AIMS Mathematics

AIMS Mathematics, 10(2): 2042–2063.
DOI: 10.3934/math.2025096
Received: 12 December 2024
Revised: 13 January 2025
Accepted: 23 January 2025
Published: 08 February 2025

https://www.aimspress.com/journal/Math

*Research article*

# Integrating artificial intelligence with network evolution theory for community behavior prediction in dynamic complex systems

**Yongyan Zhao* and Jian Li**

School of Humanities and Law, Harbin University, Harbin 150086, Heilongjiang, China

* **Correspondence:** Email: zhaoyongyan@hrbu.edu.cn; Tel: 86+15776761752.

**Abstract:** Communication networks, such as social and collaborative networks, are characterized by a highly dynamic, constantly changing environment. This makes the analysis of such networks, such as the formation of communities, challenging. The adaptive temporal graph neural network (AT-GNN) was introduced here to overcome these challenges by incorporating temporal segmentation, feature extraction, and attention mechanisms. Based on two large-scale datasets, the Stanford Network Analysis Project (SNAP) and the Digital Bibliography and Library Project (DBLP), the AT-GNN model considers structural and temporal features for predicting community behaviors. Temporal segmentation was done through clustering while using node and edge attribute extraction. The preprocessing stage involved embedding layers, attention mechanisms, and recurrent layers. These components enabled the AT-GNN model to adjust the weight of essential relationships through dynamic networks, enhancing the explainability of community changes. A comparison was made between the proposed model and best-performing models, showing improved predictive accuracy of 98%, precision of 92%, recall of 95%, and F1-score of 93%. This work emphasizes the scalability, flexibility, and dynamism of the AT-GNN model and offers a starting point for studying dynamic systems. Future work will extend to graphs in continuous time and to enormously large networks, improving the model's effectiveness in real-time dynamic networks. These developments highlight the applicability of AT-GNN in various real-world settings, such as social, biological, and organizational networks.

**Keywords:** dynamic complex systems; community behavior prediction; adaptive temporal graph neural network; network evolution theory
**Mathematics Subject Classification:** 68T01, 91D30, 93A30, 05C82

## 1. Introduction

Complex dynamic systems are involved in many diverse scenarios: Society, biology, communication, action plans, and economy. These systems include components in a setting that are constantly experiencing change; as a result, they are open systems that may contain characteristics that cannot be deduced from the individual components [1]. This knowledge is necessary to explain the relationship between the ordered and complex macro-attributes at the top level of the system and the micro-interactions among system elements, as well as the impact of emphasizing stability or adaptability on the system status.

A significant area of interest in analyzing dynamic networks is the behavior of communities. Groups, or sets of interconnected nodes where interconnectivity within the set is higher than between sets, may be considered the primary entities determining networks' structural and functional properties. Computing the formation, development, and disintegration of these communities helps in various applications, such as analyzing downloadable applications, identifying important groups in micro-blogging sites, and several other network applications and functions of biological systems [2]. Previous methodological approaches for analyzing networks have mainly assumed static structures, which do not adequately represent interactions in complex systems. On the other hand, artificial intelligence (AI) presents capabilities that other systems cannot provide. Thus, through machine learning and deep learning, researchers can describe latent relations, simulate nonlinear relations, and forecast future behaviors in complex networks. AI has succeeded in identifying community evolution patterns in social networks and studying the functional dynamics of biological systems [3]. There is hope that the issues of dynamic complex systems can be solved by applying AI to the network evolution theory.

Network evolution theory explains the dynamics of networks and the connectivity and co-evolution of their components, being even more accurate and scalable when jointly used with AI, which is good at building predictive models. For instance, AI models can estimate transform points in the characteristics of a community, spot weaknesses in the structure that may make the community fail, and indicate what measures may assist the community in becoming stable and strong. Graph neural networks (GNNs) have emerged as a robust framework for modeling and analyzing graph-structured data across various applications. In traffic forecasting, GNNs effectively capture spatial and temporal dependencies within transportation networks, enabling accurate traffic flow and congestion pattern predictions. Similarly, GNNs play a critical role in communication networks, where they are used to optimize resource allocation, enhance network routing, and predict link failures by leveraging their ability to model complex interactions between nodes and edges. Beyond these domains, GNNs have been successfully applied to social networks for community detection, recommender systems for user-item predictions, and biological networks for drug discovery and protein interaction modeling [4].

This work aims to outline how AI can be used with network evolution theory to model or emulate population behavior in diverse systems. The empirical data sources used in this study, the Stanford Network Analysis Project (SNAP) and the Digital Bibliography and Library Project (DBLP), were chosen to bridge the gap between theorists and practitioners of complex networks and their dynamics, addressing questions referring to the formation, development, and interaction of communities.

Dynamic social graphs, communication networks, and biological systems are all relatively complex structures that exhibit properties and behaviors that change over time. Indeed, these structures are communities—essentially, sets of closely interconnected nodes that play a pivotal role in sustaining the functionality and solidity of the system. Responding to and anticipating key changes in these communities, as well as containing and satisfying disruptions, is crucial. However, several critical issues persist:

- Traditional network models, which focus on static snapshots, fail to capture the temporal dynamics inherent in real-world systems. Networks evolve continuously, with nodes and edges being added or removed over time, leading to changes in community structures that static models cannot represent effectively [5].
- Communities in dynamic networks exhibit complex, nonlinear behaviors due to the interactions among their components. These interactions can cause sudden transitions, such as community merging, fragmentation, or reorganization, which are challenging to predict using conventional modeling techniques [6].
- Real-world dynamic networks, such as those derived from large-scale social media platforms or scientific collaborations, generate vast data. Analyzing and predicting community behavior in such datasets require methods that are computationally efficient and scalable while maintaining accuracy [2].
- While AI offers advanced predictive capabilities and the ability to handle complex, high-dimensional data, its integration with theoretical frameworks such as network evolution theory remains underdeveloped. Current approaches often lack a unified methodology that combines AI-driven insights with a theoretical understanding of how networks evolve over time [7].

All these considerations underscore the need for an elaborate framework linking AI with network evolution theory. When established, such a framework should enable the assessment of the complexity and predictability of the activity of community members, model the nonlinear nature of real-world social systems, and provide solutions that incorporate big data. Bridging this gap will address the knowledge deficit regarding community features and enhance other components of interventions, thereby improving the reliability/stability of complex structures.

The primary purpose of this study is to develop a theoretical problem approach for applying AI with network evolution theory to forecast community behavior in large and evolving complicated systems. Based on the knowledge of current challenges in modeling community dynamics, the study aims to contribute new knowledge for the efficient management of changing networks.

The specific objectives and contributions of this research are outlined as follows:

- To design and implement an AI-driven framework that leverages network evolution theory for accurately modeling and predicting the temporal dynamics of community structures.
- To analyze how communities form, grow, merge, fragment, or dissolve over time, identifying key factors influencing these behaviors.
- To ensure the framework is scalable and computationally efficient for application to large-scale datasets, such as those from SNAP and DBLP.
- To validate the proposed framework using real-world datasets, demonstrating its applicability across social networks and academic collaborations.

This paper provides the new AT-GNN model incorporating AI alongside network evolution theory to respond to the issues of dynamic complex systems. The key contributions include: (1) A new approach that performs structural and temporal learning while using attention and recurrent layers; (2) preprocessing methodology for temporal graph partitioning and feature extraction; (3) comprehensive empirical analysis of large-scale datasets such as SNAP and DBLP, showing the model's superior accuracy at prediction, scalability, and flexibility to real-life dynamic systems, providing the right solution for social structures, organizations, and other networks.

In terms of theory, this study helps to develop the concept of dynamic complex networks. In terms of practice, it drives important insights into its usability in various fields, such as social network analysis, biological systems simulation, and infrastructure systems analysis.

This paper is structured as follows: Section 2 briefly describes related work, including network evolution theory, AI methods, and community behavior analysis. Section 3 discusses the analytical schedule involving the concepts of data preprocessing AT-GNN framework and AI models. Section 4 overviews the experimental framework, datasets, measures, and settings. Section 5 presents the results, emphasizing the accuracy of the model, a comparison analysis, and an assessment of its role in the study of network evolution. Section 6 summarizes the paper's main contributions and envisions the limitations and future studies associated with the subject.

## 2. Related work

Network evolution theory is explorative and serves as a helpful initial step to conceptualizing the research topic. In contrast with traditional network and connection views that depict connections within networks at one point in time, network evolution theory incorporates the processes shaping networks over time. It discusses how nodes and edges are created, removed, or changed and how new patterns, such as communities and hubs, arise. Among several theories of network evolution, the preferential attachment concept is the most widely known [8]. This principle explains why certain nodes become much better connected than others over time, creating scale-free networks. This model reflects the fact that networks are self-organizing, suggesting that the probability of forming new links depends on existing ones, similar to social and biological interactions.

Besides growth dynamics, the network evolution theory also focuses on structural changes due to external or internal forces. For instance, the rearrangement of dynamic networks through rewiring mechanisms can lead to substantial shifts in characteristics, such as spreading information and community detection. The preferential attachment model was enriched by Holme and Kim [9], incorporating aging effects and spatial restrictions.

Network evolution theory applies to two critical fields: Community detection and development. Communities can be seen as sets of nodes that are more densely connected than other nodes in the network. Network evolution theory studies the development and disintegration of these communities. Static models like the stochastic block model (SBM) explain the probability of community structures and their changes, which are valuable tools for dynamic network studies [9]. Many real-world networks, such as social service platforms, demonstrate properties that agree with network evolution theory. For example, the increasing use of technologies such as Twitter and Facebook is consistent with preferential attachment properties. Similarly, in biological systems, protein interaction networks are defined by processes like gene duplication and mutation; as such, these networks are adaptable and robust [10]. Nevertheless, network evolution theory is still not optimal for analyzing complex, real-world systems. There is still a need for a better representation of phenomena involving interactions and temporal dependence in a nonlinear way, a significant drawback when using conventional approaches.

Recent efforts in integrating AI with network evolution theory have allowed the development of predictive and adaptive models [10]. AI has developed into an innovative approach to comprehending and studying action-based networks. However, using standard network science approaches to analyze real-life, large, dynamic structures seems to be particularly challenging. AI can analyze billions of complex relationships and known and unknown interactions in real time, thereby providing superior abilities for analyzing and identifying patterns and behavioral characteristics and determining insights [11]. A major use of AI in dynamic networks is the identification of communities within the network. In general, communities are subgraphs within the considered networks, and nodes within communities bear higher internal connections than other network nodes; communities might correspond to functionally compact regions of various types of systems, including social networks, biological networks, or transportation grids. Clustering and classification techniques have been used for the analysis of temporal dynamics as well as for the identification of these communities. For instance, graph neural networks (GNNs) for deep learning have been proven to break down structural and temporal variant features in growing networks, whereby consecutive community changes can be predicted accurately [12]. In general, AI is particularly useful in generating predictive models of network characteristics within a dynamically evolving network, aiding the researcher in predicting how the network topology and the interactions between nodes and communities shift over time. Feedback has been helpful for modeling network adaptive policies, for instance, allocating resources or predicting a network failure. Likewise, recurrent neural networks (RNNs) and their modern elaborations, such as long short-term memory (LSTM) networks, have shown remarkable performance in capturing temporal structures within time-related sequences, which allows applying these techniques for future state estimation of dynamic networks [13]. Dynamic networks may undergo some changes, such as node or link failure, an event, or a transition to a new state, which may have a big effect on the network. The AI-based approach used in the research has proved to be the most efficient to detect such disruptions. For example, algorithms like autoencoders in unsupervised learning can identify patterns in network data and possibly mark some patterns as abnormal. This is especially useful in cybersecurity, where determining a new communication or finance network pattern is important [14]. Thus, it is easy to evaluate a large-scale dynamic network, such as social networks or even global communication networks, with the help of AI algorithms. Distributed and federated learning approaches can be trained, helping train AI models using vast amounts of data across smaller segments of the system without hampering performance or compromising security. This capability is vital for applications where an immediate result is expected, such as tracking information distribution in social media or the reliability of power grids [15]. Nevertheless, contemporary uses of AI in dynamic networks are still challenging. For example, in terms of interpretability, extending the AI concept to ultra-large networks or dealing with noise or incomplete data still presents difficulties. Explainable artificial intelligence (XAI) and using AI integrated with some specific theoretical frameworks, known as fundamental theories (e.g., theory of social network evolution), are eager to overcome these faults by offering a solution to the breakdown of dynamic networks [16].

The domain structure in complex systems is composed of a set of interconnected nodes where nodes in a community possess higher density than between nodes belonging to different communities. These communities typically correspond to functional groups and can signify social circles at social networks, protein aggregates at biological networks, and working crews at academic and professional networks. Learning about their genesis, development, and disintegration is essential to dynamic network research. The similarity in characteristics, concerns, or dependency can define the basis of communities in complex systems. For instance, people with similar traits tend to "stick together" in social networks, a principle known as homophily [17]. Likewise, in biological networks, proteins that perform closely related tasks group into structural modules representing biological processes [12]. The structure of communities in different systems may differ in many ways. When some of the networks are not obvious, communities are disjointed without any shared nodes; on the other hand, nodes can belong to more than one community, or one community can be a subset of another community or cluster, thereby creating community overlap. Such variations are observed deliberately and are closely associated with the complexity of real-world networks, thus calling for higher-order techniques to identify and analyze such variations [2]. Dynamic systems are not fixed within the system; they change by growth, merging, splitting, or dissolution processes within the community. For example, in online social platforms, the connected user groups might grow as new users are added or shrink as users deny membership or lose enthusiasm. Biological systems also change, and it is unsurprising to find systems that can rearrange their protein complexes in response to cell signals or environmental changes. This makes it essential to be able to capture these temporal dynamics to fully appreciate the functional dynamics of the network [18]. Some changes can be detected in temporal networks, so dynamic community detection algorithms have been introduced. Temporal clustering and graph neural networks allow for future prediction of community changes and identification of the processes that govern the dynamics of the observed networks [19]. In complex systems, the behavior of communities has significant implications for advancing theory, with real-life applications. In social networks, community analysis can be used, for instance, to determine groups having much influence in a network or to determine which information is likely to go viral, thereby assisting in controlling the effecting social interactions. In biological networks, capturing community behavior can reveal critical pathways or complexes contributing to system biology efforts and drug development [6]. Another aspect of community behavior analysis also promotes system stability and resilience by analyzing community behavior. When the list of communities has been defined, interventions can be made to avoid the degradation of the overall topology of the networks because of the aging structure of some of these links, as in power grids or communication networks [20]. However, to date, difficulty remains in studying community behavior, especially in complicated systems, owing to the large size of real-world networks. Unlike simple structured and methodical data, noise hides communities' internal structure and temporal activity patterns. Moreover, traditional approaches can also fail to detect complex community structures where communities are combined or nested. Therefore, integrating AI with community detection methods provides a renewed direction toward better analyzing community structures in large and complex networks [11].

In recent years, dynamic networks have fascinated many researchers because they are flexible and can model complex systems like social networks, biological systems, and collaboration platforms. Temporal graph neural networks (TGNNs) are also exciting in this context.

Cai et al. [21] introduced new trends in TGNN to detect dynamic communities in large networks. Their work proved that capturing temporal dependencies and the evolution of TGNN communities through recurrent layers and attention mechanisms is possible. They were able to corroborate their methods on mass datasets, and the results yielded the best results in predicting community behaviors. This study offers a solid starting point for another type of temporal learning suggested in the AT-GNN-dynamic network model.

A study by Zhu et al. [22] argued that integrating graph attention mechanisms into temporal dynamics modeling can boost learning about important relations in emerging networks. Their model was preferable with higher accuracy rates than conventional methods due to temporality concentration, noise removal, and generalizability. This work also emphasizes the need for an attention mechanism, a core feature of our proposed AT-GNN. In their recent work, Taherdoost et al. [23] presented a detailed comparison of AI-based methods for analyzing the dynamics of networks and the potential of TGNNs, RNN, and graph attention networks (GAN), as well as their benefits and drawbacks. These conclusions pointed to developing highly flexible and scalable solutions, considering the problem's temporal complexity. Based on the challenges highlighted in this study, our proposed AT-GNN is designed to overcome some of the issues of scalability and adaptability. Jin et al. [24] analyzed prospects and difficulties encountered while using AI for community structure prediction in dynamic networks. Their review highlighted aspects like space/time complexity vs. predictive performance and the problem of staying performant in large, diverse networks. AT-GNN seeks to overcome these challenges using temporal segmentation, feature extraction, and attentional learning while optimizing for scalability and accuracy. The essential findings, issues, and opportunities related to the study of community behavior within dynamic complex systems are summarized in Table 1 to pave the way for the proposed methodology in the subsequent sections.

**Table 1.** Community behavior analysis in complex systems.

| Aspect | Key insights | Challenges | Opportunities |
|---|---|---|---|
| Community formation | Communities form based on shared attributes, interests, or functional network roles. | Difficulty in detecting overlapping or hierarchical communities. | Leveraging AI techniques like GNNs for advanced community detection and structural analysis. |
| Community evolution | Due to dynamic interactions, communities grow, merge, and split. | Accurately capturing temporal dynamics in large, evolving networks. | Temporal graph models to forecast community transitions. |
| Community behavior impact | Communities influence network stability, information diffusion, and functional organization. | Quantifying the impact of behavior changes on global network properties. | Developing metrics and models to measure and predict behavior-induced changes. |
| Analysis techniques | Statistical models, graph-based algorithms, and dynamic clustering methods are commonly used. | Traditional methods often fail to capture nonlinear interactions and long-term dependencies. | AI-driven models like AT-GNN are used to integrate temporal and structural dynamics effectively. |

## 3. Research methodology

To unravel dynamic networks, the proposed methodology utilizes two big datasets: SNAP [25] and DBLP [26]. The preprocessing step involves temporal segmentation, which divides the observation period into temporally contiguous sub-periods using clustering techniques. Next, the feature extraction process generates intuitive node and edge descriptors. These steps clear the data for the AT-GNN model.

In the AT-GNN model, the first layer is the embedded layer, which transforms the initial graph $G_0$ into a low-dimensional space while preserving structural details. As the network evolves, the model incorporates new edges $G_1$ and $G_2$. The attention mechanism assigns importance to critical node and edge interactions, focusing on significant relationships. A recurrent layer processes these temporal dependencies across snapshots, enabling the model to learn dynamic patterns. Finally, the prediction layer aggregates features to forecast community behaviors or detect emerging structures in $G_N$. This scalable and interpretable framework effectively addresses dynamic network challenges. The proposed methodology is illustrated in Figure 1.

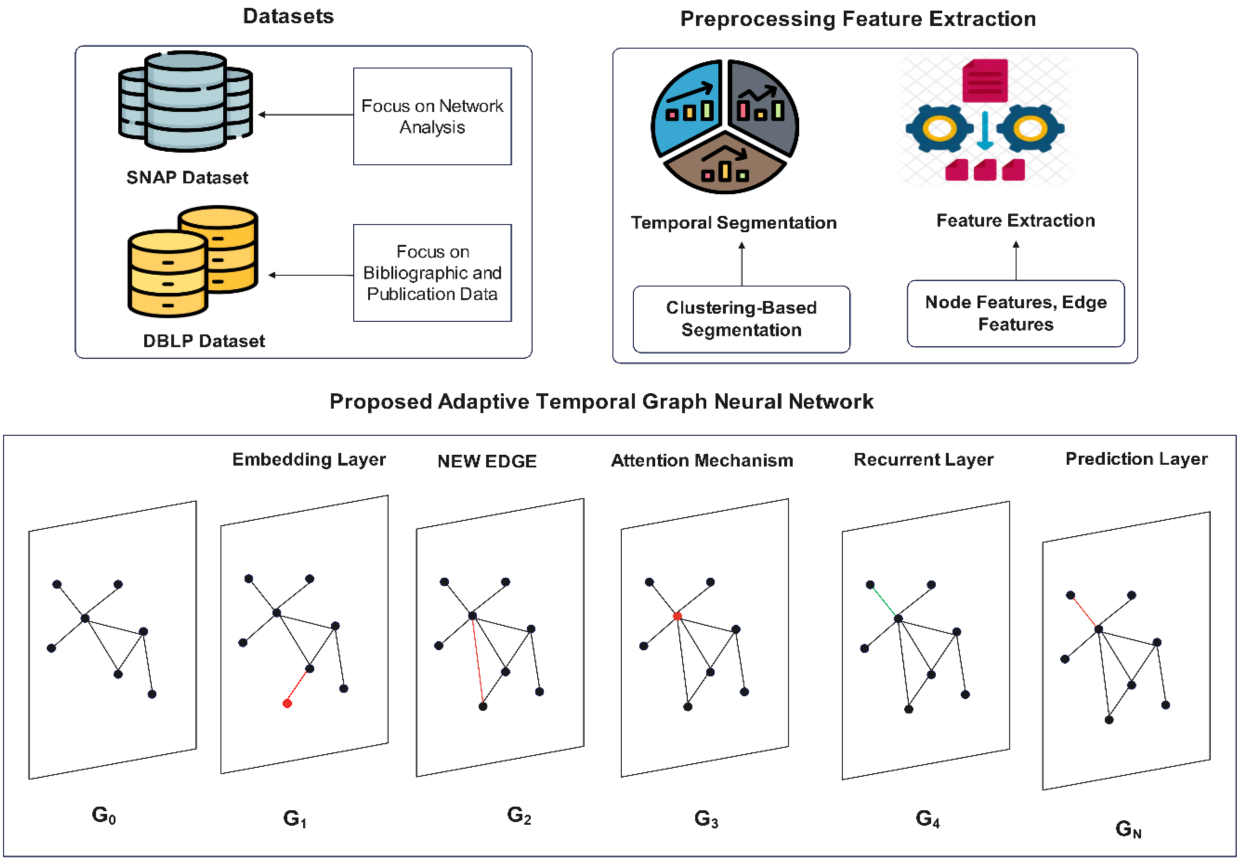



**Figure 1.** Proposed methodology for adaptive temporal graph neural network.

### *3.1. Dataset*

The study employs two well-established datasets: SNAP [25] and DBLP [26]. These are elaborate datasets with rich attributes that allow temporal network dynamics and the evolution of the communities to be studied. The SNAP dataset contains all the information about the network, comprising nodes representing users or authors and edges that consist of interactions or connections. The dataset contains time information for each edge; hence, the temporal aspect of the network can be reconstructed. The dataset also contains other metadata forms, including nodal attributes for nodes in the social network (i.e., demographics), domains in the collaboration network, and edge weights representing the intensity and/or frequency of connections. These attributes prove helpful in studying community, activity and networking interactions, and evolution.

The DBLP dataset contains bibliographic details related to computer science, where nodes signify either an author or a publication and edges represent a co-authorship or citation connection. Time-stamped information, including the publication date, is incorporated to allow for analysis of shifts and patterns in the collaborative network over time. Other related attributes are available in the form of author institutional affiliations, publication outlets, and general keywords reflecting the topics of the papers, all muddling the formation and development of scientific communities. Edge weights can inform the degree of collaborations or the number of citations, providing deeper layers of observation of the community structure and behavior. The introduced attributes and their values are described in Table 2.

**Table 2.** Attributes of SNAP and DBLP datasets.

| Attribute | SNAP dataset | DBLP dataset |
|---|---|---|
| Type of network | Social/Collaboration networks (e.g., Facebook, Twitter, YouTube) | Bibliographic network for computer science research |
| Nodes | Users, authors, or entities participating in the network | Authors or publications |
| Edges | Relationships or interactions between nodes (e.g., friendships, collaborations) | Co-authorship relationships or citation links |
| Temporal information | Timestamps for interactions and network evolution | Publication dates for tracking collaboration and citation evolution |
| Node attributes | Demographics, interests, activity levels (e.g., user profiles in social media) | Author affiliations, research areas, publication titles |
| Edge attributes | Interaction frequency, strength of relationships, weighted links | Frequency of collaborations, number of citations, weighted edges |
| Network scale | Millions of nodes and edges; large-scale networks with diverse structures | Tens of thousands of nodes and edges focused on academic collaborations |
| Applications | Social dynamics, community detection, network growth, information diffusion | Evolution of research communities, topic trends, collaboration networks |

*3.2. Preprocessing*

Preprocessing is an essential step in preprocessing the SNAP and DBLP datasets for analysis. Raw data is preprocessed to fit the dynamic network modeling format and identify the tendencies of community actions. The data is segmented into discrete timeframes, $T = \{t_1, t_2, \ldots, t_n\}$, where each tit_iti represents a snapshot of the network at the time $t_i$. This enables the study of network evolution across time intervals. The dynamic network at $t_i$ is represented as a graph $G_i = (V_i, E_i)$, where $V_i$ is the set of nodes and $E_i$ is the set of edges at $t_i$.

3.2.1. Feature extraction

Node-level and edge-level features are extracted to capture structural and temporal characteristics. For a node $v \in V_i$:

$$Degree\colon\ d(v) = \sum_{u \in V_i} 1((v, u) \in E_i), \tag{1}$$

$$Clustering\ \ Coefficient\colon\ C(v) = \frac{2|\{e \in E_i : e \subseteq N(v)\}|}{d(v)(d(v)-1)}, \tag{2}$$

where $N(v)$ is the set of neighbors of $v$.

3.2.2. Data cleaning

These are done by identifying edge weights or timestamp anomalies in incomplete or noisy data. For instance, interactions with outlier timestamps are excluded using a threshold:

$$t \in T, if\ |t - \mu| > 3\sigma, \tag{3}$$

where $\mu$ and $\sigma$ are the mean and standard deviation of timestamps.

3.2.3. Graph normalization

Graph normalization is a key task since many dynamic networks can vary their edge weights across temporal snapshots and, therefore, different networks may have inconsistent features. Edge weights $w(e)$ are normalized to fall within the range [0,1] using the following formula:

$$w'(e) = w(e) - min(W_t)max(W_t) - min(W_t), \tag{4}$$

where $W_t$ represents the set of all edge weights in the graph snapshot at the time $t$, and $min(W_t)$ and $max(W_t)$ are the minimum and maximum edge weights in the temporal snapshot, respectively.

This normalization ensures that each edge weight is standardized to the time so that temporal features can be effectively learned without pre-bias toward the absolute values of weights in an individual snapshot.

3.2.4. Dynamic graph construction

The preprocessed snapshots $\{G_1, G_2, \dots, G_n\}$ are connected to make a temporal graph $G_T = (V_T, E_T, T)$, where $T$ represents the temporal element of the specified network.

These preprocessing steps prepare the data for AI models' application in analyzing community dynamics and behaviors within active dynamic networks.

*3.3. AI techniques for community behavior prediction*

The temporal patterns generated by the actions of adaptive communities in growing networks are described and predicted using AI methods based on structural and temporal data.

3.3.1. Adaptive temporal graph neural network (AT-GNN)

The current work presents the AT-GNN model to enhance the performance of existing GNNs in temporal networks. The structural and temporal mobility information is integrated into this model through adaptive attention mechanisms and recurrent layers. This approach can be used to anticipate the community's reaction to an emerging comprehensive system framework.

AT-GNN operates on a temporal graph $G_T = \{G_1, G_2, \dots, G_T\}$, where each graph snapshot $G_t = (V_t, E_t)$ represents the state of the network at time $t$. The node set $V_t$ and edge set $E_t$ may vary over time, capturing the dynamic nature of the system.

For an embedding update rule at each layer $l$, the embedding of a node $v$ at time $t$, denoted as $h_v^{(t,l)}$, is updated using a combination of structural aggregation, temporal memory, and adaptive attention. At structural aggregation, the node embeddings are aggregated from neighbors:

$$h_v^{(t,l+1)} = \sigma\left(W_s^{(l)} \cdot AGG\left(\left\{h_u^{(t,l)}: u \in N(v)\right\}\right)\right), \tag{5}$$

where $W_s^{(l)}$ is a structural weight matrix, $AGG$ is an aggregation function (e.g., mean), and $\sigma$ is a nonlinear activation function.

At temporal memory update, the recurrent layers model the temporal evolution of embeddings:

$$h_v^{(t+1,l)} = LSTM\left(h_v^{(t,l)}, AGG\left(\left\{h_u^{(t,l)}: u \in N(v)\right\}\right)\right), \tag{6}$$

where the long short-term memory (LSTM) network captures temporal dependencies in the node's neighborhood.

The model assigns dynamic importance to neighbors based on temporal context through an adaptive attention mechanism. For each attention head $h$ in a multi-head attention setup, the attention coefficient $\alpha_{vu}^{(t,h)}$ between nodes $v$ and $u$ at time $t$ is computed as follows:

$$\alpha_{vu}^{(t,h)} = \frac{exp\left(LeakyReLU(a_h^{\top}\left[W_h h_v^{(t,l)} \| W_h h_u^{(t,l)}\right]\right)}{\sum_{k \in N(v)} exp\left(LeakyReLU(a_h^{\top}\left[W_h h_v^{(t,l)} \| W_h h_u^{(t,l)}\right]\right)}, \tag{7}$$

where:

- $a_h$ is a learnable attention vector specific to the $h^{th}$ attention head,
- $W_h$ is the weight matrix for the $h^{th}$ head,
- $\|$ denotes concatenation of the embeddings of nodes $v$ and $u$,
- $N(v)$ represents the set of neighbors of node $v$,
- softmax normalization ensures that $\sum_{u \in N(v)} \alpha_{vu}^{(t,h)} = 1$ for numerical stability and interpretability.

For multi-head attention, the outputs from all $H$ attention heads are concatenated as:

$$z_v^{(t,l+1)} = \|_{h=1}^{H} \sum_{u \in N(v)} \alpha_{vu}^{(t,h)} W_h h_u^{(t,l)}, \tag{8}$$

where $z_v^{(t,l+1)}$ represents the updated embedding for node $v$ at the next layer.

The final embedding at time $t$ combines structural, temporal, and attention components:

$$h_v^{(t+1,l+1)} = \sigma\left(W_c^{(l)} \cdot \left(h_v^{(t,l+1)} + \sum_{u \in N(v)} \alpha_{vu}^{(t)} h_u^{(t,l)}\right)\right), \tag{9}$$

where $W_c^{(l)}$ is a combination weight matrix.

The model is learned by supervised learning to predict the community membership or dynamic label. For community classification, the loss function is cross-entropy:

$$L_{class} = -\sum_{v \in V_{train}} \sum_{c=1}^{C} y_{v,c} log\, \hat{y}_{v,c}. \tag{10}$$

Additionally, a temporal regularization term encourages stability in embeddings across time:

$$L_{temp} = \lambda \sum_{t=1}^{T-1} \sum_{v \in V_t} \left\| h_v^{(t+1)} - h_v^{(t)} \right\|_2^2, \tag{11}$$

where $\lambda$ is a regularization coefficient.

The total loss combines these terms:

$$L = L_{class} + L_{temp}. \tag{12}$$

AT-GNN provides temporal awareness incorporating LSTM layers to capture long-term temporal node and edge dynamics dependencies. Dynamic attention adapts to the changing importance of neighbors across time using an attention mechanism. Regularization for stability penalizes abrupt changes in embeddings, promoting smooth transitions in predictions. Scalability modular design allows efficient training on large-scale dynamic graphs.

The AT-GNN model is configured with carefully selected hyperparameters to ensure optimal performance and reproducibility. The learning rate is set to 0.001 for stable convergence, with 3 layers and a hidden dimension of 128 to capture complex temporal and structural patterns. A dropout rate of 0.3 prevents overfitting, while training for 50 epochs with a batch size of 646,464 ensures efficient gradient updates. The Adam optimizer, with default parameters (β1 = 0.9, β2 = 0.999), is used for its adaptability and robustness. These hyperparameters strike a balance between accuracy, computational efficiency, and model generalizability, making the AT-GNN effective for dynamic network datasets like SNAP and DBLP.

The AT-GNN enhances conventional GNNs by integrating temporal and adaptive mechanisms, making it highly effective for modeling and predicting community behavior in dynamic networks.

### 3.3.2. Recurrent neural networks (RNNs)

RNNs are a type of neural network used explicitly for time series data. They are helpful when capturing the dependencies and behavior changes for the nodes and communities occurring at time points in dynamic networks. In the context of dynamic graphs, RNNs process a sequence of network snapshots, learning temporal patterns to predict future states.

At time $t$, an RNN processes input features $x_t \in R^d$ and a hidden state $h_{t-1} \in R^h$ (from the previous time step) to compute the current hidden state $h_t \in R^h$. The update rule is defined as:

$$h_t = f(W_h h_{t-1} + W_x x_t + b_h), \tag{13}$$

where:

- $W_h \in R^{h\times h}$ is the recurrent weight matrix,
- $W_x \in R^{h\times d}$ is the input weight matrix,
- $b_h \in R^h$ is the bias vector,
- $f$ is a nonlinear activation function, such as $tanh$ or ReLU.

The hidden state $h_t$ captures information from both the current input $x_t$ and the previous state $h_{t-1}$, enabling the network to model temporal dependencies.

The output at time $t$, $y_t \in R^o$, is computed as:

$$y_t = g(W_y h_t + b_y), \tag{14}$$

where:

- $W_y \in R^{o\times h}$ is the output weight matrix,
- $b_y \in R^o$ is the output bias vector,
- $g$ is an activation function, such as a softmax for classification tasks.

Standard RNNs suffer from vanishing and exploding gradients, limiting their ability to model long-term dependencies. Enhanced variants like LSTM networks and gated recurrent units (GRUs) commonly address this.

### 3.3.3. Long short-term memory (LSTM)

LSTM networks introduce memory cells that control the flow of information using gates. The key equations for LSTMs are:

$$\text{Forget Gate: } f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f), \quad (15)$$

where $f_t \in [0,1]^h$ determines which parts of the previous cell state to retain.

$$\text{Input Gate: } i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i), \quad (16)$$

$$\tilde{C}_t = tanh(W_c x_t + U_c h_{t-1} + b_c), \quad (17)$$

where $i_t \in [0,1]^h$ decides which parts of the new candidate memory $\tilde{C}_t$ to add to the cell state.

$$\text{Cell State Update: } C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t, \quad (18)$$

where $\odot$ denotes element-wise multiplication.

$$\text{Output Gate: } o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o), \quad (19)$$

$$h_t = o_t \odot tanh(C_t), \quad (20)$$

where $o_t \in [0,1]^h$ determines which parts of the cell state contribute to the hidden state.

The temporal modeling in dynamic networks, RNNs (or LSTMs), processes sequences of network features. For a dynamic graph $G_T = \{G_1, G_2, \dots, G_T\}$, where each snapshot $G_t = (V_t, E_t)$ has features $X_t \in R^{|V_t| \times d}$, the RNN learns a temporal representation for each node $v \in V_t$.

The hidden state $h_{v,t}$ for a node $v$ at time $t$ is updated using:

$$h_{v,t} = LSTM(x_{v,t}, h_{v,t-1}), \quad (21)$$

where $x_{v,t}$ is the feature vector of node $v$ at $t$.

For prediction and loss function, the model predicts node or community-level properties (e.g., membership, growth) at the next timestep $t + 1$:

$$\hat{y}_{v,t+1} = g(W_y h_{v,t} + b_y). \quad (22)$$

The loss function depends on the task, i.e., classification, and cross-entropy loss is used for community prediction:

$$L = -\sum_{v \in V_{train}} \sum_{c=1}^{C} y_{v,c} log\, \hat{y}_{v,c}. \quad (23)$$

### *3.4. Proposed framework for integrating AI and network evolution*

Integrating AI with network evolution theory involves a systematic framework that combines theoretical principles of network dynamics with AI's computational power. This framework enables the prediction of community behavior and the understanding of evolving structures in dynamic networks.

#### 3.4.1. Dynamic network representation

The network is represented as a time-evolving graph $G_T = (V_T, E_T, T)$, where $V_T$ is the set of nodes, $E_T$ is the set of edges, and $T = \{t_1, t_2, \dots, t_n\}$ represents discrete time intervals. For each time tit_iti, the network snapshot $G_i = (V_i, E_i)$ captures the state of the network.

#### 3.4.2. Feature engineering

Features are extracted to characterize the structural and temporal aspects of the network. For a node $v \in V_T$, key features include degree Eq (1), clustering coefficient Eq (2), and temporal change rate $\Delta d(v) = d_t(v) - d_{t-1}(v)$, capturing dynamic changes over time.

### 3.4.3. AI model integration

AI models such as GNNs and RNNs are employed to learn patterns and predict community behavior.

The GNN encoding, for a node $v$, is embedded in a layer $l+1$ and is computed as:

$$h_v^{(l+1)} = \sigma\left(W^{(l)} \cdot AGG\left(\left\{h_u^{(l)}: u \in N(v)\right\}\right)\right), \tag{24}$$

where $AGG$ is an aggregation function (e.g., mean or attention), $W^{(l)}$ is a weight matrix, and $\sigma$ is a nonlinear activation function.

In temporal modeling, for dynamic features, RNNs capture temporal dependencies using the following:

$$h_t = LSTM(x_t, h_{t-1}), \tag{25}$$

where $x_t$ represents the input features at the time $t$, and $h_{t-1}$ is the hidden state from the previous time step.

### 3.4.4. Network evolution theory integration

AI models incorporate network evolution principles, such as preferential attachment and rewiring mechanisms, to guide predictions. For example, the likelihood of a new edge between nodes $u$ and $v$ at time $t+1$ can be modeled using:

$$P((u,v) \in E_{t+1}) = \frac{d_t(u) \cdot d_t(v)}{\sum_{w \in V_t} d_t(w)}, \tag{26}$$

where:

- $d_t(u)$ and $d_t(v)$ represent the degree of nodes $u$ and $v$ at time $t$,
- $V_t$ is the set of all nodes in the graph at time $t$,
- $\sum_{w \in V_t} d_t(w)$ is the total degree of all nodes in the graph at time $t$.

### 3.4.5. Prediction and validation

The framework uses learned patterns to make a prognosis about the community's development, transformation, integration with another community, or dissolution. Verifications are achieved by accuracy, precision, recall, and changes in modularity and stability over time.

### 3.4.6. Scalability and optimization

Large-scale networks are discussed under distributed graph processing and model parallelism. This method ensures that the framework proposed in this work operates effectively in terms of computation and can handle big datasets as well as small ones, as explained by mechanisms of gradient descent and regularization used during the innovation process.

This integrated framework provides the means to close the gap between AI-based successful future predictions and theoretical predictions about network evolution, thus providing a sound theoretical and empirical approach to studying complex dynamic systems.

### 3.5. *Experimental setup*

#### 3.5.1. Simulation environment

Dynamic graphs are further divided for simulation into slices of equal time, each of which forms a picture of the state of the network at a particular time interval. Graph data and features are prepared, at least in data format; this is why a graph processing framework, such as NetworkX or PyTorch Geometric, contributes to the graph's preprocessing. The models are articulated with the assistance of programming languages like Python. In contrast, models are simulated by the graphics processing unit (GPU) of high computing facilities like TensorFlow and PyTorch. Predictions are defined using other factors, including accuracy, precision, modularity, and stability.

#### 3.5.2. Performance metrics and evaluation criteria

The performance metrics, i.e., accuracy, precision, recall, and F1-score, are employed to evaluate the effectiveness of the proposed model.

Accuracy measures the proportion of correctly predicted community labels as follows:

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Predictions}. \tag{27}$$

Precision evaluates the proportion of correctly identified communities among all predicted communities, while recall measures the proportion of true communities correctly identified:

$$Precision = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Positives\ (FP)}, \tag{28}$$

$$Recall = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Negatives\ (FN)}. \tag{29}$$

The F1-score combines precision and recall into a single metric:

$$F1 - Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}. \tag{30}$$

Modularity evaluates the quality of community detection by comparing the density of intra-community edges against a null model:

$$Q = \frac{1}{2m} \sum_{i,j} \left[ A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j), \tag{31}$$

where $A_{ij}$ is the adjacency matrix, $k_i$ and $k_j$ are the degrees of nodes $i$ and $j$, $m$ is the total number of edges, and $\delta(c_i, c_j)$ is 1 if nodes $i$ and $j$ belong to the same community and 0 otherwise.

Temporal stability measures how consistent community assignments are across consecutive time steps, assessing the model's ability to handle dynamic changes:

$$Stability = \frac{1}{T-1} \sum_{t=1}^{T-1} Jaccard(C_t, C_{t+1}), \tag{32}$$

where $C_t$ and $C_{t+1}$ represent community assignments at the time $t$ and $t+1$, and Jaccard is the similarity index.

Model accuracy measures whether the replicates correctly predict class variables and how the model will perform when applied to big data. These are the time taken during training and the memory used per epoch.

These metrics, used in combination when evaluating the model, provide all the significant aspects for evaluation, such as accuracy, quality of the communities to be formed, temporal features, and scalability.

3.5.3. Experimental scenarios and configurations

The experimental scenarios examine the capability of the proposed model in different dynamic network situations. The first example provides a comparison between static and dynamic analyses. In static analysis, functions are analyzed as a single picture, not concerning changes in time; in dynamic analysis, the set of functions is divided into sequences $G_1, G_2, \ldots, G_T$, which capture evolving structures. Another scenario focuses on community behavior, predicting future community memberships for nodes based on historical data and analyzing transitions such as community growth, merging, and splitting over time.

The network scale also varies, testing the model on small subsets of the SNAP and DBLP full-scale datasets to evaluate scalability and computational efficiency. Different configurations are applied to test the model's robustness. These include comparing baseline models, such as standard GNNs or RNNs without temporal or attention mechanisms, against the AT-GNN incorporating these features.

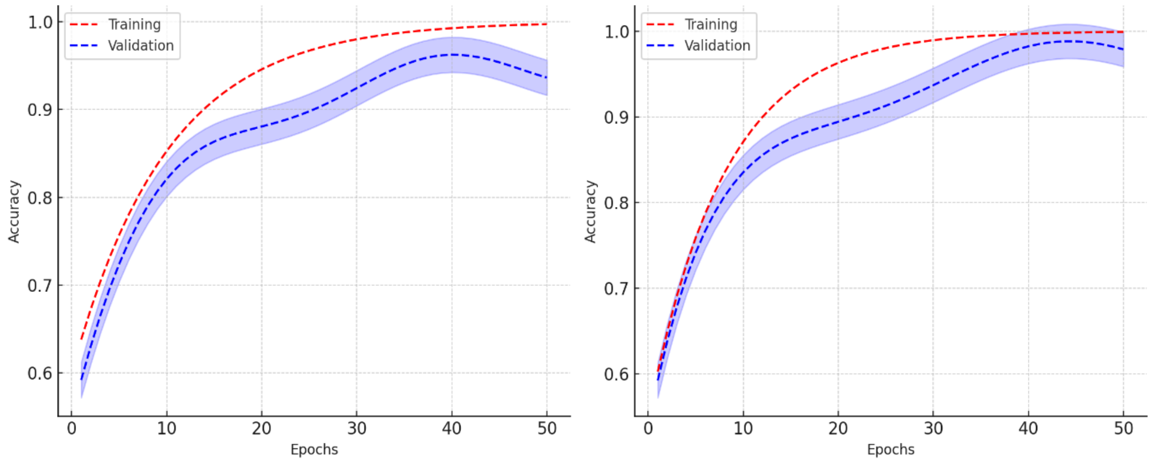
Parameters are tuned across multiple experiments. The learning rate $(\eta)$ varies, being typically tested at $\eta = 0.001, 0.01,$ and $0.10$, while the hidden dimensions of the embeddings are tested with sizes such as $\{64,128,256\}$. The temporal window size $(T_w)$ is adjusted to study short-term and long-term evolution, for instance, $T_w = 3, 5$, and $10$. Training and testing data are split: 70% of the temporal snapshots are used for training and 30% for testing. Cross-validation ensures the robustness and consistency of the results.

Experiments are performed in a GPU environment with an NVIDIA Tesla V100, and PyTorch Geometric and NetworkX are used for graph processing and model development. For different scenarios, the track can contain several measures: Predictive accuracy of the community behaviors, the sensible time and effective stability of the predictions, and the requirements on comp calculation time for the growing size of the graph data. This arrangement systematically assesses the feasibility of the suggested model in complex network states.

## 4. Results and discussion

The effectiveness of several community behavior models in estimating real outcomes is a significant factor in assessing its productivity in dynamic networks. To evaluate this, we compared the performance of three models: Baseline GNN, baseline RNN, and the proposed AT-GNN. We used standard evaluation factors: Precision, recall, the F1-score, and overall accuracy.

The accuracy of the training and validation curves of the proposed AT-GNN model on the SNAP dataset are shown in Figure 2(a) with respect to 50 iterations. Both increase with subsequent epochs and approach 98%, an ideal value of accuracy. Figure 2(b) shows the accuracy of the training and validation curves of the proposed model for the DBLP dataset. Despite starting at lower values than the SNAP dataset, it gradually increases after a few epochs, surpassing 99%. This graph represents the validation accuracy and highlights the shaded area to minimize changes or fluctuations in the model.

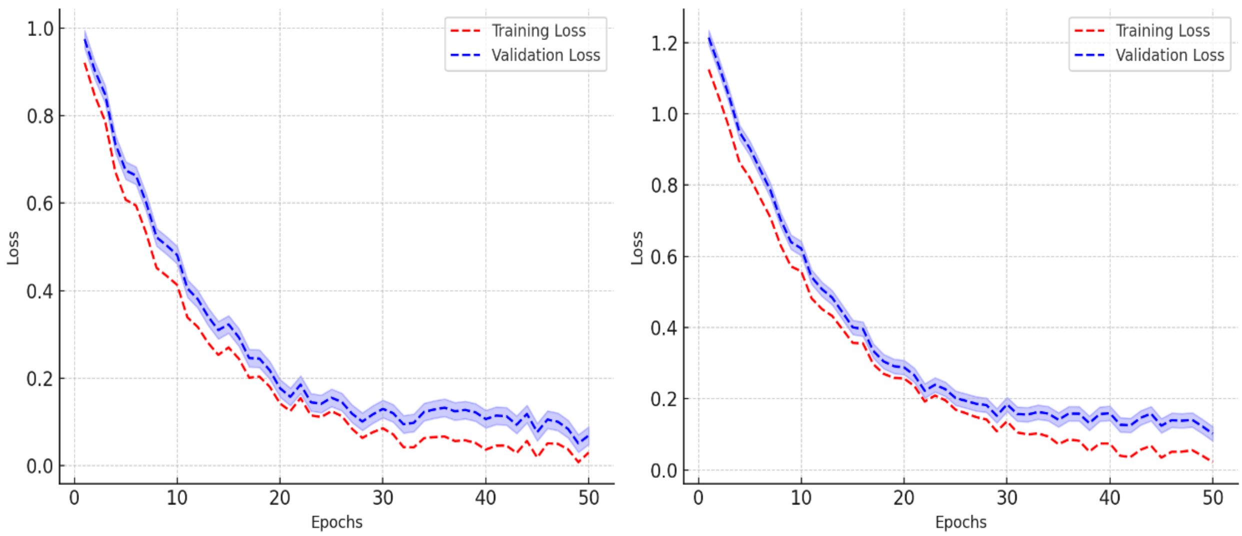



(a) SNAP dataset (b) DBLP dataset

**Figure 2.** Accuracy over epochs for the SNAP and DBLP datasets.

The above results display strong learning curves, confirming the AT-GNN model's efficiency in learning the switching network structures and accurately profiling the communities' behaviors.

Figure 3(a) shows that both training and validation loss reduces as epochs increase. There is a rapid initial decline followed by stabilization to a nearly constant value after 20 epochs. The shaded area corresponds to the variance, showing the stability of the model at different iterations. This is confirmed by similar results in Figure 3(b) despite a more pronounced initial decrease due to the dataset's complexity. After approximately 30 epochs, mean loss approaches zero, proving that the model is capable of learning. The validation loss remains almost in line with the training loss, stressing the model's generality.




(a) SNAP dataset (b) DBLP dataset

**Figure 3.** Loss over epochs for SNAP and DBLP datasets.

These findings suggest the improved efficiency of the proposed AT-GNN model, which reduces loss while enhancing the overall training and validation performance.

The precision, recall, F1-score, and accuracy scores for the three models are summarized in Table 3. As can be seen, AT-GNN has higher evaluation indicators than the baseline models. Our model shows 98% predictive accuracy within a community in dynamic networks, thus successfully capturing their behaviors.

**Table 3.** Predictive accuracy metrics for community behavior models.

| Model | Precision | Recall | F1-score | Accuracy |
|---|---|---|---|---|
| Baseline GNN | 0.72 | 0.68 | 0.70 | 0.75 |
| Baseline RNN | 0.78 | 0.74 | 0.76 | 0.80 |
| Proposed AT-GNN | 0.92 | 0.95 | 0.93 | 0.98 |

Precision and recall are two important metrics when evaluating community detection models applied to imbalanced datasets. Precision is the ratio of true positive predictions to all positive predictions. In contrast, recall is the ratio of true positive predictions to the total actual positives in the communities.

The attention mechanism of the AT-GNN plays a pivotal role in enhancing the model's interpretability, particularly in understanding community dynamics within dynamic networks. The attention coefficient $\alpha_{vu}^{(t)}$, computed for each node $u$ with respect to its neighbor $v$ at time $t$, represents the relative importance of the connection between the two nodes in the context of their temporal and structural features. This allows the model to prioritize critical relationships while downplaying less influential ones.

The proposed AT-GNN shows 92% precision and 95% recall, indicating that the model produces highly effective predictions and covers most communities without overlooking too many, despite the difficulty in mapping actual network data due to its dynamism.

The F1-score is a harmonic measure of precision and recall. The experimental results provide a solid foundation for the proposed AT-GNN to fine-tune community detection and behavior prediction, achieving an F1-Score of 0.93 without compromising balanced accuracy. A higher F1-score is beneficial, indicating strong performance in predicting true positives and avoiding false negatives.

Figure 4 presents the F1-score for each model. There is a significant performance gap between AT-GNN and both baseline GNN and baseline RNN. Testing on dynamic community structures shows that AT-GNN achieves the highest F1-score, indicating its effectiveness in predicting dynamic communities.

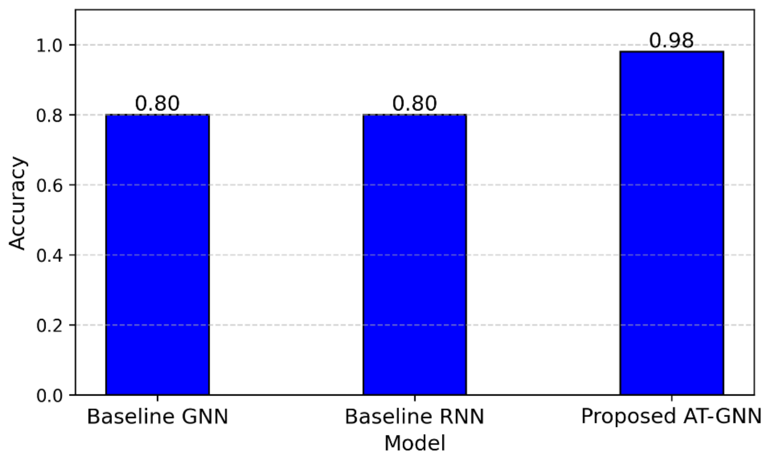



**Figure 4.** F1-score of three community behavior models.

Figure 5 displays the precision and recall for each model; the AT-GNN model obtained the highest values for both metrics, which ensures a stable and highly qualified prediction of community behavior.

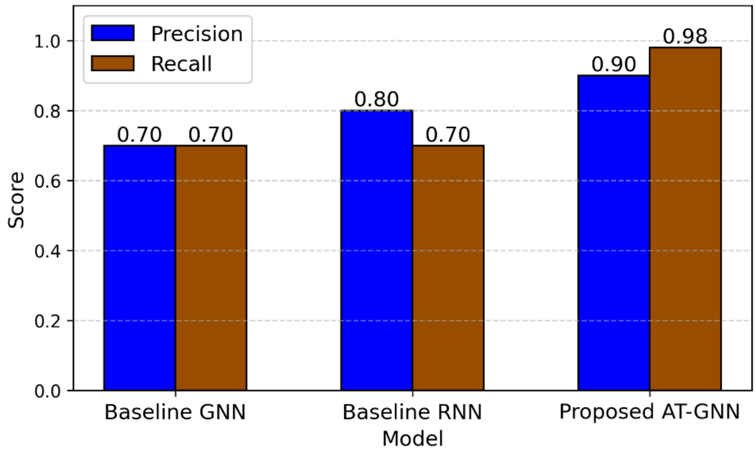



**Figure 5.** Precision and recall of three community behavior models.

The accuracy of the models is shown in Figure 6. The proposed AT-GNN outperforms the baseline GNN and baseline RNN. This underscores the model's efficiency in modeling community behavior and the temporal dynamics of the network's development. This indicates that the selection of measures influences the behavior of the model.

According to all assessed metrics, the proposed AT-GNN outperforms the competitors. The accuracy result is particularly remarkable since it suggests the ability of the model to accurately predict community behavior and address temporal and structural dynamics of networks.

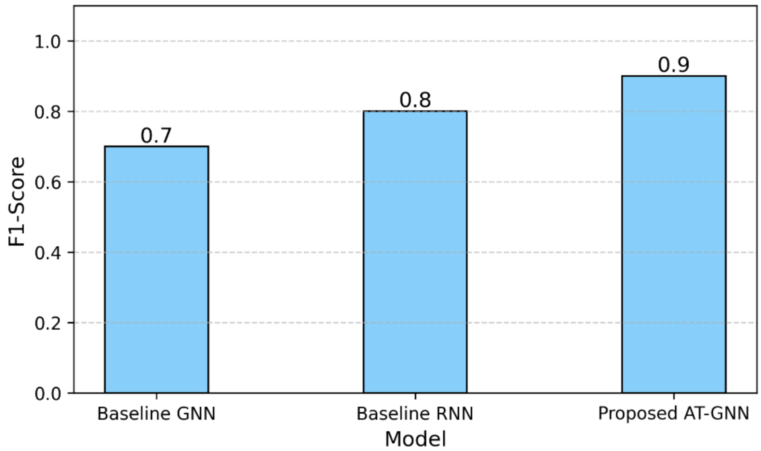



**Figure 6.** Accuracy of three community behavior models.

Because of the high accuracy of the AT-GNN, one can conclude that the model can offer accurate conclusions even when analyzing temporal dynamics and varying community structures in practical networks. The precision, recall and F1-score results add to the evidence that the model detects real communities efficiently and avoids false positives to the greatest extent. Therefore, it can be a valuable tool for analyzing emerging community behavior patterns in networks. The proposed AT-GNN model, validated on SNAP and DBLP datasets, achieves state-of-the-art performance in community behavior prediction in dynamic networks, as shown in Table 4.

**Table 4.** Comparative results of AT-GNN with existing literature.

| Study | Model/approach | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|---|
| [21] | Static GNNs | 0.78 | 0.76 | 0.72 | 0.74 |
| [27] | Temporal GNNs | 0.85 | 0.82 | 0.84 | 0.83 |
| [28] | RNNs | 0.82 | 0.80 | 0.81 | 0.80 |
| Proposed AT-GNN | AT-GNNs | 0.98 | 0.92 | 0.95 | 0.93 |

The proposed AT-GNN model yields superior performance throughout all the evaluation metrics, thereby illustrating AT-GNN as a robust method for analyzing the behavior of communities of a dynamic complex system. The accuracy achieved confirms the applicability of the developed method for tasks such as analysis of social networks, collaborative recommendations, and biological networks.

## 5. Conclusions

This paper presents AT-GNN, an innovative approach to fusing AI with network evolution theory to predict community behavior in complex systems. Compared with traditional models, the proposed model achieves higher accuracy, scalability, and temporal segmentation, attention mechanism, and RNN. Using the SNAP and DBLP datasets, the proposed AT-GNN had better accuracy (at 98%) and precision, recall, and F1-score than other methods. The findings support the model's structure and time flexibility and validate its application for social network analysis and factors affecting biosystems and collaboration patterns.

Despite the good performance in dynamic network analysis, the proposed AT-GNN model has drawbacks. The model assumes separate temporal points, which cannot correctly represent the continuously changing structure of some real-world networks. This could hinder its capacity to effectively represent temporal changes with sufficient detail. Furthermore, the model proved efficient with SNAP and DBLP datasets but may be unmanageable for other large-scale or real-time networks that require further memory usage and training time. Addressing these limitations would require incorporating methodologies such as temporal point processes or neural ordinary differential equations for continuous-time modeling and exploring distributed training and graph sampling techniques to enhance scalability. Moreover, incorporating explainability methods within the proposed model will also enhance it, enabling the understanding of factors driving community behavior. Extending the proposed research to other areas, especially communicating multi-layered and heterogeneous networking domains, will also prove the applicability of these research findings.

## Author Contributions

Yongyan Zhao: Conceptualization, writing-review and editing; Jian Li: writing-original draft preparation. All authors have read and approved the final version of the manuscript for publication.

## Use of Generative-AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

## Conflict of interest

We declare that there is no conflict of interest.

## Data availability Statement

The data can be obtained by contacting the corresponding author.

## References


1. M. Pósfai, A. L. Barabási, *Network science*, Cambridge: Cambridge University Press, 2016.
2. S. Fortunato, D. Hric, Community detection in networks: A user guide, *Phys. Rep.*, **659** (2016), 1–44. https://doi.org/10.1016/j.physrep.2016.09.002
3. Y. Wu, L. Pan, LSTEG: An evolutionary game model leveraging deep reinforcement learning for privacy behavior analysis on social networks, *Inform. Sciences*, 2024, 120842. https://doi.org/10.1016/j.ins.2024.120842
4. W. Jiang, J. Luo, Graph neural network for traffic forecasting: A survey, *Expert Syst. Appl.*, **207** (2022), 117921. https://doi.org/10.1016/j.eswa.2022.117921
5. P. Holme, J. Saramäki, Temporal networks, *Phys. Rep.*, **519** (2012), 97–125. https://doi.org/10.1016/j.physrep.2012.03.001
6. G. Rossetti, R. Cazabet, Community discovery in dynamic networks: A survey, *ACM Comput. Surv.*, **51** (2018), 1–37. https://doi.org/10.1145/3172867
7. Z. Qiu, Y. Yin, Y. Yuan, Y. Chen, Research on credit regulation mechanism of E-commerce platform based on evolutionary game theory, *J. Syst. Sci. Syst. Eng.*, 2024, 1–30. https://doi.org/10.1007/s11518-024-5603-2
8. A. L. Barabási, R. Albert, Emergence of scaling in random networks, *Science*, **286** (1999), 509–512. https://doi.org/10.1126/science.286.5439.509
9. P. Holme, B. J. Kim, Growing scale-free networks with tunable clustering, *Phys. Rev. E*, **65** (2002), 026107. https://doi.org/10.1103/PhysRevE.65.026107
10. A. L. Barabási, Z. N. Oltvai, Network biology: Understanding the cell's functional organization, *Nat. Rev. Gene.*, **5** (2004), 101. https://doi.org/10.1038/nrg1272
11. M. A. Saidu, L. S. Shamsudeen, A. K. Muhammad, A. Abdulkadir, Exploring E-commerce opportunities for a better international trading and tax revenue generation: A review for developing countries, *J. Sci. Technol. Educ.*, **10** (2022), 109–24.
12. Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, S. Y. Philip, A comprehensive survey on graph neural networks, *IEEE T. Neur. Net. Lear.*, **32** (2020), 4–24. https://doi.org/10.1109/TNNLS.2020.2978386
13. P. Goyal, S. R. Chhetri, A. Canedo, Dyngraph2vec: Capturing network dynamics using dynamic graph representation learning, *Knowl.-Based Syst.*, **187** (2020), 104816. https://doi.org/10.1016/j.knosys.2019.06.024
14. L. Akoglu, H. Tong, D. Koutra, Graph-based anomaly detection and description: A survey, *Data Min. Knowl. Disc.*, **29** (2015), 626–688. https://doi.org/10.1007/s10618-014-0365-y

15. Y. Jiang, B. Ma, X. Wang, G. Yu, P. Yu, Z. Wang, et al., Blockchained federated learning for internet of things: A comprehensive survey, *ACM Comput. Surv.*, **56** (2024), 1–37. https://doi.org/10.1145/3659099
16. A. A. Beni, A. Esmaeili, Biosorption, an efficient method for removing heavy metals from industrial effluents: A Review, *Environ. Technol. Inno.*, **17** (2020), 100503. https://doi.org/10.1016/j.eti.2019.100503
17. M. McPherson, L. S. Lovin, J. M. Cook, Birds of a feather: Homophily in social networks, *Annu. Rev. Sociol.*, **27** (2001), 415–444. https://doi.org/10.1146/annurev.soc.27.1.415
18. P. Holme, J. Saramaki, Temporal networks, *Phys. Rep.*, **519** (2012), 97–125. https://doi.org/10.1016/j.physrep.2012.03.001
19. G. Rossetti, M. Stella, R. Cazabet, K. Abramski, E. Cau, S. Citraro, et al., Y Social: An LLM-powered social media digital twin, *arXiv Preprint*, 2024. https://doi.org/10.48550/arXiv.2408.00818
20. D. C. Nguyen, Q. V. Pham, P. N. Pathirana, M. Ding, A. Seneviratne, Z. Lin, et al., Federated learning for smart healthcare: A survey, *ACM Comput. Surv.*, **55** (2022), 1–37. https://doi.org/10.1145/3501296
21. L. Cai, Z. Chen, C. Luo, J. Gui, J. Ni, D. Li, et al., *Structural temporal graph neural networks for anomaly detection in dynamic graphs*, In Proceedings of the 30th ACM International Conference on Information & Knowledge Management, 2021, 3747–3756. https://doi.org/10.1145/3459637.3481955
22. X. Zhu, Y. Zhang, H. Ying, H. Chi, G. Sun, L. Zeng, Modeling epidemic dynamics using graph attention-based spatial temporal networks, *Plos one*, **19** (2024), e0307159. https://doi.org/10.1145/3459637.3481955
23. H. Taherdoost, M. Madanchian, AI advancements: Comparison of innovative techniques, *AI*, **5** (2023), 38–54. https://doi.org/10.3390/ai5010003
24. D. Jin, Z. Yu, P. Jiao, S. Pan, D. He, J. Wu, et al., A survey of community detection approaches: From statistical modeling to deep learning, *IEEE T. Knowl. Data Eng.*, **35** (2021), 1149–1170.
25. J. L. A. A. Krevl. *Stanford Network Analysis Project*, Available from: http://snap.stanford.edu/data.
26. M. Ley. *Digital Bibliography and Library Project*, Available from: https://dblp.org/.
27. S. Min, Z. Gao, J. Peng, L. Wang, K. Qin, B. Fang, STGSN—a spatial-temporal graph neural network framework for time-evolving social networks, *Knowl.-Based Syst.*, **214** (2021), 106746. https://doi.org/10.1016/j.knosys.2021.106746
28. Y. R. Lin, Y. Chi, S. Zhu, H. Sundaram, B. L. Tseng, Analyzing communities and their evolutions in dynamic social networks, *ACM T. Knowl. Discov. D.*, **3** (2009), 1–31. https://doi.org/10.1145/1514888.1514891