Detecting Clusters of Anomalies on Low-Dimensional Feature Subsets with Application to Network Traffic Flow Data

Zhichong Qiu, David J. Miller and George Kesidis
School of EECS, The Pennsylvania State University
{zzq101,djm25,gik2}@psu.edu

Abstract

In a variety of applications, one desires to detect groups of anomalous data samples, with a group potentially manifesting its atypicality (relative to a reference model) on a low-dimensional subset of the full measured set of features. Samples may only be weakly atypical individually, whereas they may be strongly atypical when considered jointly. What makes this group anomaly detection problem quite challenging is that it is a priori unknown which subset of features jointly manifests a particular group of anomalies. Moreover, it is unknown how many anomalous groups are present in a given data batch. In this work, we develop a group anomaly detection (GAD) scheme to identify the subset of samples and subset of features that jointly specify an anomalous cluster. We apply our approach to network intrusion detection to detect BotNet and peer-to-peer flow clusters. Unlike previous studies, our approach captures and exploits statistical dependencies that may exist between the measured features. Experiments on real world network traffic data demonstrate the advantage of our proposed system, and highlight the importance of exploiting feature dependency structure, compared to the feature (or test) independence assumption made in previous studies.

Index Terms

Bonferroni correction, group anomaly detection, Gaussian Mixture Model, p-value, network intrusion detection, BotNet, dependence tree

This work supported in part by a Cisco Systems URP gift.
I. Introduction

Group anomaly detection has recently attracted much attention, with applications in astronomy [14], social media [15], disease/custom control [9][3] and network intrusion detection [11][5][4]. In this work, we focus on group anomaly detection applied to network intrusion detection, where the anomalous groups are either distributed Botnet (Zeus) or peer-to-peer (P2P) nodes generating traffic that deviates from the normal (Web traffic) behavior. Many existing intrusion detection systems (IDSs) only make sample-wise anomaly detections, e.g., in [12], the samples which deviate most from a normal (reference) model are flagged as anomalies/outliers. However, such an approach does not identify anomalous groups (e.g., a collection of BotNet flows), whose samples all exhibit similar behavior. Identifying such groups could be essential for mounting some form of system response or defense. Moreover, individual samples may only be weakly atypical. Thus, a sample-wise IDS may either fail to detect most of the anomalous samples, or may incur high false positives when a low detection threshold is used. By contrast, (weakly) anomalous samples whose anomalies are all “similar to each other” may be strongly atypical when considered in aggregate, i.e., jointly. For example, for an $N = 100$-dimensional feature space, suppose there is a sizeable collection of samples in the captured data batch that are all (even only weakly) atypical with respect to the same feature or the same (small) feature subset. There is a low probability that this occurs by chance, (i.e., under the null). Thus, such clusters of anomalies, each defined by a sample subset and a feature subset, may be strongly atypical, and hence more convincing anomalies, than individual sample anomalies. It should be noted that there is an enormous number of candidate anomalous clusters, considering the conjoining of all possible sample subsets and all possible feature subsets. Thus, a GAD scheme will require some type of heuristic search over this huge space, aiming to detect the most statistically significant cluster candidates. In the sequel, we propose such a GAD scheme. Rather than assuming individual features or outlier events are statistically independent under the null as in [9], [5], in our approach, as in [10], we capture and exploit statistical dependencies amongst the features defining a candidate cluster. Compared to previous works, as shown in our experiments, the proposed scheme is more effective in detecting group anomalies.

The paper is organized as follows. Section II defines the problem and elaborates on related works. Section III describes the proposed model. Section IV evaluates the system performance,
and compares with some recent works. We then discuss some extensions of our system and future works in section V, followed by conclusions.

II. PROBLEM DEFINITION AND RELATED WORK

We assume there is a batch of normal web traffic available at the outset as training set, \( i.e. \ X_l = \{ \tilde{x}_i, i = 1, ..., T_l, \tilde{x}_i \in \mathbb{R}^D \} \), where \( \tilde{x}_i \) is a \( D \)-dimensional feature vector representing the \( i \)-th training traffic flow, and where we assume the number of training flows \( T_l \) is large enough to learn an accurate reference model (null hypothesis). These traffic flows can either be generated and captured in a sandbox environment, or sampled from a domain of interest (data warehouse, enterprise network) in real time under normal operating conditions. Given a model of normal network traffic learned based on \( X_l \), our goal is to interrogate a capture batch of unknown traffic flows \( X_u = \{ \tilde{x}_i, i = 1, ..., T_u, \tilde{x}_i \in \mathbb{R}^D \} \), seeking to identify latent groups of Botnet or P2P traffic, with the flows in each such group exhibiting similar behavior. This has been previously considered in [5], where the authors used the samples in \( X_l \) to estimate bivariate Gaussian Mixture Models (GMMs), on all feature pairs, representing the null hypothesis. These bivariate GMMs were used to evaluate mixture-based \( p \)-values for all pairs of features. Assuming the features (tests) are statistically independent, a joint significance score function was defined for a given candidate cluster, specified by its sample subset and feature subset, with a Bonferroni correction used to account for multiple testing. Instead of exhaustively searching over feature subset candidates at order \( K^4 \), the authors proposed to trial-add individual features only to the top-ranking candidate feature subsets (in terms of the Bonferroni corrected score) at order \( K - 1 \). Furthermore, the authors showed that the computational complexity of determining the optimal (in terms of the joint score) sample subset given the feature subset fixed is linear in \( T_u \), once the samples in a given feature subset are ranked by their aggregate p-values. However, the independent test assumption used in [5] becomes grossly invalid as more and more features are included in a cluster, which limits the proposed model’s detection accuracy for increasing \( K \). A related framework was also proposed in [9], albeit assuming categorical attributes. Here, the

---

1 A flow is a bidirectional communication sequence between a pair of nodes in a network.

2 Unknown in the sense that we do not know which if any of these flows represent outliers or attacks.

3 A \( p \)-value is the probability that an event is more extreme than the given observation.

4 We use “order” to denote the maximum feature dimension considered.
authors built a single, global null hypothesis Bayesian network based on $X_i$. They then assigned categorical-based p-values to samples in $X_u$, with a cross entropy based scoring criterion used to efficiently search for the best feature and sample subset candidates. A limitation of this approach is that the statistical tests are again assumed to be independent.

We herein describe and experiment with a method of anomaly detection that extends [9], [5] and is closely related to [10]. The method captures dependencies between the features in a candidate cluster by a dependence tree structure, and uses this model to help evaluate joint p-values for cluster candidates. As in [5], the Bonferroni corrected score is used as the objective function for evaluating the best cluster candidates (defined by their sample and feature subsets). The candidate with the best such score is detected as a cluster of anomalies. Whereas in [9] a single global (null model) Bayesian network is used to assess candidate clusters, in [10] and in the current work a local, customized \textit{cluster-specific} dependence tree model is used to assess each candidate cluster.

III. PROPOSED MODEL

A. Mixture-based P-values for Singletons and Feature Pairs

Consider a (sample, feature) index pair $(i, j)$ and let $I_i^{(j)}$ be an indicator variable for the event that the $j^{th}$ feature value of the $i^{th}$ sample, $x_i^{(j)}$, is an outlier with respect to the null distribution for feature $X^{(j)}$. Let $O^{(j)}(x_i^{(j)})$ be a subset of the real line such that, $\forall y^{(j)} \in O^{(j)}(x_i^{(j)})$, $y^{(j)}$ is “more extreme” than the given observation $x_i^{(j)}$. One good definition for this set, consistent with evaluating a 2-sided p-value for a unimodal, symmetric null for $X^{(j)}$, is:

$$O^{(j)}(x_i^{(j)}; \mu^{(j)}) = \{y^{(j)} : |y^{(j)} - \mu^{(j)}| \geq |x_i^{(j)} - \mu^{(j)}|\},$$

where $\mu^{(j)}$ is a representative (mean) value for feature $X^{(j)}$. Given the component means $\mu_l^{(j)}$, $l = 1, ..., L_j$, of an $L_j$-component Gaussian mixture null, let $M^{(j)}(x)$ be a function that maps $x$ to the mixture component index set $\{1, 2, ..., L_j\}$, i.e., it indicates which mixture component generated $x$. Also, let $Y_j$ be a random variable distributed according to the mixture density $f_{X_j}(x)$. Then, for a given observation $x_i^{(j)}$, we define the binary random variable $I_i^{(j)}$, where $I_i^{(j)} = 1$ if $Y_j$ is
more extreme under the null than \( x_{i}^{(j)} \). Then, we can write the singleton mixture p-value as:

\[
P[I_{i}^{(j)} = 1] = P[Y_{j} \in \bigcup_{l=1}^{L} ((O^{(j)}(x_{i}^{(j)}; \mu_{l}^{(j)})) \cap (M^{(j)}(x_{i}^{(j)}) = l))] = \sum_{l=1}^{L} P[Y_{j} \in O^{(j)}(x_{i}^{(j)}; \mu_{l}^{(j)})]P[M^{(j)}(x_{i}^{(j)}) = l]. \tag{1}
\]

Here, an extreme outlier event is conditioned on \( x_{i}^{(j)} \) having been generated by component density \( l \). The probability \( P[Y_{j} \in O^{(j)}(x_{i}^{(j)}; \mu_{l}^{(j)})] \) is the two-sided Gaussian p-value, integrating over the region \(|y - \mu_{l}^{(j)}| \geq |x_{i}^{(j)} - \mu_{l}^{(j)}|\), while \( P[M^{(j)}(x_{i}^{(j)}) = l] \) is the \textit{a posteriori} probability that \( x_{i}^{(j)} \) was generated by component \( l \).

Similarly, for a \textit{pair} of observations \((x_{i}^{(j)}, x_{i}^{(k)})\), we have the second order mixture p-value:

\[
P[I_{i}^{(j)} = 1, I_{i}^{(k)} = 1] = \sum_{l=1}^{L} P[Y_{j} \in O^{(j)}(x_{i}^{(j)}; \mu_{l}^{(j)}), Y_{k} \in O^{(k)}(x_{i}^{(k)}; \mu_{l}^{(k)})] \cdot P[M^{(j,k)}(x_{i}^{(j)}, x_{i}^{(k)}) = l].
\]

Here, \( P[Y_{j} \in O^{(j)}(x_{i}^{(j)}; \mu_{l}^{(j)}), Y_{k} \in O^{(k)}(x_{i}^{(k)}; \mu_{l}^{(k)})] \) integrates the \( l \)-th component bivariate Gaussian density over the region

\[
\{ (y_{j}, y_{k}) : |y_{j} - \mu_{l}^{(j)}| \geq |x_{i}^{(j)} - \mu_{l}^{(j)}|, |y_{k} - \mu_{l}^{(k)}| \geq |x_{i}^{(k)} - \mu_{l}^{(k)}| \}.
\]

This region consists of the union of four unbounded rectangular regions in the plane, as illustrated in Figure 1.

In this work, a sample’s anomalousness on a given feature subset is estimated by a joint p-value, with statistical dependencies between features accounted for by a dependence tree (DT) structure [2]. Since the dependence tree [2] is based on first and second order probabilities, the joint p-value will be based on the singleton and second order mixture p-values, as given above. A smaller joint p-value indicates a sample is more anomalous under the given feature subset.

B. Scoring Clusters

Let \( \{I_{c}, J_{c}\} \) denote cluster candidate \( c \), \( I_{c} \) its sample subset and \( J_{c} \) its feature subset. Let \( T_{c} = |I_{c}|, N_{c} = |J_{c}| \). Note that p-values are uniformly distributed on \([0, 1]\) under the null. Thus,
given a cluster with feature subset $J_c$, from a test batch of size $T_u$, the probability that at least one cluster with $T_c$ samples has a smaller p-value than $P[\bigcap_{j \in J_c} (I_i^{(j)}) = 1]$ is:

$$1 - (1 - \prod_i P[\bigcap_{j \in J_c} (I_i^{(j)}) = 1])^{C(T_u, T_c)} \text{ (2)}$$

Here, $C(T_u, T_c) = \binom{T_u}{T_c}$, i.e. it is the number of combinations and implements multiple testing correction, accounting for all possible sample subset configurations in a cluster with $T_c$ samples, from a test batch of size $T_u$. In principle, (2) provides a sound basis at least for directly comparing all cluster candidates with the same feature subset $J_c$. However, it does not allow comparing pairs of cluster candidates with any configurations of $(T_c, N_c)$, because all possible feature subset configurations at a given order, $N_c$, have not yet been properly multiple-testing corrected. Also, (2) requires evaluation of the joint p-value $P[\bigcap_{j \in J_c} (I_i^{(j)}) = 1], \forall i \in I_c$, which in general depends on the joint density function for $(X_{j_1}, X_{j_2}, ..., X_{j_{N_c}}), j_m \in J_c, m = 1, ..., N_c$. When $D$ is large, it is not practically feasible to learn and store these $\binom{D}{N_c}$ joint null density functions, i.e., for all possible combinations of features up to order $N_c$. Thus, it appears some tractable representation of $P[\bigcap_{j \in J_c} (I_i^{(j)}) = 1]$ is needed. An obvious temptation is to assume that $I_i^{(j)}$ and $I_i^{(j')} \text{ independent } \forall j, j' \in J_c, j' \neq j$. But this is a very poor assumption, consistent with assuming the features are independent.

To address the above problems, we seek to modify (2) in two respects. First, we propose to multiple test correct both for the different sample and the different feature subsets, given...
a cluster candidate with \((T_c, N_c)\). In this approach, instead of the exponent being the number of combinations, it becomes the product of combinations on samples and combinations on features. Based on the Bonferroni approximation of \((2)\), we have the joint score function

\[
S(I_c, J_c) = \left(\frac{D}{N_c}\right)^T \prod_{i \in I_c} P[\bigcap_{j \in J_c} (I_i^{(j)}) = 1].
\]

For this joint significance measure, we can efficiently determine the optimal sample subset, given a fixed feature subset, by greedy sequential sample inclusion, in sorted joint p-value order. This is due to the unimodality of this Bonferroni approximated joint significance measure, as a function of the number of samples included in a cluster’s sample subset (see next subsection).

Second, a rich, tractable, joint probability mass function model that does capture statistical dependencies is a restricted form of Bayesian network, based exclusively on first and second order distributions, i.e., the dependence tree (DT), which factorizes the joint distribution \(P[\bigcap_{j \in J_c} (I_i^{(j)}) = 1]\) as a product of first and second order probabilities \((2)\). In \((2)\), it was shown that, even though there is an enormous number of unique dependence tree structures, one can efficiently find the globally optimal dependence tree, over all such structures, maximizing the dataset’s log-likelihood, by realizing that this can be recast as a maximum weight spanning tree problem, with the pairwise weights defined as the mutual information between the pairs of random variables.

The maximum weight spanning tree can be efficiently solved via Kruskal’s algorithm, with complexity \(O(N_e^2 \log(N_c))\). Hence, given any candidate feature subset \(J_c\), Kruskal’s algorithm can be applied to determine the DT that maximizes the likelihood measured on \(X_l\), i.e., the null hypothesis is determined, consistent with the given candidate feature subset \(J_c\).

Based on a given DT structure, \(P[\bigcap_{j \in J_c} (I_i^{(j)}) = 1]\) factorizes as a product of first and second order distributions, i.e., \(\forall i \in I_c:\)

\[
P[\bigcap_{j \in J_c} (I_i^{(j)})] = P[I_i^{(j_1)}]P[I_i^{(j_2)}|I_i^{(j_1)}]...P[I_i^{(j_{N_c})}|I_i^{(j_{N_c-1})}],
\]

where we use \(j_1\) to denote the root node of the DT representing \(J_c\).

It is apparent from \((3)\) that, for any feature subset, one can represent the joint p-value of a given sample by its first and second order mixture p-values. That is, for any feature pair \((j, k)\),

\[
P[I_i^{(j)}|I_i^{(k)}] = \frac{P[I_i^{(j)}, I_i^{(k)}]}{P[I_i^{(k)}]}.
\]

The numerator and denominator are, respectively, the second and first order mixture-based p-values that we defined earlier. Also note that, in order to evaluate the first order mixture p-value \(P[I_i^{(k)}]\), we marginalize feature \(j\) from the bivariate GMM for the feature pair \((j, k)\). This gives us the GMM for feature \(k\).
C. Identifying the Optimal Sample Subset \( I_c \), Given Fixed \( J_c \)

Given a fixed \( J_c \) and associated DT, we would like to choose the sample subset \( I_c \) to minimize the joint score function:

\[
S(I_c, J_c) = \left( \frac{D}{N_c} \right) \left( \frac{T_u}{T_c} \right) \prod_{i \in I_c} P[ \cap_{j \in J_c} (I_i^{(j)}) = 1].
\]

(4)

It is in fact easily shown that this objective function is globally minimized by the following procedure: i) sort the samples in increasing order of their joint p-values \( P[ \cap_{j \in J_c} (I_i^{(j)}) = 1] \); ii) sequentially include the samples on the sorted list into \( I_c \), until the objective function no longer decreases. This procedure globally minimizes over \( I_c \) given fixed \( J_c \).

D. Overall Search Algorithm

First, using the normal samples in \( X_l \), all the first and second order null GMMs are separately trained. Mutual information for all feature pairs is then calculated based on the bivariate GMMs. This is achieved by generating \( M = 10^6 \) samples from a given bivariate GMM distribution, and then estimating the mutual information by

\[
\frac{1}{M} \sum_{n=1}^{M} \log \left( \frac{f_{X_1 X_2}(x_1^{(n)}, x_2^{(n)})}{f_{X_1}(x_1^{(n)}) f_{X_2}(x_2^{(n)})} \right).
\]

We then detect clusters in \( X_u \) sequentially, in a rank-prioritized fashion, according to the joint score \( S(I_c, J_c) \). The algorithm operates on an enormous space of candidate clusters even if the feature space itself is only 5Separately learning each marginal and pairwise feature GMM using the common training set \( X_l \) will not ensure consistency with respect to feature marginalizations. Specifically, a marginal-consistent collection of univariate and bivariate density functions should satisfy the following: if we consider any feature pairs \((i, j)\) and \((j, k)\), marginalizing out feature \(i\) from the \((i, j)\) bivariate density and marginalizing out feature \(k\) from the \((j, k)\) bivariate density should lead to the same marginal density for feature \(j\). However, when the univariate and bivariate distributions are Gaussian mixtures, with a non-convex log-likelihood function (and with BIC-based model order selection separately applied to choose the number of components for each GMM), separate application of EM-plus-BIC to learn each GMM density function does not ensure a set of marginal-consistent distributions. This property is not centrally important here, however, since our main concern is only to learn marginal and pairwise density functions that allow accurate assessment of p-values. Accordingly, in this work we will apply EM-plus-BIC separately, to learn each low-order GMM.

One approach to obtain marginal-consistent low-order distributions is to simply learn the single GMM for the joint distribution on the full feature vector, \( X \). This determines (via marginalization) all lower-order distributions (which are also GMMs, and which are guaranteed to be marginal-consistent). However, this strategy suffers from the curse of dimensionality. Alternatively, we refer the interested reader to [10], where a procedure for directly, jointly learning a marginal-consistent set of low-order GMMs is elaborated.
modestly sized \((D)\). We start by sweeping over feature subset candidates at low orders and, for tractability, only the “most promising” candidates at higher orders, with candidate feature subsets at order \(K\) formed by “accreting” new features to the best-scoring candidates at order \(K - 1\). For each candidate feature subset \(J_c\), its DT is first learned and its associated, optimal subset \(I_c\) is then determined using the method described in section III.C. Evaluating all candidates at all feature subset orders, the one with the best score function value at each order \(N_c\) is recorded. The cluster with smallest Bonferroni-corrected score \(S(I_c, J_c)\) is then forwarded as detected. Its samples are then removed from the test batch. Subsequent cluster detections can then be made following the same procedure. Cluster detections are thus made (in general) in order of decreasing joint significance.

IV. Experimental Setup and Results

Our experiments focus on detecting Zeus botnet and P2P traffic among normal Web traffic. The Web packet-flows are obtained from the LBNL repository \([6]\). This dataset contains Web traffic on TCP port 80, with specified time-of-day information. Specifically, the experiments in this paper are based on three datasets named “200412215-0510.port008”, “20041215-1343.port008” and “20041215-1443.port010”. The protocols to obtain normal, P2P and BotNet network traffic are the same as in \([5]\), i.e., we used the port-mapper in \([16]\) to identify P2P traffic in these files by a C4.5 decision tree pre-trained in another domain (the Cambridge dataset \([7]\)). The Zeus Botnet traffic are obtained from another domain \([13]\).

A. Feature Space Selection and Representation

Firstly, we did not use layer-4 port number features for purposes of detection \([16]\), \([1]\). Also, we did not consider timing information herein because the Zeus activity was recorded on another domain \([1]\). In \([1]\), previous efforts were made to detect BotNet and P2P traffic using the well-known feature representation for network intrusion detection from \([8]\). The authors found that these features, though able to detect some attack activity, could not successfully discriminate BotNet or P2P from normal Web traffic, i.e., BotNet and P2P traffic appear as “normal” Web activity according to the features of \([8]\), \([1]\).

To capture the intrinsic behavior of BotNet and P2P packet-traffic, we note that most Zeus BotNet traffic involves masters giving command (control) messages, while slaves execute the
given commands. In the case of P2P, nodes often communicate in a bidirectional manner, exchanging relatively large packets in both directions. Normal/background Web traffic, on the other hand, tends to involve server-to-client communications.

Hence, we seek to preserve the bidirectional packet size sequence information as feature representation for different traffic flows. This feature representation was previously considered in [5], [11]. The authors used the first $N$ (we set $N = 10$ in our experiments) packets after the three-way hand shake of each TCP flow. Then a feature vector of dimension $2N$ is defined, specified by the sizes and directionalities of these $N$ packets. Traffic are assumed to be alternating between client-to-server (CS) and server-to-client (SC). A zero packet size is thus inserted between two consecutive packets in the same direction to indicate an absence of a packet in the other direction. For example, if the bidirectional traffic is strictly SC, a zero will be inserted after each SC packet size. This $2N$-dimensional feature representation preserves bidirectional information of a given TCP flow, which is essential for discriminating between P2P, Zeus and normal Web traffic.

B. Performance Metrics

Our algorithm detects clusters (groups) in a sequential fashion. For each extracted group, we rank the samples in the group by their associated joint p-values on the given feature subset. These samples will be sequentially removed from the test batch, with the system then continuing to extract groups until the test set is depleted. Then we sweep out an ROC curve based on these rank-ordered detected samples. A larger area under the ROC curve indicates earlier detections of anomalous groups, which implies the effectiveness of the intrusion detection system. We compare our system’s performance with a GMM based anomaly detector, trained by normal samples, on the whole feature space. For this detector, we rank the test samples based on their data likelihood under the GMM, and sweep out an ROC curve. We also compare with the approach presented in [5], which assumes significance tests are independent (denoted “Independence tests”), and with the recent work presented in [9] with a slight modification – instead of discretizing feature values consistent with [9], we use a single dependence tree null distribution learned on $X_i$ and our proposed joint p-value for continuous features, $P[\bigcap_{j \in J_c} (I_i^{(j)}) = 1]$. We denote this variation on the approach in [9] by “single Bayesian Net.” There are two generalization performance measures of interest on the test set: one is the aforementioned ROC area under curve (ROC AUC) as a function of the maximum feature subset size for a cluster, $K_{\text{max}}$. The other is the top
100 precision rate, defined as the fraction of anomalous samples amongst the first 100 detected samples. Lastly, instead of exhaustively searching over all feature subsets at order $K$, we trial-add individual features to the top candidate feature subsets from order $K - 1$. At each order $K$, starting from order 2, we only consider the top 500 candidates from order $K - 1$.

Two different sets of experiments were performed, one on synthetic data, and the other on the network data mentioned earlier. In the synthetic dataset experiment, we used one unimodal Gaussian with 10 dimensions to generate normal samples and two additional unimodal Gaussians to generate two distinct anomalous clusters. The two anomalous clusters use the same distribution as the normal distribution for nine of the ten features. Thus, they deviate from the normal (null) distribution only on a single feature dimension (this “informative” feature dimension was different for the two clusters). Their corresponding sample subsets consist of 2.5% of the whole data batch $X_u$ (so the proportion of anomalous samples in $X_u$ is 5% of the total). The variance of the informative features was chosen to be the same as that of the normal features, $\sigma_n^2$. Moreover, the mean of the informative feature under an anomalous cluster was chosen to be two standard deviations away from the mean under the normal class, i.e. $|\mu_n - \mu_a| = 2\sigma_n$, where we use subscripts $n$ and $a$ to denote ‘normal’ and ‘anomalous’, respectively. Thus, if we consider only the informative feature dimension, the Bayes error rate in discriminating normal from anomalous is 15.87%. After generating the synthetic data batch (with a size of ten thousand samples), we randomly chose 20% of normal samples as ground-truth and used them to train the null hypothesis. The remaining normal samples were used as part of the test batch, along with the samples from the two anomalous clusters. This was repeated 10 times, with the performance averaged.

For the network data, all the normal web flows from the three files were combined, making nearly ten thousand normal web flows. We randomly selected 20% of these flows as ground-truth normal samples to train the null, and treated the remaining normal flows as part of the test batch, combined with either P2P or Zeus anomalous flows. We separately experimented with P2P and Zeus flows. There were roughly 5% of either P2P or Zeus flows in a given test batch. Experiments for each scenario were averaged over 10 random train-test splits.
C. Experimental results

In Figure 2, we show the performance on the synthetic data. Note that both the proposed scheme and [5] effectively capture groups of anomalies when the maximum feature subset order is two. The first captured cluster (sample subset) consists of more than 95% anomalous samples on average. However, as the maximum feature subset order increases, the “independence tests” approach drops significantly in performance. This is because too many (assumed to be independent) pairwise tests create many redundant features that are all used to evaluate cluster anomalousness; use of these redundant features de-emphasizes, within the score function, the important (low-order) feature subset. Also, we see an early advantage of using cluster-specific DTs, compared to the single Bayesian Net approach. It appears that if an anomalous process is strictly generated from a low order subspace and normal in other feature dimensions (as is the case in this experiment) our cluster-specific DT approach outperforms a single Bayesian Net approach.

In Figure 3 a), we show the performance for normal-P2P discrimination. Compared to [5], which degrades in performance as more and more tests are included, we see superior performance for the proposed method. There is a large batch of anomalous samples captured at maximum order 6 by the proposed method, but both [2] and [5] did not capture this group effectively, as seen in the top 100 precision figure. Also, both of these methods are outperformed by the GMM baseline method. In Figure 3 b), we show the performance for normal-Zeus discrimination. Again, at maximum feature subset order 6 the proposed method captures a large portion of the anomalous flows – more than 50 Zeus flows were captured out of the first 100 flows detected.
Fig. 3: Network traffic data experiment: comparison of different schemes with P2P or Zeus anomalies

by the proposed method. [5] performs poorly in this experiment, and again we observed that as the number of tests increase, the independence assumption degrades the detection performance. The single Bayesian Net approach in [9] also performs relatively poorly on this dataset.

V. EXTENSIONS AND FUTURE WORK

In this work, we used the Bonferroni corrected score function to directly evaluate cluster candidates. Alternatively, we could try to evaluate empirical p-values for this decision statistic, by applying our detection strategy to (many) bootstrap test batches drawn from the null distribution. It would be interesting to see whether such an approach gives comparable (or even better) detection accuracy than use of the Bonferroni corrected score by itself. Such an approach could also be used to determine whether any detected clusters are truly statistically significant. In this work we showed detection accuracy as a function of the maximum feature subset size for a cluster. As the maximum feature subset size continues to increase, we observed that false positives also increase in the first detected cluster, and the objective in (4) tends to favor the maximum feature dimension over use of fewer dimensions. In future, we should propose and investigate criteria for choosing this maximum feature subset size.
VI. Conclusion

In this work, we proposed a GAD scheme to identify anomalous sample and feature sub-sets, accounting for dependencies between the features in a given subset. The proposed model outperforms previous works that assume statistical tests are independent under the null. We demonstrated the effectiveness of our proposed system on both synthetic and real world data, with the latter drawn from the network intrusion detection domain, aiming to discriminate between normal and P2P/Zeus traffic. Our future work includes empirical p-value assessment and automatic determination of the maximum feature subset size of a cluster.

REFERENCES

[1] Z. B. Celik, J. Raghuram, G. Kesidis, and D. J. Miller. Salting public traces with attack traffic to test flow classifiers. In Proc. USENIX CSET, 2011.
[2] C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. IEEE Transactions on Information Theory, 14(3):462–467, 1968.
[3] K. Das, J. Schneider, and D. B. Neill. Anomaly pattern detection in categorical datasets. In Proc. 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 169–176, 2008.
[4] E. Eskin. Anomaly detection over noisy data using learned probability distributions. In Proc. 17th International Conference on Machine Learning, pages 255–262. Morgan Kaufmann Publishers Inc., 2000.
[5] F. Kocak, D. J. Miller, and G. Kesidis. Detecting anomalous latent classes in a batch of network traffic flows. In Proc. IEEE Conf. on Information Sciences and Systems (CISS), pages 1–6, 2014.
[6] LBNL/ICSI Enterprise Tracing Project. http://www.icir.org/enterprise-tracing.
[7] W. Li, M. Canini, A. W. Moore, and R. Bolla. Efficient application identification and the temporal and spatial stability of classification schema. Computer Networks (Elsevier), 53(6):790–809, 2009.
[8] W. Li and A. W. Moore. A machine learning approach for efficient traffic classification. In Proc. IEEE Int’l Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), pages 310–317, 2007.
[9] E. McFowland, S. Speakman, and D. B. Neill. Fast generalized subset scan for anomalous pattern detection. The Journal of Machine Learning Research, 14(1):1533–1561, 2013.
[10] D. J. Miller and G. Kesidis. Detecting clusters of anomalies in a data batch that manifest on low-dimensional feature subsets via dependence-tree based evaluation of joint statistical significance. Provisional United States Patent Filing, 2014.
[11] D. J. Miller, F. Kocak, and G. Kesidis. Sequential anomaly detection in a batch with growing number of tests: Application to network intrusion detection. In Proc. IEEE Int’l Workshop on Machine Learning for Signal Processing (MLSP), pages 1–6, 2012.
[12] R. Sommer and V. Paxson. Outside the closed world: On using machine learning for network intrusion detection. In IEEE Symposium on Security and Privacy, pages 305–316, 2010.
[13] VRT Labs - Zeus Trojan Analysis. https://labs.snort.org/papers/zeus.html
[14] L. Xiong, B. Póczos, J. G. Schneider, A. Connolly, and J. VanderPlas. Hierarchical probabilistic models for group anomaly detection. In Proc. International Conference on Artificial Intelligence and Statistics, pages 789–797, 2011.
[15] R. Yu, X. He, and Y. Liu. GLAD: group anomaly detection in social media analysis. In Proc. 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 372–381, 2014.

[16] G. Zou, G. Kesidis, and D. J. Miller. A flow classifier with tamper-resistant features and an evaluation of its portability to new domains. IEEE Journal on Selected Areas in Communications, 29(7):1449–1460, 2011.