Multi-Level Anomaly Detection on Time-Varying Graph Data

Robert A. Bridges*,[1], John P. Collins*, Erik M. Ferragut*, Jason A. Laska* and Blair D. Sullivan†,[2]
* Computational Science and Engineering Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831
{bridgesra,ferragutem,laskaja}@ornl.gov, jparcoll@gmail.com
† Department of Computer Science, North Carolina State University, Raleigh, NC 27695
blair_sullivan@ncsu.edu

***Abstract*—This work presents a novel modeling and analysis framework for graph sequences which addresses the challenge of detecting and contextualizing anomalies in labeled, streaming graph data. We introduce a generalization of the BTER model of Seshadhri et al. by adding flexibility to community structure, and use this model to perform multi-scale graph anomaly detection. Specifically, probability models describing coarse subgraphs are built by aggregating node probabilities, and these related hierarchical models simultaneously detect deviations from expectation. This technique provides insight into the graphs' structure and helps contextualized detected event. For evaluation, two new hierarchical detectors are tested against a baseline Gaussian method on a synthetic graph sequence with seeded anomalies. We demonstrate that in a labeled setting with community structure, our graph statistics-based approach outperforms both a distribution-based detector and the baseline, accurately detecting anomalies at the node, subgraph, and graph levels.**

I. Introduction

Social networks play an increasingly important role in today's society, yet extracting domain insights from their analysis and visualization remains challenging—in large part due to their transient nature and the inherent complexity of many graph algorithms. Many social graphs naturally have (i) labeled nodes representing individuals or entities, and (ii) an edge set that changes over time, creating a sequence or time-series of individual snapshots of the network. A key task in understanding this data is the ability to identify patterns and aberrations across snapshots—specifically in a way that can pinpoint areas of interest, and provide context for results. The importance of context in anomaly detection is easily exemplified in a cyber-security setting, where observing an unanticipated connection (edge) between an internal IP and an external host might warrant alarm. However, providing the context that many similar IPs (i.e. nodes in a common community) regularly contact that host could save an unnecessary investigation. Unfortunately, although this time-varying labeled scenario is natural in many domains, most existing techniques for anomaly detection are either limited to static graphs or unable to "zoom in" on the reason a graph is identified as non-standard (see Section II).

Here we address the problem of identifying and contextualizing anomalies at multiple levels of granularity in the sequential graph setting. This problem is neither a special case nor an extension of the more commonly studied scenario of finding anomalous parts of a single (static) graph, since the availability of common node labels provides information not available in single-graph or unlabeled graph ensemble problems, necessitating the development of new methods. We propose and test a novel method for anomaly detection in time-varying graph data using hierarchically related distributions to detect related abnormalities at three increasingly fine levels of granularity (e.g., at graph, subgraph, and node levels). This multi-scale technique gives insight into exactly what caused the anomaly, and allows one to focus attention on the specific subgraphs involved. The probabilistic multi-scale detection relies on comparison with an underlying graph model, and we use an extension (described in Section III) of the recent BTER model [1] that enables improved prescription of community structure. To fit an instance of the model to observed graphs, we give methods for detecting communities and estimating parameters (see Section IV). Section V defines the probability calculations for two new multi-scale detectors, as well as a baseline detector similar to that of [2] (which is limited to detecting anomalies at the graph-level). Finally, to test a newly observed graph for anomalous structure, we compute hierarchically-related probabilities from the tuned model and their associated p-values using a Monte-Carlo simulation. Our workflow is a streaming detection framework, where parameters are learned from previous observations, the detector is applied to new data, then the parameters are updated to include the new graph in the observations. We note that performing anomaly detection using a graph's probability—as given by the model from which it was sampled—will often result in an inaccurate detector when node labels are known. This is a consequence of the likelihood of an unlabeled graph being shared by isomorphic copies distinguished by these labels and is discussed in Section V. We illustrate this phenomenon and provide empirical evidence that modeling a set of statistics indicative of node/subgraph interactions provides more accurate detection in two experiments described in Section VI.

II. Background & Related Work

Previous work on finding anomalies in graph data includes compression techniques such as [3], in which minimum de-

[1]Lead author. Phone: (865) 241-0319

[2]Supported in part by DARPA GRAPHS/SPAWAR Grant N66001-14-1-4063, the Gordon & Betty Moore Foundation, and the National Consortium for Data Science.

This manuscript has been authored in part by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan). Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of DOE, DARPA, SSC Pacific, the Moore Foundation, or the NCDS.

scription length is used to detect repetitive "normal" subgraphs. Because this technique detects subgraphs deviating only slightly from a found normative pattern, it performs more rigid detection than ours. Approaches based on hypothesis testing using graph statistics have also been studied (e.g. [2], [4], [5]). Methods in [4] and [5] are geared towards identifying abnormally dense regions, while our detectors are designed to identify anomalies caused not only in community density, but changes in the interactions within or between communities. A more recent hypothesis testing approach in [2] fits Gaussian distributions to three graph statistics. Due to similarities with our workflow (using a p-value estimated by a Monte-Carlo simulation from a graph model to decide anomalies), we test our method against a baseline detector using similar Gaussian estimates, although we note that [2] focused on Kronecker graphs, not the GBTER model. A recent paper of Peel & Clauset [6] also uses a hierarchical generative graph model and Bayesian hypothesis testing for change detection in time-varying networks; our work differs in the introduction of a new graph model (Section III) and focus on related anomalies at different scales (as opposed to a "shock" that changes the overall graph structure). For a survey on graph anomaly detection, we refer the readers to Akoglu et al. [7].

This paper extends the general anomaly detection workflow of Ferragut et al. [8] to hierarchically analyze graph data. The general method estimates probability models from observations and new data is declared anomalous if it has sufficiently small p-values. More precisely, if a probability distribution P is estimated from observed data $x_1, ..., x_{n-1}$, the p-value of new data, x_n, is p-value$(x_n) := P(\{X : P(X) \leq P(x_n)\})$. Generally, a threshold $\alpha \in [0, 1]$ is set, and if p-value$(x_n) \leq \alpha$, x_n is identified as anomalous. For streaming data, model parameters are iteratively updated to include the new observation, x_n. Often, α is tuned in light of labeled results to find an acceptable balance of false vs. true positives. Analysis in [8] identifies operational benefits of the method, including a theorem allowing users determine α à priori by setting an expected alert rate. We utilize the framework's accommodation of any probability model in order to apply it simultaneously at hierarchical levels.

III. The Generalized BTER Model (GBTER)

In order to perform probabilistic anomaly detection, we need a generative graph model that enables computation of probabilities for various graph configurations while accurately modeling a graph's community structure and degree sequence. Significant prior work has been devoted to developing such models and validating the importance of capturing both these aspects of a real-world data set (e.g., [1], [9], [10]). Motivated by social and cyber settings, we require a generative model that can accommodate observed hierarchical structure. A natural candidate is a Stochastic Block Model, first introduced in [11], in which community membership is defined and edge probability depends on membership of the endpoints. This achieves flexible community membership and density, but the expected degree of each node cannot be set explicitly. To improve adherence to degree distribution, one could use the Block Two-Level Erdös-Rényi (BTER) [1], [10], but we found the implicitly determined community structure of the model to be too limiting for matching real-world data. To address these challenges, we define and use a generalization of BTER that allows explicit prescription of the communities' size, membership, and approximate density.

The Generalized Block Two-level Erdös-Rényi (GBTER) model takes as input (1) the expected degree of each node, (2) community assignments of the nodes, i.e., a partition of the vertex set into disjoint subsets, $\{C_j\}$, and (3) an edge probability p_j for edges within each community C_j. In the first stage of edge generation, within-community edges are sampled from an Erdös-Rényi [12] model[4], ER($|C_j|, p_j$), for each community C_j. Note the expected degree of a node within C_j is $p_j(|C_j| - 1)$ after the first stage. In the second stage, we define the *excess expected degree* of a node i, denoted ε_i, to be the difference between the input expected degree λ_i and the expected degree after stage one. Formally, $\varepsilon_i := \max(0, \lambda_i - p_j(|C_j| - 1))$ for node i in community C_j. We then apply a Chung-Lu style model [9] on the excess expected degree-sequence, $[\varepsilon_i]_{i \in V}$. Specifically, the probability of adding the edge (i, j), is

$$P(i, j \mid \varepsilon) = \frac{\varepsilon_i \varepsilon_j}{\sum_k \varepsilon_k}. \tag{1}$$

Note that the second stage can generate both inter- and intra-community edges. It is necessary that Chung-Lu inputs, $\{\varepsilon_i\}$, satisfy $\varepsilon_i \varepsilon_j \leq \sum_k \varepsilon_k$ for Equation 1 to define a probability. A calculation shows that the expected degree of node i is indeed d_i whenever $d_i \geq p_j(|C_j| - 1)$ (i.e., the expected degree from the first-stage edges does not exceed the total expected degree of any node), thus the CL model is well-defined.

To calculate the probability of edge (i, j), we condition on whether i and j share a community. Recall, our communities partition the set of nodes, so each i is in exactly one community. If i and j are assigned to the same community, C, let p denote the internal edge probability of C, and we see

$$P(i, j \mid i, j \in C) = p + (1 - p)\frac{\varepsilon_i \varepsilon_j}{\sum \varepsilon_k}. \tag{2}$$

If i and j are assigned to different communities, the edge probability is as given in Equation 1.

GBTER differs from the original BTER model by allowing greater flexibility and assignment of community membership, size, and internal edge density (p). As indicated in [10], the expected clustering coefficient for an ER(n, p) graph is p^3. This implies that GBTER also allows pre-specification of each community's approximate clustering coefficient. Note that GBTER assumes node labels allowing specification of community assignments, while BTER only depends on the number of nodes of each expected degree, and BTER community assignment is not known/specified à priori. Since edge and graph probabilities depend on community assignments (in both models), such calculations are complicated and expensive in the BTER (as all possible community assignments must be considered), which inhibits its use for anomaly detection.

IV. Fitting Model Parameters

We now describe how to fit the GBTER model to a sequence of observed graphs with common node labels using Bayesian techniques for learning the parameters and inferring

[4]Erdös-Rényi graphs, denoted ER(n, p), have n fixed nodes and each possible edges occurs independently with probability p.

the following model inputs: (1) the community assignments, (2) the within-community edge densities, and (3) the expected node degrees. Once a specific instance of the model is deduced, probabilistic anomaly detectors are constructed, as detailed in Section V.

In this paper, a partition of the vertex set into communities is learned using the Markov Clustering (MC) algorithm [13]. We chose MC as it is known to scale well and is easy to implement. To apply MC, a weighted graph is constructed by counting occurrences of each edge in observed graphs. We note that our general method requires a partition of the nodes into communities but is blind to the algorithm used. For example, communities inferred from context (e.g., grouping nodes by a known, common affiliation) can be used to obviate this step and may provide more insightful results in a real-world setting.

Given community assignments, the within-community edge densities are estimated. Each community, C, is modeled internally by an Erdös-Rényi random graph, ER($|C|, p$), and we seek to estimate p. Letting k denoted the number of edges within the subgraph C, it follows that $k \sim \text{Binomial}(\binom{|C|}{2}, p)$. In order to use Bayesian inference, we assume $p \sim \text{Beta}(\alpha, \beta)$, with prior parameters $\alpha > 0, \beta > 0$, and then use the maximum posterior likelihood estimation (MPLE). Specifically, $(p|k_1, \ldots, k_N) \sim \text{Beta}(\hat{\alpha}, \hat{\beta})$ with posterior parameters $\hat{\alpha} = \alpha + \sum_i k_i$, and $\hat{\beta} = \beta + N\binom{|C|}{2} - \sum_i k_i$ where k_i denotes the number of edges internal to C observed in the i-th graph, G_i, for $i = 1, \ldots, N$. MPLE gives $p := (\hat{\alpha} - 1)/(\hat{\alpha} + \hat{\beta} - 2)$, the mode of the posterior.

Lastly, the expected degree sequence must be estimated from the data. For a fixed node, we assume its degree, d, is Poisson distributed with expected degree λ, i.e. $d \sim \text{Poisson}(\lambda)$. We use the conjugate prior, $\lambda \sim \text{Gamma}(\alpha, \beta)$ with prior parameters $\alpha > 1, \beta > 1$. This yields the posterior distribution, $(\lambda|d_1, \ldots, d_N) \sim \text{Gamma}(\hat{\alpha}, \hat{\beta})$ with posterior parameters $\hat{\alpha} = \alpha + \sum_i d_i$, and $\hat{\beta} = \hat{\beta} + N$, where d_i denotes the observed degree of the node in G_i. For each node, MPLE gives its expected degree, $\lambda := (\hat{\alpha} - 1)/\hat{\beta}$, the mode of the posterior Gamma.

V. Anomaly Detectors

Given an instance of a GBTER model, which defines a probability distribution on graphs, one can leverage the distribution to detect anomalies at the graph, subgraph, and node level. This section defines two multi-scale detectors, one which uses the GBTER distribution directly, and one which leverages statistics inherent to the GBTER model. The Multi-Scale Probability Detector naturally uses the graph probability as determined by the GBTER model for detection, which is then decomposed into probabilities of subgraphs and nodes for hierarchical information. Although intuitive, this detector suffers from a few limitations, discussed below, which inform construction of the Multi-Scale Statistics Detector. This second detector builds from the bottom up, defining the probability of a node based on the likelihood of its internal and external degree. Subgraph probabilities are determined by those of member nodes, so multi-scale analysis is facilitated by both models. Lastly, we describe a baseline method for detecting anomalous graphs by fitting Gaussian distributions to graph statistics. We note that the Gaussian Baseline is only used for identifying anomalous graphs, as it cannot discriminate at the subgraph or node level.

A. Multi-Scale Probability Detector

Our first method uses the graph probability, as given by the GBTER model, for anomaly detection. Specifically, given a graph $G = (V, E)$ with vertices V and edges E, the probability of G is

$$P(G) = \prod_{(i,j)\in E} P(i,j) \prod_{(i,j)\notin E} (1 - P(i,j)), \tag{3}$$

where $P(i,j)$ is the probability of the edge (i,j) under the GBTER model, as derived in Section III. In practice, given a graph G, we compute its probability using Equation 3, then use Monte-Carlo simulation to estimate its p-value.

In order to detect anomalies at different scales, the probability of a graph is decomposed into a product of subgraph probabilities. Specifically, we define the probability of node i_0 as

$$P(i_0) := \prod_{j:(i_0,j)\in E} P(i_0,j) \prod_{j:(i_0,j)\notin E} (1 - P(i_0,j)).$$

It follows that $P(G) = \prod_i P(i)^{1/2}$. Similarly, the probability of a subgraph $G' = (V', E')$ is defined to be $\prod P(i)^{1/2}$, with the product over $i \in V'$. Hence, given a partition of V into communities, $\{C_i\}$, the probability of G also breaks into a product of community probabilities, i.e., $P(G) = \prod_i P(C_i)$. This formulation allows anomaly detection of any fixed subgraph, in particular at the node, community, and graph level.

The probability of sampling a graph under a given generative model is an intuitive choice for anomaly detection. Upon further examination, this technique will yield poor results in models where the mode of the distribution varies depending on whether or not labels are regarded. As an illustrative example, consider the ER model on three labeled nodes, $V = \{1, 2, 3\}$ with $p = 1/3$. The most probable unlabeled graph under this distribution has exactly one edge, and occurs with probability $\binom{3}{1}(1/3)(2/3)^2 = 4/9$. Now labeling nodes, there are three different but isomorphic graphs with one edge each, namely, with edge $(1,2)$ or $(2,3)$ or $(1,3)$ only. But the probability of each of these one-edge graphs is $(1/3)(2/3)^2 = 4/27$, while the probability of the empty graph is $(2/3)^3 = 8/27$. Hence when labels are regarded, the mode of the distribution is the empty graph, not the one-edge graphs as in the unlabeled case; consequently, in this case the Multi-Scale Probability Model will view the expected graphs as more anomalous than the less likely empty graph! Now consider the GBTER model used in the our experiments. Because the probability of a within-community edge is greater than $1/2$ and inter-community edge is less than $1/2$ with the given parameters, the labeled-node mode of the distribution is the graph with every community as a clique and no other edges. Although this graph is unlikely to be sampled, the Multi-Scale Probability Model will regard it as the most "normal" possible graph. The conclusion of this reasoning is that using the graph's probability will produce unwarranted results, yet modeling characterizing statistics of the graph (e.g., inter- and intra-community node degrees) gives accurate detection capabilities. This is exhibited in our empirical results, and motivates the second detector.

B. Multi-Scale Statistics Detector

Our second detector is based on observing and modeling intra- and inter-community node degrees (after learning GBTER parameters). Fix a node $i_0 \in V$, and let C denote i_0's community, p denote C's intra-community edge probability, and λ the expected degree of node i_0 (all as learned from fitting the GBTER model to our observations). We set $d_{in} := |\{(i_0, j) \in E : j \in C\}| = i_0$'s internal degree, and $d_{ex} := |\{(i_0, j) \in E : j \notin C\}| = i_0$'s external degree. Following the ER($|C|, p$) assumption, we assume $d_{in} \sim$ Binomial($|C|-1, p$), and $d_{ex} \sim$ Poisson(ε), where $\varepsilon = \max(0, \lambda - p(|C|-1))$, is the excess expected degree of i_0 (see Section III). For the Multi-Scale Statistics Detector, the probability of node i_0 is defined as the joint probability of its degrees, $P(i_0) := P(d_{in}, d_{ex})$. We assume the two degrees are independent and obtain

$$P(i_0) = \binom{|C|-1}{d_{in}} p^{d_{in}} (1-p)^{|C|-1-d_{in}} \frac{e^{-\varepsilon}\varepsilon^{d_{ex}}}{d_{ex}!}. \quad (4)$$

Given a subgraph $G' = (V', E')$ we set $P(G') := \prod_{V'} P(i)$.

Note that since GBTER allows both internal and external edges to be created by the second stage of the process, the model above inflates internal degree d_{in} and deflates d_{ex} compared to GBTER. Additionally, as the range of a Poisson variable is unbounded, degrees exceeding $|V|-1$ (an impossibility) are assigned positive probability by this model. To circumvent this possibility, the truncated Poisson can be used for sampling. In our experiments, the expected degree (λ) and expected excess degree (ε) are sufficiently smaller than $|V|-1$, which implies the $P(\deg(i) > |V|-1|)$ is negligible. Testing with and without the truncation exhibited similar results.

To use either of the multi-scale detectors, we set thresholds at each level, and any node/subgraph/graph with p-value below the respective threshold is labeled anomalous. The model parameters are updated upon receipt and detection of each graph.

C. Gaussian Baseline Detector

Our baseline method fits univariate Gaussian distributions to graph statistics and uses the product of the p-values for detection. We compute three statistics from each observed graph: average node degree (X_1), average clustering coefficient (X_2), and the spectral norm (X_3). Calculating X_1 and X_2 from a given graph is straightforward. In order to calculate X_3, the GBTER model is used with parameters estimated as described above to produce the expected adjacency matrix $E(A)$, in which $E(A)_{i,j}$ gives the probability of an edge between nodes i and j. The spectral norm is defined as the maximum modulus eigenvalue of the residual matrix $A - E(A)$. After computing the observed statistics, independent univariate Gaussian distributions ($\mathcal{N}(\mu_i, \sigma_i)$) are fit to each of the three statistics. Lastly, given a newly observed graph, G, with statistics x_1, x_2, x_3, we assign p-value(G) := $\prod_{i=1}^{3} P(X_i \leq x_i | \mathcal{N}(\mu_i, \sigma_i))$. As before, p-values falling below a given threshold, α, are labeled anomalous, and the three normal distributions are updated upon receipt of each new graph.

This follows the approach of Moreno and Neville [2], although their work is based on mixed Kronecker Product graphs and uses average geodesic distance instead of the spectral norm we employ for X_3. Since the average geodesic distance is undefined for disconnected graphs, we selected the spectral norm based on prior use in network hypothesis testing and strong results for similar tests involving Chung-Lu random graphs [5]. While we consider this baseline a natural adaptation of [2], the disparity in use between their and our application inhibits direct comparison.

TABLE I: Community assignments for first GBTER experiment

	C_1	C_2	C_3
M_r	[0,1,2,3]	[4,5,6,7]	[8,9,10,11]
M_a	[0, *11*, 2, *4*]	[*3*, 5, 6, *8*]	[*7*, 9, 10, *1*]

Note: For the first experiment, the seeded-anomaly model M_a is obtained from M_r by switching the position of two nodes from each of the first three communities, C_1, C_2, C_3. Anomalous nodes shown in italicized red print, and anomalous communities are circled. Communities 4-10 are unchanged and not shown.

VI. Synthetic Graph Experiment

In order to test the anomaly detection capabilities, two hidden GBTER models are used to generate labeled data, (1) a "regular" model, M_r, for sampling non-anomalous graphs, and (2) a seeded-anomaly model, M_a, with slightly perturbed inputs to generate anomalous graphs. To begin the experiment, 100 non-anomalous graphs are sampled from M_r, and the anomaly detectors are fit to the data, as described in Section V. Prior distributions Beta(1,1), Gamma(2,2) were used. To test the streaming anomaly detection, 500 graphs are iteratively generated and observed with every fifth graph from the seeded anomaly model. Upon sampling a new graph, its p-value according to each anomaly detector is computed (and it is labeled as anomalous if it falls below a given threshold). Similarly, the hierarchical detectors label each node and community depending on its respective p-value. Lastly, each anomaly detector's GBTER parameters are updated to include observation of the new graph.

We conduct two experiments, both using networks of 40 nodes divided into ten equally-sized communities. For the "regular" model, each community is assigned a within-edge probability of $p = .8$, and the expected degrees of nodes vary in the range of five to eight according to a truncated power-law. To create the seeded-anomaly model for the first experiment, two nodes from each of the first three communities are interchanged resulting in six (of 40) anomalous nodes and three (of ten) anomalous communities per anomalous graph (see Table I). For the second experiment, community assignments are held constant, but the within-community density (p) of the first four communities is changed from 0.8 to 0.4 in the seeded-anomaly model, and the expected degree of the nodes in these four communities is increased by two. This will decrease intra-community, and increase extra-community interaction for these four communities. All together the second experiment has four (of ten) anomalous communities, and 16 (of 40) anomalous nodes per anomalous graph.

Results of the experiments are given in Table II, which includes the AUC as well as Precision, Recall, and F1 for each

TABLE II: GBTER Experiment Results (α maximizing F1)

Method	α	F1	Precision	Recall	AUC
EXPERIMENT 1					
Graph Level					
Graph Probability	0.020	0.742	0.678	0.820	0.934
Graph Statistic	0.009	**0.919**	**0.929**	**0.910**	**0.991**
Gaussian Baseline	0.029	0.526	0.418	0.710	0.785
Community Level					
Graph Probability	0.019	0.810	0.745	**0.887**	**0.987**
Graph Statistic	0.009	**0.830**	**0.840**	0.820	**0.987**
Node Level					
Graph Probability	0.020	0.298	0.239	0.393	0.877
Graph Statistic	0.017	**0.547**	**0.453**	**0.690**	**0.951**
EXPERIMENT 2					
Graph Level					
Graph Probability	0.007	0.895	0.855	**0.940**	0.984
Graph Statistic	0.011	**0.922**	**0.904**	**0.940**	**0.993**
Gaussian Baseline	0.006	0.590	0.697	0.510	0.809
Community Level					
Graph Probability	0.062	0.436	0.390	0.495	0.838
Graph Statistic	0.028	**0.654**	**0.620**	**0.693**	**0.936**
Node Level					
Graph Probability	0.053	**0.436**	0.368	**0.533**	**0.894**
Graph Statistic	0.047	0.434	**0.427**	0.442	0.821

detector at the threshold α maximizing its F1 score.[5] Recall that the Gaussian Baseline is only for graph level detection and thus does not contribute to the community or node level results. For the full-graph tests, the Gaussian Baseline is far inferior to the new models with the Multi-Scale Statistics Detector as the clear winner. Further, the results at all levels provide evidence that the Multi-Scale Statistics Detector is the superior method, as expected after the à priori analysis given in Section V-A.

VII. Conclusions and Future Work

As many applications involve representing data with known entities and time-varying relationships, this work considers a sequence of graphs with node labels and changing edges. Our goals were to investigate a method for finding abnormalities in such a graph sequence that (1) use multiple, related levels of granularity to facilitate an understanding of why/how an anomaly occurred, and (2) to leverage node labels for more accurate detection. To this end, we introduced GBTER, a generalization of the BTER graph model, that allows more accurate modeling of community structure, and built two hierarchical streaming anomaly detectors. The first intuitively uses the graph's probability as given by the model, yet more thorough analysis suggests that the inability of graph models to distinguish isomorphic copies with different node labels will inhibit detection accuracy. Secondly, a statistics-based detector, that respects the node labels in each graph is created. Our hypothesis that the statistics-based detector will give more accurate results is verified in two tests on synthetic data where ground-truth is known at the node, subgraph, and graph levels. Additionally, both detectors outperform a baseline detector that fits Gaussian distributions to observed statistics of the full graph. We believe applying this method to other time-sampled social networks will enable discovery of underlying structure and anomalies with the context in which they occur. Also, the multi-scale detection will inform an intuitive graph visualization for "zooming-in" on detected regions.

[5]AUC statistic denotes the area under the receiver operator characteristic (ROC) curve. F1 is defined as the harmonic average of Precision, P, and Recall, R. Specifically, F1:=ave$(P^{-1},R^{-1})^{-1} = 2PR/(P+R)$.

While investigations of scalability are outside the scope of this work, we expect applications of this approach to necessitate larger data. Here we identify the bottlenecks in the current implementation for future efforts. Firstly, this approach requires a partition of the nodes into communities, but is agnostic to the method used. Hence, we have the ability to optimize performance by the partitioning algorithm chosen. As mentioned above, using communities known from context can obviate this step and provide groupings that are familiar to the operator. Secondly, estimating the p-values of a given distribution can be computationally expensive, especially if it requires sampling large graphs and calculating their probabilities. In general, importance sampling, in which one over-samples from a subset of the event space, can aid in Monte-Carlo simulations, although further research is required to optimize performance gains for our needs. Thirdly, the choice of probability models of the parameters could be changed to admit easier p-value computation. For example, multinomials become robust with abundant observations. In a specific application, flexibility in the modeling may yield increased performance with negligible effects on accuracy. Lastly, adapting the overall workflow to fit a specific application may admit performance gains. For example, updating parameters less often (in a batch process periodically) or discarding anomalous data from the update observations are options that have yet to be explored. In summary, while the current implementation is suitable only for small datasets, the approach gives opportunities for scalability and should be adaptable to high-volume and/or large-network settings.

References

[1] C. Seshadhri, T. G. Kolda, and A. Pinar, "Community structure and scale-free collections of Erdős-Rényi graphs," *Physical Review E*, vol. 85, no. 5, p. 056109, 2012.

[2] S. Moreno and J. Neville, "Network hypothesis testing using mixed kronecker product graph models," in *IEEE 13th International Conference on Data Mining (ICDM)*. IEEE, 2013, pp. 1163–1168.

[3] W. Eberle and L. Holder, "Anomaly detection in data represented as graphs," *Intelligent Data Analysis*, vol. 11, no. 6, pp. 663–689, 2007.

[4] B. A. Miller, N. T. Bliss, P. J. Wolfe, and M. S. Beard, "Detection theory for graphs," *Lincoln Laboratory Journal*, vol. 20, no. 1, 2013.

[5] B. A. Miller, L. H. Stephens, and N. T. Bliss, "Goodness-of-fit statistics for anomaly detection in Chung-Lu random graphs," in *International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2012, pp. 3265–3268.

[6] L. Peel and A. Clauset, "Detecting change points in the large-scale structure of evolving networks," *arXiv preprint arXiv:1403.0989*, 2014.

[7] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: a survey," *Data Mining and Knowledge Discovery*, pp. 1–63, 2014.

[8] E. M. Ferragut, J. Laska, and R. A. Bridges, "A new, principled approach to anomaly detection," in *International Conference on Machine Learning and Applications*, vol. 2. IEEE, 2012, pp. 210–215.

[9] F. Chung and L. Lu, "The average distances in random graphs with given expected degrees," *Proceedings of the National Academy of Sciences*, vol. 99, no. 25, pp. 15 879–15 882, 2002.

[10] T. G. Kolda, A. Pinar, T. Plantenga, and C. Seshadhri, "A scalable generative graph model with community structure," *arXiv preprint arXiv:1302.6636*, 2013.

[11] P. W. Holland, K. B. Laskey, and S. Leinhardt, "Stochastic blockmodels: First steps," *Social Networks*, vol. 5, no. 2, pp. 109–137, 1983.

[12] P. Erdős and A. Rényi, "On random graphs," *Publicationes Mathematicae Debrecen*, vol. 6, pp. 290–297, 1959.

[13] S. M. Van Dongen, "Graph clustering by flow simulation," Ph.D. dissertation, University of Utrecht, 2000.