Network-wide anomaly detection via the Dirichlet process

Nick Heard
Department of Mathematics, Imperial College London
and Heilbronn Institute for Mathematical Research,
University of Bristol
Email: n.heard@imperial.ac.uk

Patrick Rubin-Delanchy
Department of Statistics, University of Oxford
and Heilbronn Institute for Mathematical Research,
University of Bristol
Email: delanchy@stats.ox.ac.uk

***Abstract*—Statistical anomaly detection techniques provide the next layer of cyber-security defences below traditional signature-based approaches. This article presents a scalable, principled, probability-based technique for detecting outlying connectivity behaviour within a directed interaction network such as a computer network. Independent Bayesian statistical models are fit to each message recipient in the network using the Dirichlet process, which provides a tractable, conjugate prior distribution for an unknown discrete probability distribution. The method is shown to successfully detect a red team attack in authentication data obtained from the enterprise network of Los Alamos National Laboratory.**

I. Introduction

A number of data sources are available for the discovery and prevention of cyber-attacks and other nefarious network activity. Whereas traditional systems of cyber-defence focus on detecting strong signatures in the data, such as standard antivirus software, there is a largely under-exploited opportunity to use more statistical, probabilistic techniques. The potential advantage of such approaches is the ability to *learn*, from historical data, normal patterns of network behaviour. Anomalies can then be detected which would not stand out otherwise, for example, network traversal using legitimate credentials [1], [2].

One of the challenges of bringing this type of methodology through to deployment is coping with the large data scales and rates involved. A balance between statistical efficiency and computational feasibility is typically needed. At the same time, many classical statistical models, perhaps because of their mathematical simplicity, seem to allow very efficient and scalable implementations.

The present article seeks to model the normal patterns of credential authentication on a computer network, when viewed as a directed interaction network with authenticated connection events between a source computer and a destination computer (which validates or refuses the user credentials). To limit the number of assumptions imposed on the model and yet still achieve computational tractability, the Dirichlet process [3] from Bayesian nonparametric statistics is used to learn the source computers which most typically connect to each destination computer. Using these destination computer models, an algorithm is then developed for detecting source computers which have connected to several unusual destination computers. The method is demonstrated on real computer network authentication data from Los Alamos National Laboratory, where a subset of the data relating to a red team exercise provide a surrogate for intruder behaviour.

The remainder of the article is structured as follows: Section II presents some exploratory data analysis, which motivates the model and algorithm developed in Sections III and IV respectively. Section V describes a *Hadoop MapReduce* implementation of the algorithm which enables deployment to large-scale data problems. Section VI presents the results of applying the algorithm to the authentication data, with some exciting success at detecting the red team presence.

II. Authentication data from Los Alamos National Laboratory

To aid research in applying data science methods to cyber-security, [4] published a comprehensive data set summarising 58 days of (anonymised) traffic on the enterprise network of Los Alamos National Laboratory (LANL). The data are freely available online at http://csr.lanl.gov/data/cyber1, and contain records of network flows, DNS look ups, user processes and authentication events. This data resource is made particularly interesting by the occurrence of a "red team" penetration testing operation during the data collection period. As a consequence, a subset of the authentication event data have been labelled as representing known red team compromise events, which according to the authors "may be used as ground truth of bad behavior that is different from normal user and computer activity". It should be noted that the red team labelling is not exhaustive, and it is very likely there many more authentication events in the data caused by red team behaviour which have not been labelled accordingly.

The methodology which will be presented in this article can be applied to any directed interaction graph, and so the network flow and authentication event data from the LANL collection would be particularly appropriate. This work chooses to focus on authentication events due to the presence of red team labelling for this particular data source, which provide an ideal target for anomaly detection testing. Each event is attributed to the particular pair of source and destination computers used in the connection, which will be colloquially referred to as (src,dst) event modelling. There are $336{,}806{,}387$ observed

authentication events in the data generated between 16,230 source computers and 15,417 unique destination computers. Two example records from the authentication data are as follows:

```
3,U31@DOM1,U31@DOM1,C663,C457,Kerberos,...
30,C829$@DOM1,C829$@DOM1,C829,C829,Kerberos,...
```

The relevant fields are boxed and respectively represent time, source and destination computer. No other fields will be used in the bulk data analysis presented here, although it would be a trivial extension to incorporate any of the remaining fields as part of the labelling scheme for connection events. For example, analysis could focus on (src:user,dst) connections, so that each source computer and user ID pairing are treated as a separate network entity.

Considering authentication connections between source and destination computers as a bipartite graph, Figs. 1 and 2 show the degree distributions of the source and destination computers in the data. It can be seen that the average outdegree exceeds the average indegree for computers in the network; there are many destination computers that have a small number of computers connecting to them. This is a factor we will seek to exploit in the anomaly detection procedure, as unusual connections to those machines have more potential to stand out.

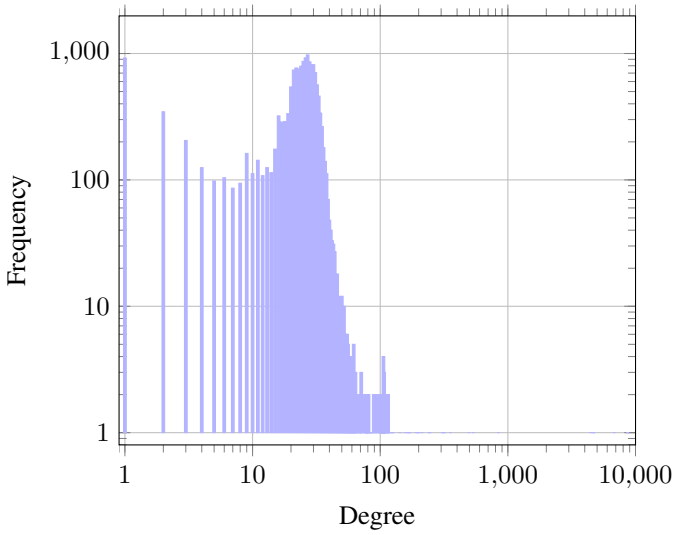

Fig. 1: Log-log plot of outdegree distribution for source computers in the LANL network, measured by the number of unique destination computers receiving authenticated connections.

The authentication event data labelled as red team events account for 48,079 of the total records. These events feature just four source computers, and Table I shows how the event data are distributed across those particular machines. For each source computer, the middle column of the table shows the number of associated authentication events, and then the number of those which are labelled as red team events; the final column shows the outdegree of the four computers, across both the full data set and then just the red team event data.

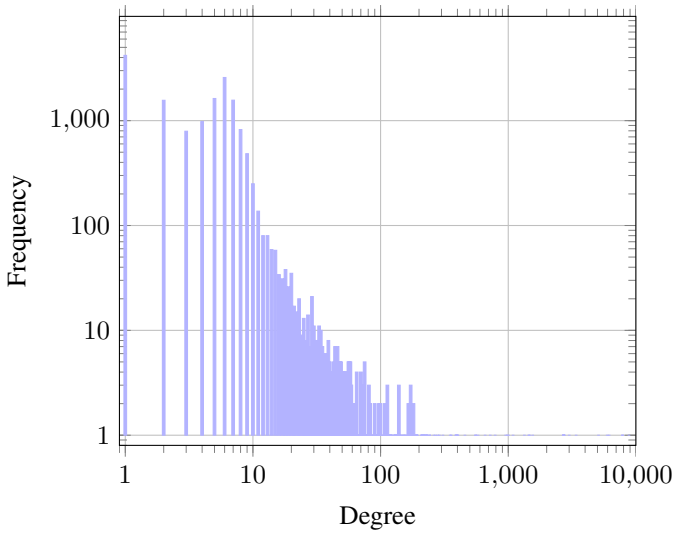

Fig. 2: Log-log plot of indegree distribution for destination computers in the LANL network, measured by the number of unique source computers making authenticated connections.

Source computer ID	Frequency	Unique destination IDs
C17693	701/1717	296/534
C18025	3/101	1/29
C19932	19/10,008	8/30
C22409	26/36,253	3/31

TABLE I: Numbers of records and unique destination computers connected to by four source computers in the LANL authentication data identified for hosting red team activity (number of red team labelled events/total number).

Simply through consideration of Table I, the source ID C17693 provides motivation for the methodology proposed in the remainder of this article. This computer makes a relatively moderate number of authenticated connections, but to a significant number of different destination computers. Now, without imposing unwanted, restrictive modelling constraints, statistically monitoring the outward connections of C17693 would not reveal a strong anomaly detection signal, as it simply behaves as a high degree computer that connects with many different machines; instead, the proposed strategy is to quantify the surprise according to each of the *destination* computers at receiving each connection, and combine those measures of surprise for the events of a source computer. Each destination computer will find a connection from C17693 relatively anomalous, and combining the surprise scores of different destination computers for that source should amplify the signal.

In the next section, a statistical model is presented that will be used for monitoring every destination computer in the network. Scores quantifying the surprise in each sequential connection to those destination computers will take the form of predictive p-values.

III. Modelling via the Dirichlet process

Consider a directed graph (V, E), where V is a set of $K = |V|$ vertices or *nodes* and $E \subset V \times V$ a set of directed edges between the vertices. In the context of this article, the graph will be a computer network with nodes denoting computers and an edge $(x, y) \in E$ representing the presence of directed connections from source computer x to destination computer y.

For each computer in $y \in V$, a separate statistical model will be constructed for the identities of the sequence of source computers $x_1, x_2, \ldots$ that connect to y as a destination. It will be assumed that $x_1, x_2, \ldots$ is an exchangeable sequence of random variables.

A. Dirichlet process model

Suppose $x_1, x_2, \ldots \overset{iid}{\sim} F$, with F an unknown distribution function on the node set V. Let $\alpha > 0$, and F_0 be some specified distribution function on V, acting as a prior estimate of F. It is said that F is a Dirichlet process with base measure αF_0, written $F \sim \text{DP}(\alpha F_0)$, if for any partition $B_1, \ldots, B_p$ of V,

$$(F(B_1), \ldots, F(B_p)) \sim \text{Dirichlet}(\alpha F_0(B_1), \ldots, \alpha F_0(B_p)).$$

The Dirichlet process provides a conjugate Bayesian model for random discrete distribution functions. After observing n elements of the sequence $x_1, x_2, \ldots$ sampled from F, the posterior distribution for the unknown distribution F is again a Dirichlet process,

$$F|x_1, \ldots, x_n \sim \text{DP}\left(\alpha F_0 + \sum_{i=1}^{n} \delta_{x_i}\right).$$

Consequently, the predictive distribution for the next element in the sequence has a very simple form:

$$p_{x_{n+1}|x_1,\ldots,x_n}(x) = \alpha_x^*/\alpha^*, \tag{1}$$

where $\alpha_x^* = \alpha F_0(x) + \sum_{i=1}^{n} \mathbb{I}_x(x_i)$ and $\alpha^* = \alpha + n$.

B. Predictive p-values

Given the predictive distribution (1) for $[x_{n+1}|x_1, \ldots, x_n]$, for anomaly detection it will be necessary to quantify the level of surprise in the realised value of x_{n+1}. Define the p-value for the observation x_{n+1} to be

$$p_{n+1} = \sum_{x \in V : \alpha_x^* \leq \alpha_{x_{n+1}}^*} \alpha_x^*/\alpha^*, \tag{2}$$

which is the predictive probability of observing a node as improbable as the realised value x_{n+1}.

Note that the p-value (2) has a discrete distribution, and is naturally conservative; whilst the model is assumed to hold, the p-value will be stochastically larger than a standard uniform random variable.

IV. Scoring edges and nodes in the network graph

Suppose a network-wide statistical scoring procedure is carried out, where each connection received by a destination computer is scored in sequence according to the Dirichlet process model of Section III and equation (2). In this way, all 336,806,387 events in the data set are assigned a p-value score which, conditional on the destination computer, measures the level of surprise at the identity of the generating source computer.

Motivated by the discussion in Section II, attention now switches to the connections along specific edges in the network from a source computer x to a destination computer y. Measures of surprise are obtained for each edge, and then finally these edge scores are combined into a single score for each source node.

A. Scoring edges

Let $p_1, p_2, \ldots$ be the sequence of p-values observed for each connection event on the edge (x, y). To provide a first reduction in the data, it is desirable to reduce these p-values to a single score for that edge, measuring the highest anomaly level observed on that edge.

To capture the maximum level of surprise observed on the edge (x, y) after m connections, the minimum of the p-values $p_1, \ldots, p_m$ is taken. Due to the discreteness of the p-values, and some possible correlation, the minimum of m p-values is only approximately distributed Beta$(1, m)$. The lower tail probability of this distribution evaluated at $\min\{p_1, \ldots, p_m\}$ provides the p-value score for that edge, denoted $p_{x,y}$.

B. Scoring nodes

For a source node of interest x, let $E_x = \{(x, y)|y \in V \cap (x, y) \in E\}$ be the set of edges in the network graph with that source node on which connections have been observed. For each edge $(x, y) \in E_x$, a p-value $p_{x,y}$ has been obtained in Section IV. Since we are interested in finding strongly anomalous behaviour on each edge emanating from the same source node, these p-values are combined into a single score using Fisher's method,

$$s_{x,y} = -2 \sum_{(x,y) \in E_x} \log p_{x,y}.$$

Under the null hypothesis of the model being true and behaviour normal, $s_{x,y}$ has an approximate $\chi^2_{2|E_x|}$ distribution, and so the upper tail probability of $s_{x,y}$ according to this distribution is taken to be the p-value score for source node x, denoted p_x.

After obtaining the score for each source node in the network, the nodes can then be ranked in their anomalousness based on these doubly combined p-values, to provide a list of the most interesting source nodes down to the least interesting.

V. Hadoop MapReduce implementation of scoring algorithm

There are several steps in the preceding sections which are required to arrive upon the eventual p-values for each source

node. Since independent statistical models are built for each destination computer, and scores are combined locally across first edges and then nodes, the method is highly parallelisable. For this reason, a streaming Hadoop implementation is developed, with python mapper and reducer programmes.

A description of the MapReduce steps performed is given below, and an example python programme for calculating the significance of the minimum p-value from data from a Dirichlet process can be obtained at https://github.com/naheard/dirichlet.git.

Algorithm 1 Sequence of MapReduce tasks for scoring source computers

Input: Entire authentication data.

1: A mapper with primary key of destination computer and a secondary sort on timestamp; reducer calculates sequential p-values for every single record, obtained from the Dirichlet process model. The source, destination and p-value are written to file.

2: A mapper applied to previous output with key given by the source and destination computer pair, and value the p-value. In the reducer, the minimum p-value is found for each source/destination key, along with the number of records for that edge. The significance of this minimum p-values is calculated according to the corresponding Beta distribution, and the source, destination and combined p-value are written to file.

3: A mapper applied to previous output with key given by the source computer of the scored source/destination edge and value the p-value. The reducer calculates the Fisher combined p-value for the sum of log p-values for each source computer key and writes the source computer and final p-value to file.

VI. Results

The proposed anomaly detection method was run on the entire LANL authentication data set using Hadoop MapReduce. The base measure F_0 for the Dirichlet Process in Section III was assumed to be the discrete uniform distribution on the node set V; the parameter α was chosen to be equal to the eventual indegree of each destination computer, as it is assumed that such information could be readily obtained prior to running the method, and higher degree computers should register less surprise at forming new edges. The network-wide distributions of the combined and doubly-combined p-values for respectively scoring edges and nodes are shown in Fig. 3. Recall that the p-values are discrete, and by construction from (2) it follows that 1 will be the most probable outcome. The presence of small p-values near zero correspond to the anomalies according to the Dirichlet process model.

The source computers were then ranked in anomalousness according to the node p-values from Fig 3. From the 16,230 source computers on the LANL enterprise network, the four computers known to be used in the red team attack $\{\text{C17693}, \text{C18025}, \text{C19932}, \text{C22409}\}$ are respectively ranked as shown in Table II. In particular, the computer C17693 discussed in Section II is ranked fifth out of 16,230, which corresponds to a p-value of approximately 3×10^{-4}. Additionally, another known red team source computer, C18025 is also ranked in the top hundred, with a p-value of 0.006. These two entries in Table II are tentatively highlighted as successful intrusion detections found by the algorithm.

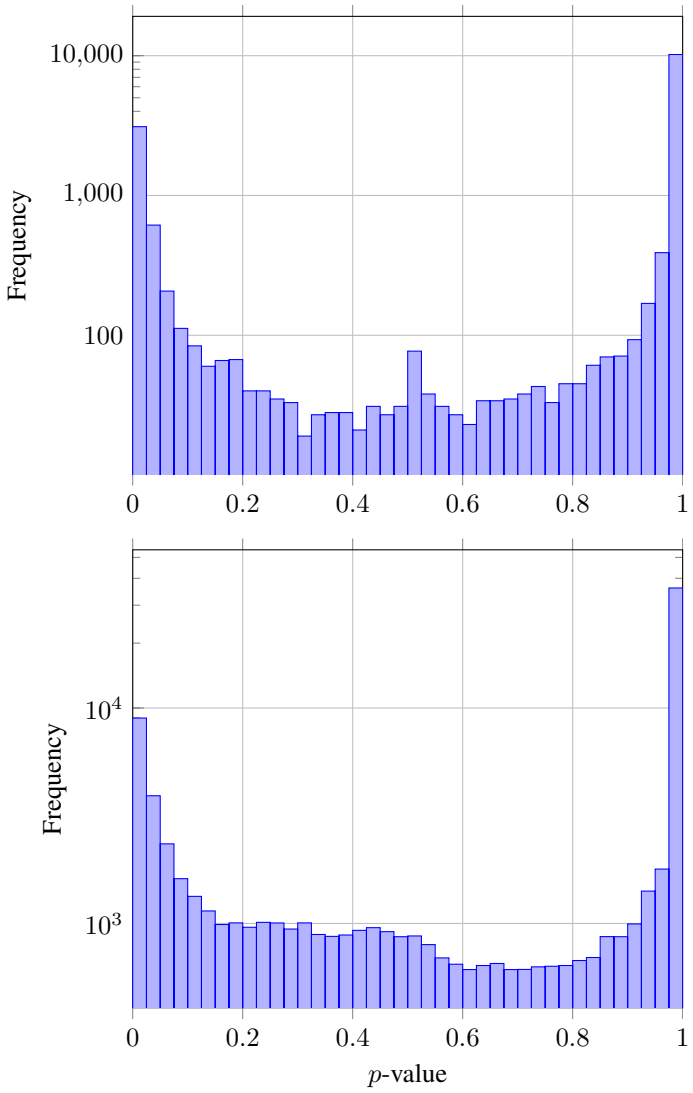

Fig. 3: Distribution of discrete p-values for nodes (top) and edges (bottom) in the LANL computer network.

Furthermore, the computer ranked first by the algorithm as the most anomalous source computer in the data, C15244, is very possibly another useful detection; as although C15244 does not appear as a source computer in the red team labelled data, it does appear as a destination computer in the labelled data. And so it is very possible that, having been compromised as a destination computer, this machine was also used anomalously as a source computer but not labelled.

Source computer ID	Anomaly ranking	Ranking p-value
C17693	5	3×10^{-4}
C18025	94	0.006
C19932	5347	0.329
C22409	7172	0.442

TABLE II: Anomaly ranking of destination computers in the LANL authentication data labelled as relating to red team events. The p-values indicate the significance of those rankings.

VII. Conclusion

A scalable method has been proposed for performing network-wide statistical anomaly detection on a computer network. The scalability of the method stems both from the simplicity of the probability model and the fact that the algorithm is fully parallelisable. To exploit this, the method has been deployed using a "Big data" computational framework, namely Hadoop MapReduce.

Most importantly, the method was found to perform interestingly well, successfully detecting two maliciously acting source computers based only on their unusual connectivity patterns. This performance was achieved on perhaps the most simple view of the data; other fields in the data set describing the authentication or network flow events could be exploited, such as the identity of the user for authentication data and the server port in network flow data. Such fields could, for example, be appended to the source or destination computer identity, allowing the algorithm to then be deployed in an identical manner.

References

[1] J. Neil, C. Hash, A. Brugh, M. Fisk, and C. B. Storlie, "Scan statistics for the online detection of locally anomalous subgraphs," *Technometrics*, vol. 55, no. 4, pp. 403–414, 2013.

[2] M. J. M. Turcotte, N. A. Heard, and J. Neil, "Detecting localised anomalous behaviour in a computer network," in *Advances in Intelligent Data Analysis XIII*, 2014, pp. 321–332.

[3] T. S. Ferguson, "A Bayesian analysis of some nonparametric problems," *Ann. Statist.*, vol. 1, no. 2, pp. 209–230, 03 1973.

[4] A. D. Kent, "Comprehensive, Multi-Source Cyber-Security Events," Los Alamos National Laboratory, 2015.