Abstract—Network data is ubiquitous in cyber-security applications. Accurately modelling such data allows discovery of anomalous edges, subgraphs or paths, and is key to many signature-free cyber-security analytics. We present a recurring property of graphs originating from cyber-security applications, often considered a ‘corner case’ in the main literature on network data analysis, that greatly affects the performance of standard ‘off-the-shelf’ techniques. This is the property that similarity, in terms of network behaviour, does not imply connectivity, and in fact the reverse is often true. We call this disassortivity. The phenomenon is illustrated using network flow data collected on an enterprise network, and we show how Big Data analytics designed to detect unusual connectivity patterns can be improved.

I. INTRODUCTION

There is growing evidence that statistical, data-oriented approaches to enterprise cyber-security can provide effective additional protection over traditional techniques [1], [2], [3], [4]. In such applications, data often have a network (or ‘graph-like’) structure, e.g. computers communicating on a corporate network [1], buyers and sellers in the underground economy [5], user authentication networks [2], botnets [6], and so on. Understanding these patterns of connectivity is key to developing many cyber-analytics for detecting, for example, nefarious network traversal and/or reconnaissance behaviour [1]. As a result, statistical methodology for network data analysis is of great importance to cyber-security. However, a property that is ubiquitous in networks encountered in cyber-security applications, yet relatively rare elsewhere, is that nodes seem to organise into clusters such that connectivity between clusters is stronger than within. Although network modelling is a booming area of Statistics [7], [8], [9], and graph theory is otherwise relatively mature in fields such as mathematics, probability or computer science, many standard methods of analysis are inadequate because of this phenomenon. In particular, Big Data analytics designed for the detection of anomalous edges, paths or subgraphs in computer network data, can be improved if this property is properly taken into account.

To make matters concrete, consider a large corporate or institutional computer network. On this network, there are workstations, Domain Name servers (DNS), web servers, printers and much more. As a rule, printers do not communicate with each other; workstations rarely connect to other workstations; web servers do not themselves browse the web. In summary, nodes that are similar, e.g. in terms of their role on the network, their behaviour, or their connectivity patterns, are often relatively unlikely to connect. Drawing from biological terminology (e.g., [10]), we call this disassortativity. This phenomenon has important consequences for the induced graph of connections. To illustrate, Figure 1 shows two graphs of communications between computers on the Los Alamos National Laboratory (LANL) network, observed over one minute on the left, and five minutes on the right. The graphs were constructed from the “network flow events” dataset [11], [12], by assigning each IP address to a node, and recording an edge if the corresponding two nodes are observed to communicate at least once over the specified time period. It is a common exploratory procedure to count the number of triangles in a graph in order to gauge clustering as, by transitivity, if similar nodes tended to communicate with each other, triangles would occur. Here, remarkably, there are no triangles in either graph.

Fig. 1. LANL network flow graph. Left: first minute. Right: first five minutes.

There are several reasons why disassortative behaviour might be observed in cyber-related data more generally. Networks are often structured into client-server relationships, where the clients are the instigators of communications, querying different servers for different services. In this model, client-to-client communications and server-to-server communications are more rare. Approximately bi-partite or k-partite network structure is also often induced by data collection mechanisms. Within a corporation, for example, most internal traffic can be recorded, as well as traffic between the inside and outside. However, obviously, outside-to-outside traffic is not observed, inducing partially bi-partite structure. Similarly, within the internal network, routers often do not record traffic between nodes on the same subnet, inducing k-partite structure. The superposition of different approximately bi-partite or k-partite network features results in a smoother continuum.
of network behaviours, but a strongly disassortative network structure remains.

Statistical literature on network data analysis is considerably more focussed on modelling assortative behaviour. In a seminal paper introducing ‘modularity’, Newman [13] writes “One issue that has received a considerable amount of attention is the detection and characterization of community structure in networks..., meaning the appearance of densely connected groups of vertices, with only sparser connections between groups”. “A tutorial on spectral clustering” [14], a reference for many data analysts and researchers worldwide (e.g. cited almost 4000 times according to Google Scholar, May 2016), considers only the first \( K \) eigenvectors of different types of Laplacian. As these can in each case be interpreted as solutions to relaxed min-cut problems [14] which, loosely speaking, seek to partition the nodes into densely connected clusters, they do not make good representations of (partly) disassortative networks.

The remainder of this article is structured as follows. In Section II, we focus on the use of graph-based Big Data analytics of computer network data for cyber-security applications. In Section III, we present a statistical analysis of real computer network data, showing how disassortative behaviour can be diagnosed automatically, and at scale. In Section IV, we demonstrate how taking proper account of disassortativity allows improved prediction of new edges on the network, so that more sensitive detection of anomalous edges, paths or subgraphs is possible. Finally, Section V concludes.

II. THE CYBER-SECURITY APPLICATION

This section provides some context and motivation for the use of graph-based Big Data analytics in cyber-security. Our main application is enterprise network defence. This is of course a multi-faceted endeavour, where protection can be talked about in many different layers of abstraction, such as the security of hardware, software protection tools (e.g. anti-virus and firewalls), risk mitigation through policy, and more. The focus here is on providing additional protection through network data analysis. The methodology we present could be incorporated in automated, real-time network defence tools such as PathScan [15], [16], but also used in a forensic capacity, for example if an intrusion is known to have occurred and the task is to identify the machines which have been compromised.

Because storing the actual content of network communications is infeasible, and live packet-inspection can be expensive, intrusive and impossible under encryption, the data available for real-time and post-hoc analysis of a network’s activity are often (much) more summary. Our paper focusses on network flow data, such as Cisco Systems’ NetFlow. These are records of communications between computers in the typical form “time, duration, source computer IP address, source port, destination computer IP address, destination port, protocol, packet count, byte count”. A number of other different types of data are typically available for analysis, for example, authentication events, Domain Name Server look-ups, host log data, and more [11], [12].

To appreciate that detection may be possible despite the limited information available in the data, consider this typical attack pattern, detailed in [4]. A computer on the network is compromised, for example, because an unwitting user clicks on an attachment in a ‘phishing’ email. An important point is that the compromised machine is rarely the ultimate target, but more likely an arbitrary and relatively low-privilege account on the network. The hacker must therefore explore the network, hopping from machine to machine, to achieve his goal. One purpose is to harvest credentials on the different machines he visits, in order to gain higher network privileges. At this exploratory stage, no malware is necessarily being used, and so hard signature-based protection systems may not flag an anomaly. However, the behaviour could be detected as being statistically abnormal given historical data. In particular, a sequence of unusual connections are made, i.e. with low \( a \) priori (or predictive) probability.

Hence, an analysis of the data where IP addresses are treated as the nodes of a graph, and connections between IP addresses form edges, may prove fruitful. Modelling this graph allows discovery of unusual edges, paths or subgraphs, that stand out relative to historical behaviour. The stronger the statistical model for how connections occur on the network, the better those anomalies can be detected. The use of such methodology for enterprise network defence has previously been demonstrated in papers [17], [1], [18], [2] and in software tools such as PathScan [16].

Graph-based analyses are useful in many other cyber-security applications. One example is discovering hit-list worms from protocol graphs, i.e. graphs of communications restricted to different protocols such as HTTP or SMTP [6]. To quote the authors, “We hypothesise that while an attacker may have a hit list identifying servers within a network, he will not have accurate information about the activity or audience for those servers.” Again, the underlying principle is to detect anomalous connectivity patterns in a graph, relative to historical behaviour. Another example is detecting anomalous web-browsing behaviour based on clickstream data [19].

Although we are treating only metadata relating to computer communications, as opposed to content, the data scales involved are large. For example, Imperial College London, a leading university in the UK, generates about 14 Terabytes of network flow data in a month, while LANL, a major US research institution, generates about 30 Gigabytes a day [20]. This calls for analysis tools that can cope with high data rates, or “Big Data analytics”, which at the same have the necessary sensitivity to handle the “needle-in-a-haystack” character of the problem, whereby only a very small proportion of events are likely to be connected with nefarious activity. The data used in this paper represents internal-only traffic on the LANL computer network over a period of 58 days. It is available for public download [11], described more fully in [12], and contains roughly 130 million events, 100 thousand unique edges, and 12 thousand unique (anonymised) nodes.
Our focus on spectral methods for data analysis and anomaly detection, in the next sections, is motivated by this need for scalability. There is a considerable body of research on performing Eigen and singular-value decompositions of large matrices (both central to spectral analysis). Many if not most algorithms also exploit graph sparsity in order to achieve huge computational savings [21], [22]. Note, for instance, that the number of edges in our network flow dataset is extremely small relative to the number of nodes squared (which is the order of the number of edges possible). The decompositions can be performed online, so that a full recomputation is not necessary at each time tick [23], facilitating real-time analysis. Finally, there is fast-growing research into exploiting parallel Big Data platforms for these problems [24].

III. SPECTRAL ANALYSIS OF NETWORK FLOW GRAPHS

In this section, we present a statistical analysis of graphs originating in network flow data. Spectral methods are used to draw out strong evidence of disassortative behaviour. Here, focus is on illustration, and so relatively small graphs are used. However, as emphasised in the previous section, spectral methods provide a scalable approach to data analysis of large networks.

Consider an undirected, simple graph. The adjacency matrix of such a graph is denoted $A$, where $A_{ij}$ is one if the nodes $i$ and $j$ share an edge, and zero otherwise. Hence, $A$ is a symmetric $n \times n$ binary matrix, where $n$ is the number of nodes. Its degree matrix is a diagonal matrix $D$ where $D_{ii} = d_i = \sum_{j=1}^{n} A_{ij}$. Its normalised Laplacian is $L = I - D^{-1/2}AD^{-1/2}$ [14], where $I$ is the identity matrix of order $n$. A common recommendation for representing the nodes of a graph as points in a space is to compute the first $K$ eigenvectors of $L$, column-bind the vectors to form an $n \times K$ matrix, and take the rows of that matrix to represent the $K$-dimensional locations of the $n$ nodes in space.

However, this is not entirely appropriate for graphs with a strong disassortative structure. To see why, consider that the eigenvectors $e^{(k)}$, $k = 1, \ldots, n$ of $L$ minimise

$$\sum_{i,j} A_{ij} \left( \frac{e_i}{\sqrt{d_i}} - \frac{e_j}{\sqrt{d_j}} \right)^2,$$

under the constraint that the vector $e = (e_1, \ldots, e_n)^T$ satisfies $\|e\| = 1$. The actual minimum is achieved at the first eigenvector, $e^{(1)}$; the second eigenvector $e^{(2)}$ provides the next best solution that is orthogonal to $e^{(1)}$, and so on. The first $K$ eigenvectors therefore embed the nodes into a space where nodes that are close are likely to connect.

For partially disassortative graphs, we have found the modified Laplacian $\tilde{L} = D^{-1/2}AD^{-1/2}$ [25] to have an easier interpretation. While the eigenvectors of $L$ and $\tilde{L}$ are the same, the eigenvalues of $\tilde{L}$ are reversed and shifted down by one. These lie between -1 and 1, with negative values indicating disassortative network behaviour [25]. The sequence in decreasing order is hereafter referred to as the graph spectrum.

Figure 2 shows the spectra of different graphs generated from network flow data on the LANL network. Ten one-minute intervals were selected uniformly over the first day, and the graph of communications over each interval was constructed. Only the largest connected component of each graph was analysed. The spectrum of each is displayed as a line in the top panel. Because each has a different number of nodes ($n \approx 1000$), percentiles rather than raw indices for the sequences are used. Three representative network flow graphs are shown in the bottom panel.

The spectra are distinctive. First, they are each almost perfectly anti-symmetric about 50%. This would not be expected for an arbitrary graph and in fact suggests that the network flow graphs are almost bipartite (although they are not exactly). More generally, the large presence of negative eigenvalues is strong evidence of disassortative network behaviour. Second, a high number of eigenvalues are identically zero. This is due to a large number of nodes having exactly the same connectivity patterns. All of this information about the graph is ‘hidden away’ in the higher eigenvalues of $L$ (zeros become ones and negative values now fall between 1 and 2), and would be lost if only the first $K$ eigenvectors of $L$ (e.g. $K = 10$) were computed, as is common for large graphs.

IV. APPLICATION: EDGE PREDICTION

In this section, we demonstrate how taking proper account of the disassortivity of computer networks yields better edge prediction, and therefore more accurate anomaly detection. As previously mentioned, the use of spectral techniques helps ensure that the methodology described, although only shown here on small examples, is scalable to much larger problems.
The algorithm presented outputs a predictive probability for an edge. A derived Big Data analytic for the detection of anomalous paths or subgraphs can be constructed by combining the probabilities of the corresponding edges. Specifically, given a predictive probability for any $n$ connections under analysis, $p_1, \ldots, p_n$, our level of surprise at observing all $n$ connections can be quantified very efficiently, e.g. by combining the probabilities through Fisher’s method [26], $\hat{p} = S^2_{\chi^2}(t)$, or its discrete analog [27],

$$\hat{p}' = \exp\left(n - t'/2 - n \log(2n/t')\right),$$

where $t = -2 \sum \log(p_i)$, $t' = -2 \sum \log(p_i/2)$ and $S^2_{\chi^2}$ is the survival function of the chi-squared distribution with $2p$ degrees of freedom, available in most mathematical packages.

In a real application, a number of additional data sources could be brought to bear on the problem. However, here, for demonstration purposes, we use only graph information, and compare an edge prediction algorithm that takes explicit account of the disassortative properties of the network to one that does not.

Given a graph of communications $G$, we consider two edge prediction algorithms. The first, naive, approach uses only the positive side of the spectrum, ignoring disassortative components. We compute the first $K$ eigenvectors of $\hat{L}$, corresponding to the highest eigenvalues, $\lambda_1, \ldots, \lambda_K$. The vectors are bound columnwise to form an $n \times K$ matrix, and node $i$ is then represented by row $i$, denoted $v_i$. We form a diagonal matrix $\Lambda$ containing the $K$ highest eigenvalues in descending order. Finally, the probability of $i$ and $j$ sharing an edge is modelled as (a monotonic function of) $v_i \Lambda v_j^T$.

In the second approach, we use the first $K$ eigenvectors corresponding to the highest $K$ eigenvalues, in magnitude. In our data, this always results in using $K/2$ eigenvectors from the negative side of the spectrum, but this would not be true in general. We proceed as in the previous algorithm, so that each node $i$ is represented by a row $v_i'$, formed by column-binding eigenvectors from the positive and negative sides of the spectrum. This construction is theoretically justified since, for example, it generates consistent (i.e. asymptotically correct) clusterings under the stochastic block model [25]. The edge probability is modelled as $v_i' \Lambda' v_j'^T$, where $\Lambda'$ is diagonal, containing the $K$ eigenvalues with their original sign, in decreasing order of magnitude. In both models, $K = 10$.

For three one-minute intervals, uniformly selected over a day, we constructed network flow communication graphs from LANL data, as described previously, and fit both types of predictive models on each. For each interval, we then selected edges occurring within the next minute that were not in the original graph, but that did involve two nodes that had been active (so that each has at least one edge in the original graph). Every such edge is scored according to both predictive models. A similar number of pairs of nodes that did not communicate were selected, at random, as negative test-cases. In Figure 3, Receiver Operating Characteristic (ROC) curves are shown for the edge prediction performance of both models, with each subfigure corresponding to one of the three time intervals. (For readers unfamiliar with this performance measure, the higher the curve, the higher the classification performance.) A false positive event is recorded whenever an edge is predicted but does not occur. A false negative event is recorded if an edge occurs when it was predicted not to. In all but the highest false positive regions (which are usually of lesser interest), the predictive model that uses both sides of the spectrum dominates.

![Figure 3: Edge prediction performance for new edges using only the positive side of the spectrum, versus both.](image)

V. Conclusion

A number of papers and software tools have shown that Big Data analytics for graph-based anomaly detection in computer network data can provide important additional protection over more traditional techniques for enterprise network defence, such as anti-virus software or firewalls. In this paper, we have highlighted an important property of many graphs encountered in cyber-security applications, namely disassortivity, which has been largely overlooked. Spectral analysis provides a scalable methodology for diagnosing and properly accounting for this property, showing strong improvements in predicting new connections. These improvements will feed through to anomaly detection tools for edges, paths and subgraphs, which rely on accurate network models.

Acknowledgment

The authors would like to thank John M. Conroy (IDA Center for Computing Sciences, Bowie, USA) and Bill Oxbury (Heilbronn Institute for Mathematical Research, University of Bristol) for useful discussions and ideas.

References


