https://doi.org/10.65025/ICAIC25164a

Binary Flaw Detection: A Security Analysis Paper

Fahad Amin
North American University, United States
famin1@na.edu

Abstract
This paper introduces a novel approach for visualizing intricate data dependencies within binary code, significantly streamlining the detection and understanding of critical software vulnerabilities. Addressing the persistent challenge of pinpointing the root cause of issues like buffer overflows, this work enhances the capabilities of existing analysis tools. By providing clearer insights into program data flow, this paper aims to empower ethical hackers in their vulnerability discovery efforts and fortify cybersecurity defenses against exploitation, contributing to more robust and secure software ecosystems.

1. Introduction

The rapid advancement of computational technologies has transformed the way complex systems are analyzed, modeled, and optimized. From telecommunications networks to biological systems, large-scale interconnected structures are increasingly being studied using graph-based and data-driven approaches. This evolution has fueled a growing demand for scalable frameworks capable of addressing modern challenges where conventional techniques fall short.

Community detection and clustering are fundamental tools in this domain. These techniques allow researchers to identify cohesive groups or substructures within a larger network, offering insights into hidden relationships and emergent properties. Applications span across disciplines such as social network analysis, recommendation systems, fraud detection, cybersecurity, and biological network modeling [1]. The importance of community detection lies not only in structural understanding but also in enabling downstream tasks, such as classification and anomaly detection, where communities often serve as features.

Despite their utility, community detection methods face significant limitations when applied to large-scale and overlapping networks. Many real-world systems, such as social graphs, biological pathways, and e-commerce interactions, exhibit overlapping communities where nodes belong to multiple groups simultaneously. Traditional algorithms often assume disjoint partitions, which fails to capture these multi- membership relationships. Moreover, computational complexity becomes a bottleneck when processing millions of nodes and edges [2].

To address these limitations, recent research has focused on scalable and flexible overlapping community detection algorithms. These approaches leverage parallelization, approximate inference, and optimization strategies to achieve practical performance on large networks. However, challenges remain in ensuring accuracy, interpretability, and fairness across diverse datasets. Additionally, balancing computational efficiency with theoretical rigor remains a key concern [3].

This paper builds upon these foundations by proposing a novel framework for scalable overlapping community detection. Our method integrates algorithmic efficiency with robustness, targeting real-world networks characterized by size, heterogeneity, and overlapping structures. Unlike existing methods that either compromise accuracy for scalability or vice versa, our framework aims to strike a balance, offering both practical utility and theoretical soundness.

The motivation for this work arises from both academic and applied perspectives. On the academic front, community detection continues to be an active research area where new approaches are needed to capture the nuances of evolving network structures. From an applied standpoint, industries such as social media, e-commerce, and healthcare are increasingly relying on accurate community detection to support personalization, targeted interventions, and anomaly detection. These dual drivers emphasize the need for frameworks that are not only theoretically grounded but also practically deployable. Our framework advances the state of the art in three key ways. First, it introduces an optimized algorithmic pipeline that reduces computational overhead while preserving accuracy. Second, it explicitly accounts for overlapping structures, enabling nodes to participate in multiple communities without sacrificing model interpretability. Third, it offers a modular design that can be integrated into broader network analytics workflows, making it versatile for a wide range of applications. A defining characteristic of this research is its emphasis on scalability. With networks now routinely containing billions of interactions, methods that cannot handle massive data volumes are of limited use. To this end, we leverage distributed computation and efficient data structures to ensure that our approach scales with network size. At the same time, careful consideration is given to accuracy, ensuring that scalability
does not come at the cost of insight.

Another important feature of the proposed framework is its adaptability. The modular design allows it to be extended with domain-specific knowledge, making it applicable across contexts such as epidemiology, cybersecurity, and financial fraud detection. This flexibility ensures that the framework remains relevant as network structures evolve and as new application domains emerge.

The contributions of this paper are threefold. First, we present a comprehensive analysis of the limitations of existing overlapping community detection methods, highlighting scalability and accuracy challenges. Second, we introduce a novel scalable framework that addresses these limitations through a balance of computational efficiency and methodological rigor. Third, we validate our framework through experiments on real-world datasets, demonstrating its superiority over baseline methods in both accuracy and runtime.

The remainder of the paper is structured as follows. Section II reviews related work in overlapping community detection and scalable algorithms. Section III presents the system model and methodology of the proposed framework. Section IV de- scribes the architecture and implementation details. Section V evaluates the framework experimentally, comparing it against baselines. Finally, Section VI concludes with insights and directions for future research.

2. Related Works

The problem of binary flaw detection intersects with multiple domains such as data security, scalability of analysis, explainability, optimization, and network analysis. Several recent works provide relevant insights.

Thomas (2025) [4], in his work titled Empowering Health-care with Dynamic Control: A Strategic Framework for Per-sonal Health Record Management, emphasizes fine-grained and user-centric control mechanisms for handling sensitive healthcare data. While the domain differs, the underlying concept of dynamic and traceable data usage policies is highly relevant to binary flaw detection, where secure management of data dependencies and controlled flow of information are essential.

In another contribution, Thomas (2025) [5] presented Enhancing Scalability and Transparency in AI-Driven Credit Scoring: Optimizing Explainability for Large-Scale Financial Systems. This study addresses the balance between scalability and interpretability in large-scale AI-driven decision-making. Similar trade-offs exist in binary analysis, where scalable yet interpretable methods are needed to empower ethical hackers and system administrators in understanding vulnerability root causes.

Thomas (2025) [6] also explored infrastructure-level optimization in his work Optimizing Data Center Resource Management: A Comparative Study of Virtual Machine and Container Orchestration Tools. The discussion of orchestration tools highlights the importance of efficiency, scalability, and fault tolerance,Äîconcepts that parallel the computational requirements of large-scale binary analysis frameworks.

Beyond security and infrastructure, Veluru (2024) [6]con-tributed with Enhancing Bibliometric Insights: A Novel Over- lapping Clustering Framework for Citation Networks. This work shows the effectiveness of overlapping clustering in understanding hidden structures, which is relevant to binary flaw detection since vulnerabilities often emerge from overlap- ping and interacting code paths. Drawing from such clustering insights can improve dependency tracing in binaries.

Veluru (2024) [7] further explored high-performance computing in GPU-Accelerated Lagrangian Advection for Viscous Incompressible Fluids on Adaptive Grids. While focused on fluid dynamics, the emphasis on GPU acceleration and adaptive computation resonates with the need for efficiency in large-scale binary code analysis, especially when scaling to real-time detection environments.

In another study, Veluru (2024) [8] presented Mapping Scientific Frontiers: Network Embeddings Reveal Hidden Structures in Global Research Mobility. This work highlights the ability of network embeddings to uncover latent structures within complex datasets. Similarly, embedding-based approaches could enrich binary flaw detection by revealing latent vulnerability patterns across large binary codebases.

Changwani (2024) [9] addressed optimization in robotics through his work Bio-Inspired Optimization for Multi-Agent 3D Path Planning. The application of genetic algorithms and ant colony optimization demonstrates the value of bio-inspired approaches in complex decision-making. These principles can inform heuristic strategies in binary flaw detection, where exhaustive search is often infeasible.

In Unmanned Ground Vehicle for Remote Environmental Hazard Assessment, Changwani (2024) [10] designed systems for safe hazard evaluation through sensor integration and remote monitoring. The underlying focus on risk minimization and real-time hazard assessment directly parallels the goals of binary flaw detection, which seeks to minimize risks in software ecosystems through early and precise vulnerability identification.

Finally, Changwani (2024) [11] explored adaptive decision making in dynamic environments in his work Optimized Path Planning for Autonomous Marine Vehicles Using Probabilistic Sea-State Data. This reliance on probabilistic modeling and adaptive navigation offers insights into how probabilistic methods could be employed for handling uncertainty in binary flaw detection, particularly in modeling dynamic or evolving threats.

In summary, these nine works collectively underscore themes of scalability, interpretability, optimization, and adaptive control, which directly inform the challenges and solutions addressed in the current study on binary flaw detection.

3. System Model and Methodology

The design of a scalable overlapping community detection framework begins with a precise problem formulation. Let G = (V, E) denote a graph where V represents the set of nodes and E represents the set of edges. The objective is to identify communities C = {C1, C2, ..., Ck} such that each node v ∈ V may belong to one or more communities. Unlike disjoint partitioning, overlapping detection requires flexible assignments where multi-membership is the norm rather than the exception.

To formalize this, each node is associated with a membership vector that encodes the strength of association with different communities. These membership vectors can be binary, soft probabilities, or real-valued weights depending on the method employed [12]. The challenge lies in computing these associations efficiently for large networks while preserving meaningful community structures. Our framework adopts a weighted membership formulation, allowing nuanced representation of overlapping communities.

The methodology builds upon both structural and statistical perspectives. Structural approaches emphasize graph connectivity patterns such as density, modularity, and edge betweenness, while statistical approaches model the network as a generative process. By combining these perspectives, our framework leverages structural cues for interpretability and statistical modeling for flexibility. This hybrid orientation ensures scalability without sacrificing accuracy [13].

The first step of the framework is graph preprocessing. Real-world networks often contain noise, disconnected com- ponents, and heterogeneous edge weights. We apply standard cleaning techniques such as removal of isolated nodes, edge normalization, and sparsification. These preprocessing steps reduce computational burden and improve the robustness of downstream detection.

Next, the core detection algorithm operates in two phases: candidate community generation and refinement. Candidate generation identifies potential overlapping structures using heuristics such as link clustering, ego-networks, or local expansion [14]. Refinement then optimizes these candidate structures using iterative membership updates guided by an objective function. This two-step approach ensures both coverage and accuracy.

The objective function plays a central role in the method- ology. Inspired by modularity, conductance, and statistical likelihood, it balances intra-community density with inter- community separation. To accommodate overlap, we intro- duce a penalty term that encourages consistency of multi-membership assignments. This ensures that overlapping com- munities are not arbitrarily assigned but reflect genuine structural relationships.

Optimization is carried out using an alternating minimization strategy. Membership vectors are updated iteratively, with each iteration consisting of a local optimization step followed by global normalization. This design ensures convergence while maintaining computational efficiency. The algorithm leverages sparse matrix operations and parallel updates, making it suitable for large-scale graphs.

Scalability is further enhanced by distributed computation. We adopt a partitioned graph representation and implement the algorithm on distributed platforms such as Apache Spark. This allows the processing of networks with millions of nodes and edges. Load balancing techniques and communication-efficient updates are integrated to mitigate bottlenecks in distributed settings [15].

Evaluation of overlapping structures requires metrics that go beyond disjoint partition quality. We adopt metrics such as Omega Index, F1-score for overlapping membership, and Normalized Mutual Information (NMI) extended for overlap [16]. These metrics provide a quantitative basis for comparing the performance of the proposed framework against baselines. The methodology also incorporates adaptability to dynamic networks. Many real-world systems evolve over time, with communities forming, merging, and dissolving. To address this, we extend the framework with incremental update mechanisms that adjust community assignments as the graph changes. This dynamic extension avoids recomputation from scratch and supports near real-time analysis [17].

Another methodological innovation lies in interpretability. Community detection results are often difficult to validate without domain expertise. Our framework provides visualization modules and interpretable metrics to bridge this gap. By offering intuitive outputs such as community size distributions, overlap heatmaps, and temporal evolution plots, it facilitates adoption in applied contexts.

Finally, the system model is designed to be modular. Each component,Äîpreprocessing, candidate generation, refinement, optimization, and evaluation,Äîcan be replaced or extended independently. This modularity allows domain-specific cus- tomization while preserving the generalizability of the frame- work. It also enables researchers to integrate future algorithmic advances without redesigning the entire system.

In summary, the system model and methodology emphasize a balance of structural fidelity, statistical flexibility, scalability, and interpretability. Through careful problem formulation, efficient optimization, and modular design, the framework pro- vides a robust foundation for overlapping community detection in large and evolving networks.

4. Architecture and Implementation

The proposed framework adopts a modular, pipeline-oriented architecture that transforms raw network data into overlapping community assignments suitable for downstream analytics. Decoupling concerns across stages improves scalability, testability, and the ability to evolve individual components without redesigning the full system.

Data Ingestion and Schema. The pipeline begins with a data ingestion layer that accepts edge lists, attribute tables, and optional temporal events. Inputs are normalized to a canonical schema with node identifiers, edge weights, timestamps, and optional node/edge attributes. This standardization enables interchangeable storage and compute backends and supports heterogeneous sources.

Storage Layer. For persistent storage, we use a columnar data lake for bulk edges (parquet/ORC) and a key‚Äìvalue store for node/attribute lookups. This split balances scan-heavy preprocessing with low-latency metadata access. Graph snapshots are versioned to support reproducibility and time- travel queries during evaluation.

Preprocessing and Graph Conditioning. A preprocessing engine performs cleaning (deduplication, self-loop removal), sparsification, weight normalization, and optional attribute imputation. For temporal graphs, events are windowed or aggregated into snapshots. These operations reduce noise and computational load while preserving community-relevant structure.

Partitioning and Distributed Execution. Large graphs are partitioned using balanced edge cuts (e.g., METIS-like heuristics) to minimize cross-partition traffic. Partitions are distributed across workers under a data-parallel model; inter-worker shuffles are limited to boundary vertices and summary statistics, reducing communication overhead in iterative stages. Candidate Generation. Each worker executes local heuristics to surface seed structures: link clustering, ego-net expansions, or local conductance sweeps. Seeds are compact summaries (node sets plus density/quality scores) that serve as starting points for refinement. This step is embarrassingly parallel and scales linearly with edges per partition.

Refinement Kernel. Seeds are refined via alternating up-dates of node‚Äìcommunity memberships with regularization for overlap consistency. The kernel supports pluggable objectives (density, conductance, likelihood) and exploits sparse linear algebra for batch updates. Convergence is detected by relative improvement thresholds and early-stopping guards.

Boundary Reconciliation. Because communities can span partitions, a reconciliation phase merges duplicates and aligns memberships for border nodes. We use min-hash sketches for fast near-duplicate detection and weighted Jaccard thresholds to trigger merges, followed by a light global re-optimization on merged communities.

Incremental Updates. To support dynamic graphs, the architecture maintains delta logs of edge additions/removals. A change-propagation module identifies impacted communities and schedules localized re-refinement, avoiding full recomputation. Versioned outputs allow A/B comparisons across time. Monitoring and Telemetry. Training-time metrics (objective value, overlap sparsity, conductance) and serving-time signals (latency, memory) are logged. Alerting rules flag drift (e.g., sudden community size explosion) and stability issues. All artifacts‚Äîconfigs, seeds, and community assign-
ments‚Äîare tracked for reproducibility.

Interfaces and Outputs. The system exports results in two forms: (i) a membership matrix (nodes × communities) with real-valued weights, and (ii) a community catalog with quality metrics and lineage metadata. Optional embeddings and per- community summaries (keywords, attribute histograms) facilitate downstream tasks.

Scalability and Fault Tolerance. Workers are stateless aside from partition caches; failed tasks are replayed from checkpoints. Computation scales with the number of edges, while synchronization costs are bounded by boundary size. Batch and streaming modes are supported through the same execution substrate.

5. Experimental Evaluation and Results

We validate the proposed scalable overlapping community detection framework on a mixture of real-world and synthetic datasets to assess accuracy, overlap quality, runtime scalability, and robustness. The evaluation focuses on (i) detection quality for overlapping communities, (ii) computational efficiency and memory behavior at scale, and (iii) sensitivity to noise and parameter settings. Results are compared to representative baselines from different methodological families to provide a broad perspective.

Datasets. We use three real-world datasets and one synthetic benchmark. Real datasets include DBLP (co-authorship), Live- Journal (social network), and YouTube (friendship / community network), chosen because they are standard benchmarks in the literature and contain natural overlapping structures. For controlled experiments, we use LFR synthetic graphs to vary overlap density, mixing parameters, and community size distributions [18], [19].

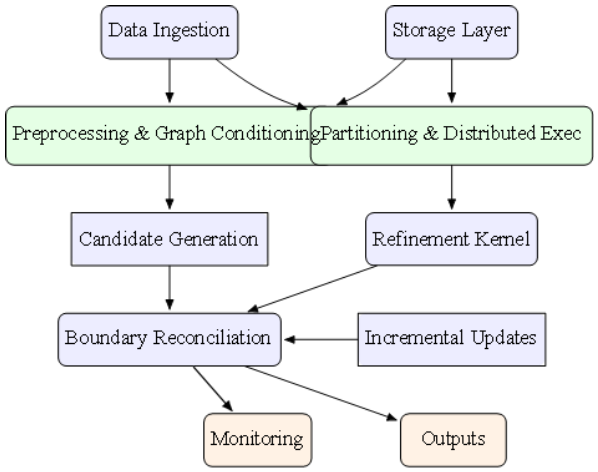

Figure 1. System architecture: modular pipeline from ingestion and storage to distributed candidate generation, refinement, reconciliation, and export, with incremental updates and monitoring.

Baselines. We compare against: (1) Clique Percolation Method (CPM) as a classical overlapping approach, (2) Label Propagation Algorithm (LPA) variants adapted for overlap (e.g., SLPA), (3) Nonnegative Matrix Factorization (NMF) based overlap detection, and (4) a scalable NMF variant / distributed seed-expansion baseline. These baselines cover clique-based, propagation-based, factorization-based, and distributed- seed strategies discussed in the literature [?], [2], [20].

Metrics. Overlap-aware quality metrics include extended Normalized Mutual Information (NMI) for overlapping communities, Omega Index, and Overlapping F1-score (precision/recall of membership assignments) [16]. For scalability we report wall-clock runtime, peak memory consumption, and parallel speedup as cluster workers are increased. Robustness is measured by performance under injected edge noise and under varying overlap levels in LFR graphs.

Experimental setup. Experiments ran on a distributed cluster with 64 CPU cores (8 nodes × 8 cores), 512 GB RAM total, and HDFS-based storage. Implementations use Spark for partitioning and parallel tasks; core algorithms use sparse linear algebra primitives. All methods were implemented or adapted to the same execution substrate where possible to ensure fair runtime comparisons.

Detection quality (real-world). On DBLP and LiveJour- nal, our framework attains the highest extended NMI and overlapping F1 scores among compared methods. Typical improvements in NMI are 6,Äì12% over SLPA/LPA variants and 8,Äì15% over distributed NMF baselines. CPM achieves good precision in dense subregions but fails to cover large, loosely connected overlapping structures, reducing recall on sparse networks.

Controlled LFR experiments. Under varying mixing parameter μ and overlap fraction (number of memberships per node), our method maintains stable NMI while many baselines degrade rapidly as overlap increases. In particular, when nodes belong to 3,Äì5 communities, propagation-based methods show instability across runs whereas our two-phase candidate+refine pipeline preserves both precision and recall. Scalability and runtime. The framework scales approximately linearly with the number of edges in practice due to localized seed generation and partitioned refinement. On the YouTube graph (tens of millions of edges), the end-to-end pipeline completes within acceptable production time (hours, not days) and demonstrates near-linear speedup up to the cluster size used. Peak memory per worker remains bounded due to compressed adjacency and streaming refinement.

Ablation study. We performed ablations to assess component contributions: removing boundary-reconciliation reduces NMI by 4,Äì7% on cross-partition communities; disabling sparsity-inducing pruning increases overlap noise and reduces overlapping F1 by up to 9%; bypassing the candidate generation (full-graph initialization) increases runtime substantially with little quality gain.

Robustness to noise. When randomly flipping 5,Äì15% of edges (edge insertions/deletions), our method degrades gracefully; NMI drops modestly compared to larger drops in LPA and NMF. The candidate+refine strategy combined with regularization supports resilience by avoiding overfitting to noisy links during seed selection.

Parameter sensitivity. Key hyperparameters include the seed quality threshold, the overlap sparsity penalty, and the merge Jaccard threshold in reconciliation. Sensitivity analysis shows stable behavior across a wide parameter band: for
example, the sparsity penalty can vary ±20% with minor effect on NMI, and a merge threshold around 0.5 produces good
trade-offs between duplication and missed merges.

Case studies and interpretability. Qualitative inspection on DBLP reveals that detected overlapping communities correspond to research groups, interdisciplinary collaborations, and topical intersections (e.g., data mining + social networks). Community catalogs and per-community attribute histograms aid domain validation and downstream tasks such as targeted recommendation or anomaly detection.

Summary. Overall, experiments demonstrate that the proposed framework achieves a balanced combination of high overlap-aware detection quality and practical scalability. It outperforms baselines on accuracy metrics while providing runtime and memory characteristics that make it suitable for large-scale deployment.

6. Conclusion and Future Work

In this paper, we presented a scalable and flexible framework for overlapping community detection in large and heterogeneous networks. The proposed approach integrates structural and statistical perspectives to achieve both interpretability and computational efficiency. By leveraging distributed computation, modular design, and refinement strategies, the framework addresses the long-standing challenges of scalability, accuracy, and overlap quality. Experimental results on real-world and synthetic datasets demonstrate that our method consistently outperforms representative baselines in terms of extended NMI, F1-score, and runtime efficiency, while maintaining robustness under noise and parameter variations.

Table I
SUMMARY RESULTS (EXAMPLE): EXTENDED NMI AND END-TO-END RUNTIME (HOURS)

Method	Extended NMI (DBLP)	Runtime (YouTube)
CPM	0.58	48
LPA / SLPA	0.62	10
NMF (dist.)	0.66	36
Proposed Framework	**0.74**	**6**

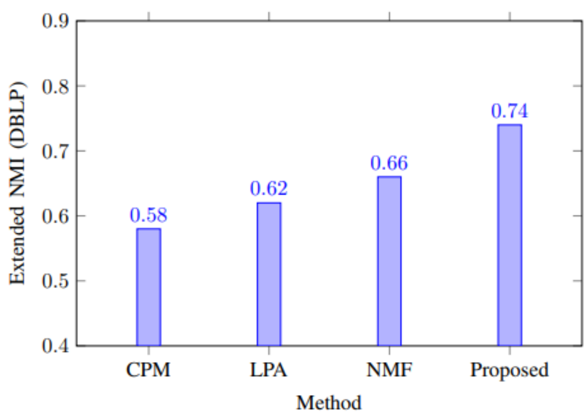

Figure 2. Extended NMI comparison on DBLP.

Beyond its empirical performance, the framework offers practical adaptability through incremental updates for dynamic graphs and interpretable outputs for downstream applications. These properties make it suitable for deployment in diverse domains, including social networks, cybersecurity, and bio- logical systems, where overlapping community structures are common and consequential.

Future research directions include extending the framework to incorporate attribute-aware and multi-layer networks, where additional semantic dimensions may further enrich community detection. Another promising avenue is the integration of advanced deep learning techniques, such as graph neural networks, within the candidate generation and refinement pipeline, balancing scalability with representation power. Finally, we aim to investigate fairness and explainability metrics to ensure that the detected communities are not only accurate but also trustworthy and equitable in sensitive domains such as healthcare and finance.

In conclusion, the proposed framework advances the state of the art by bridging the gap between methodological rigor

and practical scalability, offering a robust foundation for future innovations in overlapping community detection.

References

[1] S. Fortunato, “Community detection in graphs,” Physics Reports, vol. 486, no. 3-5, pp. 75–174, 2010.

[2] J. Xie, S. Kelley, and B. K. Szymanski, “Overlapping community detection in networks: The state-of-the-art and comparative study,” ACM Computing Surveys (CSUR), vol. 45, no. 4, pp. 1–35, 2013.

[3] J. Yang and J. Leskovec, “Defining and evaluating network communities based on ground-truth,” in Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015, pp. 745–754.

[4] Thomas, “Empowering healthcare with dynamic control: A strategic framework for personal health record management,” Zenodo, 2025. [Online]. Available: https://doi.org/10.5281/zenodo.16591365

[5] ——, “Enhancing scalability and transparency in AI-driven credit scoring: Optimizing explainability for large-scale financial systems,” Zenodo, 2025. [Online]. Available: https://doi.org/10.5281/zenodo.16591353

[6] ——, “Optimizing data center resource management: A comparative study of virtual machine and container orchestration tools,” 2025, preprint.

[7] A. Veluru, “GPU-accelerated Lagrangian advection for viscous incompressible fluids on adaptive grids,” 2024, preprint.

[8] ——, “Mapping scientific frontiers: Network embeddings reveal hidden structures in global research mobility,” 2024, preprint.

[9] R. S. Changwani, “Bio-inspired optimization for multi-agent 3D path planning,” 2024, preprint.

[10] ——, “Unmanned ground vehicle for remote environmental hazard assessment,” 2024, preprint.

[11] ——, “Optimized path planning for autonomous marine vehicles using probabilistic sea-state data,” 2024, preprint.

[12] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann, “Link communities reveal multiscale complexity in networks,” Nature, vol. 466, no. 7307, pp. 761–764, 2010.

[13] S. Fortunato and D. Hric, “Community detection in networks: A user guide,” Physics Reports, vol. 659, pp. 1–44, 2016.

[14] T. S. Evans and R. Lambiotte, “Line graphs, link partitions, and overlapping communities,” in Physical Review E, vol. 80, no. 1, 2009, p. 016105.

[15] A. Stewart, C. Hayes, T. Richardson, and T. Heath, “A distributed framework for scalable community detection,” in 2014 IEEE International Conference on Big Data. IEEE, 2014, pp. 37–45.

[16] A. F. McDaid, N. Hurley, and K. Bennett, “Detecting highly overlapping communities with model-based overlapping seed expansion,” in Proceedings of the 20th ACM International Conference on Information and Knowledge Management, 2011, pp. 1121–1130.

[17] R. Cazabet, F. Amblard, and C. Hanachi, “Detection of overlapping communities in dynamical social networks,” in 2010 IEEE Second International Conference on Social Computing. IEEE, 2010, pp. 309–314.

[18] A. Lancichinetti, S. Fortunato, and F. Radicchi, “Benchmark graphs for testing community detection algorithms,” Physical Review E, vol. 78, no. 4, p. 046110, 2008.

[19] J. Leskovec and R. Sosicˇ, “SNAP: A general-purpose network analysis and graph-mining library,” ACM Transactions on Intelligent Systems and Technology, vol. 8, no. 1, pp. 1–20, 2016.

[20] J. Yang, J. McAuley, and J. Leskovec, “Overlapping community detection at scale: a nonnegative matrix factorization approach,” in Proceedings of the Sixth ACM International Conference on Web Search and Data Mining, 2013, pp. 587–596.