Detection Theory for Graphs

Benjamin A. Miller, Nadya T. Bliss, Patrick J. Wolfe, and Michelle S. Beard

Graphs are fast emerging as a common data structure used in many scientific and engineering fields. While a wide variety of techniques exist to analyze graph datasets, practitioners currently lack a signal processing theory akin to that of detection and estimation in the classical setting of vector spaces with Gaussian noise. Using practical detection examples involving large, random "background" graphs and noisy real-world datasets, the authors present a novel graph analytics framework that allows for uncued analysis of very large datasets. This framework combines traditional computer science techniques with signal processing in the context of graph data, creating a new research area at the intersection of the two fields.

For many Lincoln Laboratory mission areas, relationships between entities of interest are important. Over the years, much work has gone into the detection of entities, such as radio emissions, vehicles in images, or people named in documents. The examination of relationships between entities (Which emitters are co-located? Which vehicles stopped at the same place? What people are mentioned in the same documents?) can provide a significant improvement in situational awareness and may allow analysts to find subtle, coordinated activity that would go undetected if the relational components of the data were not considered. In a cyber security application, for example, the volume of network traffic may not be high on any particular node, but if there are notably higher rates of communication between a small subset of nodes where the traffic is usually more diffuse, some suspicious activity may be occurring. While the entities alone may not present any detectable anomalous behavior, the relationships and interactions may indicate the presence of interesting activity.

Graphs provide a natural representation for relational data. A graph $G = (V, E)$ is a pair of sets: a set of vertices, V, that denote the entities and a set of edges, E, that represent relationships or connections between the entities. Graphs have been used, implicitly or explicitly, for hundreds of years to represent sets of objects that are somehow connected, such as points on possible travel routes, nodes in electrical circuits, or interacting particles. More recently, graphs have gained popularity in the modeling of relational data, such as social and computer networks. In the last two decades, the explosion in new data-collec-

BENJAMIN A. MILLER, NADYA T. BLISS, PATRICK J. WOLFE, AND MICHELLE S. BEARD

tion capabilities brought about by the World Wide Web and online social networking has led to the emergence of network science, in which graphs of interest frequently represent large sets of entities, often many millions, and the research questions are often data driven [1]. When relational data are cast as a graph, finding small subsets of vertices whose interactions do not fit the model of typical behavior can help in detecting and countering malicious activity. The detection of an anomalous event in the graph data can cue an analyst to pay attention to a certain portion of the network where the previously mentioned coordinated activity is occurring. Thus, detection theory for graph-based data is an important technical area.

While graph theory provides a mathematical object that naturally encodes relationships, the relationships are difficult to analyze in the context of traditional signal detection. Graphs are discrete, combinatorial objects and lack the framework that exists for data in vector spaces with additive Gaussian noise. Translation and scaling, for example, are not well defined for general graphs, and neither is the notion of addition in a way that allows an additive inverse. Therefore, new frameworks are needed to efficiently solve detection problems pertaining to graph and network data. A detection framework for graph data will be useful across myriad applications, and understanding and quantifying detectability are important to making progress in this area.

Over the last five years, researchers at Lincoln Laboratory have been building a signal processing framework for graph-based data, developing a suite of algorithms and metrics in the context of this framework, and applying those algorithms to a wide range of application datasets. This framework is based on the analysis of a graph when compared to its expected topology and projected into a low-dimensional Euclidean space in which known techniques can be applied. The signal processing chain described in this article uses this framework to enable the uncued detection and identification of anomalous subgraphs. The processing chain involves fitting observed data to a model and efficiently computing their residuals (deviations from expected values); performing temporal integration for graphs that vary over time; computing a low-dimensional subspace in which to analyze the data; calculating a detection statistic; and, if an anomalous subgraph is detected, determining the vertices involved.

Definitions and Notation

Each data observation considered in this article is in the form of a graph $G = (V, E)$. Each vertex in V is an abstract entity, with each edge in E connecting a pair of vertices in V. These pairs may or may not be ordered, depending on whether the relationships the graph encodes are directional (e.g., a "pays" relationship is typically directed, whereas an "are friends" relationship is not). Edges are weighted if the relationship has a magnitude associated with it (e.g., bytes sent and received in a computer network). The degree of a vertex is the number of edges the vertex shares with other vertices in the graph. The volume of a graph is defined as the sum of vertex degrees over all vertices in V, and its density is the number of edges divided by the possible number of edges. Graphs will be combined by taking their union, with the union of two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ defined as

$$G_1 \cup G_2 = (V_1 \cup V_2, E_1 \cup E_2).$$

In the subgraph detection framework, graphs are analyzed in their matrix form as defined in the introduction to this *Lincoln Laboratory Journal* issue, "Confronting the Challenges of Graphs and Networks." In this form,

- the vertices are labeled with an integer i, $1 \le i \le |V|$, and the ith vertex is denoted by v_i,
- the adjacency matrix $A = \{a_{ij}\}$ of G is defined as a $|V| \times |V|$ binary matrix in which a_{ij} is nonzero only if there is an edge connecting v_i and v_j, and
- the vector of degrees is denoted by k, where k_i, the ith component of k, is the degree of v_i.

In many applications, the topology of a graph varies over time. In these cases, the graph can be considered at individual, discrete time steps. The expression $G(n) = (V, E(n))$ denotes the graph at discrete time sample n. When the graph is dynamic, the vertex labeling, while arbitrary, is consistent across time samples, and $A(n)$ and $k(n)$, respectively, denote the adjacency matrix and degree vector at time n.

The Subgraph Detection Problem

The subgraph detection problem is framed in the context of traditional detection theory, as a problem of detecting a signal in noise [2]. The observations are cast as a graph whose edges are unweighted and undirected, though the analysis can be extended to weighted and directed graphs. In some cases, G consists of only typical background activity. This is the "noise-only" scenario, and the background

Graph theory
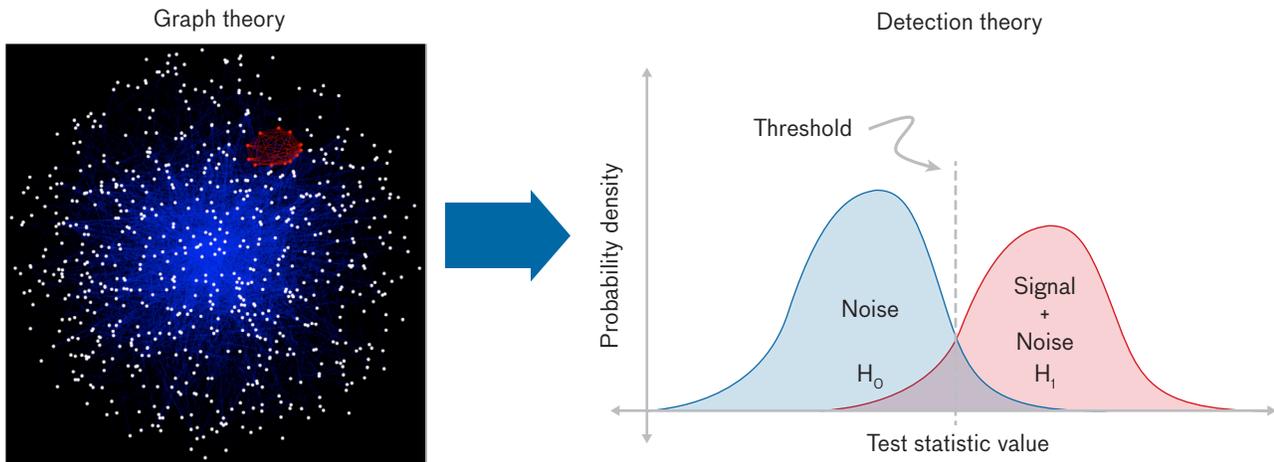
Detection theory

FIGURE 1. A notional diagram of the subgraph detection problem. Given a graph, a test statistic is computed, and the objective is to determine the distribution from which it was drawn. If there is substantial separation between the distributions under the null hypothesis (no signal present) and the alternative hypothesis (signal present), then the comparison of this statistic to a threshold will be powerful for the detection of anomalous subgraphs. This is a common approach in many signal processing application domains, and the novel framework presented in this article allows the analysis of large graphs in this context.

graph is denoted as $G_B = (V, E_B)$. In other cases, most of G exhibits typical behavior, but a small subgraph has an anomalous topology. This is the "signal-plus-noise" scenario. The signal subgraph is denoted $G_S = (V_S, E_S)$, with $V_S \subset V$. The objective, given the observation, is to discriminate between the two scenarios. Formally, we want to resolve the following binary hypothesis test:

$$\begin{cases} H_0 : G = G_B \\ H_1 : G = G_B \cup G_S \end{cases}$$

Under the noise-only hypothesis H_0, the background graph G_B is generated by a random noise process. Under the alternative hypothesis H_1, the background is generated by the same process and combined with G_S, the anomalous subgraph.

As previously noted in the problem model, because $V_S \subset V$, the vertex set is the same under both hypotheses; therefore, the presence of the signal subgraph is indicated by a significant difference in the edge set, i.e., strange connectivity. When the graph is dynamic, the background graph and anomalous foreground are denoted similarly, as $G_B(n) = (V, E_B(n))$ and $G_S(n) = (V_S, E_S(n))$, respectively.

Even in the seemingly simple scenario presented here, the problem is quite difficult. For a fixed (nonrandom) graph, detection of a small subgraph within a larger graph is the subgraph isomorphism problem, which, in

general, is NP-hard (i.e., it is in a class of problems for which there is no known solution in polynomial time [3]). The technique described in the experiments throughout this article allows reliable detection of small subgraphs in the presence of large, realistic backgrounds. As notionally depicted in Figure 1, a test statistic is computed from an observed graph and compared to a threshold. If the distributions of test statistics are sufficiently different under H_0 and H_1, then the comparison of the test statistic to a threshold will accurately detect the presence of a signal subgraph. The objective of the Signal Processing for Graphs (SPG) effort is to develop computationally efficient test statistics that enable the detection of small anomalies within large networks. That is, the test statistics must be computable from an observed graph in a short amount of time, and be sensitive to changes in the graph's topology that occur when an anomalous subgraph is embedded into the background.

An adjacency matrix of an example graph is shown in Figure 2a. The adjacency matrix has some structure to it, as is discussed in the next section. This structure adds a dimension of complexity to the noisy background behavior. The entries highlighted by red circles are the subgraph edges. In a 1024-vertex background where the average vertex has degree 11.5, a 12-vertex complete subgraph (one with all possible edges) is embedded. While only slightly more than 1% of the vertices belong to the

Simulated adjacency matrix

(a)

Empirical test statistic distribution

(b)
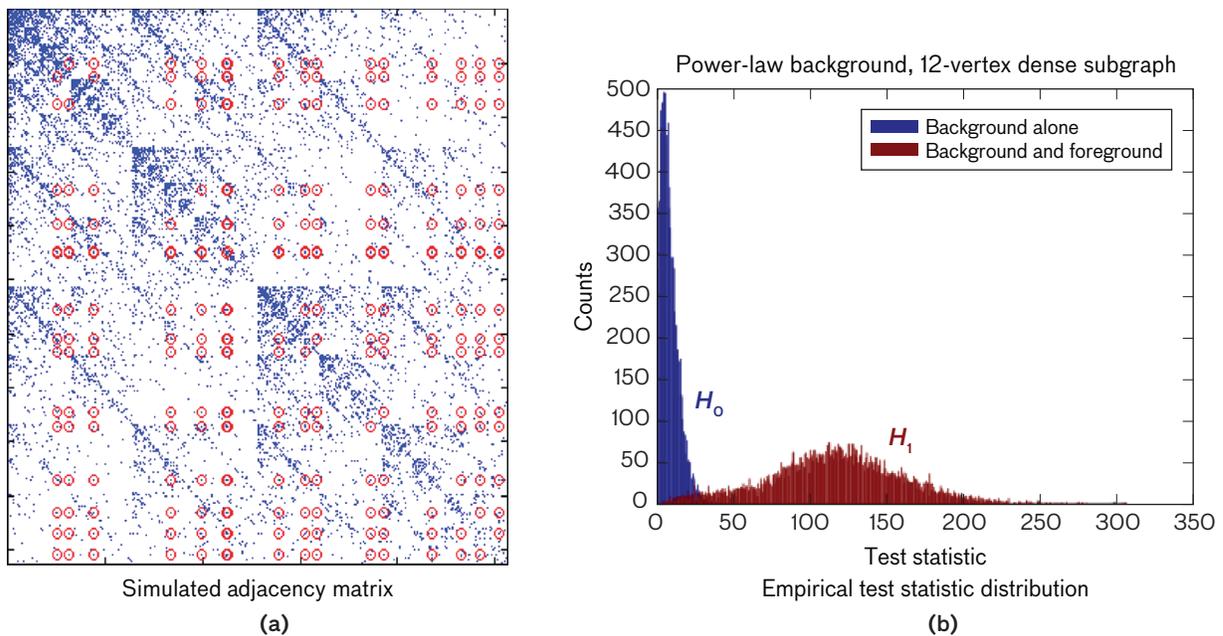
FIGURE 2. An empirical subgraph detection example. The graph in (a) is shown in the form of its adjacency matrix. The graph is sparse, so relatively few entries of the matrix are nonzero. The blue dots indicate edges arising from background (noise) activity. In this simple example, a 12-vertex subgraph with all possible edges, indicated by the red circles, is embedded into the 1024-vertex background. From the observed graph, a test statistic is computed, an example of which is in (b). Statistics that are sensitive to the embedding of an anomalous subgraph, thus yielding significantly different distributions as in the histogram, will be powerful detectors for the hypothesis test.

subgraph and the number of edges attached to each subgraph vertex is not high, we can compute a test statistic that drastically changes when the subgraph is embedded, as demonstrated by the empirical distributions in Figure 2b. This change in the distribution allows the reliable detection of relatively small anomalies. Several test statistics will be discussed later in this article.

Signal and Noise Models

The first step toward the realization of this framework requires the modeling of the signal and noise, both of which are drawn from random graph distributions. Random graph theory began in earnest with the work of Erdös and Rényi [4]. In their random graph model, each possible edge within a fixed set of vertices occurs with the same probability. While these models enable tractable theoretical analysis [5], they do not generate graphs representative of those seen in practice. In recent years, random graph models have been developed that capture properties seen in real-world networks, such as power-law degree distributions [1], small diameters [6], and community structure [7].

One class of random graph model that has gained

significant attention is the stochastic Kronecker graph, in which the edge probabilities are defined by the n-fold Kronecker product of a base probability matrix [8], thereby building graphs with a fractal-like structure. For the experiments run in this paper, we used the R-MAT (recursive matrix) generator, which iteratively chooses edges based on a stochastic Kronecker graph model [8]. An adjacency matrix of an example R-MAT graph is shown in Figure 3a. The R-MAT generator was designed to fit several of the aforementioned properties seen in real networks. The degree distribution is, as expected, an inverse power-law distribution; i.e., the number of vertices with degree d is proportional to $d^{-\beta}$ for a positive constant β. Figure 3b provides the degree distribution of an example R-MAT graph. Chakrabarti, Zhan, and Faloutsos [8] showed that when this generator's parameters are fit to several real-world graphs, the eigenvalues of the adjacency matrix and the graph diameter (longest distance between vertices) are well matched. Also, as is evident in the structure of the adjacency matrix in Figure 3a, there are communities in the graph. In each quadrant of the adjacency matrix, "com-
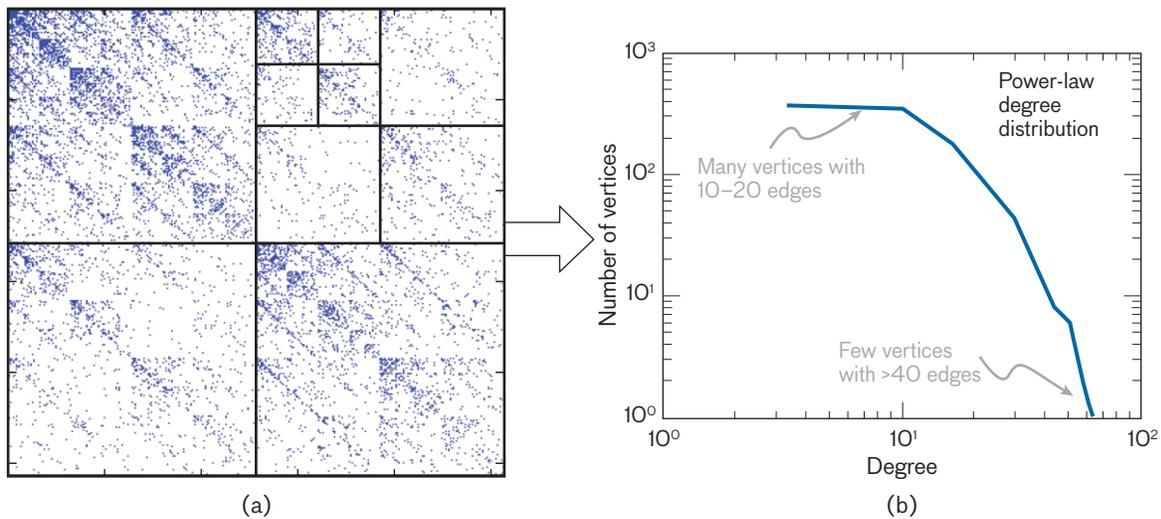
FIGURE 3. Properties of the recursive-matrix (R-MAT) generator. The adjacency matrix created by the R-MAT generator (a) exhibits a fractal-like structure, with each quadrant of the matrix having a similar structure to the full matrix. This is due to the process that generates the graph, recursively choosing a quadrant into which an edge is placed, as exemplified by the axes overlaid in the figure. In addition to the "communities of communities" visible within the adjacency matrix structure, this generator creates graphs with power-law degree distributions (b), which are seen in many graphs based on real network data.

munities of communities" were created by the recursive algorithm. Because this generator creates graphs that exhibit several interesting phenomena seen in real networks, it served well as the noise model in the formulation of the SPG framework.

Under the alternative hypothesis (H_1), a background was generated by the noise model and then combined with a subgraph G_S that is unlikely to occur in the noise-only case. For the anomalous subgraph, a variant of the Erdös-Rényi model was used. The density of the subgraph was fixed, and the edges were chosen uniformly at random from all possible edge sets with the given density. The embedded subgraph has higher connectivity than a typical subgraph of the same size in the background, and thus this simple random graph served as the signal model in the initial experiments. When the signal is embedded into the background, $|V_S|$ randomly selected vertices from V comprise the vertex set of the anomaly.

Signal Processing for Graphs

The detection framework developed at the Laboratory over the last five years is based on the analysis of graph residuals [9]. Given a graph, we fit the observation to a model for background behavior and consider the deviations from the model that exist in the observation. This process is akin to the analysis of variance in linear regression, as illustrated in Figure 4. Normal deviation from the expected value can be attributed to the presence of noise. When much more unlikely deviations occur, however, it is an indication that there is also a signal present. This principle is applied to graph analysis for anomalous subgraph detection. After the data are projected into a low-dimensional space, as shown in Figure 4b, significant deviations of the observed graph from its expected value indicate that an anomalous subgraph exists within the graph.

Figure 5 illustrates the signal processing chain for graphs. While there are several different algorithmic approaches that could be used to implement individual components, the chain is general enough to be applicable to a wide range of problems of interest, as will be demonstrated in the later section entitled "Results on Application Data." The chain begins with the input of a graph in the form of its adjacency matrix. The input, shown in Figure 5a, may be either static or dynamic. No cue to the vertices that may be of interest is given. The processing sequence is as follows: residuals computation, temporal integration, dimensionality reduction, anomaly detection, and vertex identification.
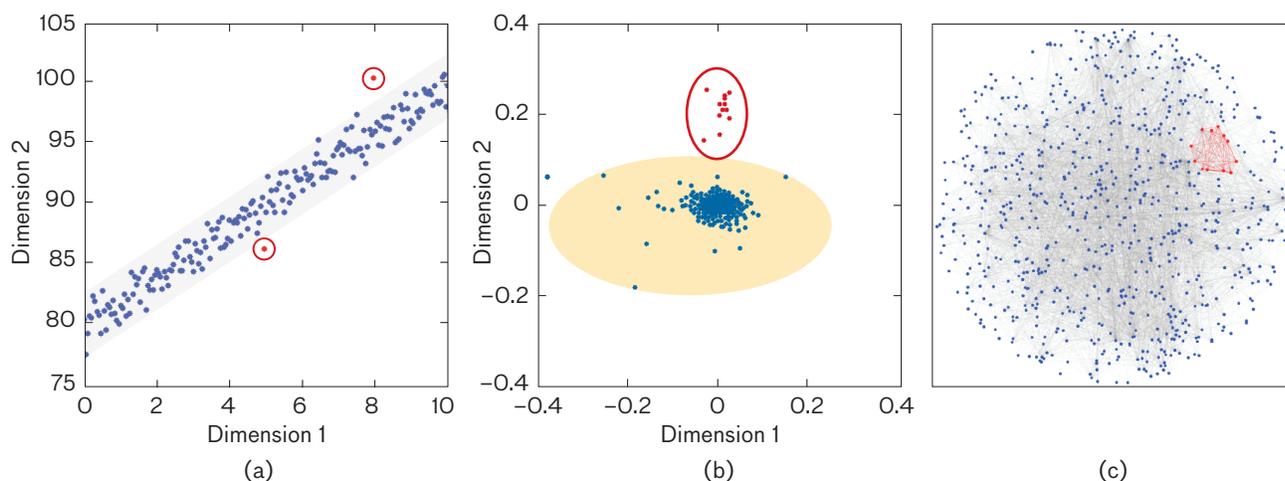
FIGURE 4. Comparison of linear regression and graph "regression." In (a), the residuals (circled in red) appearing outside the gray noise range in a linear regression problem indicate the presence of a signal in addition to noise. A similar principle can be applied for subgraph detection. In (b), a graph has been projected into two dimensions, with most points falling near the origin. One cluster of points stands significantly outside the bulk of the data, and these points correspond to an anomalous subgraph in the original graph, highlighted in red in (c).

1. Residuals Computation

The first step is fitting the observed data to a model and computing the graph residuals with respect to this model. In the experiments highlighted in this article, the modularity matrix [10] is used as a graph-residuals matrix. The modularity matrix B of an unweighted, undirected graph is given by

$$B = A - \frac{kk^{\mathrm{T}}}{2|E|} \; .$$

This expression represents the adjacency matrix of the observed graph, A, minus the expected value of a random graph where the probability of an edge from v_i to v_j is proportional to $k_i k_j$, the product of the degrees v_i and v_j. While this expression does not quite match the structure of our noise model—e.g., it does not capture the "communities of communities" visible in Figure 3a—it still enables good detection performance. Also, because this quantity can be calculated easily from observed data, the residuals can be computed from real data without first training the model. The modularity matrix is used in community detection in a method to partition the graph into two parts such that there is maximal connectivity on either side of the partition and relatively little across it. As the first step in the development of the SPG framework, we reinterpret the modularity matrix as a residuals

matrix and seek to detect a subgraph with large residuals. In ongoing work [11, 12, 13], we have considered a number of other residual models, including ones that account for directionality, dynamics, and attributes.

2. Temporal Integration

If the graph changes its topology over time, we apply a principle from traditional signal processing: matched filtering [12]. In this case, we have to hypothesize the subgraph's evolution over time, possibly with several competing alternatives (i.e., use a bank of filters, each representing a different candidate evolution pattern). Using the residuals matrices

$$B(n) = A(n) - \frac{k(n)k^{\mathrm{T}}(n)}{2|E(n)|}$$

over a finite window of L discrete time steps, the integrated residuals $\tilde{B}(n)$ are computed as

$$\tilde{B}(n) = \sum_{l=0}^{L-1} B(n-l)h_l \, ,$$

where h_ℓ are real-valued coefficients for $0 \le l < L$. When well matched to subgraph behavior, this technique enables the uncued detection of anomalies that would be undetectable by using the same techniques at any instant in time. Thus, as with other data types, temporal integration
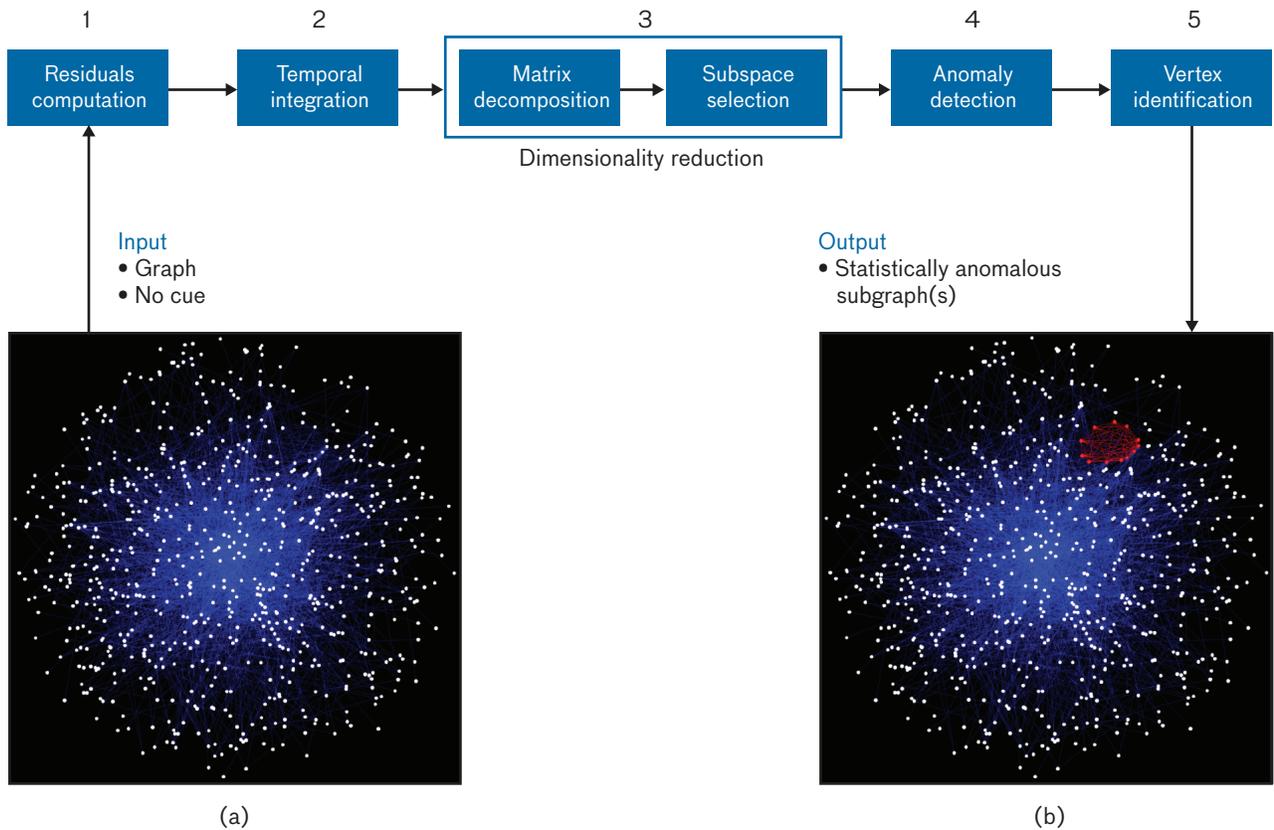
FIGURE 5. The Laboratory's Signal Processing for Graphs (SPG) framework is illustrated in the processing chain for uncued anomalous subgraph detection. A graph (a) in the form of an adjacency matrix is input into the chain. In steps 1 and 2, data are fit to the model, residuals are computed, and data are temporally integrated if necessary. In step 3, the dimensionality is reduced via a matrix decomposition. In steps 4 and 5, anomalous vertices are detected and identified. The final output is the subgraph (or subgraphs), as seen in (b).

gain can be used to increase signal power and improve the detection performance of the processing chain.

3. Dimensionality Reduction

The dimensionality reduction portion of the processing chain has two components: matrix decomposition and subspace selection. First, the residuals matrix must be broken down in a way that allows the reduction of the dimensionality. For the cases considered in this article, this is done by using an eigendecomposition to permit the projection of B (or $\tilde{B}(n)$ for dynamic graphs) into the low-dimensional subspace in which the residuals are the largest. This step is the computational bottleneck in the processing chain, and thus it dominates the asymptotic running time of the algorithms.

A few strategies are used to select the subspace in which to analyze the residuals. The first and simplest strategy is to simply choose the m-dimensional subspace

where the residuals are largest and analyze the graph there. One detection algorithm uses this approach with $m = 2$. An alternative is to use a larger value for m and choose a small subspace in which the residuals are concentrated on just a few vertices. A second algorithm takes this approach, which, in a sense, blurs the line between subspace selection and detection. A third technique solves an optimization problem to find a one-dimensional space in which the residuals reside mostly on a few vertices. This technique is the most computationally costly, but it provides the best detection performance. As discussed in the next section, approaches requiring more computational power enable the detection of smaller, subtler anomalies.

4. Detection

Three algorithms for detection all go hand-in-hand with the component-selection strategies summarized above.
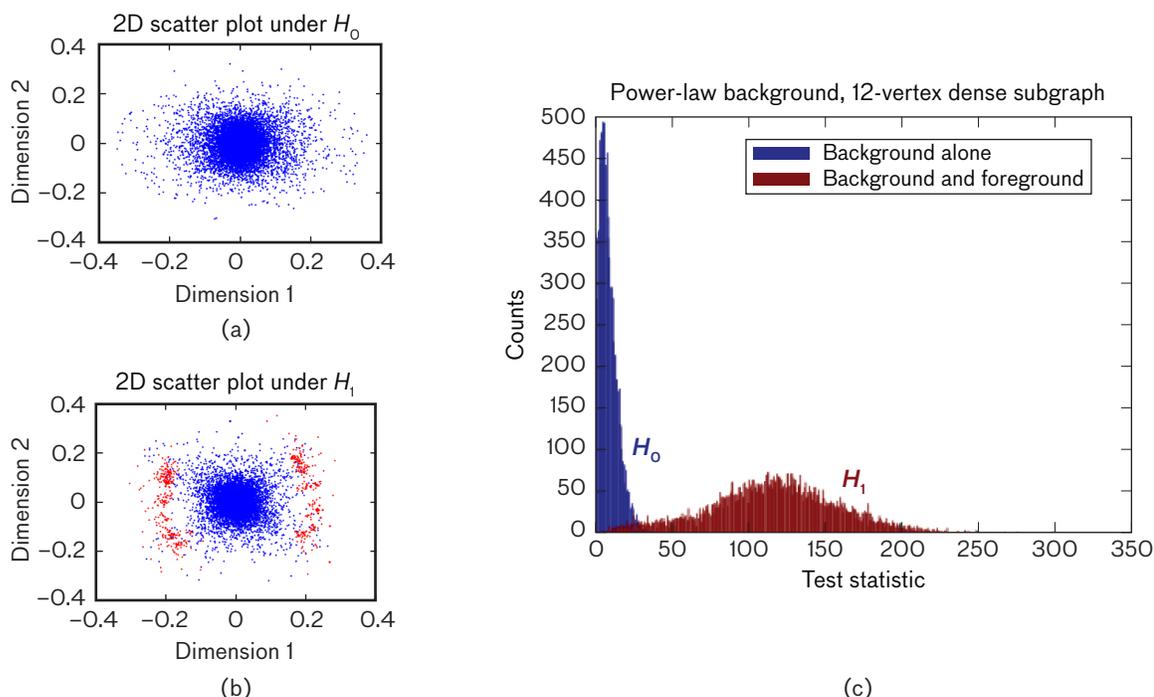
FIGURE 6. A depiction of subgraph detection using the chi-squared test statistic in the principal two dimensions. When the modularity matrix of a graph created by the R-MAT generator is viewed in the space of its two principal eigenvectors, as in (a), the projection is quite symmetric about the origin. When a small, tightly connected subgraph is embedded into the background, the symmetry of the projection is altered, as shown in (b), with red points denoting the vertices of the anomalous subgraph. The chi-squared statistic is much smaller for more radially symmetric projections, yielding a distribution of test statistics much closer to zero when no subgraph is embedded, as demonstrated by the empirical distributions in (c).

Each of the detection algorithms computes a statistic from the projection of the residuals into a low-dimensional space. The computing requirements increase as the detection statistics become more powerful.

The first algorithm considers B in the two-dimensional (2D) space where the residuals are the largest [9]. In this 2D space, a chi-squared test for independence is performed based on the number of vertices projected into each quadrant. The resulting test statistic is maximized over rotation in the plane, and thus takes on smaller values when the projection is more radially symmetric. Figure 6 demonstrates the effect of embedding a small, tightly connected subgraph on this statistic. In Figure 6a, 50 graphs generated under the null hypothesis are displayed, all overlaid in the two-dimensional plane. The projections are rather symmetric about the origin. Under the alternative hypothesis, a significant change is seen in the symmetry, as shown in Figure 6b. The embedded subgraph alters the projection, thereby increasing the value of the test statis-

tic. Histograms of the chi-squared statistic under H_0 and H_1 are shown in Figure 6c. The distributions are radically different, and thus this method will allow the detection of the embedded subgraph in this background.

The second algorithm finds a subspace in which the residuals are highly concentrated on a small subset of vertices [14]. Using this algorithm allows the detection of anomalies that are not strong enough to stand out in the principal 2D subspace. In this case, we look for an eigenvector in which the components stand out along just a few axes. The vector's L_1 norm (the sum of the absolute values of its components) can be used, when it has been unit-normalized in an L_2 sense, to detect such cases because a vector's L_1 norm tends to be smaller when concentrated on a few vertices. This technique is illustrated in Figure 7. When only noise is present, the L_1 norms of the subspace vectors tend to stay within a few standard deviations of the mean. When a subgraph is embedded, however, one of the vectors has an L_1 norm more than 10
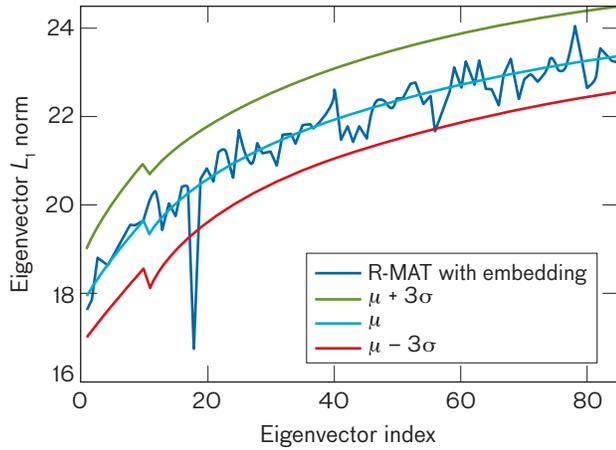
FIGURE 7. L_1 norms of residuals subspace vectors. Under the null hypothesis, the L_1 norms of the modularity matrix's eigenvectors have low variance and typically fall between the red and green curves denoting three standard deviations from the mean. When an anomalous subgraph is embedded, as shown in the plot, the L_1 norm of one of the vectors will often become significantly smaller than usual. In this case, the 18th eigenvector's L_1 norm is 10 standard deviations below the mean, indicating the presence of an anomaly.

standard deviations from the mean. This result occurred in a less prominent vector (specifically, in the eigenvector associated with the 18th largest eigenvalue), so this subgraph would not have been detected in the principal 2D subspace. By using this technique, we can detect these subtler anomalous subgraphs in an uncued fashion.

The third and most complicated algorithm explicitly attempts to find a small subset of vertices that have large residuals [15]. Rather than computing eigenvectors of B and then finding a vector that is concentrated on a small number of components, this algorithm incorporates the fact that the subgraph is relatively small into the subspace computation process. This technique uses sparse principal component analysis (sparse PCA) to find a small subset where the residuals are large. This task involves using a convex approximation to the NP-hard problem

$$\max_{\|x\|=1} x^{\mathrm{T}} B x$$

subject to $\|x\|_0 = |V_s|$, where $\|\bullet\|_0$ denotes the L_0 quasi-norm—the number of nonzero entries in a vector. A one-dimensional subspace that maximizes the residuals (while penalizing the number of entries that have large values) is found by using the convex relaxation. As a detection statistic, we used the L_1 norm of this vector.

The first two algorithms use `eigs` in MATLAB to compute the residuals subspace. To compute an m-dimensional subspace using `eigs` requires a runtime of $\mathrm{O}((|E|m + |V|m^2 + m^3)h)$ [16], where h is the number of iterations before the algorithm converges. The first algorithm requires computation of a subspace of fixed dimension ($m = 2$), so, assuming a number of iterations bounded by a constant, $\mathrm{O}(|E| + |V|)$ time is required for computation. The second algorithm is flexible in the number of dimensions used, and thus requires $\mathrm{O}(|E|m + |V|m^2)$ time under the same assumption. This running time will be linear in $|E|m$ if m does not grow faster than the average degree (volume divided by number of vertices, proportional to $|E|/|V|$) as the problem size scales. Sparse PCA is implemented using the DSPCA toolbox [17], and the running time has been shown to be $\mathrm{O}(|V|^4 \sqrt{\log|V|} / \varepsilon)$, where ε controls the error tolerance [18]. While too complex to run in real time on large graphs, this method demonstrates that still smaller subgraphs are detectable using similar principles.

5. Identification

After the presence of an anomaly is detected, the vertices engaged in behavior that triggers the detector are identified. The method used to identify the vertices is paired with the detection algorithm. In the 2D space, a k-means-based clustering algorithm is used to find a large and a small cluster, and the smaller cluster is declared the subgraph of interest, with the larger cluster assumed to be background noise. This decision allows us to find a small, concentrated set of vertices in the 2D plane. The two more complex algorithms, which rely on the vector being concentrated on a few entries (vertices), enable us to simply apply a threshold to the vector found in the detection process. In all cases, this step concludes the processing chain. The identified vertices are given as output (Figure 5b), which then suggests that an analyst focus on a particular part of the network.

Simulation Results

The graph processing chain was demonstrated on simulated data generated according to the models described earlier. These demonstrations were performed using random backgrounds with embedded anomalies of varying degrees of subtlety, and both static graphs and graphs that vary over time were considered.
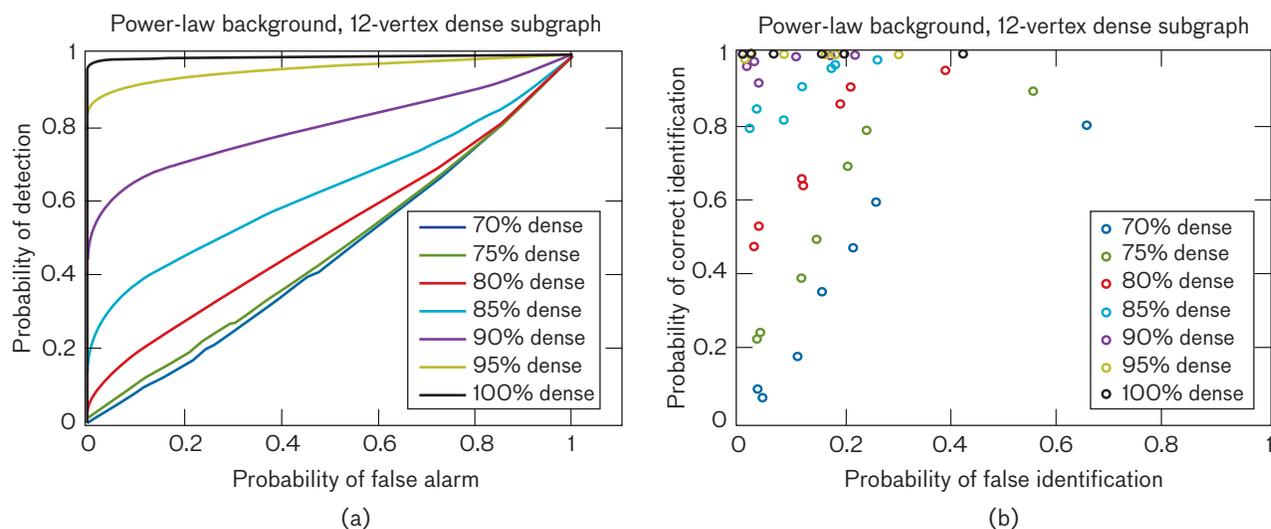
FIGURE 8. Detection and identification performance when residuals are projected into two dimensions. (a) Receiver operating characteristic (ROC) curves for detection performance using the symmetry-based chi-squared statistic when a 12-vertex anomaly is embedded into 1024-vertex power-law background. As the density of the anomalous subgraph increases from 70% to 100%, detection performance goes from no better than chance to nearly perfect. (b) ROC curves for identification performance using clustering in the plane. The individual circles correspond to the detection of clusters of different sizes, and, as with detection performance, more vertices are correctly identified as the subgraph density increases.

Detection and Identification in Static Graphs

In the static simulations, we used an R-MAT background with 1024 vertices. The average degree was approximately 11.5 and the power-law exponent was roughly –2.5. In each of the following scenarios, the distribution of test statistics under the null model was generated by a 10,000-trial Monte Carlo simulation of the background graph alone.

In the first scenario, the signal subgraph consisted of 12 vertices. The density of the subgraph increased from 70% to 100% in increments of 5%. Figure 8a shows the receiver operating characteristic (ROC) curves that demonstrate the trade-off between detections and false alarms when using the symmetry-based statistic in a 2D space. When the subgraph's density is below 80%, this statistic had zero power to detect the embedded subgraph. At 80% density, detection performance increased to slightly better than chance. Detection performance continued to increase with subgraph density, until it reached an equal error rate (EER) of 2.1% at 100% density. Even when the graph was fully connected, the degree of the subgraph vertices was lower than the average degree of the background, and thus a degree-distribution-based detector would not perform well in the detection of these embedded subgraphs.

Upon detection, the aforementioned k-means-based clustering algorithm was used to identify the subgraph vertices. The size of the cluster to be identified was varied to trade off between the number of vertices correctly identified and the number of non-subgraph vertices incorrectly identified. In each of these cases, we used a graph generated under the alternative hypothesis H_1 for each subgraph density. Identification results are summarized in Figure 8b; each discrete point is associated with a different cluster size. When the subgraph was 100% dense, near-perfect identification could be achieved with no false positives. Even when the subgraph was much less dense, the clustering method had reasonable power to discriminate between subgraph vertices and the rest of the graph, although these results assumed the presence of a subgraph as prior knowledge of the identification algorithm.

The next scenario focused on subtler anomalies. In this experiment, the embedded subgraph had 8 vertices. The density was increased from 70% to 100% in increments of 10%, and was embedded on vertices with at most three connections to the background. For this embedding, the chi-squared statistic in the principal 2D subspace yielded no detection power; therefore, a more complex algorithm was
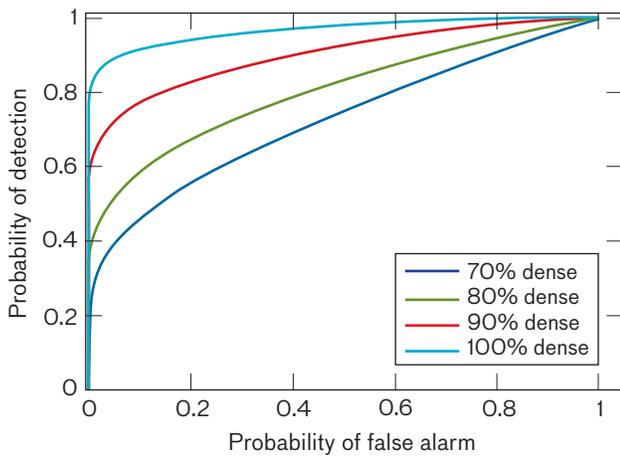
FIGURE 9. Detection performance using subspace L_1 norms. When an 8-vertex subgraph is embedded into the same 1024-vertex background as in Figure 8, the chi-squared statistic cannot reliably detect its presence. By analyzing the L_1 norms of the vectors forming the principal 100 dimensions, as depicted in this figure, the subgraph can be detected consistently, achieving an equal error rate of 9.0% when it contains all possible edges.
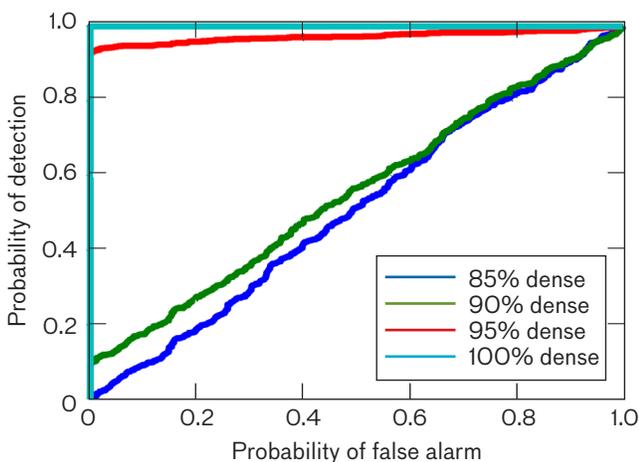
FIGURE 10. Detection performance using sparse principal component analysis (PCA). Using the same 1024-vertex random background as in Figures 8 and 9, we embedded a 7-vertex subgraph, which is not detectable in the principal 2D subspace or by L_1 norm analysis. For 7-vertex subgraphs that are at least 95% dense, sparse PCA yields near-perfect detection performance. The performance reduction for lower subgraph densities may be mitigated by tuning a parameter of the optimization procedure.

used to enable the detection of weaker signals. We computed the 100-dimensional subspace in which the residuals are largest. Under the null model, distributions of L_1 norms of the subspace vectors were built. The detection statistic was then the largest deviation below the mean among the subspace vectors. As shown in Figure 9, this method had reasonable power to detect the presence of these subgraphs.

When the subgraphs became too small to be detected by using this technique, the most complex algorithm for subgraph detection was employed. In this case, a 7-vertex subgraph was embedded into the R-MAT background on vertices that were selected without regard to degree (as in the experiments with the 12-vertex anomaly). The subgraph density was increased from 85% to 100% in increments of 5%. Analyzing eigenvector L_1 norms yielded very low detection power, with an EER of nearly 50%. By using sparse PCA, however, the denser subgraphs were detected with high accuracy, as shown in the ROC curves in Figure 10. When the graph is 100% dense, this algorithm achieved perfect detection within the 10,000-trial simulation. While much more computationally expensive than the other two algorithms, this algorithm used the same principle of low-dimensional graph residuals analysis to detect the presence of subgraphs and further demonstrated the power of this approach.

To demonstrate the performance differences between these techniques, an additional experiment was run in which each detection algorithm performed on the same data. In each of these simulations, an R-MAT background was used with similar properties as in the previous experiments (e.g., similar community structure and average degree), but only 512 vertices. An 11-vertex subgraph was embedded into the background, and detection performance is shown in Figure 11. When the embedded subgraph's density is 90%, as illustrated in Figure 11a, all detection statistics perform rather well. The symmetry-based chi-squared statistic is the lowest-performing, and its EER is about 5%. When the subgraph's density is only 70%, however, the discrepancy in algorithm performance is much more drastic. This difference is shown in Figure 11b; the chi-squared statistic yields performance no better than chance, and the L_1 norm analysis technique does not provide a great deal of improvement. Sparse PCA, on the other hand, enables reasonable detection performance in this situation, with an EER of about 12%. Here, by
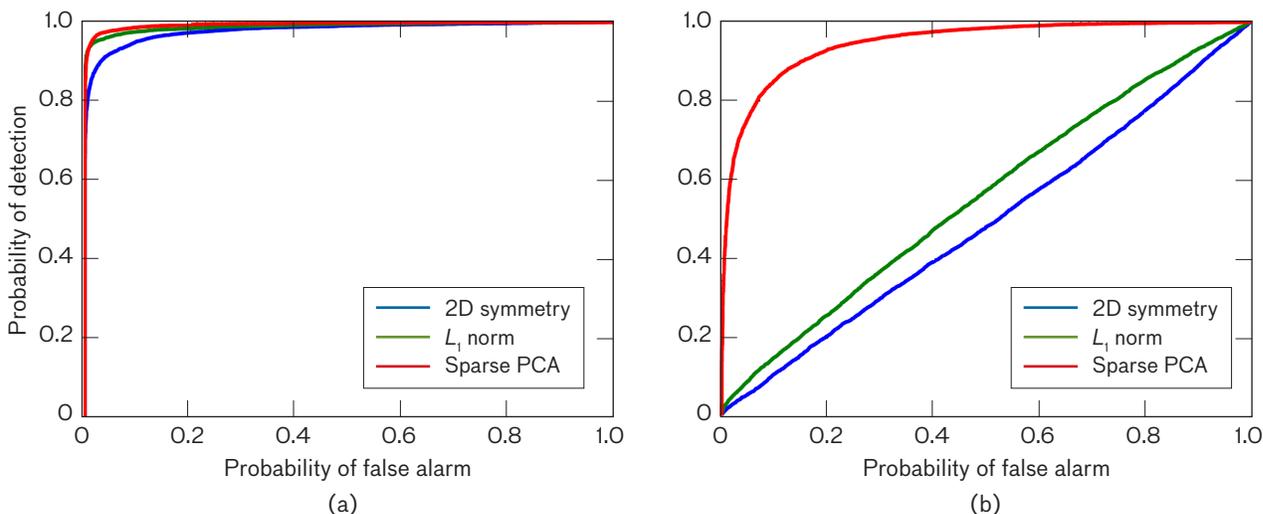
FIGURE 11. A comparison of the detection techniques on a static graph. In a 512-vertex R-MAT background, an 11-vertex subgraph was embedded. When the subgraph is 90% dense (a), the three detection algorithms all yield fairly good performance, while sparse PCA significantly outperforms the other two techniques when the subgraph is only 70% dense (b).

performing on the same dataset, we see the significant improvement in detection performance that occurs when additional computational effort is spent.

Detection and Identification in Dynamic Graphs

In the following experiments, we used simple models for graph dynamics to demonstrate the value of integration gain in this framework. As in the static simulations, the background consisted of a 1024-vertex R-MAT graph, but, in this case, independent realizations of the R-MAT were used with the same parameters over several time samples. Thus, the background noise $G_B(n)$ was independent and identically distributed, and drawn from the same random graph distribution at each time step.

In the first simulation, an independent 12-vertex subgraph with density 70% was embedded onto the same subset of the vertices at each time step. That is, the signal $G_S(n)$ was a temporally independent, identically distributed graph as well. The number of samples L was increased by powers of 2 from 1 to 32, and a uniform filter was used, i.e., $h_l = 1$ for all $0 \leq l < L$. Then, the integrated residuals matrix,

$$\tilde{B}(n) = \sum_{l=0}^{L-1} B(n-l),$$

was projected into its principal 2D subspace, and the same statistic used for the static graphs in Figure 8 was com-

puted. This experiment thus demonstrated the impact of simple averaging when there are multiple observations from the same distribution.

Detection and identification results are shown in Figure 12. In the detection plot (Figure 12a), when one sample was used, detection was no better than chance; this was the same result as for the static case with a subgraph of 70% density. Increasing to just two samples, however, increased detection performance slightly. Detection performance continued to increase as more samples were added, until an EER of 3.1% at 32 samples was reached. The multiple realizations created greater focus on the concentrated subgraph rather than on the more diffuse background residuals, and separated the subgraph vertices in the low-dimensional space. It is apparent from the identification plot (Figure 12b) that the integration technique created better separation between the background and the foreground. Identification performance increased as the number of samples increased, until near-perfect identification performance was achieved with 8 samples. The identification EER was 1% when the number of samples reached 32.

In the second simulation, the anomalous subgraph still consisted of 12 vertices selected uniformly at random from the background, but were no longer identically distributed across time samples. In each trial, 32 time samples were used, and the subgraph increased its
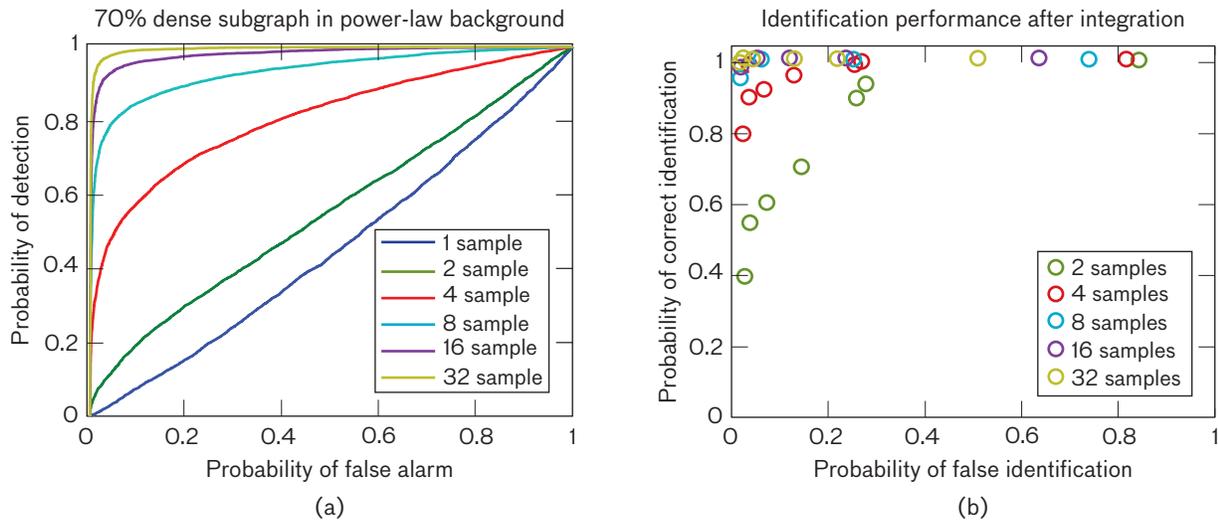
FIGURE 12. Detection and identification performance of a subgraph of fixed density as more time samples are added. As the number of samples was increased, the detection performance (a) improved until **32** samples yielded a very low error rate. The identification performance (b) also improved by integrating more time samples into the analysis.

density from 0 to a fixed density over the course of the time window, i.e., the number of edges $|E_S(n)| = cn$ for some positive constant c in the time window $0 \leq n < L$. In this case, by simply averaging over the samples as in the previous experiment, we achieved no detection power using the same statistic. Therefore, we created and used a filter that is matched to the subgraph's behavior, linearly increasing its coefficient value over the course of the time window. Because the background was generated independently at each time step, we subtracted the mean to further reduce the noise power. The filter used in the simulation, therefore, had the form $h_l = 15.5 - l$, i.e., a ramp centered at zero.

Figure 13 shows the detection and identification performance for the densifying subgraph when the ramp filter was used. When the subgraph reached only 25% density, detection performance was barely better than chance. As the density of the subgraph at the end of the window increased in 5% increments, detection performance quickly increased. When the subgraph reached 45% density, perfect detection was achieved within the 10,000 trials for the null and alternative models in this case. Compared to the previous experiment in which a denser subgraph is observed over the same number of samples, detection performance was significantly better. Thus, this experiment demonstrated the impact of integration gain when the subgraph has a growth pattern

distinct from that of the background, separating the subgraph vertices in the principal 2D space, as demonstrated in Figure 13b. Identification performance is shown in Figure 13c. Again, as we increased the density reached by the subgraph at the end of the window, the ability to discriminate between background and foreground vertices using the clustering technique improved, eventually reaching an EER of about 4%. (This rate was lower than in the previous simulation, likely because the subgraph is more spread out in the projected space, as shown in the scatterplot.) These experiments demonstrated the utility of temporal integration gain in the context of our graph processing chain, showing that this classical signal processing technique is quite powerful when working with graph data.

Results on Application Data
Social Network Analysis

Technology for counterterrorism and counterinsurgency is an important research area for Lincoln Laboratory. In applications aimed at countering terrorist activities, the objective is to find individuals intent on carrying out some sort of threat. Detecting potential threat activity is extremely difficult in an uncued setting; when no specific individual's actions indicate the intention to use violence, it is easy to conceal preparations for a violent event. Thus, to detect threatening activity without a cue,

(a)

(c)

(b)
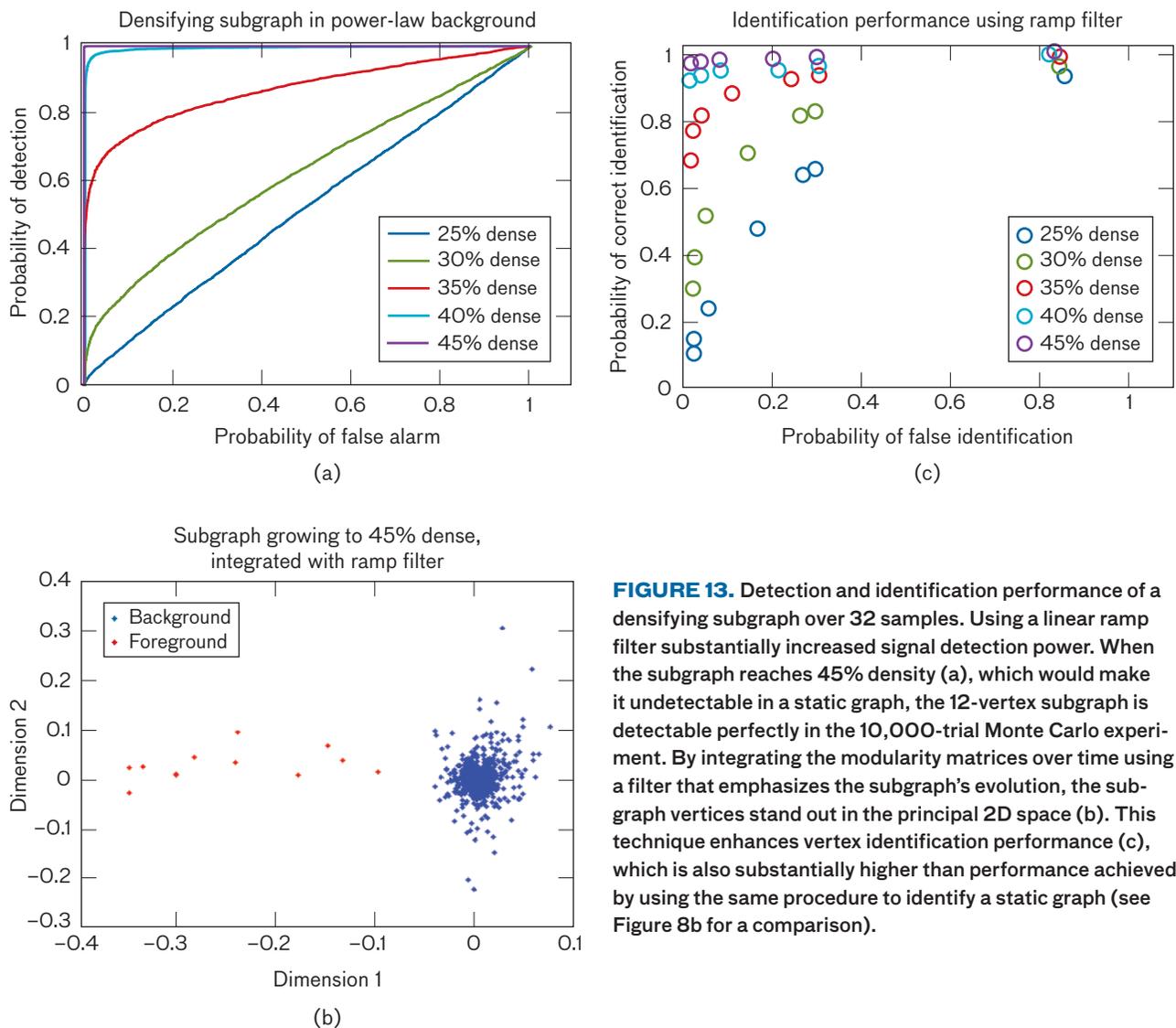
FIGURE 13. Detection and identification performance of a densifying subgraph over 32 samples. Using a linear ramp filter substantially increased signal detection power. When the subgraph reaches 45% density (a), which would make it undetectable in a static graph, the 12-vertex subgraph is detectable perfectly in the 10,000-trial Monte Carlo experiment. By integrating the modularity matrices over time using a filter that emphasizes the subgraph's evolution, the subgraph vertices stand out in the principal 2D space (b). This technique enhances vertex identification performance (c), which is also substantially higher than performance achieved by using the same procedure to identify a static graph (see Figure 8b for a comparison).

the activities of several individuals and the relationships between them must be considered.

A recent Lincoln Laboratory effort called Counter-Terror Social Network Analysis and Intent Recognition (CT-SNAIR) focused on the detection of threat activity within social networks [19]. In this context, the social network can be expressed as a graph G, with V denoting people and E denoting some relationship, such as communication or transfer of money.

Using a similar background to that used in work done by Weinstein et al. [19] and a foreground designed by the CT-SNAIR simulator to represent a threat network, we analyzed detection performance using our processing chain to detect the presence of the threat

network [20]. While previous work investigated cued techniques to identify the threat network, our interest was in determining the detectability of a small, realistic subgraph within a large background without a cue to the subgraph vertices. The foreground used is shown in Figure 14. It is not particularly tightly connected: its average degree is 2.5, and half of the 20 vertices have degree 1. Such a foreground presented a substantial challenge for the processing chain.

The objective was to detect the presence of this subgraph in an R-MAT background with 1024 vertices. The average degree of the background was 4, much larger than the average degree of the foreground. The background clustering was the dominant factor in the principal resid-
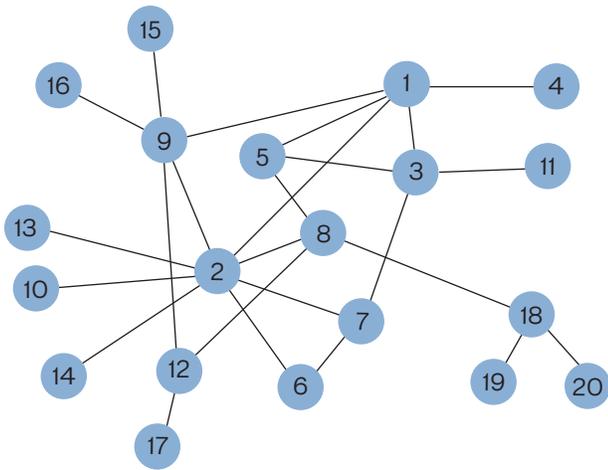
FIGURE 14. A Counter-Terror Social Network Analysis and Intent Recognition (CT-SNAIR) threat network. This foreground network was generated to simulate threat activity among individuals. It contains 20 vertices, and the average degree is 2.5. Very few vertices have connected vertices within their immediate neighborhoods, and this lack of connectivity makes detection via modularity analysis a significant challenge.
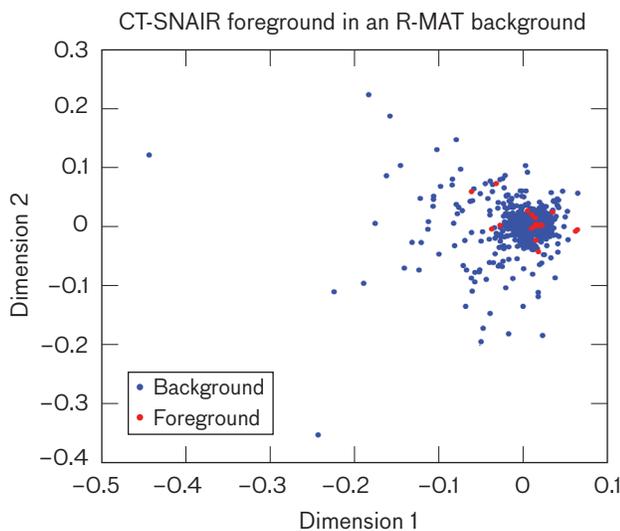
FIGURE 15. A projection of a power-law graph with the CT-SNAIR foreground embedded into two dimensions. In this case, the 20-vertex foreground is not particularly highly clustered, half of the vertices having degree 1. None of the foreground vertices stand out from the background, indicating that this subgraph will not be detectable in the principal 2D subspace.

uals subspace, as the foreground is not connected particularly tightly (very few vertices connect to the same vertices as their neighbors). As shown in Figure 15, when the subgraph was embedded into one of these backgrounds, its vertices were buried within the background noise when B was projected into its principal 2D subspace. Thus, in the static scenario, we focused on the technique using L_1 norms of the subspace vectors.

Detection performance was again tested via a 10,000-trial Monte Carlo simulation, this time varying the parameters of the background rather than the foreground. Three different variants of the background, each one having a different power-law exponent β, were considered. The $\beta \approx 8$ case created a very steep drop-off in the degree distribution, causing a relatively low maximum degree, typically 12–16. In the other cases, β was about 4 and 2.5, yielding maximum degrees around 22 and 44, respectively. The CT-SNAIR subgraph was embedded on a set of vertices that was not well correlated with the largest subspace vectors of the background graph. The correlation was determined by a threshold t when the embedding was performed, and this threshold was varied to determine its impact on detection.

Detection performance is shown in Figure 16. In Figure 16a, the background with the steepest drop-off in its degree distribution is used, and performance in this case was the best. Each plot shows the results when the threshold t was varied. As the threshold decreased, forcing the embedding to occur on a subspace more orthogonal to the subspace where the residuals of the background were large, a significant increase in detection performance was observed. As the power-law exponent was lowered, the detection performance got worse, as seen in the ROC curves, moving from (a) to (c) in Figure 16. Thus, even when we chose a subspace that is relatively uncorrelated with the background, detection performance suffered as the residuals in the background alone got stronger as β decreased. Note that, with a larger absolute value for β, the steep degree distribution implies fewer high-degree vertices, and thus the background clutter is likely to be weaker. These examples demonstrate that, as in other signal processing applications, the strength of the noise, as well as the correlation between signal and noise, has a significant impact on detection performance.

In practice, we may have an idea of how the subgraph of interest will evolve as a violent activity is planned.

In the next experiment, the CT-SNAIR subgraph was embedded with several patterns of evolution over time. In the first case, the subgraph simply increased its density over time, as in the second simulation described in the section "Detection and Identification in Dynamic Graphs." The number of edges $|E_S(n)|$ increased linearly from 0 to 25 over the $L = 32$ samples of the time window. In the second scenario, $|E_S(n)|$ was increased linearly from 0 to 25 over the first three-quarters of the window, then reduced its edges linearly over the last quarter. This "grow-and-disperse" behavior could simulate group members increasing their communication to plan an event, then ceasing communication to avoid detection. The final evolution pattern followed a sinusoidal growth and decay, with the number of edges in the subgraph at each point in time being

$$|E_S(n)| = \left\lfloor \frac{25}{2} \left(1 - \cos\left(\frac{2\pi n}{L}\right)\right)\right\rfloor, \qquad (1)$$

where $\lfloor x \rfloor$ is the greatest integer less than or equal to x.

For each foreground, independent R-MAT graphs were used as a background at each discrete time step. The R-MAT with the power-law exponent $\beta = 2.5$ was used for this experiment. In this case, V_S was chosen uniformly at random from all possible subsets of 20 vertices. The filter coefficients in each case were based on an averaging of the maximum eigenvalues of the foreground graph over several samples (the order that the edges were added was chosen at random), and the detection statistic used was the chi-squared statistic in the principal 2D subspace. This technique would not detect the subgraph in the static case, as is evident from Figure 15.

Detection results are shown in Figure 17. For the densifying subgraph, detection performance was quite good, with an EER of 7.25%. This result (shown in blue) again clearly demonstrated the impact of integration gain on subgraph detection. The chi-squared statistic performed nearly as well when detecting the grow-and-disperse network, which had an EER of 9.1% (green). Detection performance was nearly perfect for the foreground whose density follows the sinusoidal pattern [Equation (1)], as shown in the red curve. A closer inspection indicated that this subgraph's power was boosted the most by its corresponding matched filter, while suppressing the background about equally to the others. The experiments provided an example of a

(a)

(b)

(c)
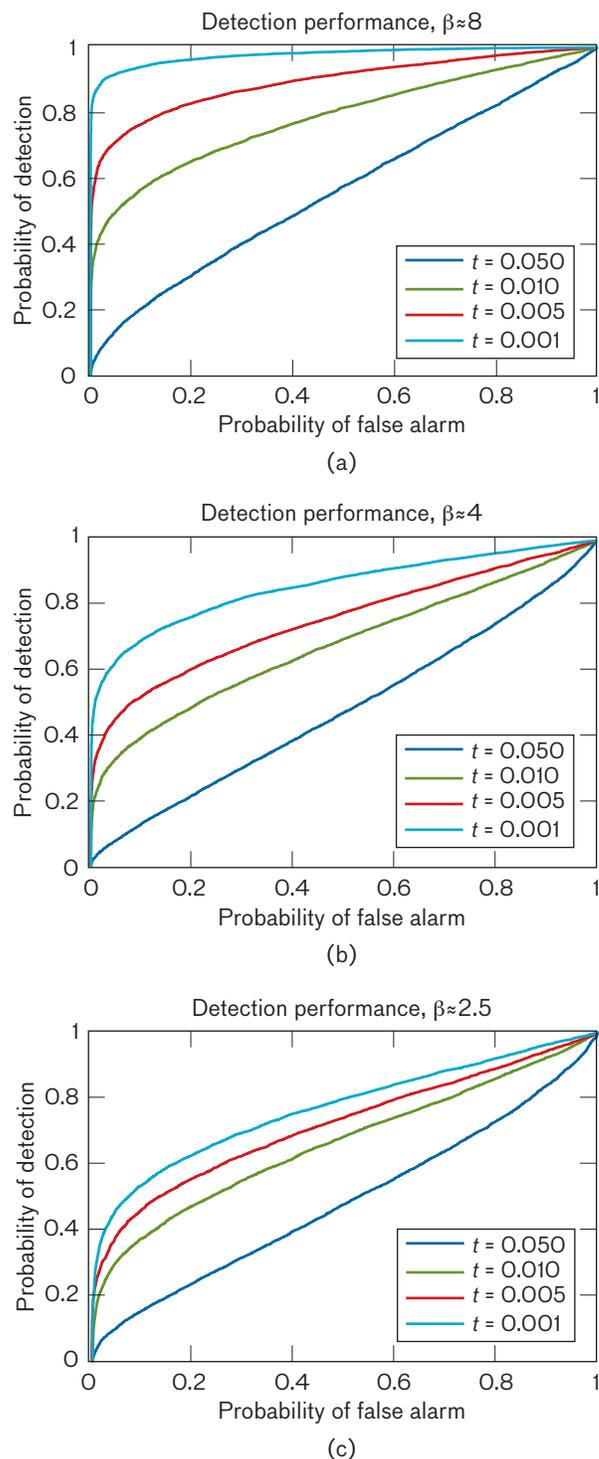
FIGURE 16. The plots showing detection performance for the CT-SNAIR subgraph embedded into power-law backgrounds illustrate that as the power-law exponent β was decreased [moving from (a) to (c)], the detection performance became worse, due to the increased clustering and higher-degree vertices. Performance also worsened as the correlation between the background and foreground increased (increasing *t* in the legends).
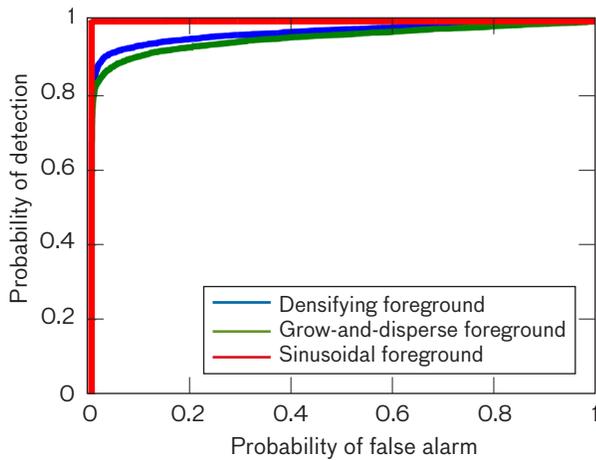
FIGURE 17. Detection of a dynamic CT-SNAIR foreground in R-MAT backgrounds. In all cases, the background was a power-law graph with an exponent of β = 2.5 and was used over 32 time samples. In each experiment, the modularity matrices over time were integrated using a filter to emphasize a different evolution pattern of the CT-SNAIR subgraph. Detection performance is shown when edges are added at a linear rate over the time window (blue), when edges are added and then removed (green), and when the number of edges follows a sinusoidal growth and decay pattern (red), which is most emphasized by its corresponding matched filter, resulting in the best detection performance.

unique pattern of growth allowing the same techniques to be used for detection of a weak, realistic foreground.

Vehicle-Track Analysis

In the intelligence, surveillance, and reconnaissance (ISR) domain, a common data source is electro-optical imagery. From such data, locations of interest on the ground can be identified and placed into a site database. When a vehicle is detected in a scene, it is typically desirable to know which sites the vehicle has visited. Understanding where vehicles stop and which vehicles visited common sites may significantly increase situational awareness.

Vehicle tracks have a natural representation as a dynamic graph: geospatial locations with connections denoting traffic. To apply graph analysis to vehicle tracks, V is designated as a set of sites, and an edge connects two vertices if a vehicle track was observed between the corresponding sites. The graph for this type of application can be dynamic; an edge $E(n)$ exists only if the track exists at time n.

Lincoln Laboratory obtained a set of vehicle-track data that had been captured by an electro-optical sensor and that had true tracks confirmed by an analyst. The graph of these data consisted of 275 vertices and 806 unique edges over the course of the period in which the tracks occurred. Among these tracks was a scenario acted out by drivers to simulate threat activity.

When the vehicle-track graph was viewed as a static dataset showing the sites that shared tracks over the course of the observation period, we noticed that the nodes composing the scripted scenario did not stand out in the 2D space corresponding to the largest residuals. Therefore, we needed to consider a technique that would enable detection of subgraphs with weaker signatures. Also, because only a *single observation*—the 275 vertices/806 edge dataset—was used, Monte Carlo simulations could not be performed to determine detectability of the scenario subgraph.

In lieu of having multiple observations, we generated a power-law graph with similar properties to the background vehicle-track network. The connections between the scenario vertices were then embedded into the simulated background graph, using statistics from the background alone to detect the presence of the embedding. As shown in Figure 18, this technique enabled the detection of the very small, weak-signature target by choosing the right subspace of the residuals for analysis. In Figure 18a, two subspace vectors have significantly smaller L_1 norms than usual for a randomly generated graph. When the graph was considered in this subspace, a substantial fraction of the scenario vertices was separated from the background (Figure 18b). The remainder of the scenario vertices may be found by using the vertices that stand out in the space determined by L_1 norm analysis as cues.

In addition to the Monte Carlo simulation for static graph analysis, temporal integration was applied to the measured dynamic sequence of vehicle-track graphs. By applying a filter that is well matched with the temporal behavior of the subgraph, we were able to detect the subgraph in the principal residuals subspace. As shown in Figure 19, this technique enabled the detection of the scripted scenario. As is apparent in Figure 19a, most of the scenario vertices did not stand out in this 2D space of the static graph. (The ones that did stand out were vertices of very high degree, which did not actually play the role of
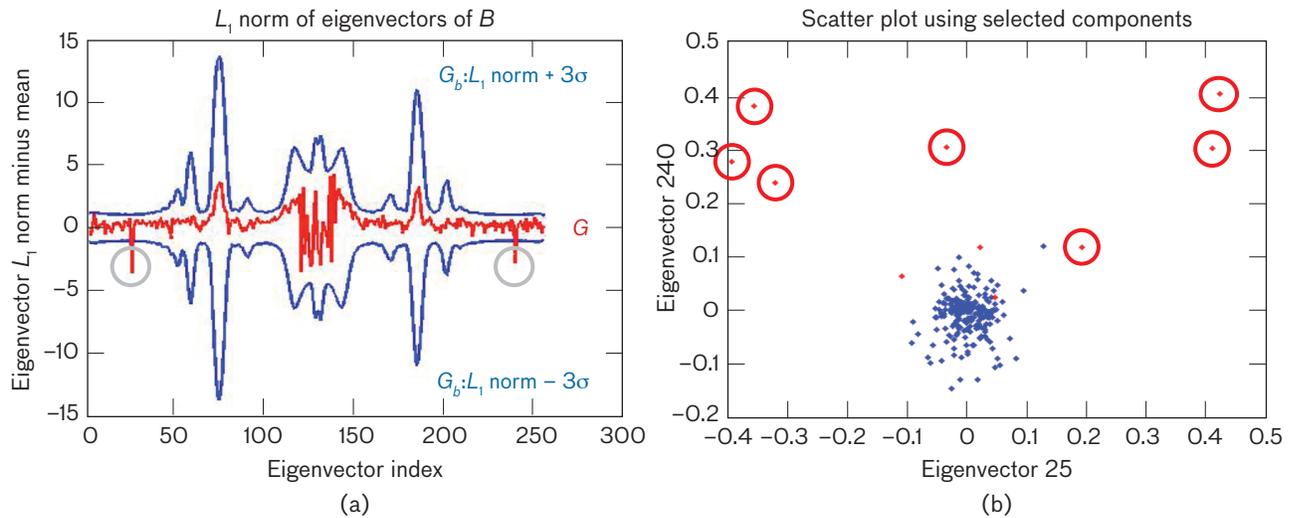
FIGURE 18. Detection of embedded vehicle-track subgraph in a representative background. As in Figure 9, the L_1 norms of the subspace vectors (a) are analyzed to determine where the subgraph resides. After selecting vectors 25 and 240 [those indicated by circles in (a)], the graph is considered in the space of these vectors (b). Most of the scenario vertices stand out from the background, and these vertices can be used as cues for further analysis.
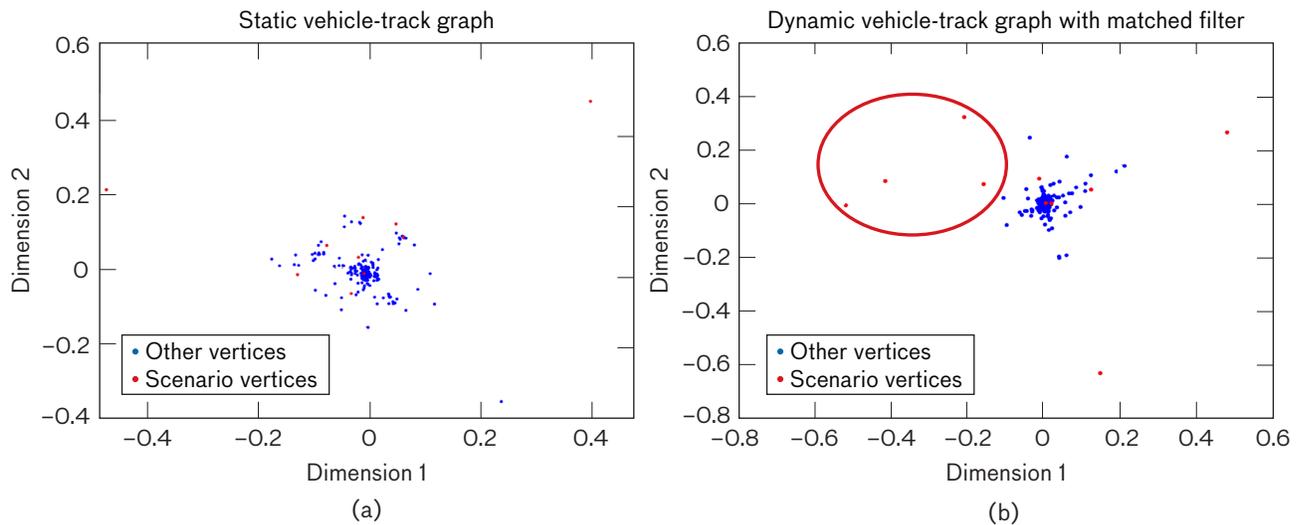
FIGURE 19. The detection of a scripted scenario in vehicle-track data using temporal integration. While the scenario vertices, with the exception of outliers that have very high degree, were buried in the background when the data were viewed in the 2D space in the static graph (a), these vertices are separated from the noise when the data were integrated over time (b) using a filter corresponding to its temporal activity pattern.

threat sites.) When a filter was applied to the sequence of residuals, however, the scenario vertices were pulled out of the background and could easily be detected by thresholding along the first dimension, as shown in Figure 19b. The primary threat sites in the scenario are the four leftmost points in the plot. Here, as in the social network case in the previous section, additional knowledge regarding the temporal evolution of the subgraph enabled discrimination of the subgraph from the background by using techniques that would fail without considering dynamics.

Future Directions

The framework and experiments detailed in this article have demonstrated the advantages of casting network problems relevant to Lincoln Laboratory mission areas in a signal processing context. In a recent Laboratory technical effort, a processing chain was developed for the detection of anomalies in large, noisy graphs. Given a graph as input, this processing chain performs a spectral analysis of the graph's residuals and computes a detection statistic. On the basis of this statistic, it is determined whether a subset of nodes in the graph acts contrary to its expected behavior, and these vertices are identified for further investigation.

Under the Signal Processing for Graphs project at Lincoln Laboratory, significant progress has been made toward creating a computationally tractable detection theory framework for graph-based data. In addition to the development of the processing chain outlined in this article, work has been done on the optimization of matched filters [21] and on parameter estimation and fitting statistics in a class of random graph models [22, 23]. Ongoing work at Lincoln Laboratory is extending this research in several dimensions. In a recent study performed for the Intelligence Advanced Research Projects Activity, the SPG framework was integrated with the Laboratory's Dynamic Distributed Dimensional Data Model (D4M) architecture [24] into a system that combines data storage, graph construction, and graph analysis [11]. This study also includes investigating the use of graph analysis techniques in cyber security [25], using graph models that include attributes of the vertices and edges [13], and demonstrating our methods on gigascale graphs [26]. Other recent efforts have used this processing chain for social network analysis and document graph analysis. Future work includes expanding to still more application areas and determining detectability bounds for the methods outlined here and in subgraph detection in general [27]. As graphs continue to grow in their utility in Laboratory mission areas, the detection-theoretic aspects of graph analysis will become increasingly important, and this research area will continue to be extremely active.

Acknowledgments

References

1. M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On Power-Law Relationships of the Internet Topology," *Proceedings of SIGCOMM*, 1999, pp. 251–262.

2. S. M. Kay, *Fundamentals of Statistical Signal Processing, Volume II: Detection Theory*. Upper Saddle River, N.J.: Prentice Hall, 1998.

3. T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. Cambridge, Mass.: MIT Press, 1990.

4. P. Erdös and A. Rényi, "On Random Graphs," *Publicationes Mathematicae*, vol. 6, 1959, pp. 290–297.

5. B. Bollobás, *Modern Graph Theory*. New York: Springer, 1998.

6. D.J. Watts and S.H. Strogatz, "Collective Dynamics of 'Small-World' Networks," *Nature*, vol. 393, 1998, pp. 440–442.

7. M.E.J. Newman, "Detecting Community Structure in Networks," *European Physical Journal B*, vol. 38, no. 2, 2004, pp. 321–330.

8. D. Chakrabarti, Y. Zhan, and C. Faloutsos, "R-MAT: A Recursive Model for Graph Mining," *Proceedings of the SIAM International Conference on Data Mining*, vol. 6, 2004, pp. 442–446.

9. B.A. Miller, N.T. Bliss, and P.J. Wolfe, "Toward Signal Processing Theory for Graphs and Non-Euclidean Data," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010, pp. 5414–5417.

10. M.E.J. Newman, "Finding Community Structure in Networks Using the Eigenvectors of Matrices," *Physical Review E*, vol. 74, no. 3, 2006.

11. B.A. Miller, N. Arcolano, M.S. Beard, J. Kepner, M.C. Schmidt, N.T. Bliss, and P.J. Wolfe, "A Scalable Signal Processing Architecture for Massive Graph Analysis," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2012, pp. 5329–5332.

12. B.A. Miller, M.S. Beard, and N.T. Bliss, "Matched Filtering for Subgraph Detection in Dynamic Networks," *Proceedings of the IEEE Statistical Signal Processing Workshop*, 2011, pp. 509–512.

13. N. Arcolano and B.A. Miller, "Statistical Models and Methods for Anomaly Detection in Large Graphs," SIAM Annual Meeting, Minisymposium "Massive Graphs: Big Compute Meets Big Data," 2012.

14. B.A. Miller, M.S. Beard, and N.T. Bliss, "Subgraph Detection Using Eigenvector L_1 Norms," *Advances in Neural Information Processing Systems 23: Proceedings of the 24th Annual Conference on Neural Information Processing Systems*, J. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, eds. 2010, pp. 1633–1641.

15. N. Singh, B.A. Miller, N.T. Bliss, and P.J. Wolfe, "Anomalous Subgraph Detection via Sparse Principal Component Analysis," *Proceedings of the IEEE Statistical Signal Processing Workshop*, 2011, pp. 485–488.

16. R. Lehoucq and D. Sorensen, "Implicitly Restarted Lanczos Method," in *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, eds. Philadelphia: SIAM, 2000, ch. 4.5.

17. R. Luss, A. d'Aspremont, and L.E. Ghaoui, "DSPCA: Sparse PCA Using Semidefinite Programming," 2005, available online at http://www.cmap.polytechnique.fr/ aspremon/ DSPCA.html.

18. A. d'Aspremont, L.E. Ghaoui, M.I. Jordan, and G.R.G. Lanckriet, "A Direct Formulation for Sparse PCA Using Semidefinite Programming," *SIAM Review*, vol. 49, no. 3, 2007, pp. 434–448.

19. C. Weinstein, W. Campbell, B. Delaney, and G. O'Leary, "Modeling and Detection Techniques for Counter-Terror Social Network Analysis and Intent Recognition," *Proceedings of the 2009 IEEE Aerospace Conference*, 2009, pp. 1–16.

20. B.A. Miller, M.S. Beard, and N.T. Bliss, "Eigenspace Analysis for Threat Detection in Social Networks," *Proceedings of the 14th International Conference on Information Fusion*, 2011, pp. 1–7.

21. B.A. Miller and N.T. Bliss, "Toward Matched Filter Optimization for Subgraph Detection in Dynamic Networks," *Proceedings of the IEEE Statistical Signal Processing Workshop*, 2012, pp. 113–116.

22. N. Arcolano, K. Ni, B.A. Miller, N.T. Bliss, and P.J. Wolfe, "Moments of Parameter Estimates for Chung-Lu Random Graph Models," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2012, pp. 3961–3964.

23. B.A. Miller, L.H. Stephens, and N.T. Bliss, "Goodness-of-Fit Statistics for Anomaly Detection in Chung-Lu Random Graphs," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2012, pp. 3265–3268.

24. J. Kepner, W. Arcand, W. Bergeron, N.T. Bliss, R. Bond, C. Byun, G. Condon, K. Gregson, M. Hubbell, J. Kurz, A. McCabe, P. Michaleas, A. Prout, A. Reuther, A. Rosa, and C. Yee, "Dynamic Distributed Dimensional Data Model (D4M) Database and Computation System," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2012, pp. 5349–5352.

25. M.C. Schmidt, "Detection of Anomalous Events in Large-Scale Graphs," *Proceedings of the Graph Exploitation Symposium*, 2012.

26. E. M. Rutledge, B.A. Miller, and M.S. Beard, "Benchmarking Parallel Eigen Decomposition for Residuals Analysis of Very Large Graphs," *Proceedings of the IEEE High Performance Extreme Computing Conference*, 2012, available online at ieee-hpec.org/2012/index_htm_files/Rutledge.pdf .

27. S.T. Smith, K.D. Senne, S. Philips, E.K. Kao, and G. Bernstein, "Network Detection Theory and Performance," *Lincoln Laboratory Journal*, vol. 20, no. 1, 2013, pp. 47–61.

About the Authors

Benjamin A. Miller is a member of the technical staff in the Computing and Analytics Group at Lincoln Laboratory. His primary research focus is the development of theory and methods for anomaly detection in large, heterogeneous datasets, with applications to a wide variety of disciplines. In 2005, he joined the Laboratory as an associate staff member in the Embedded Digital Systems Group, where he developed digital signal processing techniques for efficient real-time linearization of wideband radio-frequency devices and researched new methods for sparse signal recovery in compressed sensing. He holds a bachelor's degree (with highest honors) and a master's degree in computer science from the University of Illinois at Urbana-Champaign.

Nadya T. Bliss is the Director of Strategic Projects Development at Arizona State University (ASU). In that role, she is responsible for developing and maintaining strategic relationships with funding agencies, working with university leaders on developing large-scale, interdisciplinary proposals, and monitoring the university's strategic investments. She also leads and participates in sponsored research. Prior to joining ASU, she spent 10 years at Lincoln Laboratory, most recently as the founding group leader of the Computing and Analytics Group. In that capacity, she developed, led, and managed research initiatives in advanced analytics, high-performance computing systems, and computer architectures to address the challenges facing the Department of Defense and the intelligence community. In 2011, she received the inaugural MIT Lincoln Laboratory Early Career Technical Achievement Award, recognizing her work in parallel computing, computer architectures, and graph processing algorithms and her leadership in anomaly detection in graph-based data. She has authored or co-authored 60 publications and presentations, holds a patent, and has served as a chair and member of multiple technical and organizational committees. She received bachelor's and master's degrees in computer science from Cornell University and is a Senior Member of IEEE and a member of the Association for Computing Machinery.

Patrick J. Wolfe holds chairs in statistics and computer science at University College London, and is adjunct professor of electrical and computer engineering at his alma mater, the University of Illinois at Urbana-Champaign. His research is focused on statistical theory and methods for network data analysis, for which he is recognized as a UK Royal Society Research Fellow and an Engineering and Physical Sciences Research Council (EPSRC) Established Career Fellow in the Mathematical Sciences. He received a doctoral degree from Cambridge University in 2003, after which he joined the faculty of Harvard University, receiving the Presidential Early Career Award from the White House in 2008 for contributions to signal and image processing. Among several ongoing efforts in network modeling and inference, he currently leads a large Multidisciplinary University Research Initiative on the statistical analysis of graphs and social networks. He has worked closely with Lincoln Laboratory researchers on related topics over the past several years.

Michelle S. Beard is an assistant technical staff member in the Computing and Analytics Group at Lincoln Laboratory. She received her bachelor's degree (with honors) in computer science at Bryn Mawr College in 2010. Her technical experience includes software engineering and testing, high-performance computing, anomaly detection, and graph analytics.