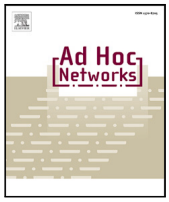
Ad Hoc Networks 157 (2024) 103436

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc



Multi-content time-series popularity prediction with Multiple-model Transformers in MEC networks☆

Zohreh Hajiakhondi Meybodi [a,1], Arash Mohammadi [b,*,2], Ming Hou [c,2], Elahe Rahimian [b,1], Shahin Heidarian [a,1], Jamshid Abouei [d,2], Konstantinos N. Plataniotis [e,3]

[a] Department of Electrical and Computer Engineering, Concordia University, Montreal, QC, H3G-2W1, Canada
[b] Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, H3G-2W1, Canada
[c] Defence Research and Development Canada (DRDC), Ottawa, Toronto, Canada
[d] Department of Electrical Engineering, Yazd University, Yazd, 89195-741, Iran
[e] Electrical and Computer Engineering (ECE), University of Toronto, Toronto, H3G-2W1, Canada

ARTICLE INFO

Keywords:
Mobile Edge Caching (MEC)
Popularity prediction
Deep neural network (DNN)
Machine learning
Transformer

ABSTRACT

Coded/uncoded content placement in Mobile Edge Caching (MEC) has evolved as an efficient solution to meet the significant growth of global mobile data traffic by boosting the content diversity in the storage of caching nodes. To meet the dynamic nature of the historical request pattern of multimedia contents, the main focus of recent researches has been shifted to develop data-driven and real-time caching schemes. In this regard and with the assumption that users' preferences remain unchanged over a short horizon, the Top-K popular contents. These contents refer to the most requested content in the upcoming period. Most existing data-driven popularity prediction models, however, are not suitable for the coded/uncoded content placement frameworks. On the one hand, in coded/uncoded content placement, in addition to classifying contents into two groups, i.e., popular and non-popular, the probability of content request is required to identify which content should be stored partially/completely, where this information is not provided by existing data-driven popularity prediction models. On the other hand, the assumption that users' preferences remain unchanged over a short horizon only works for content with a smooth request pattern. To tackle these challenges, we develop a Multiple-model (hybrid) Transformer-based Edge Caching (MTEC) framework with higher generalization ability, suitable for various types of content with different time-varying behavior, that can be adapted with coded/uncoded content placement frameworks. In this work, we consider Top-K content as the output of the 1st Stage of the proposed MTEC framework, which includes both popular and mediocre content. Simulation results corroborate the effectiveness of the proposed MTEC caching framework in comparison to its counterparts in terms of the cache-hit ratio, classification accuracy, and the transferred byte volume.

1. Introduction

Mobile Edge Caching (MEC) [1–5] has emerged as a promising solution for potential deployment in the Sixth Generation (6G) of communication networks to meet the phenomenal growth of global mobile data traffic. The main idea behind MEC networks is to provide low-latency communication for Internet of Things (IoT) devices by bringing multimedia content closer to users [6,7,7–9]. In this context, if content requested by an IoT device can be found in the storage of one of the nearby caching nodes, low-latency communication will be established and cache-hit occurs, otherwise, the IoT device will experience high latency. Due to the limited storage of caching nodes, one of the efficient approaches to improve the cache-hit ratio is to increase the content diversity. This can be fulfilled by implementing a coded/uncoded content placement in an integrated MEC network [10,11]. This hybrid approach intelligently selects which content to cache in coded form and

☆ This Project was partially supported by Department of National Defence's Innovation for Defence Excellence & Security (IDEaS) program, Canada.
* Corresponding author.
E-mail addresses: zohreh.hajiakhondimeybodi@concordia.ca (Z.H. Meybodi), arash.mohammadi@concordia.ca (A. Mohammadi), ming.hou@drdc-rddc.gc.ca (M. Hou), e_ahimia@encs.concordia.ca (E. Rahimian), shahin.heidarian@concordia.ca (S. Heidarian), abouei@yazd.ac.ir (J. Abouei), kostas@ece.utoronto.ca (K.N. Plataniotis).
[1] IEEE Student Member.
[2] IEEE Senior Member.
[3] IEEE Fellow.

https://doi.org/10.1016/j.adhoc.2024.103436
Received 19 September 2023; Received in revised form 16 January 2024; Accepted 5 February 2024
Available online 8 February 2024

which in uncoded form, optimizing storage efficiency, content delivery speed, and network reliability. Recent advancements in heterogeneous cluster-centric cellular networks [12,13] have drawn focused research attention given provided considerable improvements in content diversity, which is due to the integration of the coded/uncoded content placement and Coordinated Multi-Point (CoMP) technology. Besides, integrating Unmanned Aerial Vehicles (UAVs) as flying caching nodes into the cluster-centric MEC networks [6,14–16] can further improve the network's Quality of Service (QoS) due to the high-quality Line of Sight (LoS) links, high mobility of UAVs, and their wide transmission range.

Despite all the research works conducted to develop an efficient coded/uncoded content placement strategy, the dynamic and time-varying topology of MEC networks presents new challenges when it comes to the design of an optimal real-time caching scheme [17]. On the one hand, integrated MEC networks are unpredictable in real-world scenarios due to the high mobility of both mobile users and edge caching devices [18]. Additionally, mobile users' preferences are time-varying, depending on different variables such as content popularity, geographical region, and the users' contextual information [19]. On the other hand, the local storage capacity of cache-enabled edge devices is limited, therefore, it is essential to dynamically observe the users' request pattern to regularly update the storage of edge devices with the most upcoming popular contents. To accommodate these critical aspects of MEC networks, it is of significant practical importance to augment MEC networks with a data-driven popularity prediction model to continuously monitor and analyze time-varying request patterns of multimedia contents. The paper aims to further advance this emerging field.

Literature Review: Recently, several promising approaches have been developed to predict the popularity of multimedia content, including but not limited to (*i*) Statistical models [20,21], such as collaborative filtering, item-to-item correlation systems, and content-based filtering; (*ii*) Machine Learning (ML)-based architectures [22], such as Generalized Linear Model (GLM) [23], Decision Tree (DT) [24], and Random Forest (RF) [25], and; (*iii*) Deep Neural Networks (DNN) [26–34], such as Convolutional Neural Network (CNN) [35], and Long Short Term Memory (LSTM) [29,36]. Despite all the benefits that come from existing statistical and ML-based frameworks, they suffer from sparsity and cold-start problems, which arise when sufficient information is not provided about a new mobile user/multimedia content. Moreover, having a well-trained ML architecture relies on an efficient feature engineering model to extract several contextual information associated with both mobile users and multimedia content. While ML-based caching schemes suffer from poor scalability across different scenarios, DNN-based models can capture the user's interests from raw historical request patterns without the need for feature extraction and pre-processing. Consequently, the main focus of recent research has been shifted to DNN-based frameworks to monitor/predict content popularity using the historical request pattern of contents.

With the focus on DNN models, Yu et al. [37] introduced an auto-encoder architecture to learn the latent representation of historical request patterns of content to predict the users' preferences in the upcoming time. Although auto-encoder is an unsupervised (self-supervised) learning model and there is no need for ground truth labels of content popularity, it suffers from several drawbacks. On the one hand, auto-encoder needs a large number of training data, resulting in a high training overhead, i.e., excessive training complexity. On the other hand, auto-encoder attempts to capture as much information as possible rather than capturing the relevant information. Therefore, in a scenario where the relevant information is just a small part of the data, the model fails to train well. Zheng et al. [38] proposed an unsupervised and privacy-preserving popularity prediction framework for MEC-enabled Industrial Internet of Things (IIoT). Challenges include spatio-temporal variability, data deficiency, costly manual labeling, and non-I.I.D. user behaviors. The model introduces concepts of local and global popularities, employed a model-free Markov chain for time-varying popularity, and utilizes an Unsupervised Recurrent Federated Learning (URFL) algorithm for privacy preservation and unsupervised training. Additionally, a federated loss-weighted averaging (FedLWA) scheme addressed non-I.I.D. user behaviors, contributing to high popularity prediction accuracy in MEC-enabled IIoT. Moreover, Liu et al. [39] addressed content popularity dynamics, complexities, and user privacy challenges in MEC networks through a Privacy-Preserving Distributed Deep Deterministic Policy Gradient (P2D3PG) framework. The proposed approach maximized the cache-hit rates by converting distributed optimizations into model-free Markov decision process problems and employing a privacy-preserving federated learning method for popularity prediction. Ndikumana et al. [40] used CNN architecture to capture the contextual information of users, such as age, emotion, and gender to calculate the probability of requesting a content using Multi-Layer Perceptron (MLP). To have an efficient DNN-based caching scheme, however, both temporal and spatial correlations of content should be captured from time-variant and sequential request patterns of multimedia content. More precisely, spatial correlation represents different users' preferences, relying on regional information, content popularity, and users' contextual information. The temporal correlation is related to the time-variant behavior of the request pattern. While CNN-based caching schemes can capture local spatial correlations, they fail to properly capture temporal features of historical request patterns. Furthermore, such models require a data pre-processing stage to provide some additional information to be used as the input features and improve the cache performance.

To address the above-mentioned challenges associated with CNN-based caching schemes, Recurrent Neural Networks (RNNs), such as LSTM [29,36], were introduced to capture the temporal dependencies of sequential request patterns. While LSTM models have advantages for processing time-series data, their computation complexity, unsuitability for capturing long-term dependencies, and difficulties for parallel computations make them problematic. To tackle the aforementioned issues, Transformer architectures [41] have been developed. Similar to RNN models, Transformers are designed to process sequential data, while there is no need to analyze the sequential data in the same order, resulting in higher parallelization, and reduced training complexity. Nguyen et al. [42] proposed the Attention-based Non-Recursive Neural Network (ANRNN) to forecast future content requests, i.e., predicting the precise number of requests. In contrast, our study in [19] concentrated on predicting the Top-K popular content using historical data instead of forecasting the exact number of upcoming requests. More precisely, we introduced a Vision Transformer (ViT)-based Edge (TEDGE) caching framework for an uncoded content placement, which to the best of our knowledge, was being studied for the first time to predict the Top-K popular content in MEC networks using attention mechanism. Our prior work [19] and other existing data-driven uncoded popularity prediction models are, however, unsuitable for coded/uncoded content placement frameworks. Zhang et al. [43] introduced a DNN-based approach for coded content placement, which, however, proved to be unsuitable for hybrid content placement scenarios where content can be stored in both coded and uncoded formats. The reason is that the existing classification models classify content into two groups, i.e., popular and non-popular, in which popular contents are stored completely. In a coded/uncoded content placement, however, multimedia content should be stored completely/partially based on its popularity, i.e., the request probability. In addition, the assumption that users' preferences remain unchanged over a short time horizon, only works for contents with smooth request patterns. The paper addresses these gaps as outlined below.

Contributions: To tackle the aforementioned challenges, in this paper, we develop the Multiple-model Transformer-based Edge Caching (MTEC) framework as a multi-content and time-series popularity prediction model. The MTEC framework captures both temporal and spatial correlation of multiple contents via multi-channel Transformer

architectures, where the sequential request patterns of each content are given to a channel of the Transformer model. Our first objective for the development of the MTEC framework is to introduce a data-driven popularity prediction model with higher generalizability compared to existing works that predict the Top-K popular content relying on the historical request pattern. Predicting popularity in MEC networks is a crucial task, requiring anticipation of the most sought-after content in the near future. In ML models, a fundamental requirement is the establishment of input–output relationships, where historical content requests serve as input, and the anticipated content popularity becomes the output. It is essential to note that multimedia content exhibits diverse behaviors; some content displays a steady request pattern, while others show significant fluctuations. Most existing research works [22, 31,32] rely on the historical request data as input, a strategy effective for multimedia content with consistent fluctuations. However, in our work, we challenge this assumption. Instead of directly using previous requests as input, we first predict the upcoming content request volume. Subsequently, we utilize this predicted quantity to classify content as either popular or unpopular. This departure from conventional methods allows us to enhance the overall accuracy, particularly for content with irregular request patterns. By adopting such an approach, we achieve a more precise classification of popular content, even when faced with unpredictable request probabilities. The second objective is to adapt the data-driven prediction model within the coded/uncoded content placement approaches. To achieve these objectives, the proposed MTEC framework is built upon the Transformer architecture, consisting of two parallel paths, which takes the historical request pattern of multiple contents as its input:

- The first path of the MTEC framework is a Transformer network, responsible for identifying the Top-K popular contents in the upcoming time, using the historical request pattern of contents. This part of the architecture is efficient for contents that their request patterns are smoothly changed over a short horizon.
- The second path of the MTEC framework is introduced to boost the generalizability of the learning model. In this path, for applicability to various types of content with different time-varying behavior, we relax the assumption of unchanged request patterns of content. Moreover, within the context of coded/uncoded content placement, we take one step forward and relax the assumption that the probability of content requests is known a-priori (i.e., Zipf distribution) [44]. While traditional MEC networks typically rely on Zipf distribution to represent the probability of content requests, our approach takes a different path. Instead of using Zipf distribution, we opt for a dynamic approach by predicting the request probability of content in the future. This prediction is the output of our "Request Probability Prediction" block. In essence, our model captures the dynamic nature of user preferences, allowing us to move beyond the static representation offered by Zipf distribution. Through this approach, we consider and incorporate evolving user preferences into our predictions. In this regard, the second path is compromised of two stages, where the first stage is a Transformer network responsible for predicting the probability of requesting multiple contents in the upcoming time. The estimated request probability as the output of the first stage will be used for the coded/uncoded content placement to identify which content should be stored partially/completely in the storage of caching nodes. Next, it will be concatenated with the historical request pattern (input of the first stage), provided as the input to the second stage, which analyzes the popularity of all contents simultaneously.
- The final output, which is the combination of the two parallel paths, is the Top-K popular contents in the upcoming time, which is applicable to various types of content with different time-varying behavior. The effectiveness of the proposed MTEC framework is evaluated through comprehensive studies on the real-trace multimedia request pattern, in terms of the classification accuracy, cache-hit ratio, and the transferred byte volume. Simulation results corroborate the effectiveness of the proposed MTEC framework in comparison to its counterparts over all the aforementioned aspects.

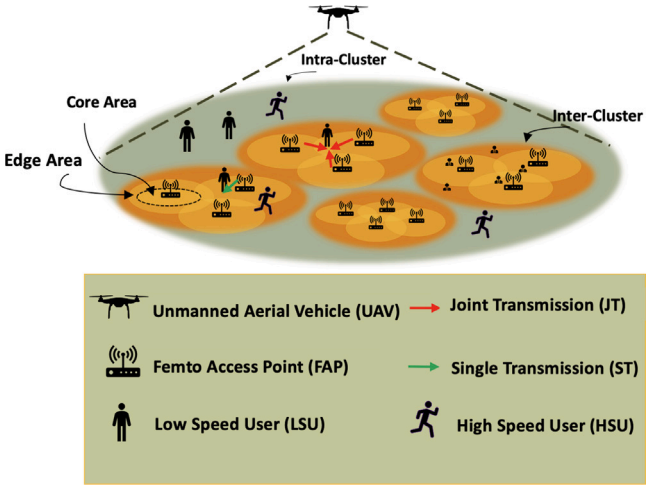


Fig. 1. A typical structure of a cluster-centric and UAV-aided cellular networks.

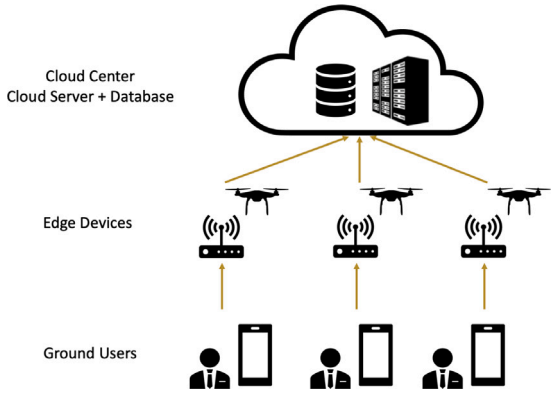


Fig. 2. Edge caching architecture.

The remainder of the paper is organized as follows: In Section 2, the system model is described and the main assumptions required for the implementation of the proposed framework are introduced. Section 3 presents the proposed MTEC scheme. Simulation results are presented in Section 4. Finally, Section 5 concludes the paper.

2. System model and problem description

We consider a cluster-centric UAV-aided cellular network in both indoor and outdoor residential areas. As shown in Fig. 1, there are N_f number of FAPs, denoted by f_i, for $(1 \leq i \leq N_f)$, distributed based on the Poisson Point Processes (PPPs) in the environment [10], N_u number of UAVs, denoted by u_k, for $(1 \leq k \leq N_u)$, and the cloud center. Illustrated in Fig. 2, the cloud center, comprising the cloud server seamlessly integrated with a dedicated database, serves as the location for the content library within it. This underscores the conformity of our approach with the current research standards. When a user initiates a content request, the system checks if the desired content is available in the storage of a nearby edge device, such as a FAP or UAV. If the content is present locally, the edge device directly manages the request. However, if the content is not available in the nearby storage, the cloud center intervenes and provides the requested content.

Followed by Gaussian mixture distribution, there exist N_g number of ground users, denoted by GU_j, for $(1 \leq j \leq N_g)$, moving through the network with the velocity of $\upsilon_j(t)$. To avoid frequent handover in terrestrial infrastructure and UAV's signal attenuation in indoor areas, indoor requests are managed by FAPs, while outdoor users' requests are handled through FAPs/UAVs, depending on their movement speed (i.e., Low-Speed Users (LSU) are served through FAPs, otherwise, they are managed by UAVs) [44]. More precisely, In Fig. 1, the High-Speed User (HSU) represents a mobile user with rapid mobility, swiftly traversing various cells or clusters within the network. This scenario mirrors real-world situations, such as when a user is in transit, like in a moving vehicle. The rapid movement of users, characterized by fast mobility, introduces challenges for content caching and delivery strategies, as users may move away from the current caching node before completing the full content download. The existence of HSU underscores the critical need for efficient edge caching algorithms, a challenge addressed by the proposed MTEC framework. To adeptly handle the requests of both LSU and HSU, we adopt a cluster-centric approach in a UAV-aided cellular network within the MTEC framework. The following two noteworthy features are considered in the network setup: Firstly, in a cluster-centric network, coded content placement enhances content diversity by storing different content segments in distinct caching nodes. This is particularly advantageous for HSU, allowing it to access multiple segments of the desired content as it moves. Secondly, the utilization of UAVs enhances caching delivery, leveraging the extended transmission range of UAVs compared to FAPs. As a result, we assume that HSU should be primarily managed by UAVs, unless there is no available UAV in its proximity, in which case the request should be handled through FAPs. Due to the high mobility of users, the location of UAVs is determined using the K-means clustering algorithm [16]. Each UAV covers one intra-cluster and hovers at its location while delivering a request [45]. Besides, in contrast to conventional femtocaching schemes, where each FAP acts as a single unity, we assume that the storage of $N_b < N_f$ number of nearby FAPs belonging to an inter-cluster is known as a component.

It is worth mentioning that the proposed MTEC framework applies to both UAV-aided cellular networks, known for their complexity, and traditional MEC-based cellular networks. Generally speaking, traditional MEC-based cellular networks exhibit key features that prioritize low latency, efficient resource utilization, and enhanced user experiences. By deploying storage at the network edge, MEC networks ensure proximity to end-users, minimizing latency and supporting real-time applications. Content caching, resource optimization, and scalability are integral aspects, allowing dynamic allocation of resources based on application demands. Our proposed MTEC framework, initially designed for UAV-aided cellular networks, seamlessly extends its applicability to traditional MEC-based cellular networks. With features like content caching optimization and adaptive content prediction, MTEC aligns with the objectives of traditional MEC networks, enhancing their efficiency, responsiveness, and overall performance in a variety of scenarios. The framework accommodates the diverse requirements of MEC architectures, contributing to improved resource management and network capabilities in the context of traditional cellular infrastructures.

There is a content library $\mathcal{C} = \{c_1, \ldots, c_{N_c}\}$, where $N_c = |\mathcal{C}|$ is the cardinality of contents in the network, and each content c_l is segmented into N_s encoded parts, denoted by c_{ls}, for $(1 \leq s \leq N_s)$. While the storage capacity of FAPs and UAVs, denoted by C_f and C_u, respectively, is limited, the cloud center has the whole library of multimedia content. Multimedia contents are conventionally classified using the Zipf distribution [10,46] into three groups, i.e., popular, mediocre, and non-popular. It is essential to note that Ref. [10] employed the Zipf distribution, whereas Ref. [46] utilized the Mandelbrot–Zipf distribution. Both distributions play a role in modeling frequency distribution for popularity prediction, reflecting the popularity of content. The Zipf distribution assumes a power-law relationship characterized by a few highly popular items and many less popular ones In contrast, the Mandelbrot–Zipf distribution extends the Zipf distribution by incorporating a dispersion parameter, allowing for smoother adaptability to variations in popularity. In this work, we advanced these methodologies by introducing the MTEC framework, wherein the output of the "Request Probability Prediction" block is utilized to forecast the probability of requesting a content. It is worth mentioning that the classification of content into popular, mediocre, and non-popular categories involves a three-step process. Initially, the content's request probability must be determined. Subsequently, the content is sorted based on these probabilities. Content labeled as "Popular" experiences frequent and high-demand requests, "Mediocre" content encounters moderate demand, while "Non-popular" content sees low demand. Finally, storage allocation involves dedicating a portion to popular content, with the remaining storage assigned to mediocre content. In our recent work [44], we employed the Zipf distribution to calculate the request probability of content. The probability p_l, representing the likelihood of content c_l being requested, is calculated as

$$p_l = \frac{l^{-\gamma}}{\sum_{r=1}^{N_c} r^{-\gamma}}, \tag{1}$$

where γ serves as the skewness parameter. The variable r denotes the rank of content c_r when all contents are arranged in descending order based on their popularity. In this work, the methodology for categorizing content as popular, mediocre, and non-popular aligns with our recent work [44]. The distinguishing factor lies in the approach used here, where instead of utilizing p_l as the output of the Zipf distribution, we predict the probability of content requests as the output of the "Request Probability Prediction" block in the proposed MTEC framework. This block, grounded in Transformer architecture, processes the time-series historical request patterns of content. Consequently, our model captures the dynamic shifts in user preferences over time, surpassing the static representation offered by the Zipf distribution. After classifying content based on their popularity, the storage capacity of FAPs, identified as C_f, is partitioned into two parts. The α portion of the storage is dedicated to storing complete popular contents, encompassing the range $1 \leq l \leq \lfloor \alpha C_f \rfloor$, where $l = 1$ designates the most popular content. Additionally, the $(1-\alpha)$ portion of the cache is allocated to store various segments of mediocre content, covering the range $\lfloor \alpha C_f \rfloor + 1 \leq l \leq N_s(C_f - \lfloor \alpha C_f \rfloor)$. The optimal value for α is determined through experimental evaluation [44], where in this study, it is set to 0.4 [44].

To deliver multimedia content from inter-clusters to users, two transmission schemes, i.e., Single Transmission (ST) and Joint Transmission (JT), are utilized based on the link quality of the user and the popularity of the requested content [44]. More precisely, if the average Signal-to-Interference-plus-Noise Ratio (SINR) of user GU_j and FAP f_i, denoted by $\overline{S}_{i,j}(t)$, is higher than a pre-defined threshold S_{th}, then user GU_j is known as the cell-core of FAP f_i. In this case and regardless of the popularity of the content, this request will be managed by FAP f_i, i.e., the ST scheme. On the other hand, if the requested content is popular and the user GU_j is marked as the cell-edge of FAP f_i, this request will be jointly served by all FAPs belonging to the corresponding inter-cluster, i.e., the JT scheme. This completes the description of the system model, which is used after predicting the content popularity by the proposed MTEC multiple-model (hybrid) Transformer architecture. Next, we introduce our proposed MTEC framework.

3. Multiple-model transformer-based edge caching (MTEC) framework

The main goal of the proposed MTEC architecture is to predict the Top-K popular content using the historical request pattern of the underlying contents. In this context, we use MovieLens Dataset [47], where leaving a comment after watching a movie is considered a request [29,48,49]. In this section, we first describe the dataset preprocessing phase, and then present different blocks of the proposed MTEC architecture.

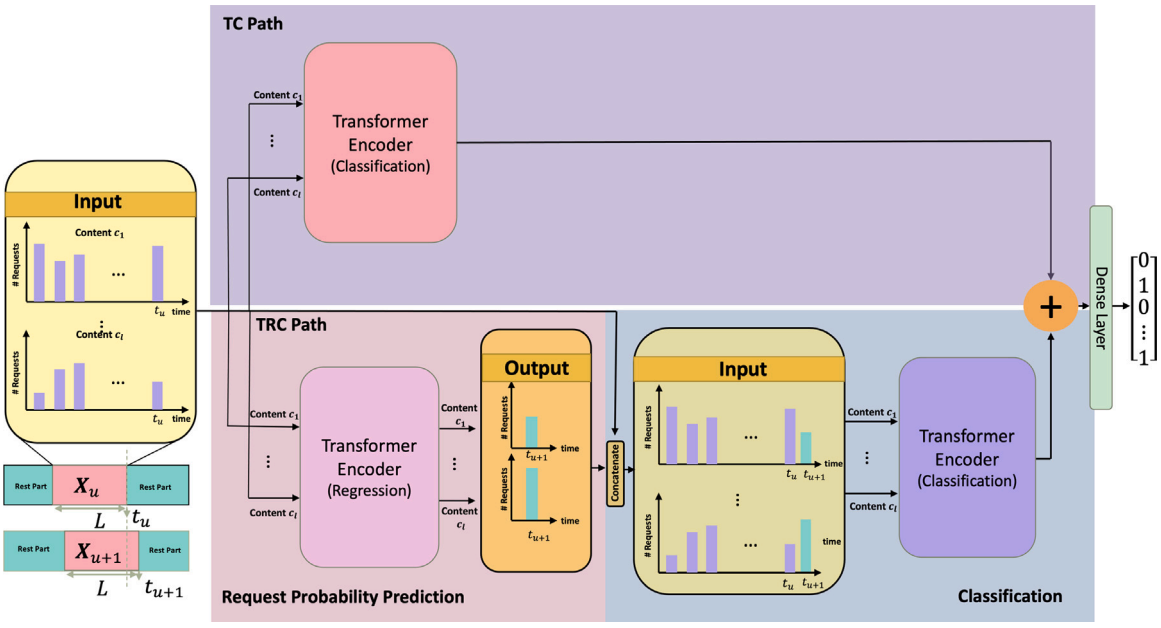


Fig. 3. Block diagram of the proposed MTEC architecture.

3.1. Dataset pre-processing

MovieLens Dataset includes users' contextual information, such as gender, age, and occupation together with their geographical information, i.e., ZIP code. To determine users' location during their requests, we convert ZIP codes to longitude and latitude coordinates [29]. Given the caching nodes' location and their transmission range, therefore, a set of caching nodes in the vicinity of each user will be determined. To adopt MovieLens Dataset to our popularity prediction model, we perform the following four steps:

Step 1 - Request Matrix Formation: Since the proposed MTEC multiple-model Transformer architecture is a time-series forecasting model, we first sort the requests of content c_l, for $(1 \le l \le N_c)$, in the ascending order of time. With the assumption that there are T timestamps (i.e., seconds) and N_c number of contents, $\mathbf{R} \in \mathbb{R}^{N_c \times T}$ represents an indicator request matrix for each caching node as follows

$$\mathbf{R} = \begin{bmatrix} 1 & \dots & 0 \\ 0 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \dots & 0 \end{bmatrix}_{N_c \times T} \tag{2}$$

where $r_{l,t} = 1$ means that content c_l is requested at time t; otherwise, $r_{l,t} = 0$.

Step 2 - Time Windowing: Considering the fact that content placement will be performed during the off-peak time (updating time t_u) [50], it is assumed that time T is discretized into N_w number of time intervals with a length of $\mathcal{W}$, where $\mathcal{W}$ represents the time duration between two off-peak times. Consequently, we form a window-based request matrix, denoted by $\mathbf{R}^{(\mathcal{W})} \in \mathbb{R}^{N_c \times N_w}$, where $r^{(w)}_{l,t_u} = \sum_{t=(t_{u-1})\mathcal{W}+1}^{t_u \mathcal{W}} r_{l,t}$ is the cumulative requests of content c_l during two consecutive updating times $t_u - 1$ and t_u.

Step 3 - Data Segmentation: To predict the Top-K popular content at updating time t_u, the input data should be a historical request pattern of content during the past updating times with the length of L. Given $\mathbf{R}^{(\mathcal{W})}$ from the previous step, the window-based request matrix $\mathbf{R}^{(\mathcal{W})}$ is segmented via an overlapping sliding window of length L to provide input samples $\mathbf{X}_u \in \mathbb{R}^{N_c \times M}$, where $M = \frac{N_w}{L}$ is the total number of segments (input samples). We, therefore, have $D = \{(\mathbf{X}_u, \mathbf{y}_u)\}_{u=1}^{M}$, with $\mathbf{y}_u \in \mathbb{R}^{N_c \times 1}$ as the output (label) of the learning model, where $\sum_{l=1}^{N_c} y_{u,l} = K$, and $y_{u,l} = 1$ means that content c_l would be popular at t_{u+1}, otherwise, it would be zero.

Step 4 - Data Labeling: Relying on the historical request pattern of content at updating time t_u, denoted by $\mathbf{X}_u$, our goal is to predict the Top-K popular content, which is a multi-class classification problem. To identify (label) content as popular/non-popular ones, we have the following two criteria:

(i) *Probability of Requesting a Specific Content*: Probability of requesting c_l, for $(1 \le l \le N_c)$, at updating time t_u, denoted by $p_l^{(t_u)}$, is given by

$$p_l^{(t_u)} = \frac{r^{(w)}_{l,t_u}}{\sum_{l=1}^{N_c} r^{(w)}_{l,t_u}}, \tag{3}$$

where $r^{(w)}_{l,t_u}$ represents the number of requests of content c_l in time window with length of L.

(ii) *Skewness of the Request Pattern*, which is a commonly used metric in time-series forecasting models to accelerate the popularity prediction of the first appearance content [51]. Term ζ_l represents the skewness of content c_l, where negative skew means that the number of requests of content c_l increases over time. Therefore, the Top-K content c_l, for $(1 \le l \le K)$, with negative skew and the highest probability, will be labeled with $y_{u,l} = 1$ and identified as the Top-K popular content.

It should be noted that, as we will describe shortly, the outcome from the 1st Stage of the proposed MTEC architecture is a binary classification used to distinguish non-popular content (labeled as zero) from popular and mediocre content (labeled as 1). Afterward, the output of the "Request probability Prediction" block in the second path is used to distinguish between popular and mediocre content.

3.2. MTEC architecture

In this subsection, we present the constituent components of the proposed MTEC framework, where the main architecture is developed based on the Transformers (see Fig. 3). There are following drawbacks to the existing research works that motivate us to develop the MTEC framework:

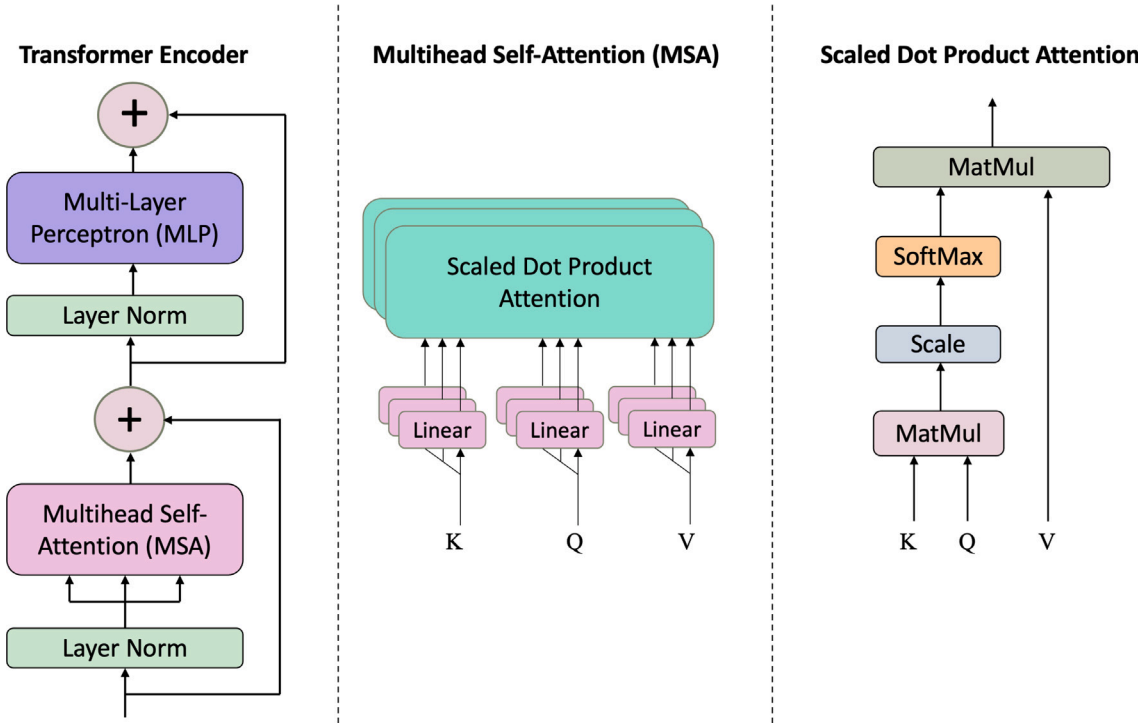


Fig. 4. Left: Architecture of the Transformer encoder, Middle: Multihead Self-Attention (MSA); Right: Scaled dot product attention.

(*i*) With the assumption that users' preferences remain unchanged over a short horizon, existing works [19,35,40] predicted the Top-K popular content in the updating time t_{u+1}, where the input of the learning model was the request pattern of all contents in a time window with the length of L, ending at the updating time of t_u. This assumption works for content whose request pattern smoothly changes over time. It is, therefore, essential to develop a popularity prediction model with higher generalization ability, suitable for various types of content with different time-varying behavior.

(*ii*) In a coded/uncoded content placement, the request probability of content in an upcoming time is also required to classify the Top-K popular content into two groups, i.e., popular and mediocre, while existing classification frameworks [19,36,37,40] classified content as popular/non-popular.

To tackle the aforementioned challenges, we propose the MTEC framework, consisting of two parallel paths, where the first path is a Transformer-based Classification (TC) model, and the second path includes two series of Transformer-based blocks, named Transformer-based Regression and Classification (TRC) network. These are followed by a fully connected layer, as a fusion center combining the output of the two parallel paths to estimate the Top-K popular content. It should be noted that although the output of these two paths is similar in nature, simulation results illustrate that considering such an architecture improves the popularity prediction. Next, we introduce each of these blocks.

3.2.1. TC path

This path is a multi-label classification model based on the Transformer model, where the input is the historical request pattern of multiple contents at time t_u and the output is the Top-K popular content at time t_{u+1}. The Transformer is a type of Machine Learning (ML) model suitable for learning from sequential and time series data. Generally speaking, Transformers outperform the LSTM models because: (*i*) Although LSTM models are capable of learning long-term dependencies, they suffer from short-term memory over long sequences. Transformers, however, capture the connection/dependency between sequential components that are far from one another, resulting in higher accuracy; (*ii*) Due to the Multi-head Self-Attention (MSA) mechanism, Transformers can process data in parallel, reducing the training time; and, (*iii*) The attention mechanism of the Transformer eliminates the need to analyze data in the same order. Consequently, positional embedding is used to preserve the position information of an entity in sequential data.

The multi-content training sample $\mathbf{X}_u \in \mathbb{R}^{N_c \times M}$ includes M feature vectors $\mathbf{x}_u \in \mathbb{R}^{N_c \times 1}$. Using the min–max normalization method, the feature vectors are first normalized and then linearly projected into a d-dimensional vector space, where d represents the model dimension. The input of the learning model $\mathbf{v}_u \in \mathbb{R}^d$ is given by

$$\mathbf{v}_u = \mathbf{W}_p \mathbf{x}_u + \mathbf{b}_p, \tag{4}$$

where $\mathbf{W}_p \in \mathbb{R}^{d \times N_c}$ and $\mathbf{b}_p \in \mathbb{R}^d$ are learnable parameters. Followed by Ref. [52] and to preserve the temporal correlations, we use a 1D-convolutional layer with d output channels to build vector $\mathbf{v}_u$. The positional embedding $\mathbf{E}^{pos} \in \mathbb{R}^{M \times d}$ is then appended to the input vector $\mathbf{V} \in \mathbb{R}^{M \times d} = [\mathbf{v}_1, \dots, \mathbf{v}_M]$, to form the input of the transformer encoder $\mathbf{Z}_0 = \mathbf{V} + \mathbf{E}^{pos}$ [41].

Transformer Encoder: As shown in Fig. 4, there are L layers in transformer encoder, consisting of MSA and the MLP modules, as follows

$$\mathbf{Z}'_l = MSA(LayerNorm(\mathbf{Z}_{l-1})) + \mathbf{Z}_{l-1}, \tag{5}$$

$$\mathbf{Z}_l = MLP(LayerNorm(\mathbf{Z}'_l)) + \mathbf{Z}'_l, \tag{6}$$

where $\mathbf{Z}'_l$ and $\mathbf{Z}_l$ represent the output of the MSA and MLP modules associated with layer l, for $(1 \le l \le L)$, respectively. Note that, before applying MSA and MLP modules, we use a layer-normalization to address the degradation problem [53]. Moreover, the Gaussian Error Linear Unit (GELU) is used as the activation function in the MLP module with two linear layers. This completes the description of the Transformer encoder developed for the design of the MTEC architecture. Considering the fact that the MSA module is defined based on the Self-Attention (SA) mechanism, next, we present the description of SA and MSA modules.

1. Self-Attention (SA): The SA module [41] in the Transformer architecture is used to capture the dependency of different vectors in

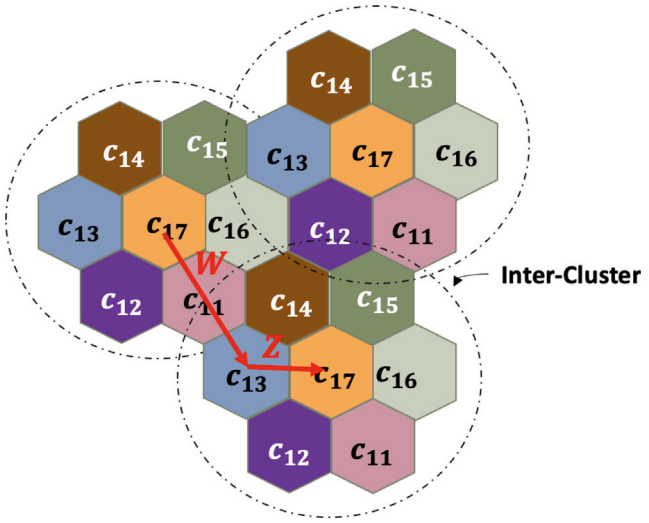


Fig. 5. Coded content placement scheme associated with mediocre content in single/multiple inter-clusters.

$\mathbf{Z} \in \mathbf{R}^{M\times d}$, where $\mathbf{Z}$ consists of M vectors, each with size of d. In this regard, Query $\mathbf{Q}$, Key $\mathbf{K}$, and Value $\mathbf{V}$ matrices are calculated as follows

$$[\mathbf{Q}, \mathbf{K}, \mathbf{V}] = \mathbf{Z}\mathbf{W}^{QKV}, \tag{7}$$

where $\mathbf{W}^{QKV} \in \mathbf{R}^{\mathbf{d}\times 3d_h}$ is a trainable weight matrix and d_h represents the dimension of $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ matrices. The output of the SA block $SA(\mathbf{Z}) \in \mathbf{R}^{M\times d_h}$ is calculated as a weighted sum of the values $\mathbf{V}$, with the weights assigned to each value, determined by a compatibility function between the query and the relevant key, as follows

$$SA(\mathbf{Z}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_h}}\right)\mathbf{V}, \tag{8}$$

where the term $\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_h}}$ is the scaled dot-product of $\mathbf{Q}$ and $\mathbf{K}$ by $\sqrt{d_h}$, and softmax is used to convert it to the probability values ranged between zero and one.

2. Multihead Self-Attention (MSA): The MSA module comprises of h heads with different trainable weight matrix $\{W_i^{QKV}\}_{i=1}^{h}$, performed h times in parallel. Given the outputs of h SA modules, the output of the MSA module is given by

$$MSA(\mathbf{Z}) = [SA_1(\mathbf{Z}); SA_2(\mathbf{Z}); \dots; SA_h(\mathbf{Z})]W^{MSA}, \tag{9}$$

where $\mathbf{W}^{MSA} \in \mathbf{R}^{hd_h\times d}$ and d_h is set to d/h.

Although this block is capable of capturing the request pattern of content with predictable behavior, i.e., with smooth changes over time in uncoded content placement, it would not be effective for content with sudden changes in a coded/uncoded manner. For these reasons, the second path is required.

3.2.2. TRC path

The second path consists of the following two blocks:

(*i*) **Request Probability Prediction Block:** The first block of the second path is used to predict the request probability of content at time t_{u+1} using the historical request pattern of content at time t_u (input data is the same as the first path). The output of this block will be used to classify the Top-K popular content (the final output of the MTEC framework) as the popular/mediocre one in the coded/uncoded content placement. As discussed in Section 2, the Top-K popular content will be sorted in descending order, where $N_p = \lfloor \alpha C_f \rfloor$ and $N_a = N_s(C_f - \lfloor \alpha C_f \rfloor)$ are the cardinality of popular and mediocre content, respectively. Following our prior work [44], vector $\mathbf{z} = [z_1, \dots, z_{N_a}]^T$ represents mediocre contents, where N_a denotes the cardinality of mediocre content. To identify which segments of mediocre content c_l, denoted by c_{ls} for $(1 \le l \le N_a)$ and $(1 \le s \le N_s)$, is cached in each FAP f_i for $(1 \le i \le N_b)$ belonging to an inter-cluster, an indicator matrix $\mathbf{Z}^{(f_i)}$ associated with FAP f_i is formed, where the lth row of $\mathbf{Z}^{(f_i)}$, represented by $\mathbf{z}_l^{(f_i)} = [0, \dots, 0, 1]_{(1\times N_s)}$ is corresponding to the segments of file c_l cached in FAP f_i. Note that, $\sum_{s=1}^{N_s} z_{ls}^{(f_i)} = 1$, where $z_{ls}^{(f_i)} = 1$, if the sth segment of file c_l is cached in FAP f_i, otherwise, it would be zero. Therefore, the mediocre content of other FAPs f_j, for $(1 \le j \le N_b, j \ne i)$, belonging to an inter-cluster, is given by

$$\mathbf{z}_l^{(f_i)}{\mathbf{z}_l^{(f_j)}}^T = 0, \quad i = 1, \dots, N_b, j = 1, \dots, N_b, i \ne j. \tag{10}$$

Following the above discussion, different segments of mediocre content will be stored in an inter-cluster. Next, the same content as FAP f_i in one inter-cluster, is allocated to FAP f_k in the nearby inter-cluster, where k is given by

$$\mathbf{Z}^{(f_k)} = \mathbf{Z}^{(f_i)} \quad \text{if} \quad k = w^2 + wz + z^2, \tag{11}$$

where w is the number of FAPs required to move from FAP f_i in any direction, after which z number of FAPs should be moved by turning 60 degrees counterclockwise to reach FAP f_k [44] (see Fig. 5). Please refer to our previous work [44] for a detailed description of the coded/uncoded content placement. Finally, the estimated request probability is appended to the original input samples to generate the input of the next block, which is used for the classification, i.e., identifying the Top-K popular content.

(*ii*) **Classification Block:** In comparison to the CT block in the first path, the input of this block is both the historical request pattern of content at time t_u and the estimated one at t_{u+1}, resulting in higher classification accuracy for such content with sharp changes in their request pattern. Then, the output features of both paths are added, which are used as the input of the fusion layer (dense layer). The output of the dense layer is a vector $\mathbf{y}_u \in \mathbb{R}^{N_c}$, with K ones, where 1′s indicates the Top-K popular content (i.e., popular/mediocre one) and 0′s are non-popular content. Finally, the estimated Top-K popular contents are categorized into popular/mediocre one according to the output of the request probability prediction block.

Finally, we use Mean Squared Error (MSE) for the request probability prediction block, and binary cross-entropy as the loss function for the CT path, the classification block in the second path, and the fusion path. The MSE is calculated as the average of the squared differences between the predicted and the actual request probability for the lth content, denoted by $\hat{p}_l^{(t_u)}$ and $p_l^{(t_u)}$, respectively. The MSE of the request probability prediction block, denoted by $\mathcal{L}_{RPP}$, is calculated as follows

$$\mathcal{L}_{RPP} = \frac{1}{N_c}\sum_{l=1}^{N_c}(\hat{p}_l^{(t_u)} - p_l^{(t_u)})^2. \tag{12}$$

The overall loss function $\mathcal{L}$ of the proposed MTEC is given by

$$\mathcal{L} = w_1\mathcal{L}_{RPP} + w_2\mathcal{L}_{C_I} + w_3\mathcal{L}_{C_{II}} + w_4\mathcal{L}_F, \tag{13}$$

where $\mathcal{L}_{RPP}$, $\mathcal{L}_{C_I}$, $\mathcal{L}_{C_{II}}$, and $\mathcal{L}_F$ represent the loss function associated with the request probability prediction block, CT block, the classification in the second path, and the fusion path, respectively, and w_i, $i = \{1, \dots, 4\}$ is the loss weight of each block.

4. Simulation results

To evaluate the performance of the proposed MTEC framework, we consider a cluster-centric UAV-aided cellular network with 138,493

Table 1
Variants of the MTEC popularity prediction framework.

Model ID	Layers (L)	Model dimension (d)	MLP layers	MLP size	Number of heads	Params	Accuracy
1	1	32	1	256	8	444,073	84.32%
2	1	32	1	256	16	652,457	88.15%
3	1	64	1	256	8	775,049	88.23%
4	2	64	1	256	8	1,438,556	92.78%
5	1	64	1	512	8	885,384	90.01%
6	2	64	2	256	16	2,882,140	94.13%
7	1	32	2	256	8	460,242	85.94%

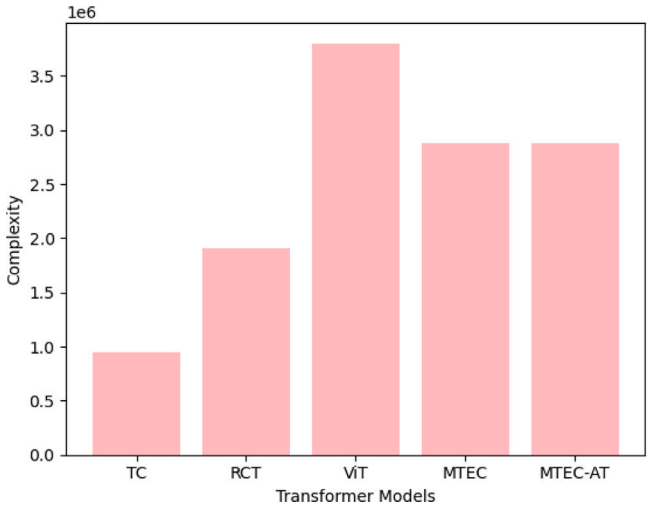


Fig. 6. Comparison of Transformer Models Complexity.

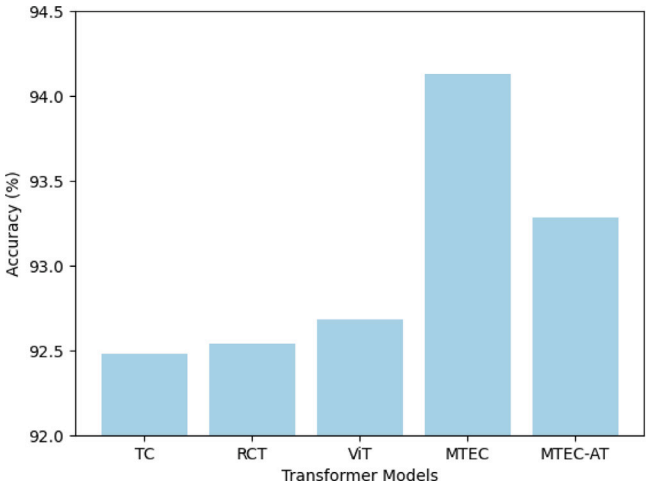


Fig. 7. Comparison of Transformer Models Classification Accuracy.

Table 2
The accuracy of the Top-K popular content using different window size (9, 49, and 99 days) for different variants of the proposed MTEC framework.

Model ID	Accuracy		
	9 Days	49 Days	99 Days
1	81.09%	84.32%	85.14%
2	86.09%	88.15%	88.86%
3	85.94%	88.23%	89.21%
4	89.45%	92.78%	93.03%
5	87.54%	90.01%	90.98%
6	92.87%	94.13%	94.54%

Table 3
The accuracy of the Top-K popular content using different loss weights.

Model ID	w_1	w_2	w_3	w_4	Accuracy
L1	0.2	0.4	0.1	0.3	94.13%
L2	0.3	0.2	0.1	0.4	93.08%
L3	0.0	0.0	0.0	1.0	90.54%

GUs and 27,278 number of distinct contents. Following the common assumption [8], we consider the scenario that the storage capacity of caching nodes is 10% of the total content, where the size of all multimedia contents is the same. Considering the GUs' location, determined based on their ZIP code [40], it is assumed that there are 21, and 3 terrestrial, and areal caching nodes, respectively, where each intercluster consists of $N_s = 7$ number of FAPs. Note that the classification accuracy is averaged over all caching nodes. To determine the best architecture of the multiple-model Transformer-based architecture, we first evaluate different versions of the proposed MTEC popularity prediction framework through trial and error. Moreover, the performance of a single classification model, i.e., Path 1 or Path 2, is evaluated, when they are trained independently. In all the experiments, the Adam optimizer is employed with learning rate of 0.0001 and weight decay of 0.00001. The activation function of the MLP layers in all Transformer models is ReLU, while it is sigmoid as the output layer. In classification blocks, the multi-content time-series request pattern data is converted to a sequential set of images, using the Gramian Angular Field (GAF) technique [54] to preserve the temporal correlations of the input data.

4.1. Effectiveness of the MTEC architecture

This subsection evaluates the performance of the proposed MTEC architecture. Considering different hyperparameters, such as the number of heads, number of transformer layers, model dimension, and MLP size, we compare different variants of the proposed MTEC architecture in terms of accuracy and the number of parameters (complexity). As it can be seen in Table 1, increasing the number of heads from 8 to 16 (Models 1 and 2) and model dimension from 32 to 64 (Models 1 and 3) increase the accuracy of the proposed MTEC framework, while increasing the complexity of the learning model. Similarly, the classification accuracy is improved by increasing the number of transformer layers from 1 to 2 (see Models 3 and 4) and the MLP size from 256 to 512 (Models 3 and 5). Moreover, increasing the number of MLP layers from 1 to 2 (Models 1 and 7) results in a slight increase in the classification accuracy. Figs. 6 and 7 illustrate the classification accuracy and complexity of Transformer-based models. According to the information provided in Table 1, the best architecture for the proposed MTEC framework is Model 6. Subsequently, we assess the learning efficiency of the MTEC framework on Model 6, revealing train time, test time, and maximum allocated memory values of $17.21e-3$ s, $6.21e-3$ s, and $27e+9$ bytes, respectively. It should be noted that although the number of parameters in Model 6 is higher than the others, the classification accuracy is higher as well. For that reason, we choose this model to compare it with other state-of-the-art.

We also evaluate the effect of window size, which is used for data segmentation during the dataset pre-processing phase. The window

Table 4
Comparison between our methodology (MTEC framework) and other versions of the Transformer-based architectures.

Model name	Layers (L)	Model dimension (d)	MLP layers	MLP size	Number of heads	Parameters	Accuracy
TC	2	64	2	256	16	946,580	92.48%
RCT	2	64	2	256	16	1,909,860	92.54%
ViT	2	64	2	256	16	3,798,320	92.68%
MTEC	2	64	2	256	16	2,882,140	94.13%
MTEC-AT	2	64	2	256	16	2,882,396	93.28%

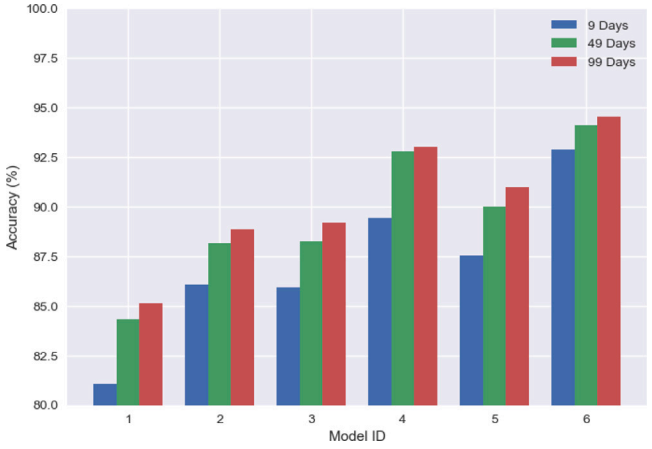


Fig. 8. Accuracy of Top-K Popular Content for Different Window Sizes.

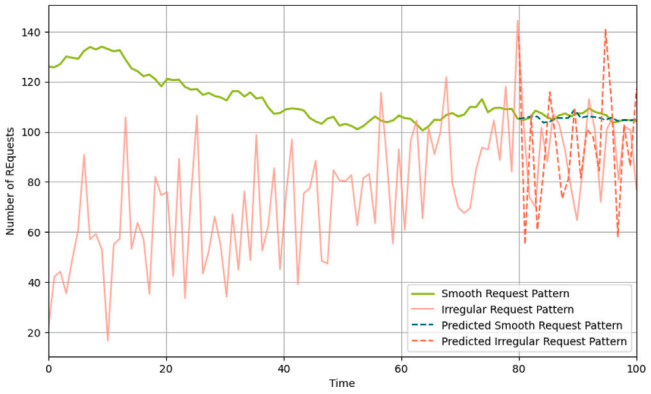


Fig. 9. Diverse historical request patterns of content, including smooth and irregular request patterns.

size, in this context, signifies the quantity of past request patterns for content used as input samples to predict the Top-K popular content in the future. For example, with a time window set at 9, it implies that the nine most recent requests for each content will contribute to the input sample. It is crucial to note that determining the optimal window size involves a trial-and-error approach. In our investigation, we explore three distinct values, i.e., a small window size (9), a medium size (49), and a large size (99), to assess the impact on the output. As indicated in Table 2, opting for a smaller window size leads to diminished classification accuracy. The primary reason is that a smaller window size requires the model to discern content popularity based on a limited number of requests, making it inefficient to capture temporal information over time. Upon increasing the window size from 9 to 49, and incorporating more timestamps into the input sample, we observe a notable enhancement in the classification accuracy. However, the gain in accuracy is marginal when further increasing the window size from 49 to 99. This experiment highlights that augmenting the window size correlates with an increase in classification accuracy, but excessively large window sizes may not be particularly efficient. Fig. 8 illustrates the classification accuracy over window sizes. Moreover, we assess the training time across different window sizes for Model 6. As detailed previously, Model 6 emerged as the best architecture, prompting further exploration in subsequent sections of the document. It is noteworthy that while increasing the window size enhances classification accuracy, it also prolongs the training time. For instance, in Model 6, the training times for window sizes 9, 49, and 99 are $4.98e-3$, $17.921e-3$, and $39.65e-3$ s, respectively. Moreover, Fig. 9 demonstrates how our RPP block predicts the future number of requests, which is then employed for the precise classification of content as popular or unpopular. This innovative approach allows us to achieve more accurate results, even when faced with unpredictable request probabilities.

We evaluate the effect of different loss functions on classification accuracy. According to Eq. (13), we consider different values for loss weight of each block, denoted by w_i, $i = \{1, \ldots, 4\}$, to illustrate the effect of loss weight on the overall classification accuracy. The total loss is calculated according to Eq. (13), with RPP and C1 losses associated with Request Probability Prediction and the Classification block in the first path, respectively. Additionally, C2 and Fusion losses represent the Classification loss in the second path and the fusion layer, respectively. More precisely, to delve deeper into the impact of various loss functions on classification accuracy, we explore different configurations of loss weights for each block, represented as w_i, where $i = 1, \ldots, 4$. The results, detailed in Table 3, illustrate the accuracy achieved by the MTEC framework under different weight assignments. For instance, in Model L1 with weights $w_1 = 0.2$, $w_2 = 0.4$, $w_3 = 0.1$, and $w_4 = 0.3$, the accuracy reaches 94.13%. Similarly, in Model L2, with a different weight distribution ($w_1 = 0.3$, $w_2 = 0.2$, $w_3 = 0.1$, $w_4 = 0.4$), the accuracy is recorded as 93.08%. Additionally, Model L3, characterized by weights $w_1 = 0.0$, $w_2 = 0.0$, $w_3 = 0.0$, and $w_4 = 1.0$, achieves an accuracy of 90.54%. These findings provide valuable insights into the sensitivity of the MTEC model to different loss weight configurations, guiding the optimization of the framework for enhanced accuracy based on specific priorities in content caching applications. Moreover, the convergence of the proposed MTEC framework is illustrated in Fig. 10. As shown in Fig. 10, increasing the number of epochs decreases the model loss, which shows that the model is well trained.

Moreover, to illustrate the superiority of the proposed MTEC architecture in comparison to a single Transformer, we compare it with different Transformer-based networks, such as the single TC model, corresponding to the first path of the MTEC model, the TRC network associated with the second path, the Vision Transformer (ViT) model [19], and the MTEC-AT, which is the proposed MTEC architecture with a self-attention layer as the fusion layer instead of the fully connected one. As indicated in Table 4, the classification accuracy for the TC architecture stands at 92.48%—a figure surpassed by the proposed MTEC architecture, which achieves an accuracy of 94.13%. It is crucial to underscore that a single multi-classification model, such as the TC architecture, is primarily suited for uncoded content placement. In this scenario, Top-K popular content is stored entirely in the storage of edge devices. However, when dealing with coded/uncoded content placement, the limitations of a single model become evident. To effectively classify the Top-K popular content into popular and mediocre files, it is imperative to consider the request probability of the content—a task seamlessly addressed by the MTEC framework. It is worth highlighting that while the training time for a standalone Transformer model is 5.89×10^{-3} – which is less than the

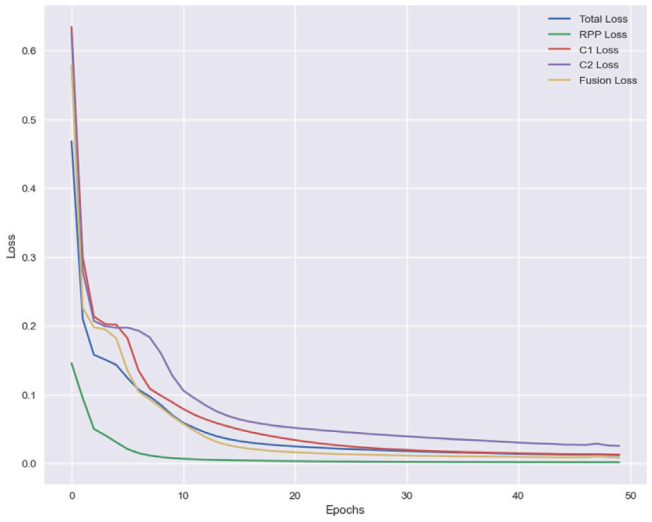


Fig. 10. The convergence of the proposed MTEC framework.

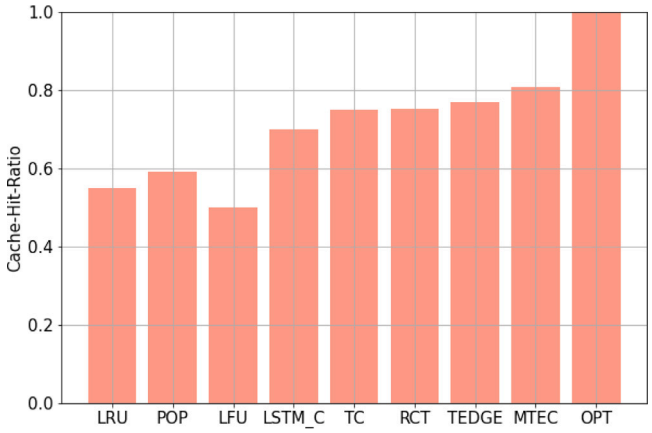


Fig. 11. A comparison with state-of-the-arts based on the cache-hit ratio.

17.21×10^{-3} needed for the MTEC framework – the decision between opting for a single Transformer architecture or the proposed MTEC framework is entirely contingent on the specific application. If the emphasis lies on minimizing training complexity and implementing uncoded content placement alone, the single Transformer architecture proves to be a favorable choice. Conversely, when the requirement involves coded/uncoded content placement in an MEC network, with a significant emphasis on enhancing content diversity and achieving critical classification accuracy, the proposed MTEC architecture emerges as the superior option.

4.2. Performance comparisons

For comparison purposes, we applied seven state-of-the-art caching schemes to the Movielens dataset [47], including Least Recently Used (LRU) [55], Least Frequently Used (LFU) [55], PopCaching [49], LSTM-C [29], the TC scheme, the RCT scheme, and the TEDGE caching scheme [19], which is based on the ViT architecture. Fig. 11 compares the performance of the proposed uncoded MTEC scheme with other baselines listed above in terms of the cache-hit ratio [44], known as one of the widely used metrics in MEC networks. This metric indicates the number of requests managed by caching nodes versus the total requests made across the network. Note that, the proposed MTEC scheme can be used for both coded/uncoded content placement and the conventional uncoded one. Other baselines, however, are based on uncoded content placement, which is one of the main drawbacks of the existing data-driven caching schemes. In such a case that the multimedia content is partially stored in the storage of caching nodes, the cache-hit ratio would not be an accurate metric for the coded/uncoded content placement framework. For this reason and to be compatible with other state-of-the-arts, we first evaluate the performance of the proposed uncoded MTEC scheme in terms of the cache-hit ratio. Then, we define another metric suitable for the coded/uncoded content placement, known as the transferred byte volume [44], illustrating the ratio of the data volume, transmitted by caching nodes versus the total volume of the requested contents managed by caching nodes. As shown in Fig. 11, the optimal (OPT) strategy [29] is defined as a caching scheme, where all requests through the network are served by caching nodes, which cannot be obtained in reality. More precisely, OPT represents an ideal caching strategy wherein all network requests are fulfilled by caching nodes, resulting in the highest achievable cache-hit ratio and transferred byte volume—performance levels unattainable in practical scenarios. This limitation arises from the fact that the storage capacity of edge devices is considerably smaller than the total content volume, preventing the storage of all content on edge devices. According to the results in Fig. 11, the proposed MTEC caching framework achieves the highest cache-hit ratio in comparison to other state-of-the-art counterparts.

In terms of the transferred byte volume, it is assumed that $\alpha = 30$ percent of the storage of FAPs is associated with the popular contents, stored completely, and 70% of the storage is assigned to mediocre content, stored partially according to the content placement strategy described in Section 2. As shown in Fig. 12, the byte volume transferred by caching nodes in the proposed MTEC framework is higher than other counterparts. In comparison to the Cluster-centric and Coded UAV-aided Femtocaching (CCUF) [44] framework, the coded/uncoded content placement in the CCUF is performed based on the historical request probability of content, while the proposed MTEC and RCT frameworks use the predicted one. Moreover, since the classification accuracy of the RCT model is lower than the proposed MTEC architecture, the MTEC framework outperforms in terms of the transferred byte volume.

We evaluate the classification accuracy of the proposed MTEC framework through a comprehensive comparison with various self-supervised, unsupervised, and supervised learning models. The results, systematically presented in Table 5, underscore the exceptional performance of the MTEC architecture compared to several baseline models. In the realm of unsupervised approaches, MTEC outperforms competitors such as Adaptive Genetic Neural Network (AGNN) [56] (81%) and Artificial Neural Network (ANN) with modified K-Means [56] (74%), achieving a notable accuracy of 94.13%. This represents a substantial improvement of 13.13% and 20.13%, respectively, over AGNN and ANN with modified K-Means. Additionally, when contrasted against state-of-the-art self-supervised contrastive learning methods, namely Self-Supervised Contrastive Learning using Random Feature Corruption (SCARF) [57] (87.17%) and Contrastive learning Popularity (CoPo) [58] (92.99%), the MTEC framework consistently demonstrates superior performance, showcasing a performance advantage of 6.96% over SCARF and 1.14% over CoPo. Furthermore, in a direct comparison with a supervised learning model, DLCC (92.81%), specifically tailored for content caching applications, the MTEC model surpasses DLCC with a substantial difference in accuracy, confirming its effectiveness in predicting the popularity of multimedia content.

5. Conclusion

In this paper, we presented an efficient multi-content time-series popularity prediction model referred to as the Multiple-model Transformer-based Edge Caching (MTEC) framework with the application to the cluster-centric Mobile Edge Caching (MEC) networks. Due to

Table 5
Comparison with state-of-the-art based on the classification accuracy.

Model	Self-supervised/unsupervised learning				Supervised learning	
	CoPo [58]	AGNN [56]	ANN + Modified K-Means [56]	SCARF [57]	MTEC	DLCC [59]
Accuracy	92.99%	81%	74%	87.17%	94.13%	92.81%

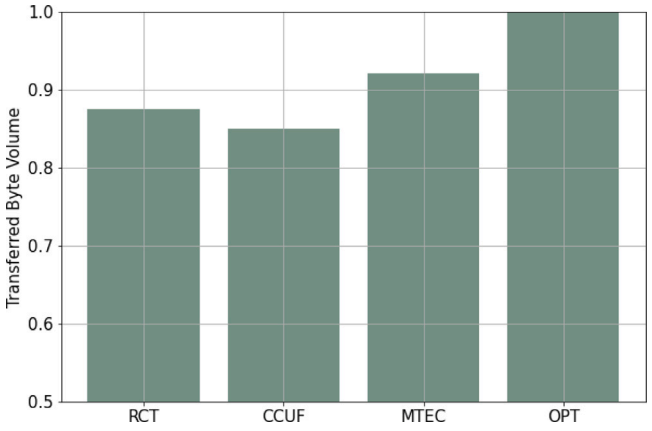


Fig. 12. A comparison with state-of-the-arts based on the transferred byte volume.

the lack of predicted request probability, existing data-driven caching strategies were inefficient for coded/uncoded content placement. To tackle this issue, we developed a multiple-model Transformer-based architecture to not only predict the upcoming Top-K popular content but also estimate the request pattern of multiple contents simultaneously, which was used to determine which contents should be stored partially or completely. Simulation results showed that the proposed MTEC caching scheme improves the cache-hit ratio and the transferred byte volume when compared to its state-of-the-art counterparts. A fruitful direction for future research is to implement an integrated Digital Twin (DT)-based popularity prediction model for MEC networks, where mobile users and edge caching nodes cooperatively learn to manage and adapt to changing users' preferences.

CRediT authorship contribution statement

Zohreh Hajiakhondi Meybodi: Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Data curation, Conceptualization. **Arash Mohammadi:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Methodology, Funding acquisition, Formal analysis, Conceptualization. **Ming Hou:** Writing – review & editing, Conceptualization. **Elahe Rahimian:** Validation, Software, Data curation. **Shahin Heidarian:** Software, Data curation, Conceptualization. **Jamshid Abouei:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology. **Konstantinos N. Plataniotis:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] M.K. Somesula, R.R. Rout, D.V.L.N. Somayajulu, Greedy cooperative cache placement for mobile edge networks with user preferences prediction and adaptive clustering, Ad Hoc Netw. 140 (2023) 103051.
[2] R. Singh, R. Sukapuram, S. Chakraborty, A survey of mobility-aware multi-access edge computing: Challenges, use cases and future directions, Ad Hoc Netw. 140 (2023) 103044.
[3] P. Liu, Y. Zhang, T. Fu, J. Hu, Intelligent mobile edge caching for popular contents in vehicular cloud toward 6G, IEEE Trans. Veh. Technol. 70 (6) (2021) 5265–5274.
[4] N. Abbas, Y. Zhang, A. Taherkordi, T. Skeie, Mobile edge computing: A survey, IEEE Internet Things J. 5 (1) (2018) 450–465.
[5] L.U. Khan, I. Yaqoob, N.H. Tran, S.M.A. Kazmi, T.N. Dang, C.S. Hong, Edge-computing-enabled smart cities: A comprehensive survey, IEEE Internet Things J. 7 (10) (2020) 10200–10232.
[6] A.A. Chowdhury, I. Islam, M.I.A. Zahed, I. Ahmad, An optimal strategy for UAV-assisted video caching and transcoding, Ad Hoc Netw. 144 (2023) 103155.
[7] A. Tout, S. Sharafeddine, N. Abbas, UAV-assisted multi-tier computing framework for IoT networks, Ad Hoc Netw. 142 (2023) 103119.
[8] Z. Hajiakhondi-Meybodi, J. Abouei, A.H.F. Raouf, Cache replacement schemes based on adaptive time window for video on demand services in femtocell networks, IEEE Trans. Mob. Comput. 18 (7) (2019) 1476–1487.
[9] Z. HajiAkhondi-Meybodi, J. Abouei, M. Jaseemuddin, A. Mohammadi, Mobility-aware femtocaching algorithm in D2D networks based on handover, IEEE Trans. Veh. Technol. 69 (9) (2020) 10188–10201.
[10] Z. Chen, J. Lee, T.Q.S. Quek, M. Kountouris, Cooperative caching and transmission design in cluster-centric small cell networks, IEEE Trans. Wireless Commun. 16 (5) (2017) 3401–3415.
[11] P. Lin, Q. Song, A. Jamalipour, Multidimensional cooperative caching in CoMP-integrated ultra-dense cellular networks, IEEE Trans. Wirel. Commun. 19 (3) (2020) 1977–1989.
[12] A.C. Kazez, T. Girici, Clustering-based device-to-device cache placement, Ad Hoc Netw. 84 (2019) 170–177.
[13] P. Lin, K.S. Khan, Q. Song, A. Jamalipour, Caching in heterogeneous ultradense 5G networks: A comprehensive cooperation approach, IEEE Veh. Technol. Mag. 14 (2) (2019) 22–32.
[14] K.Y. Tsao, T. Girdler, V.G. Vassilakis, A survey of cyber security threats and solutions for UAV communications and flying ad-hoc networks, Ad Hoc Netw. 133 (2022) 102894.
[15] R. Masroor, M. Naeem, W. Ejaz, Resource management in UAV-assisted wireless networks: An optimization perspective, Ad Hoc Netw. 121 (2021) 102596.
[16] Z. Hajiakhondi-Meybodi, A. Mohammadi, J. Abouei, Deep reinforcement learning for trustworthy and time-varying connection scheduling in a coupled UAV-based femtocaching architecture, IEEE Access 9 (2021) 32263–32281.
[17] Y. Dai, Y. Zhang, Adaptive digital twin for vehicular edge computing and networks, J. Commun. Inf. Netw. 7 (1) (2022) 48–59.
[18] K. Zhang, J. Cao, S. Maharjan, Y. Zhang, Digital twin empowered content caching in social-aware vehicular edge networks, IEEE Trans. Comput. Soc. Syst. 9 (1) (2022) 239–251.
[19] Z. Hajiakhondi-Meybodi, A. Mohammadi, E. Rahimian, S. Heidarian, J. Abouei, K.N. Plataniotis, TEDGE-caching: Transformer-based edge caching towards 6G networks, in: IEEE International Conference on Communications, ICC, 2022, Accepted.
[20] B.M. Marlin, R.S. Zemel, S.T. Roweis, M. Slaney, Recommender systems: missing data and statistical model estimation, in: Twenty-Second International Joint Conference on Artificial Intelligence, 2011.
[21] A. Odic, M. Tkalcic, J.F. Tasic, A. Kosir, Predicting and detecting the relevant contextual information in a movie-recommender system, Interact. Comput. 25 (1) (2013) 74–90.
[22] S.M.R. Abidi, X. Yonglin, N. Jianyue, W. Xiangmeng, W. Zhang, Popularity prediction of movies: from statistical modeling to machine learning techniques, Multimedia Tools Appl. 79 (47) (2020) 35583–35617.
[23] V.K.Y. Ng, R.A. Cribbie, The gamma generalized linear model, log transformation, and the robust Yuen-Welch test for analyzing group means with skewed and heteroscedastic data, Comm. Statist. Simulation Comput. 48 (8) (2019) 2269–2286.

[24] R.R. Kabra, R.S. Bichkar, Performance prediction of engineering students using decision trees, Int. J. Comput. Appl. 36 (11) (2011) 8–12.
[25] G. Mendez, T.D. Buskirk, S. Lohr, S. Haag, Factors associated with persistence in science and engineering majors: An exploratory study using classification trees and random forests, J. Eng. Educ. 97 (1) (2008) 57–70.
[26] K.N. Doan, T. Van Nguyen, T.Q.S. Quek, H. Shin, Content-aware proactive caching for backhaul offloading in cellular network, IEEE Trans. Wireless Commun. 17 (5) (2018) 3128–3140.
[27] L. Ale, N. Zhang, H. Wu, D. Chen, T. Han, Online proactive caching in mobile edge computing using bidirectional deep recurrent neural network, IEEE Internet Things J. 6 (3) (2019) 5520–5530.
[28] Q. Fan, X. Li, J. Li, Q. He, K. Wang, J. Wen, PA-cache: Evolving learning-based popularity- aware content caching in edge networks, IEEE Trans. Netw. Serv. Manag. 18 (2) (2021) 1746–1757.
[29] C. Zhang, et al., Toward edge-assisted video content intelligent caching with long short-term memory learning, IEEE Access 7 (2019) 152832–152846.
[30] S. Rathore, J.H. Ryu, P.K. Sharma, J.H. Park, DeepCachNet: A proactive caching framework based on deep learning in cellular networks, IEEE Netw. 33 (3) (2019) 130–138.
[31] Y. Lin, C. Yen, J. Wang, Video popularity prediction: An autoencoder approach with clustering, IEEE Access 8 (2020) 129285–129299.
[32] C. Zhong, M.C. Gursoy, S. Velipasalar, Deep reinforcement learning-based edge caching in wireless networks, IEEE Trans. Cognit. Commun. Netw. 6 (1) (2020) 48–61.
[33] P. Wu, J. Li, L. Shi, M. Ding, K. Cai, F. Yang, Dynamic content update for wireless edge caching via deep reinforcement learning, IEEE Commun. Lett. 23 (10) (2019) 1773–1777.
[34] Y. Wang, Y. Li, T. Lan, V. Aggarwal, DeepChunk: Deep Q-learning for chunk-based caching in wireless data processing networks, IEEE Trans. Cognit. Commun. Netw. 5 (4) (2019) 1034–1045.
[35] K.C. Tsai, L. Wang, Z. Han, Mobile social media networks caching with convolutional neural network, in: IEEE Wireless Communications and Networking Conference Workshops, 2018, pp. 83–88.
[36] H. Mou, Y. Liu, L. Wang, LSTM for mobility based content popularity prediction in wireless caching networks, in: IEEE Globecom Workshops, 2019, pp. 1–6.
[37] Z. Yu, J. Hu, G. Min, Z. Zhao, W. Miao, M.S. Hossain, Mobility-aware proactive edge caching for connected vehicles using federated learning, IEEE Trans. Intell. Transp. Syst. 22 (8) (2021) 5341–5351.
[38] C. Zheng, S. Liu, Y. Huang, W. Zhang, L. Yang, Unsupervised recurrent federated learning for edge popularity prediction in privacy-preserving mobile-edge computing networks, IEEE Internet Things J. 9 (23) (2022) 24328–24345.
[39] S. Liu, C. Zheng, Y. Huang, T.Q.S. Quek, Distributed reinforcement learning for privacy-preserving dynamic edge caching, IEEE J. Sel. Areas Commun. 40 (3) (2022) 749–760.
[40] A. Ndikumana, N.H. Tran, D.H. Kim, K.T. Kim, C.S. Hong, Deep learning based caching for self-driving cars in multi-access edge computing, IEEE Trans. Intell. Transp. Syst. 22 (5) (2021) 2862–2877.
[41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. (2017) 5998–6008.
[42] M.-T. Nguyen, D.H. Le, T. Nakajima, M. Yoshimi, N. Thoai, Attention-based neural network: A novel approach for predicting the popularity of online content, in: IEEE International Conference on High Performance Computing and Communications, China, 2019, pp. 329–336.
[43] Z. Zhang, M. Tao, Deep learning for wireless coded caching with unknown and time-variant content popularity, IEEE Trans. Wireless Commun. 20 (2) (2021) 1152–1163.
[44] Z. Hajiakhondi-Meybodi, A. Mohammadi, J. Abouei, M. Hou, K.N. Plataniotis, Joint transmission scheme and coded content placement in cluster-centric UAV-aided cellular networks, IEEE Internet Things J. 9 (13) (2022) 11098–11114.
[45] Z.M. Fadlullah, N. Kato, HCP: Heterogeneous computing platform for federated learning based collaborative content caching towards 6G networks, IEEE Trans. Emerg. Top. Comput. (2020).
[46] X. Wang, C. Wang, X. Li, V.C.M. Leung, T. Taleb, Federated deep reinforcement learning for internet of things with decentralized cooperative edge caching, IEEE Internet Things J. 7 (10) (2020) 9441–9455.
[47] F.M. Harper, J.A. Konstan, The movielens datasets: History and context, Acm Trans. Interact. Intell. Syst. 5 (4) (2015) 1–19.
[48] S. Dernbach, N. Taft, J. Kurose, U. Weinsberg, C. Diot, A. Ashkan, Cache content-selection policies for streaming video services, in: IEEE International Conference on Computer Communications, INFOCOM, 2016, pp. 1–9.
[49] S. Li, J. Xu, M. van der Schaar, W. Li, Popularity-driven content caching, in: IEEE International Conference on Computer Communications, INFOCOM, 2016, pp. 1–9.
[50] G. Vallero, M. Deruyck, W. Joseph, M. Meo, Caching at the edge in high energy-efficient wireless access networks, in: IEEE International Conference on Communications, ICC, 2020, pp. 1–7.
[51] C. Joseph, H. Hong, J.C. Stein, Forecasting crashes: Trading volume, past returns, and conditional skewness in stock prices, J. Financial Econ. 61 (3) (2001) 345–381.
[52] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, C. Eickhoff, A transformer-based framework for multivariate time series representation learning, in: Proceedings of the ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 2114–2124.
[53] J.L. Ba, J.R. Kiros, G.E. Hinton, Layer normalization, 2016, arXiv preprint arXiv:1607.06450.
[54] Y. Hong, J.J.F. Martinez, A.C. Fajardo, Day-ahead solar irradiation forecasting utilizing gramian angular field and convolutional long short-term memory, IEEE Access 8 (2020) 18741–18753.
[55] A. Giovanidis, A. Avranas, Spatial multi-LRU caching for wireless networks with coverage overlaps, ACM SIGMETRICS Perform. Eval. Rev. 44 (1) (2016) 403–405.
[56] C. Selvi, E. Sivasankar, A novel adaptive genetic neural network (AGNN) model for recommender systems using modified k-means clustering approach, Multimedia Tools Appl. 78 (2019) 14303–14330.
[57] D. Bahri, H. Jiang, Y. Tay, D. Metzler, Scarf: Self-supervised contrastive learning using random feature corruption, 2022, arXiv preprint arXiv:2106.15147.
[58] Z. HajiAkhondi-Meybodi, et al., CoPo: Self-supervised contrastive learning for popularity prediction in MEC networks, IEEE Digit. Signal Process. (2023).
[59] S. Bhandari, N. Ranjan, P. Khan, H. Kim, Y.S. Hong, Deep learning-based content caching in the fog access points, Electronics 10 (4) (2021) 512.

Zohreh Hajiakhondi-Meybodi received the B.Sc. degree in Communication Engineering from Yazd University, Yazd, Iran and the M.Sc. degree in Communication Systems Engineering (with the highest honor) from Yazd University, Yazd, Iran in 2013 and 2017, respectively. She is a Ph.D. degree candidate at Electrical and Computer Engineering (ECE), Concordia University, Montreal, Canada. Since 2019, she has been an active member of I-SIP Lab at Concordia University. Her research interests include general areas of wireless communication networks with a particular emphasis on Femtocaching, Internet of Things (IoT), Indoor Localization, Optimization Algorithms, and Multimedia Wireless Sensor Networks (WMSN).

Arash Mohammadi (S'08-M'14-SM'17) received B.Sc. degree form ECE Department at University of Tehran, Tehran, Iran, in 2005, the M.Sc. degree from BME Department at Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran, in 2007, and Ph.D. degree from EECS Department at York University, in 2013. From 2013 to 2015, he was a Post-Doctoral Fellow at the Multimedia Lab, in the ECE Department, at the University of Toronto. He is currently an Associate Professor with Concordia Institute for Information Systems Engineering (CIISE), Concordia University, Montreal, QC, Canada. He is a registered Professional Engineer in Ontario. He was Director of Membership Developments of IEEE Signal Processing Society (2018–2021), and the General CoChair of 2021 IEEE International Conference on Autonomous Systems (ICAS). Additionally, he was a member of the Organizing Committee of 2023 IEEE Intelligent Vehicles Symposium (IV 2023), 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), and 2021 IEEE International Conference on Image Processing (ICIP). Currently he is the Program Chair of IEEE International Conference on Human–Machine Systems (IEEE ICHMS) and is on the editorial board of IEEE Signal Processing Letters and Nature Scientific Reports. Dr. Mohammadi is recipient of several distinguishing awards, including the Eshrat Arjomandi Award for outstanding Ph.D. dissertation from EECS, York University, 2022 Concordia University Research Award of Excellence, 2018 Concordia President's Excellence in Teaching Award, and 2019 and 2022 Gina Cody School of Engineering and Computer Science's Research awards.

Elaheh Rahimian received the B.Sc. degree in Electrical Engineering from Isfahan University of Technology, Iran, the M.Sc. degree in Electrical Engineering from Amirkabir University of Technology - Tehran Polytechnic, Iran, and the Ph.D. degree in Information System Engineering from Concordia University, Montreal, Canada, in 2012, 2016, and 2022, respectively. Her research interests include Meta Learning, Machine Learning, Deep Learning, AI, and Neurorehabilitation Technologies.

Shahin Heidarian received the B.Sc. degree in Electrical Engineering from Iran University of Science and Technology, Iran and the M.Sc. degree in Electrical and Computer Engineering from Concordia University, Montreal, Canada, in 2019, and 2021, respectively. His research interests include Deep Learning, Medical Image Processing, and Machine Learning.

Jamshid Abouei (S05, M11, SM13) received the B.Sc. degree in Electronics Engineering and the M.Sc. degree in Communication Systems Engineering (with the highest honor) both from Isfahan University of Technology (IUT), Iran, in 1993 and 1996, respectively, and the Ph.D. degree in Electrical Engineering from University of Waterloo, Canada, in 2009. He joined with the Department of Electrical Engineering, Yazd University, Iran, in 1996 (as a Lecturer) and was promoted to Assistant Professor in 2010, and Associate Professor in 2015. From 1998 to 2004, he served as a Technical Advisor and Design Engineer in the R & D Center and Cable Design Department in SGCC, Iran. From 2009 to 2010, he was a Postdoctoral Fellow in the Multimedia Lab, in the Department of Electrical & Computer Engineering, University of Toronto, Canada, and worked as a Research Fellow at the Self-Powered Sensor Networks (ORF-SPSN) consortium. During his sabbatical, he was an Associate Researcher in the Department of Electrical, Computer and Biomedical Engineering, Ryerson University, Toronto, Canada. Dr Abouei was the International Relations Chair in 27th ICEE2019 Conference, Iran, in 2019. Currently, Dr Abouei directs the research group at the Wireless Networking Laboratory (WINEL), Yazd University, Iran. His research interests are in the next generation of wireless networks (5G) and wireless sensor networks (WSNs), with a particular emphasis on PHY/MAC layer designs including the energy efficiency and optimal resource allocation in cognitive cell-free massive MIMO networks, multi-user information theory, mobile edge computing and femtocaching. Dr Abouei is a Senior IEEE member and a member of the IEEE Information Theory. He has received several awards and scholarships, including FOE and IGSA awards for excellence in research in University of Waterloo, Canada, MSRT Ph.D. Scholarship from the Ministry of Science, Research and Technology, Iran in 2004, Distinguished Researcher award in province of Yazd, Iran, 2011, and Distinguished Researcher award in Electrical Engineering Department, Yazd University, Iran, 2013. He is a recipient of the best paper award for the IEEE Iranian Conference on Electrical Engineering (ICEE 2018).

Konstantinos N. (Kostas) Plataniotis is a Professor and the Bell Canada Chair in Multimedia with the ECE Department at the University of Toronto. He is the founder and inaugural Director-Research for the Identity, Privacy and Security Institute (IPSI) at the University of Toronto and he has served as the Director for the Knowledge Media Design Institute (KMDI) at the University of Toronto from January 2010 to July 2012. His research interests are: knowledge and digital media design, multimedia systems, biometrics, image & signal processing, communications systems and pattern recognition. Among his publications in these fields are the books entitled 'WLAN positioning systems' (2012) and 'Multi-linear subspace learning: Reduction of multi-dimensional data' (2013). Dr. Plataniotis is a registered professional engineer in Ontario, Fellow of the IEEE and Fellow of the Engineering Institute of Canada. He has served as the Editor-in-Chief of the IEEE Signal Processing Letters, and as Technical Co-Chair of the IEEE 2013 International Conference in Acoustics, Speech and Signal Processing. He was the IEEE Signal Processing Society Vice President for Membership (2014–2016). He is the General Co-Chair for 2017 IEEE GlobalSIP, the 2018 IEEE International Conference on Image Processing (ICIP 2018), and the 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2021).