connections, and the other cluster is still usually mostly normal connections, on average 75.84% normal connections. However, overall the purity of the output of the information theoretic co-clustering is not quite as pure as that from SMR co-clustering, at least with respect to the 'malicious' co-cluster.

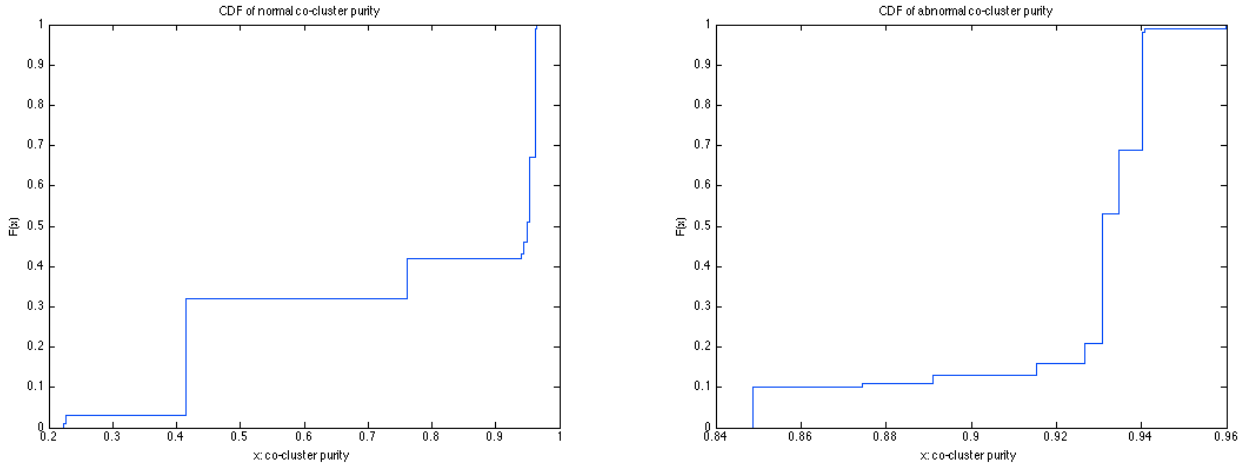
As with the SMR co-clustering, we attempted to isolate certain parameters as being consistent indicators of anomalous connections. However, when recording which parameters were associated with the "anomalous cluster," we found that each parameter was considered an indicator approximately an equal number of times across the 100 runs. We are approximating an NP-hard problem, and as a consequence, it is very probable that the algorithm will get stuck in a locally optimal solution. Thus, we choose to run multiple instances of the algorithm, with different initializations, in order to obtain some solutions that are hopefully close to the global optimum. These locally optimal solutions are also probably the reason why we weren't able to isolate a consistent set of parameters using this algorithm.

Even though the information theoretic co-clustering did not provide clear parameters as indicators, we can use the algorithm's ability to analyze the full data set by combining it with the parameters given by the SMR co-clustering. Because



(a) Normal (b) Anomalous

Fig. 3. CDF on purity of clusters from SMR co-clustering.



(a) Normal (b) Anomalous

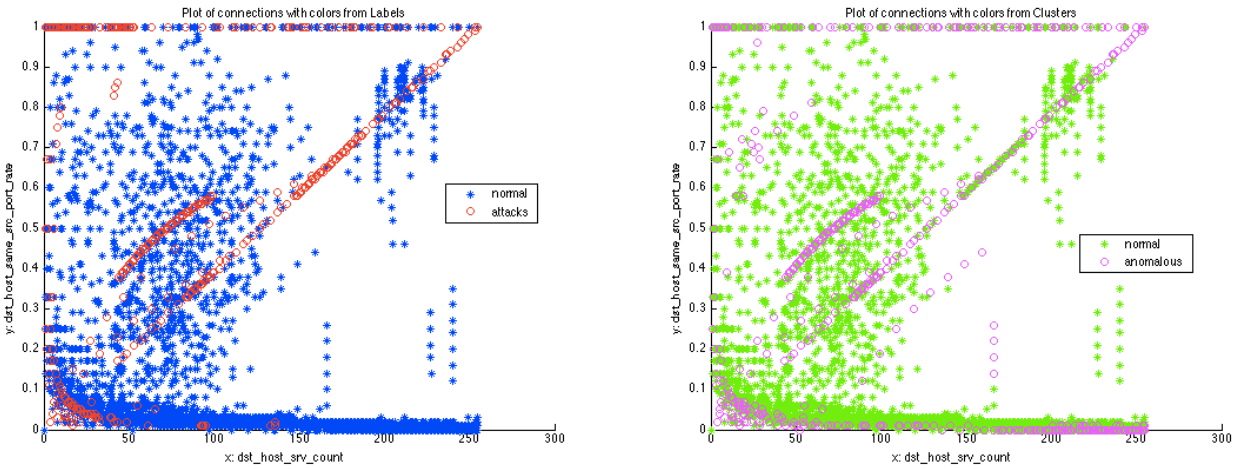
Fig. 5. CDF on purity of clusters from information theoretic co-clustering.

the information theoretic co-clustering provides a hard co-clustering with non-overlapping clusters, and our data set contains numerous types of attacks, we tried running the algorithm with $K = L = 5$, allowing for a more fine-grained break down of the data set. The purity of each cluster is shown in Table II.

Cluster	Number of Connections	Percent Normal	Percent Attacks
1	20,156	97.74%	2.26%
2	116,822	5.30%	94.70%
3	29,591	93.34%	6.66%
4	281,437	0.21%	99.79%
5	46,014	93.85%	6.15%

TABLE II
PURITY OF CLUSTERS WHEN RUNNING INFORMATION THEORETIC CO-CLUSTERING WITH $K = L = 5$.

We then took pairs of clusters and plotted the points in these clusters as a scatter plot of two parameters given from the SMR co-clustering, coloring the points based on which co-cluster the points were in. We compared this plot with the coloring given by the labels themselves. In many cases we found that the patterns that emerge from the labels were the same patterns found by coloring points based on the clusters given by the algorithm. For example, in Figure 6, we see a scatter plot of `dst_host_srv_count` versus `dst_host_same_src_port_rate`. In Figure 6(a) points are colored by the labels given, and in (b) points are colored based on the clusters found in information theoretic co-clustering. The same odd patterns stand out, indicating that little information could be gained in this case from knowing



(a) Labels

(b) Clusters

Fig. 6. A plot of dst_host_srv_count versus dst_host_same_src_port_rate. In (a) colors are based on labels for connections, and in (b) colors are based on the clusters computed by the information theoretic co-clustering.

the true labels over that given by the computed clusters.

Finally, we compare our results with hard co-clustering against the top performing algorithm from the KDD Cup 99 Competition, for which the data set was released [12]. We note, that for our work, we focused on detecting anomalies through a purely unsupervised algorithm. As a result, we do not label any cluster as an attack, but merely point out that its properties, along certain connection parameters, differs from that of the rest of the data set. This differs from the work done by teams competing in the KDD Cup 99, who were expected to label the clusters they found. In particular, the winning entry [24], being in an entirely different spirit than the present approach, uses a classifier that minimizes the *conditional risk*, along with a combination of *bagging* and *boosting*

However, we found that our co-clustering approach produced a cluster purity on par with the winning approach to the KDD cup 99. The cluster labelled as "normal" for the winning entry was 74.6% pure. This is on par, with our results for $L = K = 2$, for which one cluster was 75.84% pure, and less pure than our results with $L = K = 5$ for which we have three clusters with over 90% of the points being "normal." In labeling attacks, cluster purity for the winning entry ranged from 64.8% to 99.9% depending on the attack type. This is again on par with our results, where for $L = K = 2$ we have one cluster 92.44% malicious connections and for $L = K = 5$ we have two "malicious clusters," with purities of 94.7% and 99.79%.

This finding is important for practitioners, since our approach is by no means tailored for the specific problem, as opposed to the winning entries of the competition; thus, it can be easily generalized for a variety of problems, not necessarily pertaining to computer network anomaly detection, by just reformulating the application at hand to the appropriate matrix representation.

V. Related Work

In this section, we present a brief, but comprehensive literature survey of the prior art in anomaly detection, pertaining both to computer networks and social networks. In [20], the interested reader may find a survey of various techniques used for network intrusion detection, whereas in [23] one can find a collection of methods used for anomaly detection in general.

To the best of our knowledge, this work is the first to address the network intrusion detection problem in the co-clustering framework. However, as we pointed out in the introduction, [16] illustrates the usefulness of selecting a subset of the features of a connection, as sufficient in order to identify a malicious connection over a normal one.

In addition, there is a wealth of previous work on intrusion/anomaly detection in computer networks that employs clustering techniques, the majority of which, such as our work, operate in an unsupervised manner. For instance, [25] proposes an unsupervised clustering technique based on single-linkage clustering. In [17] the authors propose an unsupervised density-based and grid-based clustering algorithm, that, according to the authors is able to discover "unforeseen" attacks/anomalies. In [13], the authors introduce a method for discovering hosts that operate as botnets by clustering the communication traffic. In [26] a fuzzy clustering approach for network intrusion detection is proposed, and in [14] a variation of K-means clustering is used, for the same purpose.

Anomaly detection in communication or heterogeneous networks has only recently attracted substantial attention by the research community. For example, [7] proposes a parameter-free method that spots outliers/anomalies on graphs (where graphs may represent a vast range of networks). On a similar pace, [15] introduces methods for anomaly detection taking real-time constraints into consideration. Finally [18] proposes a tensor based approach for anomaly extraction in heteroge-

neous networks.

VI. Conclusions

As explained in our results, both SMR co-clustering and information theoretic co-clustering proved to be powerful tools for detecting anomalous connections out of a large data set without training the algorithms. SMR co-clustering could not compute clusters over the full data set but is effective in finding parameters that are strong indicators of abnormality in the data set and produces clusters that are highly pure. Information theoretic co-clustering on the other hand does not isolate certain parameters consistently and does not always produce *as* pure clusters, but can run on the full data set and be used to plot the full set of connections, against the parameters calculated by SMR co-clustering, colored similarly to that by the labels. Therefore, by combining both methods, co-clustering could be a strong technique for system administrators to review network connections, searching for and reviewing anomalies.

Acknowledgements

The authors would like to thank Prof. Christos Faloutsos for his insightful comments and discussions about this work. This research was funded in part by NSF under award number CNS-1040801 and a National Science Foundation Graduate Research Fellowship. Research was sponsored by the Defense Threat Reduction Agency and was accomplished under contract No. HDTRA1-10-1-0120. Funding was provided by the U.S. Army Research Office (ARO) and Defense Advanced Research Projects Agency (DARPA) under Contract Number W911NF-11-C-0088. Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding parties

References

[1] Bregman co-clustering code on-line. http://www.lans.ece.utexas.edu/facility.html.

[2] Google hack attack was ultra sophisticated, new details show. http://www.wired.com/threatlevel/2010/01/operation-aurora/.

[3] K-means algorithm, wikipedia. http://en.wikipedia.org/wiki/K-means_clustering.

[4] Kdd 99 cup dataset. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[5] Smr co-clustering code on-line. http://www.models.life.ku.dk/cocluster.

[6] Twitter restores service after attack. http://bits.blogs.nytimes.com/2009/08/06/twitter-overwhelmed-by-web-attack/.

[7] L. Akoglu, M. McGlohon, and C. Faloutsos. Oddball: Spotting anomalies in weighted graphs. *Advances in Knowledge Discovery and Data Mining*, pages 410–421, 2010.

[8] A. Banerjee, I. Dhillon, J. Ghosh, S. Merugu, and D.S. Modha. A generalized maximum entropy approach to bregman co-clustering and matrix approximation. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 509–514. ACM, 2004.

[9] H. Cho, I.S. Dhillon, Y. Guan, and S. Sra. Minimum sum-squared residue co-clustering of gene expression data. In *Proceedings of the fourth SIAM international conference on data mining*, volume 114, 2004.

[10] I.S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 269–274. ACM, 2001.

[11] I.S. Dhillon, S. Mallela, and D.S. Modha. Information-theoretic co-clustering. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 89–98. ACM, 2003.

[12] Charles Elkan. Results of the kdd'99 classifier learning contest. http://cseweb.ucsd.edu/~elkan/clresults.html.

[13] G. Gu, R. Perdisci, J. Zhang, and W. Lee. Botminer: Clustering analysis of network traffic for protocol-and structure-independent botnet detection. In *Proceedings of the 17th conference on Security symposium*, pages 139–154. USENIX Association, 2008.

[14] Y. Guan, A.A. Ghorbani, N. Belacel, et al. Y-means: A clustering method for intrusion detection. 2003.

[15] K. Henderson, T. Eliassi-Rad, C. Faloutsos, L. Akoglu, L. Li, K. Maruhashi, B.A. Prakash, and H. Tong. Metric forensics: a multi-level approach for mining volatile graphs. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 163–172. ACM, 2010.

[16] P. Kabiri and G.R. Zargar. Category-based selection of effective parameters for intrusion detection. *International Journal of Computer Science and Network Security (IJCSNS)*, 9(9):181–188, 2009.

[17] K. Leung and C. Leckie. Unsupervised anomaly detection in network intrusion detection using clusters. In *Proceedings of the Twenty-eighth Australasian conference on Computer Science-Volume 38*, pages 333–342. Australian Computer Society, Inc., 2005.

[18] K. Maruhashi, F. Guo, and C. Faloutsos. Multiaspectforensics: Pattern mining on large-scale heterogeneous networks with tensor analysis. In *Proceedings of the Third International Conference on Advances in Social Network Analysis and Mining*, 2011.

[19] David Moore, Geoffrey Voelker, and Stefan Savage. Inferring internet denial-of-service activity. In *In Proceedings of the 10th Usenix Security Symposium*, pages 9–22, 2001.

[20] B. Mukherjee, L.T. Heberlein, and K.N. Levitt. Network intrusion detection. *Network, IEEE*, 8(3):26–41, 1994.

[21] E.E. Papalexakis and N.D. Sidiropoulos. Co-clustering as multilinear decomposition with sparse latent factors. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 2064–2067. IEEE, 2011.

[22] E.E. Papalexakis, N.D. Sidiropoulos, and M.N. Garofalakis. Reviewer profiling using sparse matrix regression. In *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*, pages 1214–1219. IEEE, 2010.

[23] A. Patcha and J.M. Park. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks*, 51(12):3448–3470, 2007.

[24] Bernhard Pfahringer. Winning the kdd99 classication cup: Bagged boosting. http://www.sigkdd.org/explorations/issues/1-2-2000-01/pfahringer.pdf.

[25] L. Portnoy, E. Eskin, and S. Stolfo. Intrusion detection with unlabeled data using clustering. In *In Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001*. Citeseer, 2001.

[26] H. Shah, J. Undercoffer, and A. Joshi. Fuzzy clustering for intrusion detection. In *Fuzzy Systems, 2003. FUZZ'03. The 12th IEEE International Conference on*, volume 2, pages 1274–1278. IEEE, 2003.

[27] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.