Evolution of Anime Co-Viewership Networks (2006–2018)

Iaroslav Sagan (66661)
University of Lisbon
Lisbon, Portugal
yaroslav_sagan@gmail.com

Anna Maksymchuk (66662)
University of Lisbon
Lisbon, Portugal
anna.i.maksymchuk@gmail.com

Maria Samosudova (66663)
University of Lisbon
Lisbon, Portugal
samosudova@gmail.com

ABSTRACT

This study analyzes the structural evolution of the MyAnimeList platform (2006–2018). Utilizing time-sliced networks, we examine the dynamics of taste communities and user navigation. We employ the Leiden algorithm to trace the genealogy of clusters from a cohesive core to a fragmented landscape. Regarding user modeling, we find that stochastic random walker agents diverge from empirical trajectories, indicating the limitations of purely topological simulation. In contrast, machine learning experiments successfully predict user migration patterns. Crucially, we demonstrate that the inclusion of community-based clustering metrics significantly improves model accuracy, highlighting the predictive value of mesoscale network features.

1 INTRODUCTION

1.1 The MyAnimeList Ecosystem as a Complex Network

The digital aggregation of cultural preferences has transformed the study of computational sociology. MyAnimeList (MAL), established in 2006, serves as a primary registry for anime consumption. Unlike generic social networks where edges represent declared friendships, the primary structure of MAL is a bipartite interest graph connecting users to titles. This structure provides an optimal environment for analyzing "affinity networks," where community formation is driven by taste homophily rather than geographic proximity.

The period between 2006 and 2018 represents a critical epoch in the globalization of Japanese animation, characterized by the transition from a niche subculture to a mainstream entertainment force. Analyzing this period allows for the observation of a complete evolutionary cycle of a complex network—from a cohesive core to a fragmented, multi-polar topology.

1.2 Objectives and Research Questions

The primary objective of this study is to model the MAL community as a dynamic, evolving system. Traditional static graph analysis fails to capture the temporal fluidity of user interests. To address this, we employ a multi-stage analytical framework combining dynamic community detection, stochastic simulation, and predictive modeling.

A key focus of our investigation is the evaluation of topological agents. We test the hypothesis that standard Random Walker models, which rely solely on structural connectivity, are sufficient to simulate user navigation in affinity networks. Furthermore, we explore whether the predictive power of machine learning models can be enhanced by incorporating mesoscale network features derived from community detection.

The research is guided by the following key questions:

- **Topological Evolution:** How did the structural properties of the MAL graph change during the community's expansion (2006-2018)?
- **Community Dynamics:** How do taste clusters evolve over time, and can we identify distinct phases of fragmentation using dynamic community detection (Leiden)?
- **Limits of Topological Simulation:** To what extent do empirical user trajectories diverge from theoretical Random Walk models?
- **Predictive Modeling:** Can we predict user migration between communities using supervised learning, and does the inclusion of clustering metrics improve model performance?

2 DATA AND METHODOLOGY

2.1 Dataset Acquisition and Preprocessing

The data originates from a publicly available kaggle datasets aggregated from MAL profiles [1, 3], covering the period from 2006 to 2018. Given the platform's predominantly international user base, the dataset reflects global consumption patterns rather than domestic Japanese trends.

To ensure data quality, we applied a multi-stage filtering pipeline:

- **Bot Detection:** Removal of inactive accounts and profiles exhibiting automated behavior.
- **Percentile-based Truncation:** We excluded users falling into the extreme tails of the activity distribution. This removes users with too few votes (insufficient signal for clustering) and those with implausibly high vote counts, ensuring the analysis focuses on human-scale consumption patterns.

The final processed dataset comprises approximately **85,000 unique users** and **6,500 anime titles**.

2.2 Graph Projection and Topology Construction

We model the system as a bipartite graph which is subsequently projected into two distinct monopartite networks. The detailed construction pipelines are described in Sections **??** and **??**, respectively.

2.2.1 Anime-Anime Network. In this projection, an edge exists if two titles share a common voter. To account for varying audience sizes, we utilized the **Jaccard Similarity** index as the edge weight.

$$J(A, B) = \frac{|U_A \cap U_B|}{|U_A \cup U_B|}$$

Given the extreme density of the raw projection (where a single popular anime could fully connect thousands of users), we applied a hard threshold of $J > 0.05$. This effectively prunes weak links formed by random coincidences while preserving significant genre or fandom connections.

2.2.2 User-User Network. Here, an edge connects two users if they have rated the same anime. The edge weight is defined as the raw count of shared titles (co-votes). A major challenge in this projection is the variance in edge weights, which range from negligible values (2-3 shared items) to tens of thousands (10^4). To address this, we implemented a cutoff threshold: edges were retained only if users shared more than **3 titles**.

Anyway, even after thresholding, the user-user raw network projection suffered from extreme density saturation. Popular "blockbuster" titles (e.g., *Death Note*, *Attack On Titan*) act as super-hubs; a single vote for such a title effectively connects a user to thousands of others, creating a near-clique structure that obscures genuine taste communities. So, this titles had to be deleted from the dataset prior to projection.

2.2.3 Further Sparsification Attempts. More aggressive sparsification techniques (e.g., Backbone extraction, k-NN) were tested but did not reveal significantly distinct structural patterns. Consequently, we retained the simpler approach to avoid unnecessary information loss while maintaining structural clarity.

2.3 Resulting Topology

These thresholding strategies proved effective in mitigating the "hairball" phenomenon common in social graphs. The resulting networks exhibited a graph density in the range of $0.2 - 0.3$, striking a balance between sparsity (for efficient clustering) and connectivity (preserving the Giant Connected Component).

Only after these topological corrections is the graph subjected to the Leiden community detection algorithm and Random Walk simulations.

2.4 Computational Framework and Reproducibility

To ensure reproducibility and handle large-scale temporal networks efficiently, we developed a dedicated modular Python framework `project_cda` [2], which is a core part of the open-source repository MARS_1.0. The framework is designed with a modular architecture to support the full research lifecycle:

Graph Construction Engine. The `AnimeGraphBuilder`, `UserGraphBuilder` modules utilize streaming JSON parsers (‘ijson‘) to process massive user interaction logs with minimal memory footprint. They implement the projection logic described in Section 3.1 and support vectorized graph operations via the ‘igraph‘ C-core, ensuring high performance even for dense snapshots.

Simulation and Analysis Modules. The framework includes specialized components for dynamic analysis:

- **CommunityTracker:** Implements a greedy matching algorithm based on Jaccard similarity to trace cluster lineage across time steps.
- **RandomCrowd:** An agent-based simulation engine that deploys stochastic walkers to probe network topology and user navigation patterns.
- **ClusterEvaluation:** Encapsulates the calculation of structural (Modularity), semantic (Purity), and information-theoretic (Entropy) metrics.

Data Management and Caching. Given the computational cost of generating 13 annual snapshots with varying hyperparameters, we implemented a robust caching system managed by a `PathManager`. This module enforces idempotency: each experimental configuration (edge weights, sparsification, clustering algorithm) generates a unique hash signature. If a serialized artifact exists for a given configuration, it is loaded instantly ("lazy evaluation"), preventing redundant computations. The exact data schema required to run the pipeline is documented in the repository's `README.md`.

3 TOPOLOGICAL EVOLUTION OF PROJECTED NETWORKS

In this section[1], we analyze the structural evolution of the projections derived from the bipartite graph. We first examine the Anime-Anime network to understand how content relationships shifted over time, followed by an analysis of the User-User network.

3.1 Anime-Anime Network

The projected anime-anime network experienced explosive growth over the analyzed period (2006–2018), transforming from a compact, niche community into a sprawling, heterogeneous ecosystem. This transformation is defined by three primary phenomena: the densification-sparsification paradox, increasing taste divergence, and the crystallization of a "rich-club" core.

3.1.1 The Densification-Sparsification Paradox. The network underwent a dramatic scale expansion: the number of nodes (anime titles) increased from 732 in 2006 to 6,129 in 2018, while the volume of connections (edges) surged from ∼64,000 to ∼819,000. However, this volumetric growth reveals a fundamental structural shift.

While the absolute number of connections increased by an order of magnitude, the potential number of connections grew quadratically (N^2). Consequently, the global Graph Density declined precipitously from 0.2387 (2006) to 0.0436 (2018).

This indicates a transition from a "Village" topology—where the community is small enough for high interconnectedness—to a "Metropolis" structure. In the modern era, the ecosystem has become highly specialized; while the total volume of interactions is higher, individual anime titles connect to a significantly smaller fraction of the total population. The network has shifted from a monolithic block to a spread-out, sparse landscape.

3.1.2 Increasing Social Distance and Taste Divergence. To quantify the "cost" of traversing this expanding network, we analyzed weighted path metrics. Since edge weights represent similarity (Jaccard), the weighted distance can be interpreted as "social distance" or taste divergence.

The evolution of these metrics is presented in **Figure 1**. As shown in the *upper-left panel*, the average weighted path length rose sharply from 7.1 in 2006 to 44.4 in 2018. This metric represents the "resistance" to navigation: connecting a fan of a niche genre to a mainstream hit now requires passing through significantly more intermediaries.

Simultaneously, the network diameter (*upper-right panel*) expanded from 29 to 188.5 weighted units. This confirms that the

[1]The complete experimental pipeline and visualization tools are available in the project repository: `project_cda/1_general_graph_metrics.ipynb`.

"taste universe" is expanding. Distinct clusters (e.g., modern idols vs. vintage mecha) are moving mathematically further apart, creating deep topological fissures.

3.1.3 Local Cohesion and the "Fandom" Effect. Despite the global sparsification, the network maintains robust local connectivity. The *bottom-left panel* of **Figure 1** illustrates the Average Clustering Coefficient. After an initial adjustment, the metric stabilized at a remarkably high value of ≈0.59. This indicates that the "Small-World" property is preserved locally. If Anime A is connected to B and C, there is a consistent ∼60% probability that B and C are also connected. This proves that the sparsification did not destroy community cohesion; instead, the landscape fractured into tight, self-reinforcing "genre bubbles" (fandoms).

3.1.4 Structural Phase Shift: The 2006 Anomaly. The year 2006 represents a distinct topological phase. In this nascent period, the network exhibited disassortative mixing (Degree Assortativity ≈ −0.11), suggesting a star-like structure where popular titles served as hubs connecting primarily to niche nodes. From 2007 onward, the network flipped to positive assortativity (≈ 0.50), signaling the emergence of the "Rich-Club" phenomenon.

3.1.5 Intensity vs. Topology. Finally, we examine the distribution of influence using Node Strength. The log-log plot in the *bottom-right panel* of **Figure 1** confirms the scale-free nature of the modern network, following a clear Power Law distribution. The network is dominated by a few "mega-hubs," validating the "preferential attachment" growth model.

However, the Assortativity of Strength (≈ 0.18) is consistently lower than the Assortativity of Degree (≈ 0.50). This implies that while popular shows are structurally connected, the strongest taste affinities are located in the niche clusters, not the mainstream core.

3.2 User-User Network

In stark contrast to the Anime content network—which became "sparse" and harder to traverse as it grew—the User interaction network exhibits the classic properties of Network Densification. As the community expanded, the social distance between users collapsed, making the network significantly more interconnected.

While the "universe" of users grew, the social structure did not fragment into isolated islands. Instead, it evolved into a tight, integrated "global village," where new users actively connected to existing hubs rather than the periphery.

3.2.1 Global Integration and the "Shrinking World". The analysis of weighted path metrics reveals a community that is becoming functionally smaller and easier to traverse, despite growing in physical size.

The evolution of these metrics is presented in the *upper panels* of **Figure 2**. As shown in the *upper-left panel*, the average weighted path length dropped sharply from ∼0.45 in 2006 to ∼0.31 by 2009, maintaining this lower baseline through 2018. This reduction is a hallmark of the "Small World" effect: as the platform matured, users formed bridging connections, accelerating the flow of information across the graph.

Similarly, the network diameter (*upper-right panel*) contracted from ∼1.08 to < 0.99. Unlike the Anime graph, where taste divergence created massive gaps, the social graph's diameter is shrinking. This indicates that even socially distant groups (e.g., distinct language communities) are becoming more connected to the mainstream core.

3.2.2 Local Cohesion and Community Structure. While the network became globally smaller, the local structure evolved to balance rapid growth with intimate social circles. The *bottom-left panel* of **Figure 2** shows the Average Clustering Coefficient. It peaked in 2009 (∼0.73) during the platform's initial boom, followed by a gentle decline to ∼0.66.

A score of 0.66 remains exceptionally high for a large social network. The slight decline suggests a natural dilution of cliques as users added diverse friends, but the high retention proves the community is fundamentally built on strong, overlapping friend groups rather than loose acquaintances.

3.2.3 Influence and Inequality. The distribution of influence confirms a highly stratified social hierarchy. The log-log plot in the *bottom-right panel* demonstrates a strict linear descent, characteristic of a Scale-Free Network ($P(k) \sim k^{-\gamma}$).

The graph is dominated by a tiny fraction of "Super-Users" (hubs) who possess nearly $1,000\times$ the connectivity of the average user. These hubs act as the structural "glue" that holds the giant component together, enabling the short path lengths observed in Section 3.2.

4 MESOSCALE ANALYSIS AND COMMUNITY DETECTION EXPERIMENTS

Having characterized[2] the global topological evolution, we shift our focus to the mesoscale level—identifying the distinct taste communities that constitute the anime ecosystem. This section details the experimental framework used to select the optimal clustering algorithm, ensuring that the detected communities are structurally robust, semantically meaningful, and suitable for the subsequent migration analysis.

4.1 Experimental Setup

For each annual snapshot G_t of the projected Anime-Anime network (2006–2018), we conducted a comparative analysis of five primary community detection algorithms:

1. **Leiden (Modularity):** Tested with resolution parameters $\gamma \in [0.5, 2.0]$.
2. **Infomap (Map Equation):** Tested with Markov time parameters $T \in \{10, 25, 50\}$.
3. **Leading Eigenvector:** A spectral method based on modularity maximization.
4. **Label Propagation (LPA):** Used as a baseline heuristic.

The quality of partitions was evaluated using a composite set of metrics, prioritizing **Modularity (Q)** for structural definition, **Genre Purity** for semantic alignment, and **Number of Clusters (N_{cl})** for interpretability.

[2]The complete experimental pipeline and visualization tools are available in the project repository: `project_cda/2_clusterization.ipynb`.

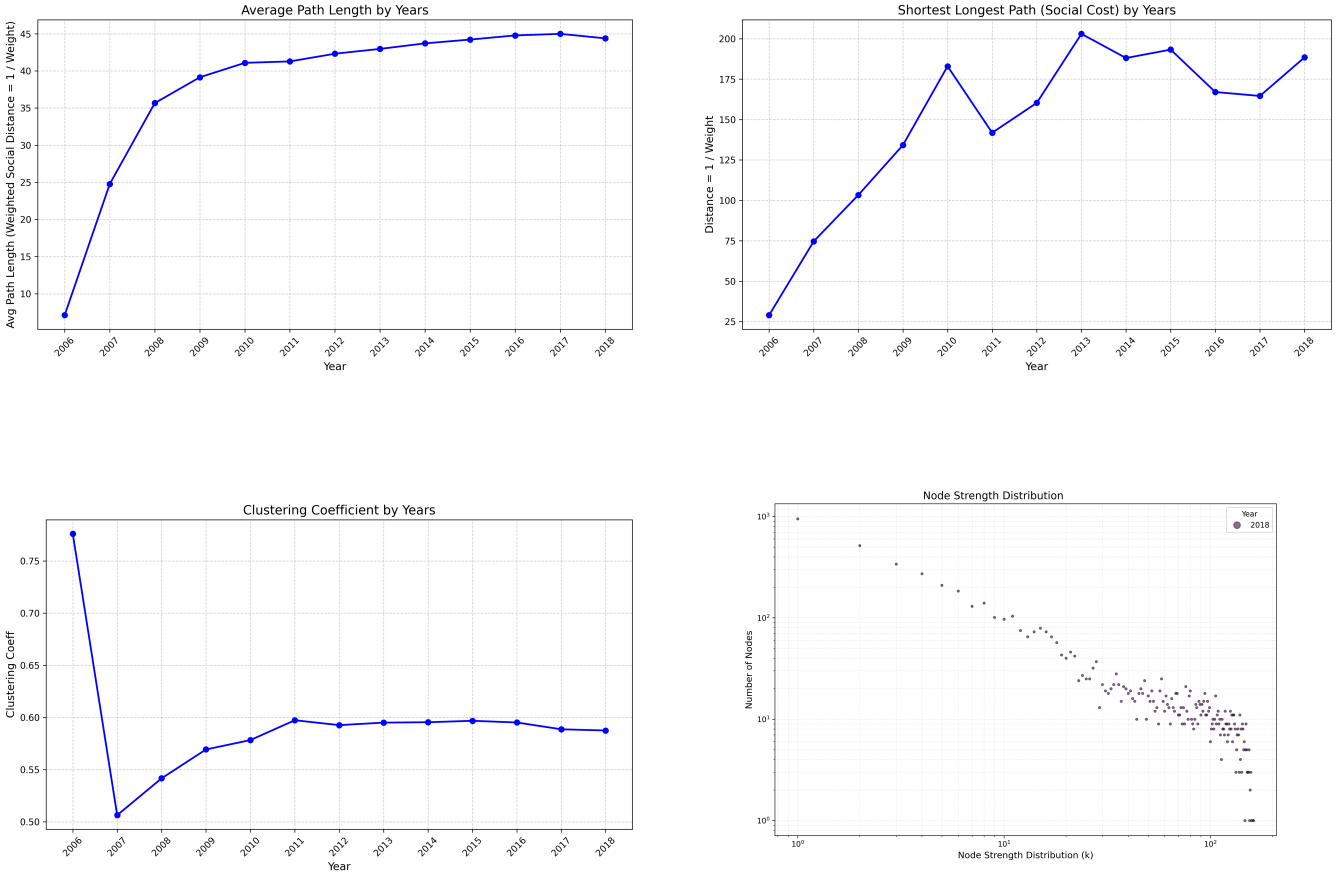


Figure 1: Evolution of Anime Network Topology (2006–2018). Upper-left: Average Weighted Path Length showing increased navigation difficulty. Upper-right: Network Diameter indicating the expansion of the "taste universe". Bottom-left: Average Clustering Coefficient stabilizing around 0.59, suggesting persistent local cohesion. Bottom-right: Node Strength Distribution (2018) confirming the scale-free ($P(k) \sim k^{-\gamma}$) nature of the modern network.

4.2 Comparative Analysis: The Structure-Granularity Trade-off

The experiments revealed a clear trade-off between semantic precision and structural compactness (see Table 1).

Label Propagation (LPA) proved to be an outlier with suboptimal performance. It demonstrated the lowest Modularity ($Q \approx 0.03$), failing to detect the dense community structure known to exist in the network.

Leading Eigenvector showed extreme instability. While its average modularity (≈ 0.30) was acceptable, the number of clusters fluctuated wildly across years (ranging from 4 to 108), indicating high sensitivity to minor topological changes. Such volatility makes it unsuitable for longitudinal tracking.

Infomap exhibited behavior highly dependent on the Markov time parameter T:

- At $T = 10$ ("short walks"), it resulted in **hyper-fragmentation**, producing over 200 micro-communities with high semantic purity (~ 0.63) but poor modular structure ($Q \approx 0.19$).
- At $T = 50$, it achieved the highest semantic accuracy overall (Genre Purity ~ 0.61) while maintaining decent modularity ($Q \approx 0.35$). However, it still divided the network into ≈ 29 clusters, which is too granular for macro-level migration analysis.

Leiden (Modularity) demonstrated superior structural definition. It consistently achieved the highest Modularity scores ($Q \approx 0.36$). Crucially, at $\gamma = 1.0$, it struck a "Golden Mean":

- **High Modularity:** 0.358 (comparable to the best results).
- **Acceptable Purity:** 0.57 (close to Infomap's 0.61).
- **Optimal Compactness:** It identified $\approx$ 10 stable macro-communities.

4.3 Selection Logic

Based on these results, we prioritized structural interpretability over maximal semantic purity. For the task of predicting user migration, it is more valuable to track transitions between a manageable number of distinct "Macro-Communities" (e.g., "Mainstream Action"

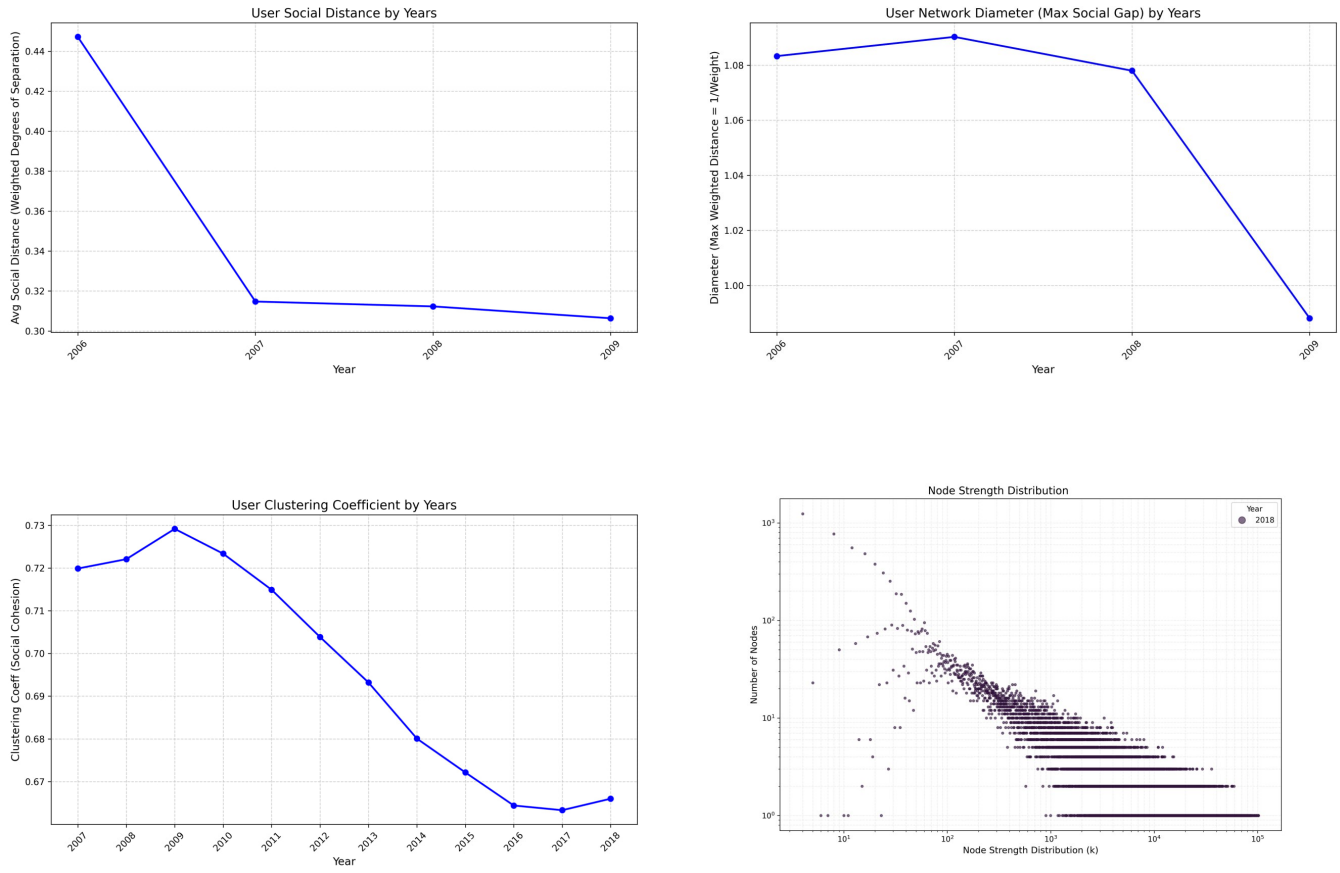


Figure 2: Evolution of User Network Topology (2006–2018). Upper-left: Avg. Weighted Path Length decreasing, indicating higher integration. Upper-right: Network Diameter contracting, showing the "shrinking world" phenomenon. Bottom-left: Clustering Coefficient peaking in 2009 and slowly stabilizing, reflecting the balance between clique formation and expansion. Bottom-right: Node Strength Distribution (2018) following a strict Power Law, highlighting the dominance of "Super-Users".

Table 1: Average Performance of Clustering Algorithms (2006–2018)

Algorithm	Modularity	Genre Purity	Source Purity	N Clusters	Verdict
Label Propagation	0.033	0.31	0.35	6	Underfitting
Leading Eigenvector	0.301	0.50	0.45	34	**Unstable**
Infomap ($T = 10$)	0.192	**0.63**	**0.54**	~ 150	Hyper-segmentation
Infomap ($T = 50$)	0.345	0.61	**0.54**	29	Over-segmentation
Leiden ($\gamma = 0.9$)	0.357	0.00*	0.00*	6	Semantic Mismatch
Leiden ($\gamma = 1.0$)	**0.358**	0.57	0.53	**10**	**Selected**

* Values of 0.00 indicate data corruption for specific years due to tag misalignment during evaluation.

vs. "Vintage Mecha") rather than tracking noise across 100+ fragmented micro-clusters (as seen in Infomap T=10 or Eigenvector). Therefore, we selected **Leiden ($\gamma = 1.0$)** as the primary algorithm.

4.4 Resolution Parameter Sweep

Having selected the Leiden algorithm, we performed a detailed sensitivity analysis to confirm the stability of the chosen resolution $\gamma = 1.0$. We constructed heatmaps for Modularity and Cluster Counts across all years for $\gamma \in [0.1, 3.0]$.

As illustrated in **Figure 3**, the combined analysis confirms our choice:

- **Modularity Landscape:** We observe high modularity values persisting across the range $\gamma \in [0.8, 1.2]$, suggesting that the

strong community structure is not an artifact of a specific parameter point.

- **Cluster Count Plateau:** Simultaneously, the number of clusters stabilizes in this range. Unlike higher resolutions ($\gamma > 1.5$), where the network splinters, $\gamma = 1.0$ consistently yields ~ 10 distinct communities.

This topological stability provides a solid foundation for the longitudinal tracking of user migration.

5 PREDICTIVE MODELING OF USER MIGRATION

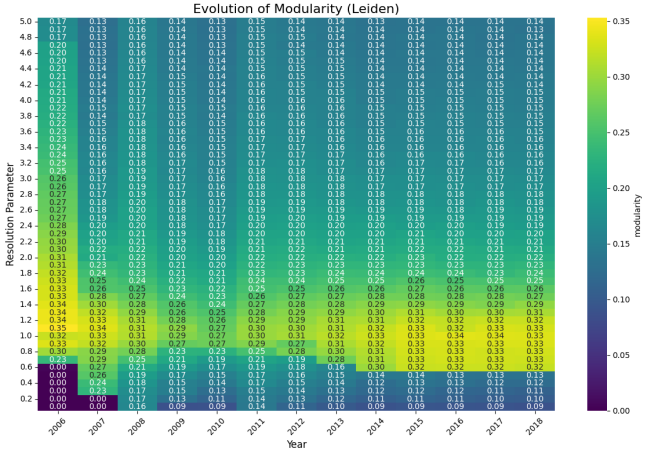
In this chapter[3], we address the problem of user migration between communities, a fundamental challenge in dynamic social network analysis. We model the MAL user network as an evolving graph $G_t = (V_t, E_t)$, where nodes represent unique users and weighted edges denote social proximity based on shared anime consumption history. Since the statistical and topological properties of the network fluctuate significantly over time, our analysis focuses on transitions between consecutive temporal snapshots, denoted as t and $t + 1$.

The migration prediction task is formalized as a binary classification problem. Let $C(u, t)$ denote the community assignment of user u at time t. A user is labeled as a *migrant* if their community membership changes between snapshots:

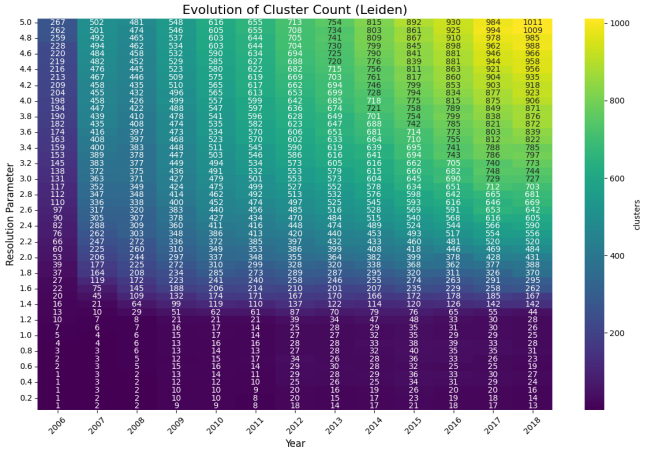
$$y_u = \begin{cases} 1 & \text{if } C(u, t+1) \neq C(u, t) \\ 0 & \text{if } C(u, t+1) = C(u, t) \end{cases} \quad (1)$$

Features extracted at time t are utilized to predict the state y_u at time $t + 1$.

[3]The complete experimental pipeline and visualization tools are available in the project repository: `project_cda/3_user_migration_pipeline.ipynb`.



(a) Evolution of Modularity (Q).



(b) Evolution of Cluster Count (N).

Figure 3: Multi-resolution analysis of the Leiden algorithm (2006–2018). (a) Modularity remains robust across the optimal range. (b) The number of clusters exhibits a stable "plateau" around $\gamma = 1.0$ (~ 10 communities).

5.1 Motivation for Community Detection Algorithm Choice

To identify distinct user communities within each temporal snapshot, we employ the **Leiden algorithm**. While sharing the objective of modularity maximization with the classical Louvain method, Leiden is selected for its superior stability, scalability, and topological guarantees on large-scale networks.

The primary advantage of Leiden over Louvain lies in its handling of community connectivity, scalability and reliability on large networks. While Louvain is efficient, it can produce poorly connected or even disconnected communities, Leiden addresses this through a distinct *refinement phase*, guaranteeing that all resulting clusters are well-connected. This is critical for our study, as membership in a poorly connected cluster could represent an algorithmic artifact rather than genuine social cohesion.

Alternative community detection methods were evaluated but rejected due to specific limitations:

- **Edge-betweenness (e.g., Girvan–Newman):** Computationally prohibitive with a complexity of $O(|V||E|^2)$, rendering it impractical for our network ($|V| \approx 50k$, $|E| \approx 20M$).
- **Infomap:** While efficient, it is sensitive to edge weight distributions and tends to generate highly fragmented micro-communities in dense graphs.
- **Label Propagation:** Although extremely fast, its non-deterministic nature leads to unstable partitions, making longitudinal tracking unreliable.

Leiden combines efficiency, stability, and high-quality partitions. Its near-linear scalability allows us to handle the network effectively, while the refinement phase ensures structurally meaningful communities.

Leiden is a hierarchical algorithm, producing a dendrogram of partitions controlled by the resolution parameter γ.

- Lower values ($\gamma < 1$) favor larger, macro-communities (e.g., broad genre preferences).
- Higher values ($\gamma > 1$) reveal finer micro-communities (e.g., specific niche groups).

In this study, we fixed $\gamma = 1$. This choice provides a balanced granularity, capturing meaningful community structures without

merging distinct groups or isolating noise-driven micro-clusters. It also establishes a robust baseline for migration analysis.

To bridge the gap between topology and semantics, we constructed a correlation matrix between the detected user communities and anime genre clusters based on historical rating data. The resulting heatmap, complemented by alluvial flow diagrams, reveals which anime categories dominate within each user group. This approach allows us to uncover community-specific preference profiles and explore the content-driven factors underlying user migration.

5.2 Feature Engineering

To capture the factors driving migration, we constructed a multidimensional feature space describing the user's state at time t. The feature vector $X_u^{(t)}$ comprises four distinct categories:

(1) **Graph Structure (Global & Local):** Measures of centrality and influence, including Degree, Weighted Strength, PageRank, and K-core decomposition.
(2) **Local Cohesion:** This is captured by the weighted clustering coefficient; low values indicate weak neighborhood integration and a potentially higher migration risk.
(3) **Community Embeddedness:** Metrics quantifying the boundary position of a user. Key among these is the *Intra-Community Ratio (ICR)*, defined as the fraction of a user's edges that connect to nodes within the same community.
(4) **Temporal Dynamics:** Delta features representing the year-over-year change in structural metrics (e.g., ΔDegree, ΔICR), capturing the trajectory of a user's engagement.
(5) **Demographics & Activity:** Static attributes (gender, age, location) integrated with dynamic indicators, such as the number of watched titles and rating fluctuations over time.

This results in a comprehensive per-user, per-year feature vector integrating static, structural, and dynamic signals for predictive modeling.

5.3 Model Specification and Performance Assessment

We utilized **CatBoost**, a gradient boosting algorithm on decision trees, for the classification task. CatBoost was selected for its native handling of categorical features (e.g., community ID, location) without one-hot encoding, robustness to missing values and different feature scales, and ability to model complex non-linear interactions between structural and behavioral signals. Crucially, it offers native support for class imbalance via internal weighting, allowing the algorithm to effectively handle the rarity of migration events.

Considering the task complexity and strong class imbalance (migration events constitute ≈ 17% of the data), the trained model demonstrates strong predictive performance. It achieves a **ROC AUC of 0.915**, indicating strong discrimination capability.

Table 2: Classification Performance Metrics

Class	Precision	Recall	F1-Score
Non-Migrant (0)	0.93	0.92	0.93
Migrant (1)	0.64	0.68	0.66
Overall Accuracy		0.88	

These results (Table 2) confirm that user migration is not a stochastic process but a predictable phenomenon driven by observable network and behavioral characteristics. This validates the model's utility for early-warning detection, offering threshold adjustments to balance recall and precision.

5.4 Feature Importance and Behavioral Interpretation

To explain the model's decisions, we analyzed SHAP (SHapley Additive exPlanations) summary plot. Figure 4 illustrates the global feature importance and directional impact on the prediction.

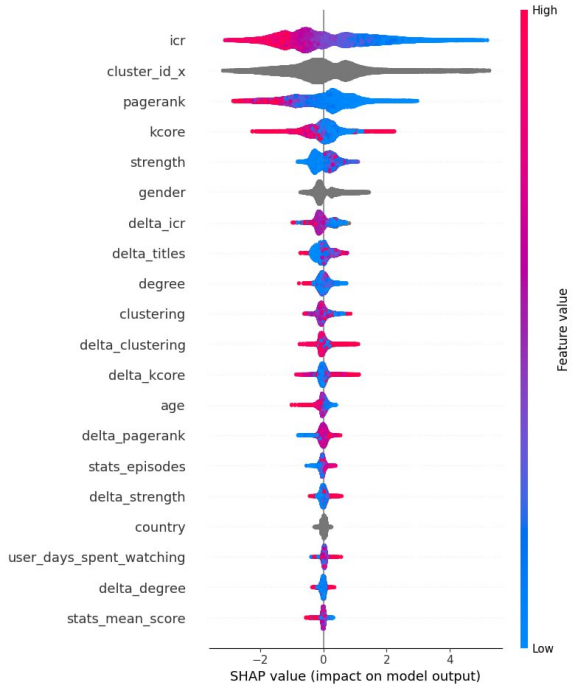


Figure 4: Distribution of SHAP values indicating the contribution of each feature to the model's predictions.

The analysis reveals that **Intra-Community Ratio (ICR)** is the dominant predictor. A high ICR acts as a stabilizing force, anchoring the user within their group, whereas a low ICR serves as a strong precursor to migration. Notably, the *temporal decrease* in ICR is the most informative early-warning dynamic signal.

The **current community assignment** itself also has a strong effect, indicating that baseline migration risks differ significantly across communities (i.e., some communities are inherently more unstable or "leaky" than others).

PageRank exhibits a non-linear stabilizing effect: highly influential users are less likely to migrate, suggesting that social capital creates "inertia." Demographically, older users show lower migration propensities.

These results emphasize that user migration cannot be captured by singular metrics. Only a comprehensive analysis combining network topology, temporal changes, and behavioral features reveals the true mechanisms driving community shifts.

6 RANDOM-WALK MODEL FOR USER TRAJECTORY SIMULATION

6.1 Motivation

User activity on a large interaction graph can be interpreted as a sequence of transitions between nodes (e.g. items, topics, or communities). Given such a sequence for each user, our goal is to construct a probabilistic model that captures the *structural tendencies* of user navigation. This model is later used to generate synthetic trajectories—"random walkers"—that approximate the observed behavior of the real user. The ensemble of walkers provides a natural way to measure how typical or atypical a given user trajectory is, relative to the structure of the graph.

Because the underlying graph is large (on the order of thousands of nodes and millions of edges), all computations must be local and efficient. We exploit the fact that the graph evolves year by year, so a user trajectory is implicitly aligned with a sequence of yearly graphs.

6.2 Definition of the Random Walker

Let $G_t = (V_t, E_t)$ denote the interaction graph in year t. For a user u we observe a trajectory

$$\mathbf{x}^{(u)} = (x_0, x_1, \ldots, x_T),$$

where $x_t \in V_t$ is the node visited by the user in year t. We construct a *random walker* whose behavior in year t is governed only by the local structure of G_t and the user's starting point x_0.

Formally, for each year t the walker occupies a state $X_t \in V_t$. Conditioned on $X_t = v$, the walker chooses its position in year $t+1$ according to a probability distribution over the neighbors of v in G_t:

$$\mathbb{P}(X_{t+1} = w \mid X_t = v) = \frac{1}{\deg_{G_t}(v)} \quad \text{for all } (v, w) \in E_t.$$

That is, the walker performs a uniformly random step along the edges that exist in the corresponding yearly graph.

A single random walker generates one synthetic trajectory

$$\mathbf{Y} = (Y_0, Y_1, \ldots, Y_T), \qquad Y_0 = x_0.$$

To model uncertainty and to obtain stable statistical estimates, we simulate an ensemble of K independent walkers for each user.

6.3 Asynchrony and Year-Level Dynamics

A key detail is that the walkers evolve *asynchronously*. Each walker only moves when the global simulation clock advances to a year in which that walker still has remaining steps. This design is necessary because real user trajectories can have different lengths, and the yearly graphs G_t may differ substantially in size and connectivity.

Thus the simulation proceeds by iterating over years $t = 0, 1, \ldots, T$ and, for each walker whose trajectory length is greater than t, performing exactly one step in G_t. Walkers whose length is shorter than the current year simply remain inactive.

6.4 Ensemble-Based Evaluation

Given a user u with observed trajectory $\mathbf{x}^{(u)}$ and an ensemble of simulated trajectories $\{\mathbf{Y}^{(k)}\}_{k=1}^{K}$, we can quantify how well the random-walk model explains the user's behavior.

Let $d(\cdot, \cdot)$ be a similarity or distance measure between two trajectories. In this work we primarily use a weighted node-overlap metric that penalizes long-distance mismatches. The average similarity of the user to the ensemble is

$$\bar{s}^{(u)} = \frac{1}{K} \sum_{k=1}^{K} s\big(\mathbf{x}^{(u)}, \mathbf{Y}^{(k)}\big),$$

with an accompanying variance

$$\mathrm{Var}^{(u)} = \frac{1}{K} \sum_{k=1}^{K} \Big(s\big(\mathbf{x}^{(u)}, \mathbf{Y}^{(k)}\big) - \bar{s}^{(u)} \Big)^2.$$

These metrics estimate how "typical" the user is relative to repeated realizations of the random-walk model.

Such quantities naturally extend to population-level statistics: distributions of similarities, identification of outliers, and hypothesis testing against the null model provided by random walks.

6.5 Consensus Models

To summarize the global behavior of the entire ensemble, we consider two forms of consensus:

Markov Consensus. All walker trajectories across all users define empirical transition counts

$$C_{vw} = \#\{\text{times a walker moves } v \to w\}.$$

Normalizing the rows yields an empirical transition matrix

$$P_{vw} = \frac{C_{vw}}{\sum_{w'} C_{vw'}}.$$

The matrix P defines a global Markov model that captures the average transition tendencies dictated by the graph structure and the distribution of starting points. This model can be used to compute likelihoods of real user trajectories, to generate new synthetic walkers, or to build a deterministic "most probable" consensus path by greedy selection.

Medoid Trajectory. As a complementary summary, we compute the *medoid* of a set of walker trajectories—the trajectory that minimizes the total distance to all others:

$$k^* = \arg\min_{k} \sum_{j=1}^{K} d\big(\mathbf{Y}^{(k)}, \mathbf{Y}^{(j)}\big).$$

The medoid offers an interpretable representative path that arises from an actual random walker, as opposed to the probabilistic object represented by P.

6.6 Interpretation

The random-walk construction provides an explicit null model driven solely by the graph's local connectivity. If a real user's trajectory significantly deviates from the ensemble predicted by the graph, the deviation may reveal hidden structure, atypical behavior, or external influences that are not captured by topology alone.

Conversely, if the random-walk ensemble closely matches the user, the graph alone is sufficient to explain the observed behavior.

This duality—graph-driven randomness versus user-specific structure—is the central object of analysis in the subsequent sections of the report.

7 MARS_1.0 PROJECT TREE

```
MARS_1.0/
 data/
    anime_ranks/
       anime.csv
       animelist.csv
       anime_with_synopsis.csv
       rating_complete.csv
       watching_status.csv
    anime_timestamps/
       anime_timestamps.csv
    cities_population_and_location/
       cities_population_and_location.csv
    myanimelist_countries_distribution/
       myanimelist_countries_distribution.csv
    users/
        profiles.csv
 db_tools/
    ...
    ...
 fds_tools/
    __init__.py
    data_cleaner.py
    fake_user_generator.py (class FakeUsersGenerator)
    fds_main.py
    project_latex/
        main.tex
        chapters/
           introduction.tex
           overview.tex
           fake_user_generation.tex
           dataset_curriculum.tex
          ...
           references.bib
        out/
            main.pdf
            ...
 cda_tools/
    ...
    ...
 __init__.py
 .env
 .gitignore
 common_tools.py (classes CommonTools, PandasTools, DBTools)
 datasets.json
 LICENSE
 README.md
 requirements.txt
```

REFERENCES

[1] Azathoth. Anime recommendation database 2020. https://www.kaggle.com/datasets/azathoth42/myanimelist, 2018.
[2] Iaroslav Sagan. Mars_1.0: Manga and anime research software. https://github.com/BlackSabbitch/MARS_1.0/tree/main/project_cda, 2025. Accessed: 2025-10-27.
[3] Hernan Valdivieso. Anime recommendation database 2020. https://www.kaggle.com/datasets/hernan4444/anime-recommendation-database-2020, 2020.