# A Graph Embedding Approach to User Behavior Anomaly Detection

Alexander Modell*
School of Mathematics
University of Bristol
Bristol, UK
Email: alexander.modell@bristol.ac.uk

Jonathan Larson
Microsoft
Redmond, WA, USA
Email: jolarso@microsoft.com

Melissa Turcotte
Microsoft
Redmond, WA, USA
Email: melissa.turcotte@microsoft.com

Anna Bertiger
Microsoft
Redmond, WA, USA
Email: anberti@microsoft.com

***Abstract*—Identifying suspicious user behavior within an enterprise network is vital to maintaining strong cyber security defenses. This paper presents a scalable approach to detecting anomalous user behavior in event logs, which we frame as a dynamic, bipartite interaction network of users and resources. Graph embedding is used to obtain vector representations of users, which are updated over time and used to model the profile of the users who typically access each resource. A standard nearest neighbor anomaly detection method is then employed to score new interactions. The approach is applied to a dataset of interaction events between users and SharePoint sites within Microsoft's internal corporate network.**

***Index Terms*—anomaly detection, graph embedding, cyber security**

## I. Introduction

User behavior anomaly detection refers to a collection of network security techniques that aim to detect unusual patterns of user activity. Detection and evaluation of such patterns are essential to the identification of security breaches.

Traditional security systems provide a vital first layer of defense, employing predefined signatures and rules to identify known threats. While these systems are invaluable, they are unable to identify unknown threats and so-called zero-day attacks and are increasingly evaded by the most sophisticated actors.

Anomaly detection systems form a second line of defense, by monitoring users and notifying of any activities which do not conform to normal behavior on the network. In this way, malicious behavior that can bypass signatures, such as network traversal using stolen credentials, or attacks exploiting previously undiscovered vulnerabilities, can be detected.

There are many practical challenges when deploying anomaly detection systems in cyber security. Users often perform a wide range of actions and malicious activity is often subtle. Any model must be flexible enough to capture the full range of normal activity so that significant amounts of benign behavior are not flagged as suspicious, and specific enough that malicious behaviors stand out and are discovered. In addition, such systems typically must process large amounts of data at very high rates, so must be simple enough to be computationally tractable at these scales. Detected anomalous behaviors may serve as indicators that can gain strength when combined with other evidence to indicate a cyber-attack, or suggest further investigation from an analysis team.

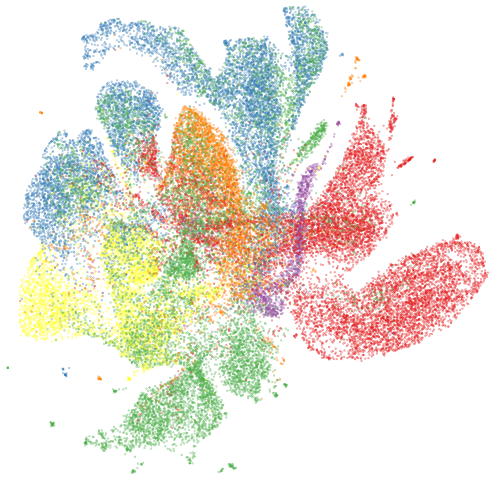

Fig. 1. Uniform manifold approximation and projection (UMAP) of the 12-dimensional space of user embeddings, colored according to the user's organization.

In this paper, we propose a framework for detecting anomalies in event logs describing access patterns to a collection of resources by a set of users. The approach fundamentally targets peer-based anomalies, in which the behavior of a user is surprising in the context of the behavior of their peers. We frame this data as a dynamic bipartite graph, in which timestamped edges represent interaction events between users and resources. We represent the users in a feature space, which we propose to learn entirely from the graph structure, using

*work performed during an internship at Microsoft Research, Redmond.

a graph embedding technique. These representations encode the peer structure of the users: a distance between two users encodes their similarity.

Graph embedding is a general-purpose machine learning tool that computes vector representations of the nodes of a graph that reflect the connection patterns observed in it. Spectral embedding refers to an embedding procedure that employs the spectral decomposition of a matrix representation of the graph. There are many variants involving different matrix representations, regularization which induces a more balanced embedding, and degree-correction which removes the dependence of degree from the embedding. These methods are well understood from a statistical perspective and benefit from being fast to compute.

Based on these representations, we define an independent non-parametric model for each resource, describing the users who typically access it, and employ a nearest-neighbor anomaly detection scheme to detect anomalous interactions. The resulting score quantifies the level of surprise, from the perspective of the resource, at receiving an interaction from the user in question.

## II. Related work

### A. Graph-based anomaly detection

Graph-based anomaly detection has been approached in a variety of ways. Akoglu et al. [1] and Ranshous et al. [2] provide comprehensive surveys. In the context of bipartite graphs, previous approaches include subgraph-based methods [3]–[5], which identify substructures in the graph to detect malicious connections, community-based methods [6]–[10], which detect nodes that do not respect community boundaries, and Bayesian approaches [11]–[13], which measure anomalies against a learned statistical model.

Our approach allows us to make minimal assumptions about the nature of the bipartite graph. We do not assume community structure, but we can make use of such structure if it exists in the graph data. In addition, the assumptions we make on the nature of anomalies are simple, interpretable, and do not require feature engineering. The minimal assumptions on which our framework is based allow it to be used in a wide array of applications.

### B. Spectral graph embedding

Spectral graph embedding is very well studied from a statistical perspective, with consistency results and central limits theorems available for adjacency and Laplacian embedding [14]–[19]. Modifications to the standard algorithm have been proposed to improve empirical performance. Degree correction [20], [21] involves removing information about the number of connections made by a node, so nodes whose connectivity preferences are similar, but whose activity levels are different are embedded into similar positions. Regularization [21]–[26] involves artificially inflating the node degrees prior to embedding and has been shown to improve performance in the presence of severe degree heterogeneity.

While the majority of the literature is focused on unipartite graphs, embedding of bipartite graphs via a truncated singular value decomposition [27]–[29] has been studied and employed extensively in practice [30], [31].

Alternative approaches to spectral graph embedding include those based on random walks [32]–[34], deep learning [35], [36] and alternative matrix factorizations [11], [37], [38].

### C. Anomaly detection

Chandola et al. [39] give a survey of anomaly detection methods for point cloud data. Nearest-neighbor based methods, which we employ in our methodology, are a simple family of methods which assume that an anomalous data point is one which has few close neighbors, making no parametric assumptions on the data. They have been used widely in intrusion detection systems [40]–[42] and their statistical properties have been studied [43].

## III. Approach

Consider a dynamic bipartite graph with $m$ user nodes $V_u$, $n$ resource nodes $V_r$ and timestamped edges $E \subset V_u \times V_r \times \mathbb{R}$. Here, an edge $(u, r, t) \in E$ represents an edge between user $u$ accessing resource $r$ at time $t$. For a time $t \in \mathbb{R}$, let $\mathbf{A}^{(t)} \in \mathbb{R}^{m \times n}$ denote a (potentially weighted) biadjacency matrix which represents a snapshot of the graph at time $t$.

One such construction is to set $\mathbf{A}^{(t)}_{ur} = 1$ if for any $s < t$, $(u, r, s) \in E$ and $\mathbf{A}^{(t)}_{ur} = 0$ otherwise. Alternatively, edges may be weighted to increase the importance of frequently occurring interactions, such as using log counts (see [44] for guidance on choosing edge weights). Edges may also be removed after a fixed period of time if another interaction has not occurred, or the edge weights set to decay with time so that more recent interactions are weighted more highly than less recent ones.

The general framework for scoring a new edge $(u, r, t)$ is as follows:

1) *Graph embedding.* Using $\mathbf{A}^{(t)}$, compute an embedding of the user nodes, $X^{(t)}_1, \ldots, X^{(t)}_m \in \mathbb{R}^d$.
2) *Anomaly detection.* Let $\mathcal{X}^{(t)}_r = \{X^{(t)}_v : (v, r, s), s < t\}$ denote the set of user embeddings for users who have accessed resource $r$ before time $t$. Return the anomaly score $s_{(u,r,t)}$, a distance from $X_u$ to its nearest neighbor in $\mathcal{X}_r$.

From hereon, we drop the dependence on $t$.

### A. Graph embedding

Following [28], we perform spectral embedding using the regularized bi-Laplacian matrix and subsequently project the resulting embeddings onto the unit sphere. The regularized bi-Laplacian matrix $\mathbf{L}_\tau$, with regularization parameter $\tau \in \mathbb{R}_+$, is defined as

$$\mathbf{L}_\tau = (\mathbf{D}^{(1)} + \tau \mathbf{I}_m)^{-1/2} \mathbf{A} (\mathbf{D}^{(2)} + \tau \mathbf{I}_n)^{-1/2}$$

where $\mathbf{D}^{(1)}$ and $\mathbf{D}^{(2)}$ are the diagonal user and resource degree matrices with $\mathbf{D}^{(1)}_{uu} = \sum_r \mathbf{A}_{ur}$ and $\mathbf{D}^{(2)}_{rr} = \sum_u \mathbf{A}_{ur}$, and $\mathbf{I}_m$ and $\mathbf{I}_n$ are the $m \times m$ and $n \times n$ identity matrices. Given the

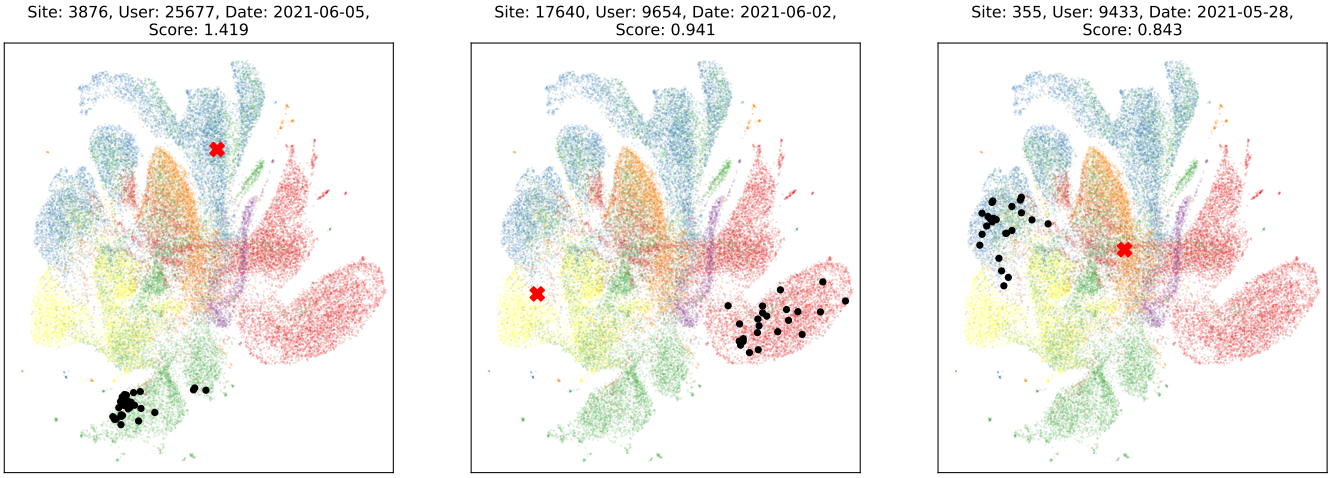


Fig. 2. Three examples of user-resource pairs which received high anomaly scores. Each panel shows the UMAP projection of the user embedding, colored by organization, as small points. Overlaid are large black points representing the users who had previously accessed the resource, and a red cross, representing the user who raised the anomaly.

regularized bi-Laplacian matrix and an embedding dimension $d$, the embedding algorithm is as follows:

1) Denote the rank-$d$ singular value decomposition of $\mathbf{L}_\tau$ as $\mathbf{USV}^\top$ and let

$$\mathbf{X}^* = [X_1^*, \ldots, X_m^*]^\top = \mathbf{US}^{1/2} \in \mathbb{R}^{m \times d},$$

$$\mathbf{Y}^* = [Y_1^*, \ldots, Y_n^*]^\top = \mathbf{VS}^{1/2} \in \mathbb{R}^{n \times d}$$

2) Define $X_u = X_u^*/\|X_u^*\|$ and $Y_r = Y_u^*/\|Y_u^*\|$ as the projection of $X_u^*$ and $Y_r^*$ onto the unit sphere, respectively.

The vectors $X_1, \ldots, X_m \in \mathbb{R}^d$ are embeddings of the users and $Y_1, \ldots, Y_n \in \mathbb{R}^d$ are embeddings of the resources. Our approach only requires the user embeddings.

The embedding dimension $d$ is a hyper-parameter that needs to be chosen. It is recommended to choose $d$ by examining a plot of the singular values of the graph adjacency matrix, known as a scree plot, and identifying an "elbow" where the singular values level off. This can be done by eye or using an automated method such as the profile likelihood method of Zhu and Ghodsi [45].

It has been recommended that a good, general choice for the regularization parameter is the average degree of the graph [21], [28]. Regularization improves the performance of spectral embedding which can ordinarily perform poorly in the presence of severe degree heterogeneity [25], [26], a common feature of these kinds of data [46]. The regularization parameter here plays a similar role to that in ridge regression [47]. The second stage of the algorithm, projecting the embedding onto the unit sphere, performs degree correction, that is, it removes the dependence of a node's degree from its position in the embedding space. In this way, the implicit notion of similarity between users contains only information about the profile of the resources accessed and not their level of activity. This is a recommendation that has been made extensively in the literature [20], [21].

### B. Anomaly detection

To score a new edge $(u, r, t)$, we employ a simple nearest-neighbor anomaly detection algorithm. Let $\mathcal{X}_r = \{X_v : (v, r, s), s < t\}$ denote the set of user embeddings for users who have accessed resource $r$ before time $t$. Given a choice of metric, such as Euclidean or cosine distance, the anomaly score for an edge is given by the distance from $X_u$ to its nearest neighbor in $\mathcal{X}_r$.

If a user $u$ has previously accessed a resource $r$ before time $t$, an edge $(u, r, t)$ will receive an anomaly score $s_{(u,r,t)} = 0$, since $X_u \in \mathcal{X}_r$. Otherwise $s_{(u,r,t)} > 0$. In an anomaly detection setting, it is typical to set a threshold $\alpha \in \mathbb{R}$, and to flag an edge as anomalous if its anomaly score is greater than $\alpha$. Practically, such a threshold is chosen so that the number of detections is reasonable, relative to the capacity for them to be investigated. Setting $\alpha = 0$ is equivalent to flagging an edge whenever a user accesses a resource they have not accessed before.

### C. Practical considerations

There are some practical considerations to consider when implementing this approach. Firstly, in practice, it is not necessary to update the embedding every time a new edge is observed. The positions remain relatively stable over time, so they may instead be recomputed at regular intervals such as daily or weekly.

For very large or dense graphs, it may be desirable to avoid recomputing the singular value decomposition each time an update is required. In this case, the embedding may be updated approximately using an out-of-sample method [48].

With this in mind, due to the simplicity of the anomaly detection step, it is possible to score edges quickly and in

real-time, a property that is critical in many cyber security applications.

Finally, it is advisable to train the algorithm for a period of time prior to scoring edges, so that the embedding contains enough information to reflect the characteristics of the nodes. This amounts to computing an initial embedding based on data from this period. Without this, the initial scores are likely to be unreliable.

## IV. SharePoint access data from Microsoft corporate network

We have applied our approach to a dataset of event logs of users and Microsoft SharePoint sites within Microsoft's internal corporate network. Microsoft SharePoint is a collaborative document management tool for sharing information within an organization. A SharePoint site is a location where a team can share content related to a specific project.

We consider a subset of users corresponding to a sub-organization of Microsoft, and a subset of SharePoint sites with at least 5 and at most 5,000 users in the time period we consider. These are the sites that we deem to be of interest from a security perspective. For example, sites with a large number of users are likely to be commonly referenced and are unlikely to contain sensitive information. In total, this comprises 42,643 users and 29,279 SharePoint sites. An edge represents any interaction between a user and a SharePoint site.

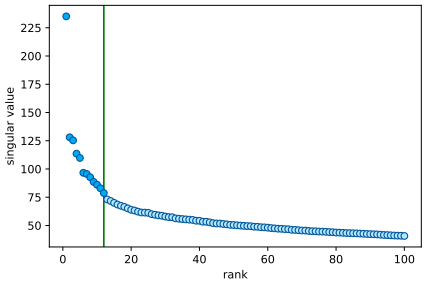


Fig. 3. Plot of the singular values of the biadjacency matrix for the induced graph of the first 28 days of activity.

We initially train the system on 28 days of logs. Figure 3 shows the scree plot of the top 100 ordered singular values of the biadjacency matrix of the induced graph of this period. We choose $d = 12$ as our embedding dimension based on the elbow identified in the plot.

Figure 1 shows the user embedding of the graph, which for visualization has been reduced to two dimensions using the uniform manifold approximation and projection (UMAP) [49] non-linear dimensionality algorithm. The colors correspond to the manager of each user at a particular level of the organizational hierarchy.

For the subsequent 28 days, we score the user-site interactions using our methodology, updating the embedding each day. We score a total of 5,936,732 edges, 81.9% of which receive a score of zero, indicating that the edge has occurred previously. Figure 4 shows the distribution of the non-zero anomaly scores, using Euclidean distance.

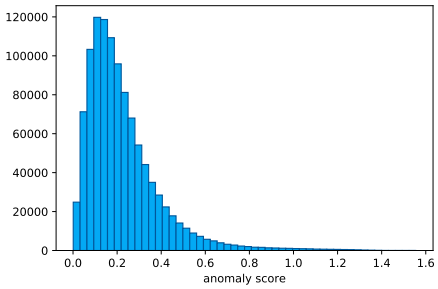


Fig. 4. Distribution of the non-zero anomaly scores.

### A. Evaluation of the method

In general, quantitative evaluation of anomaly detection methods in cyber-security is complicated by the fact that ground truth labels of malicious behavior are typically not available in real data. To overcome this, we construct a set of surrogate labels based on meta-data which represent a notion of peer-based anomalous behavior that we might want to detect.

Employees at Microsoft are organized hierarchically into organizations. We define an *organizational anomaly* to be an interaction between and a user and a site for which no other member of the user's organization, at a specified level, has interacted with previously.

The are 20,018 organizational anomalies in our data, a similar amount to thresholding our method at 0.75 (19,368).

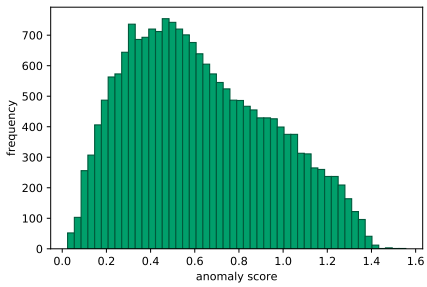


Fig. 5. Distribution of the anomaly scores associated with interactions labeled as organizational anomalies.

Figure 5 shows the distribution of the anomaly scores computed using our method for edges labeled as organizational anomalies, and Figure 6 shows the proportion of these edges which are flagged as anomalies using our approach for varying thresholds. It can be seen that organizational anomalies are typically assigned a high anomaly score using our approach.

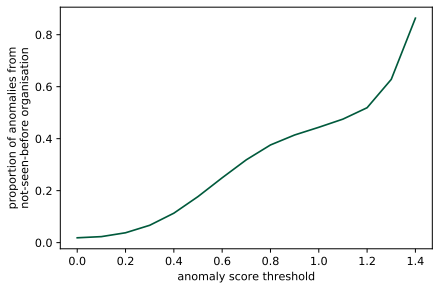


Fig. 6. Proportion of anomalies flagged using our method which are organizational anomalies at each threshold.

## V. Conclusion

We have presented a simple approach to detecting anomalies in a dataset of interaction events between users and resources. The framework we develop is very general and our choices of graph embedding and anomaly detection algorithms may be modified to suit the problem at hand. For example, the graph embedding may be performed using any other suitable algorithm and the nearest-neighbor anomaly detection scheme may be replaced by the $k$th nearest neighbor, for example, in order to increase its robustness to outliers and contaminated data.

We applied our approach to an internal dataset of Microsoft SharePoint event logs within the Microsoft organization and evaluated our approach against anomalies inferred from metadata.

In the future, the framework could be extended to incorporate additional information into the graph, such as the type of interaction. In addition, the sensitivity of the method to the choice of dimension could be investigated and frameworks for evaluation could be explored which would allow for quantitative comparison with other methods.

## References


[1] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: a survey," *Data mining and knowledge discovery*, vol. 29, no. 3, pp. 626–688, 2015.

[2] S. Ranshous, S. Shen, D. Koutra, S. Harenberg, C. Faloutsos, and N. F. Samatova, "Anomaly detection in dynamic networks: a survey," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 7, no. 3, pp. 223–247, 2015.

[3] J. Sun, H. Qu, D. Chakrabarti, and C. Faloutsos, "Neighborhood formation and anomaly detection in bipartite graphs," in *Fifth IEEE International Conference on Data Mining (ICDM'05)*. IEEE, 2005, pp. 8–pp.

[4] L. Akoglu, M. McGlohon, and C. Faloutsos, "Oddball: Spotting anomalies in weighted graphs," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2010, pp. 410–421.

[5] D. Eswaran, C. Faloutsos, S. Guha, and N. Mishra, "Spotlight: Detecting anomalies in streaming graphs," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 1378–1386.

[6] X. Xu, N. Yuruk, Z. Feng, and T. A. Schweiger, "Scan: a structural clustering algorithm for networks," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2007, pp. 824–833.

[7] Y. Chen and B. Malin, "Detection of anomalous insiders in collaborative environments via relational analysis of access logs," in *Proceedings of the first ACM conference on Data and application security and privacy*, 2011, pp. 63–74.

[8] Q. Ding, N. Katenka, P. Barford, E. Kolaczyk, and M. Crovella, "Intrusion as (anti) social communication: characterization and detection," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2012, pp. 886–894.

[9] M. Eslami, G. Zheng, H. Eramian, and G. Levchuk, "Deriving cyber use cases from graph projections of cyber data represented as bipartite graphs," in *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 2017, pp. 4658–4663.

[10] K. Xu, F. Wang, and L. Gu, "Behavior analysis of internet traffic via bipartite graphs and one-mode projections," *IEEE/ACM Transactions on Networking*, vol. 22, no. 3, pp. 931–942, 2013.

[11] M. Turcotte, J. Moore, N. Heard, and A. McPhall, "Poisson factorization for peer-based anomaly detection," in *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*. IEEE, 2016, pp. 208–210.

[12] N. Heard and P. Rubin-Delanchy, "Network-wide anomaly detection via the dirichlet process," in *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*. IEEE, 2016, pp. 220–224.

[13] F. S. Passino and N. A. Heard, "Modelling dynamic network evolution as a pitman-yor process," *Foundations of Data Science*, vol. 1, no. 3, p. 293, 2019.

[14] K. Rohe, S. Chatterjee, and B. Yu, "Spectral clustering and the high-dimensional stochastic blockmodel," *The Annals of Statistics*, vol. 39, no. 4, pp. 1878–1915, 2011.

[15] J. Lei and A. Rinaldo, "Consistency of spectral clustering in stochastic block models," *The Annals of Statistics*, vol. 43, no. 1, pp. 215–237, 2015.

[16] A. Athreya, C. E. Priebe, M. Tang, V. Lyzinski, D. J. Marchette, and D. L. Sussman, "A limit theorem for scaled eigenvectors of random dot product graphs," *Sankhya A*, vol. 78, no. 1, pp. 1–18, 2016.

[17] P. Rubin-Delanchy, J. Cape, M. Tang, and C. E. Priebe, "A statistical interpretation of spectral embedding: the generalised random dot product graph," *arXiv preprint arXiv:1709.05506*, 2017.

[18] M. Tang and C. E. Priebe, "Limit theorems for eigenvectors of the normalized laplacian for random graphs," *The Annals of Statistics*, vol. 46, no. 5, pp. 2360–2415, 2018.

[19] A. Modell and P. Rubin-Delanchy, "Spectral clustering under degree heterogeneity: a case for the random walk laplacian," *arXiv preprint arXiv:2105.00987*, 2021.

[20] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in neural information processing systems*, 2002, pp. 849–856.

[21] T. Qin and K. Rohe, "Regularized spectral clustering under the degree-corrected stochastic blockmodel," *arXiv preprint arXiv:1309.4111*, 2013.

[22] K. Chaudhuri, F. Chung, and A. Tsiatas, "Spectral clustering of graphs with general degrees in the extended planted partition model," in *Conference on Learning Theory*. JMLR Workshop and Conference Proceedings, 2012, pp. 35–1.

[23] A. A. Amini, A. Chen, P. J. Bickel, and E. Levina, "Pseudo-likelihood methods for community detection in large sparse networks," *The Annals of Statistics*, vol. 41, no. 4, pp. 2097–2122, 2013.

[24] A. Joseph and B. Yu, "Impact of regularization on spectral clustering," *arXiv preprint arXiv:1312.1733*, 2013.

[25] C. M. Le, E. Levina, and R. Vershynin, "Concentration and regularization of random graphs," *Random Structures & Algorithms*, vol. 51, no. 3, pp. 538–561, 2017.

[26] Y. Zhang and K. Rohe, "Understanding regularized spectral clustering via graph conductance," *arXiv preprint arXiv:1806.01468*, 2018.

[27] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, 2001, pp. 269–274.

[28] K. Rohe, T. Qin, and B. Yu, "Co-clustering directed graphs to discover asymmetries and directional communities," *Proceedings of the National Academy of Sciences*, vol. 113, no. 45, pp. 12 679–12 684, 2016.

[29] A. Jones and P. Rubin-Delanchy, "The multilayer random dot product graph," *arXiv preprint arXiv:2007.10455*, 2020.

[30] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American society for information science*, vol. 41, no. 6, pp. 391–407, 1990.

[31] O. Levy and Y. Goldberg, "Neural word embedding as implicit matrix factorization," *Advances in neural information processing systems*, vol. 27, pp. 2177–2185, 2014.

[32] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 701–710.

[33] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.

[34] Y. Dong, N. V. Chawla, and A. Swami, "metapath2vec: Scalable representation learning for heterogeneous networks," in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 2017, pp. 135–144.

[35] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 1225–1234.

[36] S. Cao, W. Lu, and Q. Xu, "Deep neural networks for learning graph representations," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[37] A. Saade, F. Krzakala, and L. Zdeborová, "Spectral clustering of graphs with the bethe hessian," *Advances in Neural Information Processing Systems*, vol. 27, pp. 406–414, 2014.

[38] M. Ou, P. Cui, J. Pei, Z. Zhang, and W. Zhu, "Asymmetric transitivity preserving graph embedding," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 1105–1114.

[39] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[40] Y. Liao and V. R. Vemuri, "Use of k-nearest neighbor classifier for intrusion detection," *Computers & security*, vol. 21, no. 5, pp. 439–448, 2002.

[41] J. Tang, Z. Chen, A. W.-C. Fu, and D. W. Cheung, "Enhancing effectiveness of outlier detections for low density patterns," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2002, pp. 535–548.

[42] D. Pokrajac, A. Lazarevic, and L. J. Latecki, "Incremental local outlier detection for data streams," in *2007 IEEE symposium on computational intelligence and data mining*. IEEE, 2007, pp. 504–515.

[43] X. Gu, L. Akoglu, and A. Rinaldo, "Statistical analysis of nearest neighbor methods for anomaly detection," *arXiv preprint arXiv:1907.03813*, 2019.

[44] I. Gallagher, A. Jones, A. Bertiger, C. Priebe, and P. Rubin-Delanchy, "Spectral embedding of weighted graphs," *arXiv preprint arXiv:1910.05534*, 2019.

[45] M. Zhu and A. Ghodsi, "Automatic dimensionality selection from the scree plot via the use of profile likelihood," *Computational Statistics & Data Analysis*, vol. 51, no. 2, pp. 918–930, 2006.

[46] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Reviews of modern physics*, vol. 74, no. 1, p. 47, 2002.

[47] C. M. Le, K. Levin, P. J. Bickel, and E. Levina, "Comment: Ridge regression and regularization of large matrices," *Technometrics*, vol. 62, no. 4, pp. 443–446, 2020.

[48] K. Levin, F. Roosta, M. Mahoney, and C. Priebe, "Out-of-sample extension of graph adjacency spectral embedding," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2975–2984.

[49] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.