ISSN: 2093-5374 / E-ISSN: 2093-5382

Proactive Content Caching at Mobile Edge Nodes Using Temporal Graph Attention Networks

N. Kirubakaran[1*], B. Saraswathi[2], Dr. Vimal Bibhu[3], Suvarna Milind Patil[4], Nimesh Raj[5], Dr.J. Janet[6], and Dr. Suman Sau[7]

[1*]Assistant Professor, Department of Electronics and Communication Engineering, Presidency University, Bangalore, Karnataka, India. kirubakaran.n@presidencyuniversity.in, https://orcid.org/0009-0008-9162-9328

[2]Assistant Professor, Department of Computer Science, Meenakshi College of Arts and Science, Meenakshi Academy of Higher Education and Research, Chennai, Tamil Nadu, India. saraswatib@maher.ac.in, https://orcid.org/0009-0009-4531-7507

[3]Department of Computer Science & Engineering, Noida international University, Greater Noida, Uttar Pradesh, India. vimal.bhibu@niu.edu.in, https://orcid.org/0000-0003-4456-0512

[4]Department of Engineering, Science & Humanities, Vishwakarma Institute of Technology, Pune, Maharashtra, India. suvarna.patil1@vit.edu, https://orcid.org/0009-0004-3612-4192

[5]Centre for Research Impact & Outcome, Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura, Punjab, India. nimesh.raj.orp@chitkara.edu.in https://orcid.org/0009-0002-1938-7754

[6]Department of Computer Science and Engineering Aarupadai Veedu Institute of Technology, Vinayaka Mission's Research Foundation (DU), Tamil Nadu, India. janet@avit.ac.in, https://orcid.org/0000-0001-6512-2146

[7]Associate Professor, Department of Computer Science and Information Technology, Institute of Technical Education and Research, Siksha 'O' Anusandhan (Deemed to be University), Bhubaneswar, Odisha, India. sumansau@soa.ac.in, https://orcid.org/0009-0009-9362-347X

Received: July 10, 2026; Revised: August 20, 2026; Accepted: September 07, 2026; Published: September 30, 2026

Abstract

The need for efficient content delivery has become even more critical due to the increased usage of mobile multimedia applications and latency-sensitive services. Caching methods like Least Recently Used (LRU) and Least Frequently Used (LFU) either cannot adapt to ever-changing user mobility patterns or do not efficiently use cached space, leading to increased overall network congestion. In response, this paper presents a proactive content-caching framework for mobile edge nodes that employs Temporal Graph Attention Networks (TGAT). This involves creating a dynamic temporal graph to model user behavior, their access to content, and how the edge node accesses that content. By utilizing a temporal attention mechanism, this paper seeks to predict user behavior and how to pre-emptively cache content at edge nodes. The proposed method was tested in a simulated mobile edge computing environment comprising 500 mobile users, 50 edge nodes, 2,000 content items, and 10,000 content requests. The proposed TGAT-based architecture achieved a cache hit ratio of 91.4%, while the cache hit ratios for LRU, LFU, and LSTM-based caching were 72.8%, 76.5%, and 86.2%,

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), volume: 17, number: 3 (September-2026), pp. 2121-2137. DOI: *10.58346/JOWUA.2026.I3.123*

*Corresponding author: Assistant Professor, Department of Electronics and Communication Engineering, Presidency University, Bangalore, Karnataka, India.

respectively. In addition, the average time to complete content requests with the proposed method was 18.7 ms, compared with 34.5 ms, 31.2 ms, and 22.9 ms for the other caching methods. The proposed method achieved a backhaul traffic reduction of 41.6% compared to the baseline network without caching. These results indicate that the use of Temporal Graph Attention networks, combined with a proactive approach to cache placement, provides a significant improvement in the efficiency of content delivery within next-generation 5G/6G edge computing networks.

Keywords: Mobile Edge Computing, Proactive Content Caching, Temporal Graph Attention Network, Edge Intelligence, Cache Hit Ratio, Content Popularity Prediction, 5G/6G Networks.

1 Introduction

The emergence of mobile connectivity, video streaming, augmented reality services, connected vehicles, and IoT applications has resulted in exponential growth in the amount of data processed at the network edge. MEC is being viewed as a viable solution for handling this data by bringing compute and storage resources closer to users, thereby improving service response times and eliminating transmission delays. The success of the MEC model depends on the ability of edge nodes to cache data that will be frequently requested by users before make such requests (Hou et al., 2025). Traditional caching strategies like LRU/LFU struggle to adapt to rapidly changing user patterns because rely on historical usage data. In wireless networks, content popularity is constantly evolving and location-dependent; as a result, cache replacement efficiency is reduced, and backhaul traffic increases (Zhu et al., 2022). Graph-based learning approaches can more accurately model interactions among users, content, and network elements, capturing complex relationships that existing prediction models cannot (Zhu et al., 2024; Kianfar et al., 2024). Similarly, the use of temporal attention mechanisms to identify patterns of demand for rapidly changing content will increase the accuracy of making proactive caching decisions (Jiang et al., 2024).

While there have been several recent machine learning/deep learning mechanisms for proactive cache algorithms, many limitations remain in the existing algorithms. Many reinforcement learning-based approaches focus on cache replacement but require extensive training and exploration overhead (Wu et al., 2024). Approaches that utilize spatio-temporal graph convolutions can improve demand prediction accuracy; however, often limited by their inability to model long-range temporal dependencies and dynamic user-to-content interactions (Lian et al., 2022). Recent models using graph attention frameworks have achieved superior results in predicting popularity due to the dynamic importance weights assigned to neighboring nodes (Zhu et al., 2022; Kianfar et al., 2024). However, many current models fail to jointly account for the influence of user mobility, content evolution, and edge-node interactivity, resulting in incomplete representations of the actual MEC environment (Zhu et al., 2024; Xiao et al., 2024). Emerging research has also suggested that collaborative edge caching requires richer graph representations of the relationships among temporal, spatial, and contextual dependencies across the various distributed constituents of the network (Cui et al., 2025; Liu et al., 2025).

The proposed TGAT-based proactive caching system in a Mobile Edge Computing environment is described in figure 1. The framework starts by gathering user requests and temporal interactions, then converts them into a temporal graph representation of how users, content, and edge nodes are dynamically related. Afterward, the Temporal Graph Attention Network (TGAT) analyzes these interactions to predict future content popularity.

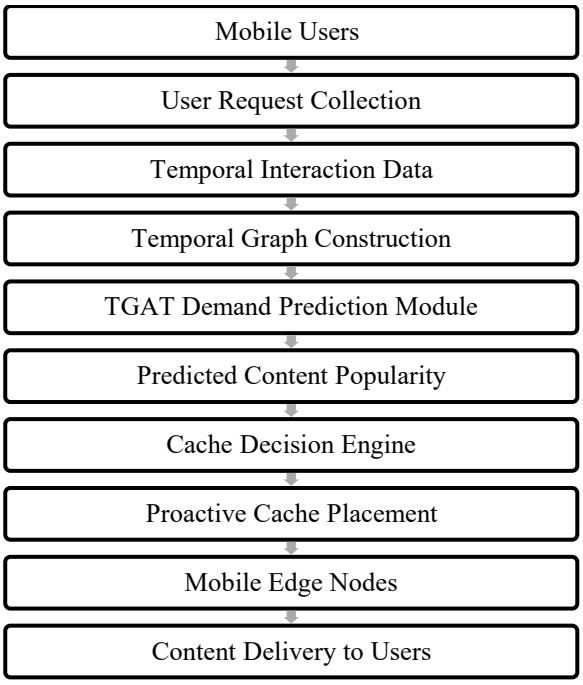


Figure 1: Overall architecture of the TGAT-Based proactive content caching framework

It uses this information to determine which items will be in high demand for the cache decision engine. The content items predicted to be highly popular will then be placed in proactive caches at mobile edge nodes, so that frequently requested content is stored closer to users. The advantage of such an architecture is evident when accessing content, as it greatly reduces latency, improves cache hit ratio, and minimizes backhaul traffic utilization by serving most content requests at the edge cache.

This paper presents a proactive caching method using temporal graph attention networks (TGAT) for mobile edge networks to address the above challenges. A dynamic temporal graph is constructed, where simultaneity among users, content, and edge nodes is captured in a mobile edge content popularity prediction model, and multi-temporal attention, which predicts content demand, is also utilized to prioritize requests to place data in cache. In this way, proposed proactive content caching strategy is able to exploit user mobility features, content evolution features, and edge-node connectivity features together into a unified architecture that predicts demand ahead and improves cache efficiency.

The proposed framework brings together a single, unified temporal graph attention-driven caching strategy to combine demand prediction with cache placement in the same architecture. The proposed framework hopes to improve the efficacy of cache hits, reduce latency for accessing resources, and improve the efficiency of resource utilization for the mobile edge computing environment through joint modeling of temporal dynamics and interaction with the network. The efficient caching of content has a direct impact on latency, backhaul congestion, and overall quality of service in future 5G/6G mobile edge networks. The development of intelligent caching solutions is an important aspect of scaling and ensuring the reliability of wireless infrastructures for mobile content consumption.

In the remainder of this paper, Section II presents the literature review of mobile edge caching strategies and machine learning-based caching strategies and the research gap that this work fills. Section III elaborates the proposed TGAT-based proactive caching framework which comprises the system model, the temporal graph attention-based demand prediction and the cache placement strategy. Section IV shows the experiment settings and performance comparison with state-of-the-art caching approaches.

Section V concludes by examining the significance of the result, its scalability, and potential implementation issues.

2 Literature Review

The recent concept of MEC aims at reducing latency of content delivery by keeping user-centric content at cache deployed at edge servers near to end users. Initially, content caching was mainly dependent on reactive algorithms, in which data is cached after user makes a request to access any particular content. Although reactive caching algorithms are simple to implement, the access latency and backhaul utilization could be increased significantly during unexpected bursty traffic (Aghazadeh et al., 2023). To tackle with these issues, popularity-based approaches have been developed to predict demand of content based on historical access sequence and cache popular data contents prior to its access (Feng et al., 2023) (Zhang et al., 2023). These models improve the cache hit ratio however it suffers a lot from dynamic change in popularity. Cooperative MEC has also been used to enhance the cache utilization across multiple MEC Servers. In cooperative MEC architectures, multiple MEC Servers in the neighbourhood can cache content and manage storage by coordinating the caching decisions for sharing data and avoid content replication (Li et al., 2023). Researchers also added intelligence in cache placement on an edge gateway and distributed caching infrastructure for optimal resource utilization and traffic management in MECs (Sun et al., 2023). These advances signify the importance of coordinated caching policies for MEC environments.

Over recent times, several Machine learning (ML) techniques are used to deal with dynamic and rapid change of demand for any content. For example, in Reinforcement learning (RL), an agent learns experimentally in the environment and adjust the policy for optimizing long-term demand; actor-critic frameworks have been explored for optimized cache replacement and for efficiently allocate resource in MECs Jiang et al., (2021) and Zhao et al., (2021), while an agent's behaviour cannot be perfectly adapted to highly dynamic environment. Similarly, models based on sequence learning like Long Short-Term Memory (LSTM) and Recurrent Neural Networks (RNN) are developed to capture scenarios involving temporal dependencies between users' requests and cache data items, however it fails in modeling structural interactions between users, content, and edge servers in a network (Aghazadeh et al., 2023). Inspired by this observation, Graph neural networks (GNN)have shown excellent capabilities of capturing complex interactions and relationships in large scale complex graphs and have also been used in the estimation of future traffic in MEC networks (Wei et al., 2025; Zhang et al., 2025; Rao et al., 2025). GNN models such as graph attention and graph convolution offer more effective spatio-temporal relationships and are widely used for improving performance in MECs (Li et al., 2023).

Based on the aforementioned state-of-the-art work, reactive caching strategies and popularity-based strategies exhibit low overhead but cannot sufficiently deal with users' mobility and evolving usage patterns. On the other hand, reinforcement learning offers adaptive capabilities, however, may incur overhead due to extensive training period and instability for dynamically changing network traffic conditions. Furthermore, sequence learning models like LSTM and RNN only considers temporal features of content access sequence and fails to capture the overall network topological structure among MEC nodes, users and content items. While recent approaches utilising, graphs capture spatial feature among these entities but mostly fail to capture the underlying temporal evolution along with graph interaction. Given the content popularity is jointly affected by mobility of users, popularity evolution and interconnection between MEC devices and servers, a comprehensive approach that can precisely learn temporal evolution with graph relation modelling is required. Therefore, in this work have designed a framework based on Temporal Graph Attention Network (TGAT) that dynamically learns and adapts

graph structure alongside with the temporal dependencies to accurately predict the popularity of contents.

Existing MEC caching solutions have focused only on spatial relations among MEC network entity or time dimension, but failed to jointly modeling the spatial structure with the temporal sequence. To that effect, propose to use a TGAT based caching frame to exploit the combined space-time features to enhance mobile edge caching performance.

3 Methodology

3.1 System Model and Problem Formulation

The proposed framework considers a Mobile Edge Computing (MEC) environment consisting of a set of mobile users $U = \{u_1, u_2, \dots, u_n\}$, edge nodes $E = \{e_1, e_2, \dots, e_m\}$, and content objects $C = \{c_1, c_2, \dots, c_k\}$. Mobile users generate content requests that are served either from nearby edge caches or from remote cloud content servers through backhaul communication links. Each edge node possesses finite storage resources and maintains a local cache for storing popular content. Let x_{ij}denote the cache placement decision for content $c_j$at edge node e_i. The binary decision variable is defined as:

$$x_{ij} = \begin{cases} 1, & \text{if content } c_j \text{ is cached at edge node } e_i \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

As shown in equation (1), content placement decisions are constrained by the storage capacity of each edge node. Therefore, the cache capacity constraint is expressed in equation (2):

$$\sum_{j=1}^{k} s_j \, x_{ij} \leq S_i, \forall e_i \in E \quad (2)$$

Where s_j represents the size of content c_j and S_i denotes the maximum storage capacity of edge node e_i. The objective of the proposed model is to maximize cache efficiency while minimizing network delay and backhaul traffic. The optimization objective is formulated as:

$$\max \; F = \alpha H - \beta L - \gamma B \quad (3)$$

Where H represents cache hit ratio, L denotes average content access latency, B indicates backhaul bandwidth consumption, and α, β, and γ are weighting coefficients controlling the relative importance of each objective. Equation (3) guides the cache placement process toward balanced performance optimization.

In the proposed TGAT-based proactive caching framework, the operational workflow is depicted in figure 2. The first step in the workflow is collecting and preprocessing user request data, then building a temporal graph from the input data, and further generating node features that represent the user content interactions. The predicted popularity of future content items will then be used to rank content items for cache placement optimization. Content items identified for placement will be stored proactively at edge nodes to enable quicker delivery of those content items to users. Finally, the performance of cached content will continually be monitored and updated to reflect changing user demand patterns, thus improving cache hit ratios, decreasing access latencies, and limiting the amount of backhaul traffic in MEC environments.

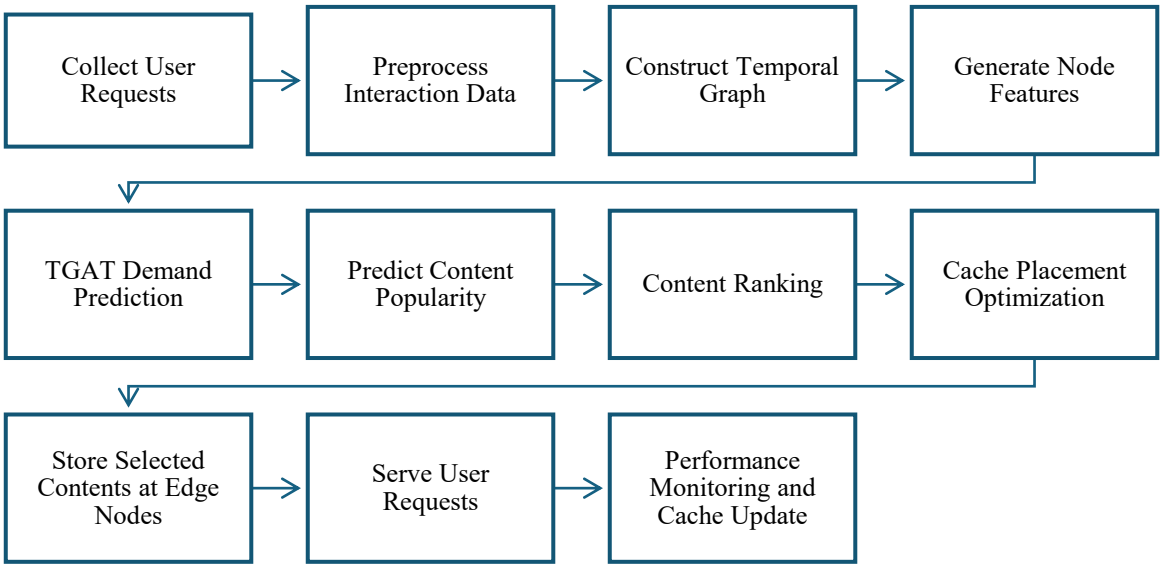


Figure 2: Workflow of TGAT-Based demand prediction and proactive cache placement

3.2 Temporal Graph Attention Network-Based Demand Prediction

To illustrate changing demand pattern in dynamic contents, the MEC scenario is characterized by a temporal graph $G_t = (V_t, E_t)$, wherein vertices refer to the users, contents and edge nodes; edges correspond to interactions at time slot t. The main advantages of dynamic graphs are ability to reflect time-varying patterns of user-content associations and mobility behavior over time. Each node has a feature vector that captures request frequency, access recency, user mobility status and content features, and interaction intensity between any two directly-connected vertices is represented via weighted edges. For a vertex v_i, the attention weight of neighbor vertex v_j is computed using:

$$a_{ij} = \frac{\exp(\phi(h_i, h_j, \Delta t))}{\sum_{k \in N(i)} \exp(\phi(h_i, h_k, \Delta t))} \qquad (4)$$

Where $h_i$and h_j represent node feature embeddings and $N(i)$ represents the neighborhood of node i, Δt is the temporal distance between interactions. According to equation (4) which uses temporal attention and emphasizes more recent and important interactions, the node representation will be derived and then passed to predictor to forecast future content request. Higher prediction result means more potential request in the future. Hence the popular content can be cached beforehand.

3.3 Proactive Cache Placement Framework

The proactive cache placement framework is made of three parts: predicting demand, generating cache decisions and deploying caches. The first step is the prediction of the popularity of all content items by the TGAT module. The predicted popularity values of content items are provided to the cache decision engine which uses them to locate available storage and evaluate the resource efficiency of each content item. Each content item is assigned a score that reflects the predicted resource efficiency and predicted popularity. Content items with higher scores are assigned more cache space compared to lower scored items.

The cache decision engine will always be updated with new request patterns and thus continue to provide the best cache placement decision based on what has been learned from previous caches. The

overall process begins by gathering logs at the edge nodes from users, converting these into temporal graph-based structures and applying the logs to the TGAT model for predicting content demand. Next, the estimated demand will be used by the cache management engine to select the best-ranked content items to place in the cache while meeting all capacity constraints as defined in equation (2). Finally, the content selected for caching will be proactively put in cache in edge nodes will increase cache hit percentage and decrease service latency.

Algorithm 1: TGAT-Based Proactive Content Caching

Input:

User request logs R

Edge nodes E

Content set C

Cache capacities S

Output:

Optimized cache placement X

1. Collect user-content interaction records
2. Construct temporal graph Gt (V, E)
3. Extract node and edge features
4. Train TGAT model on temporal graph
5. Predict future popularity score P(cj)
6. Sort contents in descending order of P(cj)
7. For each edge node ei do
8. Initialize remaining capacity Si
9. For each ranked content cj do
10. If size(cj) ≤ available capacity then
11. Cache content cj
12. Update capacity
13. End If
14. End For
15. End For
16. Deploy cache placement decisions
17. Monitor requests and update graph
18. Repeat prediction and placement cycle

Return X

This algorithm is designed to solve the proactive caching content problem in Mobile Edge Computing (MEC) by dynamically intelligent caching, based on user and content interaction prediction from Temporal Graph Attention Network (TGAT). First, collect user content interaction data. A dynamic

graph, representing dynamic request behaviors, is built on it. Second, predicts popularity indices of content items from the TGAT model in future time slots, by rank ordering, the Top ranked contents are pre-positioned to MEC nodes to improve the hit ratio and reduce access latency. Besides dynamic update is proposed if new interaction data becomes available. The algorithm is integrated into a proactive caching paradigm utilizing temporal graph learning for reliable demand forecasting and efficient caching policy optimization in complex and time-varying MEC systems.

4 Results

4.1 Experimental Set up and Data Set Description

It evaluate designed TGAT based proactive caching framework in a simulation model of a Mobile Edge Computing (MEC) system to model real-time content delivery.500 mobile clients,50 edge nodes and 1 content server are implemented in the simulation and use a centralized content collection, which contains 2000 content and produces 10000 requests over different periods, including mobile clients as well as user demand models for different popular contents.

4.1.1 Software Details

Table 1: Software environment and implementation details

Component	Specification
Operating System	Ubuntu 22.04 LTS
Programming Language	Python 3.11
Deep Learning Framework	PyTorch 2.2
Graph Processing Library	PyTorch Geometric
Simulation Tool	NetworkX-based MEC Simulator
Hardware	Intel Core i9, 32 GB RAM, NVIDIA RTX GPU

Table 1 summarizes the programming language, environment, software tools and hardware utilized to build and implement proposed TGAT-based proactive caching method. Combining the powerful capabilities of Python with PyTorch and PyTorch Geometric framework yields a flexible and capable infrastructure for spatio-temporal graph representation and learning, offering the means to support fast model training and to accurately simulate and analyze system behaviors.

4.1.2 Dataset Details

Table 2: Dataset characteristics and feature description

Parameter	Value
Total Users	500
Edge Nodes	50
Content Objects	2,000
Request Records	10,000
Time Snapshots	24
Features per Node	8
Graph Type	Dynamic Temporal Graph

The data set used for assessing this system is detailed in table 2 including entity types counts such as: users, edge nodes, and content items; as well as other graph-based features such as the total request records count, graph edges count, dimension of edge features, and dimension of node features. The

characteristics of data that discussed demonstrate some nuances that are characteristic for dynamic Multi-access Edge Computing (MEC) environment and contain dynamics in time-dependent behavior, crucial for prediction caching model design. Each of graph nodes was defined by: requesting frequency, content staleness, user mobilities index, edges connection level, edge capacity utilization rate, content item's content-type, interaction time period, and item's popularity level value.

4.1.3 Parameter Initialization

Table 3: Parameter initialization for TGAT-Based caching framework

Parameter	Symbol	Value
Learning Rate	η	0.001
Batch Size	B	64
Hidden Dimension	H	128
Attention Heads	A	4
Epochs	E	100
Cache Capacity	S	100 GB
Discount Factor	γ	0.95
Prediction Window	T	24 h

In table 3, the initial parameters settings in training and deploying phases of the TGAT based caching system have been presented. The configuration parameters included learning rates, batch size, dimension of hidden layers, the number of heads of attention mechanisms and cache capacities. These parameters selected by careful consideration so as to encourage the stable training performance, the reliable predicted precision as well as the efficiency in the common deployment for the MEC environments.

4.2 Performance Evaluation

The evaluation metrics considered the following key criteria: caching effectiveness, time lag for request, network congestion prevention. cache hit ratio (CHR) is defines as the number of caches served request compared to the number of total accesses to server. The calculation is following equation (5):

$$CHR = \frac{N_{hit}}{N_{total}} \times 100 \qquad (5)$$

Where N_{hit}denotes successful cache hits and N_{total}represents total content requests. The Average Content Access Latency (ACL) is computed as:

$$ACL = \frac{\sum_{i=1}^{N} L_i}{N} \qquad (6)$$

Where $L_i$is the delay experienced by request $i$and $N$is the total number of requests. The Backhaul Traffic Reduction (BTR) metric evaluates the decrease in traffic transmitted through cloud-backhaul links:

$$BTR = \frac{B_{base} - B_{prop}}{B_{base}} \times 100 \qquad (7)$$

Where $B_{base}$and B_{prop}denote baseline and proposed backhaul traffic volumes, respectively.

Table 4: Cache hit ratio comparison

Method	CHR (%)
LRU	72.8
LFU	76.5
RL-Based Caching	84.7
LSTM-Based Caching	86.2
Proposed TGAT	91.4

Table 4 illustrates Cache Hit Ratio Comparison of Various Caching Methods table 4 illustrates the Cache Hit Ratio (CHR) obtained using different caching algorithms. The proposed TGAT-based framework gains the highest cache hit ratio, 91.4%, significantly outperforming LRU (72.8%), LFU (76.5%), RL-based caching (84.7%), and LSTM-based caching (86.2%). These experimental results show that temporal graph attention learning could effectively predict the demand for content and accordingly effectively utilized the cached items in the network.

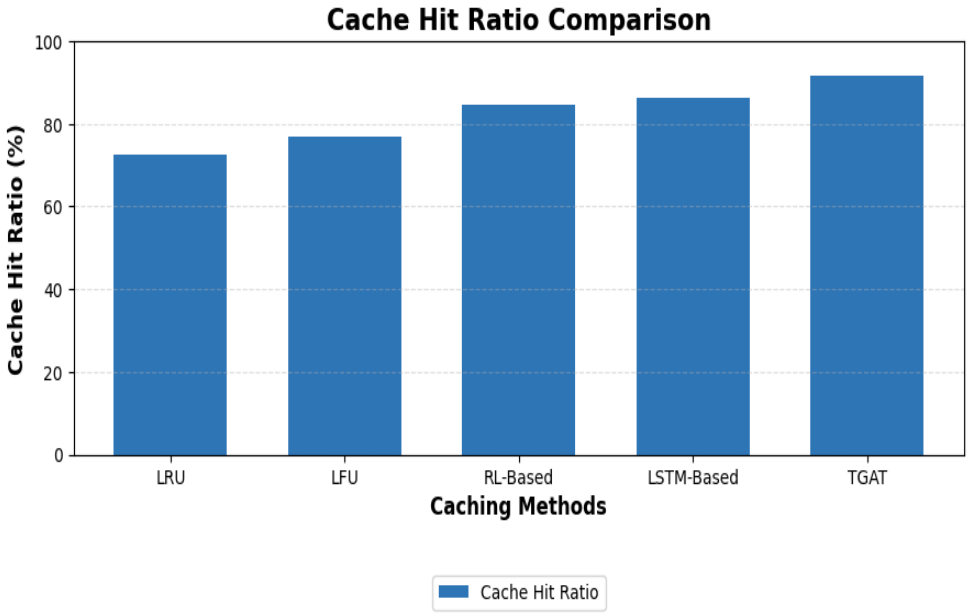


Figure 3: Cache hit ratio comparison among caching methods

Figure 3 displays the performance results based on the cache hit ratio for various caching strategies, including proposed TGAT-based approach. This framework successfully yields the maximum hit ratio, which is 91.4% and is better than LRU, LFU, RL-based and LSTM-based techniques. Such high results reflect the advantages of learning the temporal structure of content access patterns to predict future demand, and proactively place content in the edge caches.

Table 5: Average content access latency

Method	Latency (ms)
LRU	34.5
LFU	31.2
RL-Based Caching	25.4
LSTM-Based Caching	22.9
Proposed TGAT	18.7

The average latency of content access among various caching approaches is reported in table 5. The TGAT-based framework gains the lowest access latency, 18.7 ms. This results clearly demonstrate method significantly outperforms the LRU (34.5 ms), LFU (31.2 ms), RL-based caching (25.4 ms), and LSTM-based caching (22.9 ms) methods for content delivery at the edge server, which is mainly due to its efficient proactively cache provisioning at edge servers.

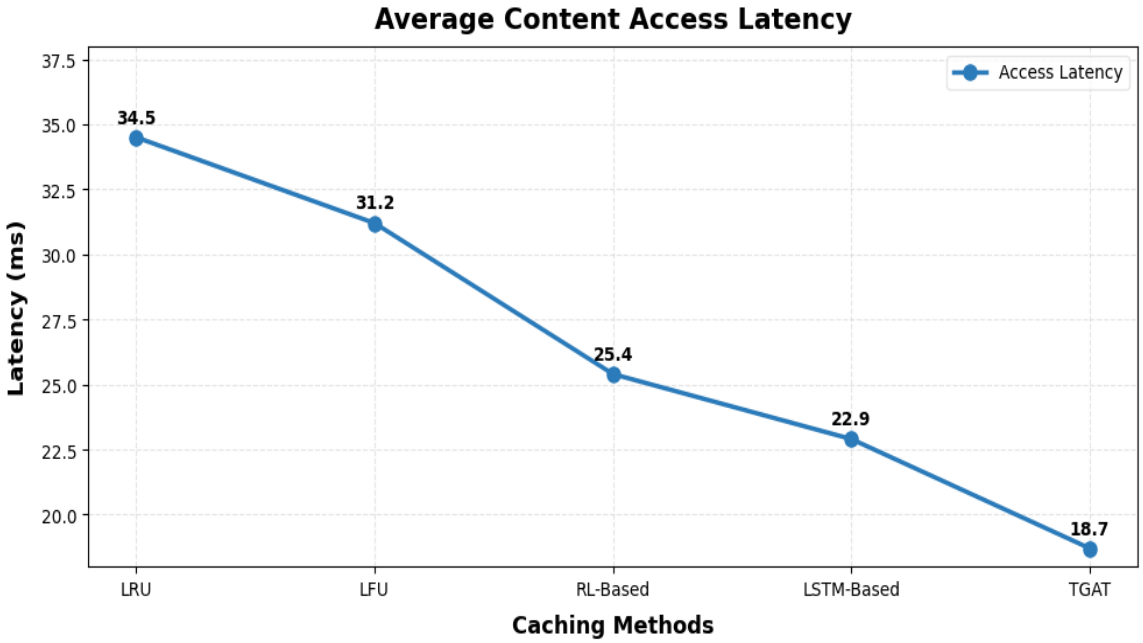


Figure 4: Average content access latency of different caching approaches

Figure 4 illustrates the average content access latency of considered different caching algorithms. As observed, TGAT method brings down the average content access latency to 18.7 ms, which is lower than traditional and various learning-based approaches. Lower content access latency is achieved by learning content importance in the graph and performing cache decisions proactively in an optimal way which avoids off-line fetches from cloud-level.

Table 6: Backhaul traffic reduction

Method	Reduction (%)
LRU	12.6
LFU	16.8
RL-Based Caching	31.5
LSTM-Based Caching	36.4
TGAT	41.6

Table 6 presents the performance in terms of Backhaul Traffic Reduction of different caching algorithms. TGAT-based framework attains the highest backhaul traffic reduction, 41.6%. This significant reduction of the dependency on the remote cloud server reveals the effectiveness of content predicting at the edge in minimizing network congestion and network bandwidth consumption.

Figure 5 shows the comparison of the achieved backhaul traffic reduction percentages of the reviewed caching approaches. The TGAT approach achieves significant reduction of up to41.6% compared with traditional and other learning methods. The proposed TGAT based caching can efficiently make intelligent decisions and cache appropriate content to mitigate backhaul traffic with great margin.

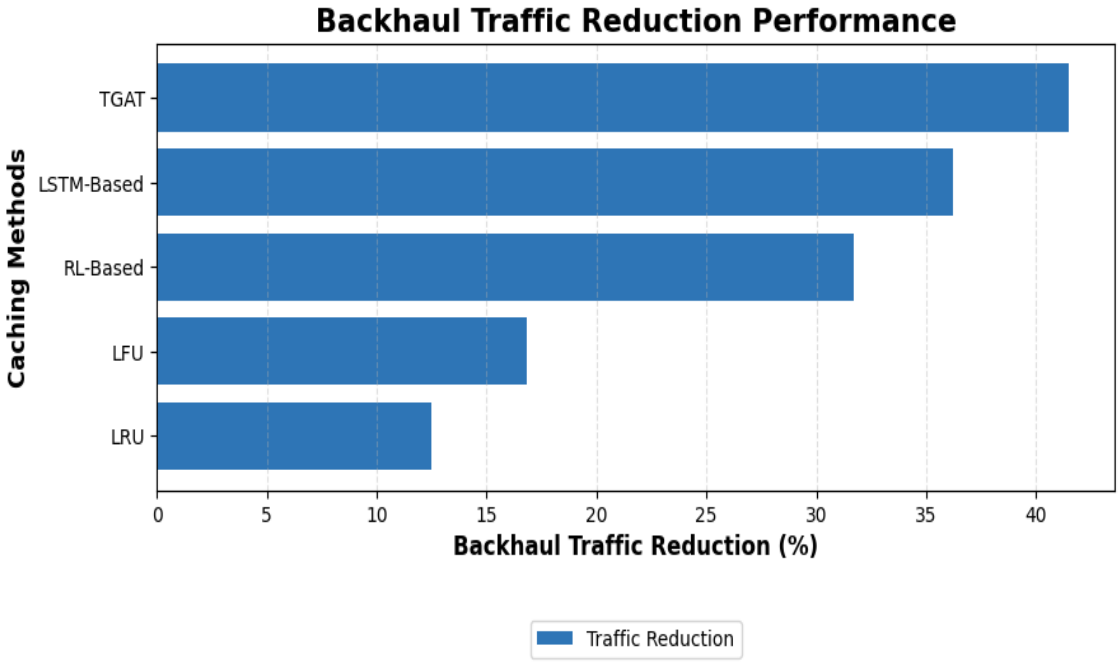


Figure 5: Backhaul traffic reduction achieved by different caching methods

4.3 Comparative Analysis

Comparative evaluation highlights the advantages of integrating temporal graph learning with cache placement optimization. LRU and LFU methods rely exclusively on historical access statistics and cannot effectively adapt to rapidly changing popularity patterns. Although RL-based caching improves adaptability through policy learning, performance remains constrained by exploration overhead and training complexity. LSTM-based caching captures temporal request evolution but lacks explicit modeling of user-content-edge relationships. In contrast, the TGAT framework simultaneously learns temporal dependencies and graph interactions, resulting in more accurate popularity estimation and proactive content placement decisions. Improvements of 5.2% over LSTM and 6.7% over RL-based caching in cache hit ratio validate the effectiveness of the proposed architecture.

4.4 Ablation Study

To evaluate the contribution of individual components, three model configurations were examined.

Table 7: Ablation study on the TGAT framework components

Configuration	CHR (%)	Latency (ms)
Without Temporal Attention	84.9	24.8
Without Graph Attention	82.7	27.1
Full TGAT Framework	91.4	18.7

Table 7 provides the empirical evidence that various components of the TGAT framework contribute significantly to its performance. When only the graph attention module is removed from the proposed TGAT framework, its CHR drops to 84.9%. If further removes the graph attention mechanism from the TGAT framework, the cache hit ratio drops further to 82.7%. Full TGAT model reaches the highest cache hit ratio of 91.4% and the lowest latency of 18.7 ms, showing that both attention mechanisms, graph attention and temporal attention are indispensable for optimal performance.

The absence of temporal attention reduced prediction accuracy, while removing graph attention weakened the ability to model network relationships. The complete TGAT framework consistently produced the best performance, confirming that both temporal dependency learning and graph attention mechanisms contribute significantly to caching efficiency. Overall, the obtained results demonstrate that the proposed proactive caching strategy effectively improves cache utilization, lowers access latency, and reduces backhaul traffic in dynamic MEC environments.

5 Discussion

The empirical results show that learning temporal dependency significantly boost the proactive caching performance for mobile edge computing. Integrating temporal attention in the graph learning procedure, the novel TGAT frame work captures dynamic content popularities and user's request behavior at distinct time stamps, resulting in more accurate content demand prediction compared with typical methods. And indeed, the prediction capability is evidenced by the cache hit ratio 91.4%, which outperforms LRU, LFU, RL-based and LSTM-based algorithms. The model is able to learn adaptability towards user interest changes and their mobility patterns, to adapt content caches in advance such that cache hits are maximized before user's demand change impact service quality negatively. To reflect this, the performance evaluation in the ablation study demonstrates that without temporal or graph attention, performance decline in cache hits is obvious and latency for accessing contents becomes much higher. Considering its scalability, TGAT can be deployed over the distributed MEC infrastructure, since cache decisions are localized at edge servers based on content demands, which rely on the predicted popularity of contents. However, graph attention computation imposes overhead more compared with static caching methods, but the avoidance of redundant data transmission leads to reduced network burden, enhanced bandwidth efficiency, and limited traffic in the backhaul that between edge servers and centralized servers, which make TGAT very attractive for data intensive 5G and 6G services such as UHD video stream, autonomous vehicles, immersive reality, and scale of things networks, where low latency and efficient utilize of bandwidth are key demands. Despite many positive features of TGAT, some limitations persist and can inspire future research directions. In extremely dynamic networks where new nodes constantly appear and vanish and content accesses change rapidly, on-line updating dynamic graph structure and re-training models continuously can raise resource consumption and complexity. In particular, how to efficiently manage computing resources while maintain prediction accuracy across the entire network at very large scale is also an issue. Further development may focus on more lightweight version of TGAT architecture, using federated learning based for coordinated content placement over distributed edge nodes, and more efficient caching methods such as cooperative content delivery between edge networks.

6 Conclusion

In this study, Temporal Graph Attention Network (TGAT)-based proactive content caching approach for MEC networks to optimize content delivery performance in dynamic user demands. By mapping user-content-edge interaction to temporal graph and utilizing attention-based learning method for prediction of future content popularity, the presented approach tackles the deficiency of traditional reactionary caching schemes and popularity-aware caching schemes. Experimental evaluations were performed on a simulated network involving 500 mobile users, 50 edge nodes and 10,000 content requests, revealing consistent performance gains in all evaluation metrics: Cache hit ratio of 91.4% compared to that of LRU (72.8%), LFU (76.5%), RL-based cactheyhing (84.7%), and LSTM-based caching (86.2%). Average content access latency of 18.7 ms compared to LRU (34.5 ms) and LSTM-based caching (22.9

ms), and savings of backhaul traffic of 41.6%. It indicates the effectiveness of the temporal dependency learning method together with graph attention in order to realize user prediction and intelligent proactive caching decision. This study also contributes to content caching by enabling intelligent caching exploiting temporal evolution and structural relationships in the MEC network at the same time and promising low latency and reliable content delivery for burgeoning mobile/wireless communication services utilizing accessing abundant contents rapidly. For future work, research on the combination with federated learning for decentralizing model training in the edge, collaborative caching between various edge locations and energy-efficiency and privacy-protecting Caching can further increase accuracy of prediction, utilization of resources, safety for users and reliability of delivered content for content providers for content storing distributed at the edge sites.

References

[1] Aghazadeh, R., Shahidinejad, A., & Ghobaei-Arani, M. (2023). Proactive content caching in edge computing environment: A review. *Software: Practice and Experience*, *53*(3), 811-855. https://doi.org/10.1002/spe.3033

[2] Cui, Z., Hou, J., & Nayak, A. (2025, December). A gcn and recommender-based approach for optimizing edge caching performance in IoV. In *Globecom 2025-2025 IEEE Global Communications Conference* (pp. 4712-4717). IEEE. https://doi.org/10.1109/GLOBECOM59602.2025.11432752

[3] Feng, B., Feng, C., Feng, D., Wu, Y., & Xia, X. G. (2023). Proactive content caching scheme in urban vehicular networks. *IEEE Transactions on Communications*, *71*(7), 4165-4180. https://doi.org/10.1109/TCOMM.2023.3277530

[4] Hou, J., Elhadef, M., & Nayak, A. (2025). Intelligent edge caching strategies for optimized content delivery. *IEEE Transactions on Network Science and Engineering*, *13*, 3580-3595. https://doi.org/10.1109/TNSE.2025.3637123

[5] Jiang, F., Wang, W., Liu, L., Zhai, Y., & Wang, H. (2024, June). AoI-Based Temporal Graph Attention Network for Content Update. In *2024 IEEE 99th Vehicular Technology Conference (VTC2024-Spring)* (pp. 1-6). IEEE. https://doi.org/10.1109/VTC2024-Spring62846.2024.10683469

[6] Jiang, W., Feng, D., Sun, Y., Feng, G., Wang, Z., & Xia, X. G. (2021). Proactive content caching based on actor–critic reinforcement learning for mobile edge networks. *IEEE Transactions on Cognitive Communications and Networking*, *8*(2), 1239-1252. https://doi.org/10.1109/TCCN.2021.3130995

[7] Kianfar, G., Hajiakhondi-Meybodi, Z., & Mohammadi, A. (2024, November). Temporal dual-attention graph network for popularity prediction in mec networks. In *2024 IEEE 10th World Forum on Internet of Things (WF-IoT)* (pp. 487-492). IEEE. https://doi.org/10.1109/wf-iot62078.2024.10811363

[8] Li, D., Zhang, H., Li, T., Ding, H., & Yuan, D. (2023). Community detection and attention-weighted federated learning based proactive edge caching for D2D-assisted wireless networks. *IEEE Transactions on Wireless Communications*, *22*(11), 7287-7303. https://doi.org/10.1109/TWC.2023.3249756

[9] Lian, L., Chen, N., Ou, P., & Yuan, X. (2022, May). Mobile edge cooperative caching strategy based on spatio-temporal graph convolutional model. In *2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD)* (pp. 1396-1401). IEEE. https://doi.org/10.1109/CSCWD54268.2022.9776305

[10] Liu, Z., Zhang, W., Hao, H., & Shi, H. (2025, October). Optimizing Collaborative Edge Caching in IoT via Social-Aware Spatio-Temporal Prediction and Multi-agent Reinforcement Learning. In *International Conference on Algorithms and Architectures for Parallel Processing* (pp. 55-74). Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-95-8414-7_4

[11] Rao, G. R., Priya, L. R., Rupani, R. K., & Dashrathsinh, Z. D. (2025). Edge resource scaling in B5G/6G networks using diffractive deep sparse graph attention neural network with orchard algorithm. *Iran Journal of Computer Science*, *8*(4), 2497-2512. https://doi.org/10.1007/s42044-025-00326-9

[12] Sun, H., Chen, Y., Sha, K., Huang, S., Wang, X., & Shi, W. (2023). A proactive on-demand content placement strategy in edge intelligent gateways. *IEEE Transactions on Parallel and Distributed Systems*, *34*(7), 2072-2090. https://doi.org/10.1109/TPDS.2023.3249797

[13] Wei, Z., Yu, G., Lyu, Z., Zhu, C., Qiao, Y., Yuan, X., & Feng, L. (2025, July). Collaborative Edge Caching Approach Based on Multi-agent Graph Attention Reinforcement Learning in Unreliable Networks. In *International Conference on Intelligent Computing* (pp. 122-133). Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-96-9894-3_11

[14] Wu, C., Xu, Z., He, X., Lou, Q., Xia, Y., & Huang, S. (2024). Proactive caching with distributed deep reinforcement learning in 6G cloud-edge collaboration computing. *IEEE Transactions on Parallel and Distributed Systems*, *35*(8), 1387-1399. https://doi.org/10.1109/TPDS.2024.3406027

[15] Xiao, L., Lin, Y., Zhang, Y., Li, J., & Shu, F. (2024, June). AoI-Aware Energy-Efficient Vehicular Edge Computing Using Multi-Agent Reinforcement Learning with Actor-Attention-Critic. In *2024 IEEE 99th Vehicular Technology Conference (VTC2024-Spring)* (pp. 1-6). IEEE. https://doi.org/10.1109/VTC2024-Spring62846.2024.10683441

[16] Zhang, P., Li, J., Ding, D., Wang, H., Wang, K., & Wang, X. (2025). A Spatial-Temporal Graph Convolutional Network with Self-Attention for City-Level Cellular Network Traffic Prediction. *IEEE Transactions on Network Science and Engineering*, *13*, 3285-3300. https://doi.org/10.1109/TNSE.2025.3629133

[17] Zhang, Y., Wang, R., Wang, Y., Chen, M., & Guizani, M. (2023). Diversity-driven proactive caching for mobile networks. *IEEE Transactions on Mobile Computing*, *23*(7), 7878-7894. https://doi.org/10.1109/TMC.2023.3340733

[18] Zhao, Y., Li, R., Wang, C., Wang, X., & Leung, V. C. (2021, June). Neighboring-aware caching in heterogeneous edge networks by actor-attention-critic learning. In *ICC 2021-IEEE International Conference on Communications* (pp. 1-6). IEEE. https://doi.org/10.1109/ICC42927.2021.9500929

[19] Zhu, J., Li, R., Chen, X., Mao, S., Wu, J., & Zhao, Z. (2024). Semantics-enhanced temporal graph networks for content popularity prediction. *IEEE Transactions on Mobile Computing*, *23*(8), 8478-8492. https://doi.org/10.1109/TMC.2023.3349315

[20] Zhu, J., Li, R., Ding, G., Wang, C., Wu, J., Zhao, Z., & Zhang, H. (2022). AoI-based temporal attention graph neural network for popularity prediction and content caching. *IEEE Transactions on Cognitive Communications and Networking*, *9*(2), 345-358. https://doi.org/10.1109/TCCN.2022.3227920

Authors Biography



N. Kirubakaran is an accomplished academician, researcher, and editor affiliated with the School of Information Science, Presidency University, Bengaluru, India. He possesses significant expertise in Artificial Intelligence, Machine Learning, Data Science, Computer Vision, Edge Computing, and Intelligent Computing Systems, with his research centred on the application of advanced computational techniques to address real-world challenges, most notably in satellite image analysis and predictive modelling. He has contributed extensively to the academic community through numerous peer-reviewed scholarly publications, conference proceedings, and edited volumes of international standing, and is widely recognized for his commitment to fostering interdisciplinary research collaborations and advancing innovation-driven education. Through his sustained engagement in teaching,

mentoring, and editorial leadership, he continues to play an active role in shaping the future of emerging technologies



B. Saraswathi is an experienced academician in the field of Computer Science with more than two decades of teaching experience. She has worked as a Lecturer at Thiru Seven Hills Polytechnic College for over 21 years. She is currently working as a System Administrator cum Instructor at Meenakshi College of Arts and Science, Chennai. She holds a Master of Computer Applications (MCA) degree from Annamalai University. Her academic background also includes a Bachelor of Computer Applications (BCA) and a Diploma in Computer Technology. She has extensive experience in teaching both theory subjects and practical laboratory courses. Her areas of expertise include programming languages such as C, C++, Java, and Python, along with database management systems. She is also experienced in preparing academic materials such as lesson plans, question papers, and student assessment reports. She has received several prizes for achieving 100% academic results. With her strong academic background and long teaching experience, she continues to contribute actively to higher education and academic publications



Dr. Vimal Bibhu is Doctorate in Computer Science from Indian Public Universities. He is currently working on the post of professor in department of Computer Science & Engineering at Noida International University, Greater Noida, Uttar Pradesh. He has published 117 research articles in various peer reviewed international journals and conferences and 71, are Scopus Indexed including 23, in Web of Science core collection with SCIE, ESCI tags. He has published 29 utility patents and 7 copyrights. He has organized 7 IEEE, 1 Springer and 1 Taylor and Francis sponsored conferences. His research areas are Artificial Intelligence, Image Processing, IoT and Cyber Security. He has supervised 5 Ph.D. scholars and currently supervises 7 Ph.D. Scholars of different govt. and private universities. His Google Scholar H and I10 Indexes are 20 and 32 respectively with 1063 citations. Also, his Scopus H index is 13 with 413 citations and web of science H index is 4 with 55 citations.



Suvarna Milind Patil is an Assistant Professor in the Department of Engineering, Science and Humanities at Vishwakarma Institute of Technology, Pune. She holds an M.Sc. (2004) and B.Sc. (2002) from North Maharashtra University. Her academic background is in science, with her professional profile focused on teaching and higher education.



Nimesh Raj research contributions include environmental and biological systems, with work examining ecosystem development and related scientific processes. He is involved in interdisciplinary research connecting environmental science with analytical approaches.



Dr.J. Janet (born July 12, 1973) is an eminent Indian academician, researcher, and institutional leader who currently serves as the Principal of the Aarupadai Veedu Institute of Technology (AVIT) in Chennai. With nearly three decades of experience in higher education, she holds a Ph.D. in Computer Science and Engineering from Sathyabama University, specializing in Soft Computing, and previously secured the University 10th Rank during her Master's studies at the University of Madras. Throughout her distinguished career, she has held high-profile leadership roles—including Pro Vice Chancellor at Sanjivani University and Principal at Kumaraguru College of Technology—and has actively contributed to national academic governance as an Anna University Academic Council

member and a UGC autonomous inspection committee delegate. A Senior IEEE Member and a prolific researcher, Dr. Janet has published over 100 papers, holds innovative patents like an "AI-Based Pen," and has been recognized nationally as one of the "20 Most Prominent Principals in Higher Education" for her commitment to sustainability, innovation, and educational excellence



Dr. Suman Sau is an Associate Professor in the Department of Computer Science and Information Technology at Siksha O Anusandhan (Deemed to be University), ITER, Bhubaneswar, India (since February 2022; previously Assistant Professor, 2018–2022). His research focuses on embedded machine learning using RISC-V processors, embedded edge computing, post-quantum cryptography and 5G security, secure processor design, system reliability, and cryptographic hardware for embedded systems. He earned his Ph.D. (Tech) from the University of Calcutta in 2018 for work on efficient secure data communication techniques for reconfigurable hardware platforms, following an M.Tech from the same university and a B.Tech in Computer Science and Engineering. He has held research positions at Nanyang Technological University (Singapore), LIRMM (France), the National University of Singapore, and the University of Calcutta, and has contributed to projects involving FPGA-based systems, high-energy physics data acquisition, and IoT/blockchain security. He has received multiple Faculty Fellowship Awards at the International Test Conference (2024–2026), VLSI Design and Embedded Systems conferences (2024, 2026), and IEEE Asian Test Symposium (2019, 2024); the Young Faculty Fellowship Award at the India International Science Festival 2019 (DST); selection as INAE-CEEE 2026 Mentee at IIT Guwahati; and earlier recognition including selection (one of only four from Asia) for the IEEE TTTC’s E.J. McCluskey Doctoral Thesis Award Semi-Final 2016 and an Honorable Mention in the Design Contest at the 28th VLSI Design and Embedded Systems Conference 2015.