CoPo: Self-supervised Contrastive Learning for Popularity Prediction in MEC Networks

Zohreh Hajiakhondi Meybodi
Electrical and Computer Eng.
Concordia University
Montreal, Canada

Arash Mohammadi
Concordia Ins. for Info. Sys. Eng.
Concordia University
Montreal, Canada

Jamshid Abouei
Dep. of Electrical Eng.
Yazd University
Yazd, Iran

Konstantinos N. Plataniotis
Electrical & Computer Eng.
University of Toronto,
Toronto, Canada

***Abstract*—Mobile Edge Caching (MEC) technology aims to provide high-quality multimedia content to mobile users by bringing storage and computation resources closer to the edge of the network. MEC networks, however, face several challenges such as limited storage capacity, dynamic network conditions, and the need for low-latency content delivery. To address these challenges, recent research has focused on integrating MEC networks with Deep Neural Networks (DNNs), in particular, supervised learning models. One significant limitation of supervised popularity prediction models is the requirement for manual labeling of contents as popular or unpopular by investigating users' past behavior, which can be a time-intensive task. This paper proposes a self-supervised learning algorithm called Contrastive learning Popularity (CoPo) prediction framework to predict the dynamic content popularity in a MEC network. The framework utilizes the distinguishing aspect of the Contrastive Learning (CL) paradigm to recognize differences among input samples, including users' contextual information and is based on the Long Short Term Memory (LSTM) model to capture temporal information. Simulation results illustrate that the proposed CoPo framework outperforms the self-supervised/unsupervised state-of-the-art methods.**

***Index Terms*—Mobile Edge Caching, Popularity Prediction, Self-supervised Learning, Contrastive Learning**

I. Introduction

Mobile Edge Caching (MEC) is a new paradigm in mobile networks that aims to address the growing demand for high-quality multimedia content by bringing storage closer to mobile users [1], [2]. MEC technology has the potential to revolutionize the way we consume data on mobile devices by reducing latency and improving the overall user experience. MEC networks, however, face several challenges, such as limited storage capacity, dynamic network conditions, and the need to provide low-latency content delivery to users. These challenges can make it difficult to optimize caching decisions and ensure that users receive the content they need in a timely and efficient manner. To address these challenges, the focus of recent researchers has shifted to integrating MEC networks with Deep Neural Networks (DNNs) [3]–[15]. More precisely, DNNs can optimize caching decisions in MEC networks by processing large amounts of data, adapting to changing network conditions in real time, and personalizing content delivery based on user preferences.

This Project was partially supported by the Department of National Defence's Innovation for Defence Excellence & Security (IDEaS), Canada.
979-8-3503-3959-8/23/$31.00 ©2023 IEEE

A large variety of DNN models [3]–[15] has been recently developed for popularity prediction in MEC networks, including but not limited to Vision Transformers (ViT) [8], [9], Long Short Term Memory (LSTM) [10], [11], and Convolutional Neural Network (CNN) [15]. Despite the various advantages of existing DNN techniques, a significant limitation of most of these models is their reliance on supervised learning. This means that these models require labeled samples to train, making it difficult to apply them to real-world problems where large amounts of labeled data may not be readily available. In cases where the dataset used for a study is unlabeled, manual labeling becomes necessary, which can be time-consuming and expensive. This need for manual labeling highlights one of the significant challenges of existing DNN-based popularity prediction frameworks, emphasizing the importance of developing highly accurate unsupervised/self-supervised learning models that can learn from unlabeled data, making them more adaptable to real-world situations. The paper aims to further advance this emerging field.

Literature Review: Recently, supervised learning models have been widely used to predict the popularity of contents in MEC networks. For instance, Doan *et al.* [16] introduced a CNN-based content-aware popularity prediction framework. Ndikumana *et al.* [17] employed CNN and Multi-Layer Perceptron (MLP) in their study to make predictions about both the characteristics of users and the likelihood of content requests. Additionally, Reinforcement Learning (RL) has been used to model real-time interactions between users and edge devices. For instance, Tang *et al.* [18] utilized a framework based on Deep Reinforcement Learning (DRL) to simulate and predict the requesting behaviors of users. Sadeghi *et al.* [19] proposed an adaptive caching framework for hierarchical networks that utilized a DRL model to represent the interactive influence between cloud and edge devices. These models, however, are not suitable for highly dynamic practical networks, as they assume that the popularity of content will remain constant over time.

Recent studies have shifted their focus towards developing supervised prediction models that can process sequential and time-variant historical request patterns. For instance, the LSTM model has been used in various studies, including those by [10], [11], [20], to predict the number of requests of contents in the future. While LSTM is effective at learning long-term dependencies, it is not able to account for the spatial correlation between multiple contents. Transformer architecture [21], [22] is another type of time-series learning model that does not require sequential data to be analyzed in the same order. In our previous work [12], we utilized a multi-channel Transformer architecture that assigns each channel to the historical request pattern of specific content to capture the spatial correlation of contents. In our other works [8], [9], we employed the ViT architecture to simultaneously predict multiple contents popularities by creating 2D images as the input samples, where each column of the 2D image was linked to the request pattern of one specific content. While the supervised learning models proposed in the aforementioned works [8], [9], [12] provide high accuracy for classifying contents as popular/unpopular, their complexity increases exponentially with an increase in the number of contents. Furthermore, the absence of contextual information regarding users, such as age and gender, makes it challenging to effectively capture the unique interests of individual users. Finally, a significant limitation of these models is the requirement for manual labeling of contents by investigating users' past behavior, which can be a time-intensive task. This paper aims to tackle these issues.

Contributions: In light of the above discussion, the objective of this study is to design a self-supervised learning algorithm to predict the dynamic content popularity in a MEC network. Referred to as the Contrastive learning Popularity (CoPo) prediction framework, it utilizes the distinguishing aspect of the CL paradigm, which involves recognizing differences among input samples, thereby eliminating the need for simultaneously input request patterns of all contents to capture spatial correlations. More precisely, the historical requests of each content are considered as the input sample, including users' contextual information, and are fed into a shared encoder which is based on the LSTM model to capture the temporal information. The simulation results demonstrate that, while the CoPo framework's classification accuracy is slightly lower than that of supervised learning popularity prediction models, the CoPo framework does not need manual labeling of content in a supervised manner. Furthermore, the CoPo framework outperforms the unsupervised state-of-the-art methods.

II. System Model and Problem Description

We study an MEC network that is supported by Unmanned Aerial Vehicles (UAVs) acting as aerial caching nodes and Femto Access Points (FAPs) as terrestrial infrastructure, and both have a limited storage capacity, denoted by K. The network consists of two sets: the set of UAVs, denoted by u_s, where s ranges from 1 to N_u; and the set of FAPs, denoted by f_i, with i ranging from 1 to N_f. The users in the network are denoted by u_l, where l varies from 1 to U, and they request different multimedia contents, identified by c_m, where m ranges from 1 to M. This completes the presentation of the system model. Next, we present the dataset used in this study.

Dataset: In this study, the proposed CoPo framework is evaluated using the MovieLens dataset [23], which is a popular movie recommendation service. The MovieLens dataset includes a document named *u.user*, which presents users' contextual information such as gender, age, occupation, and ZIP code, as depicted in Fig. 1(a). The ZIP codes are transformed into latitude and longitude coordinates to retrieve the locations of the users during their requests. Additionally, the dataset provides another document named *u.data*, comprising the user ID, item ID, the corresponding rating given by the user, and the timestamp when the user watched and rated the content.

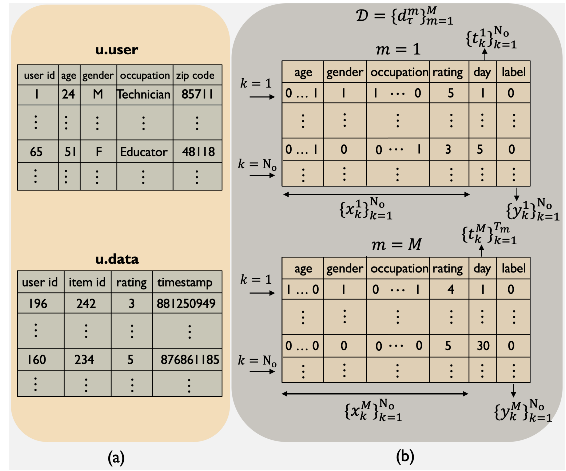

Fig. 1. (a) A typical sample of the Movielens dataset, and (b) the adopted version of the Movielens dataset used for the CoPo framework.

To prepare the Movielens dataset for use with the CoPo framework, several steps are taken. First, the *u.data* and *u.user* documents are combined based on the shared column "user ID", as depicted in Fig. 1(b), and the ZIP code information is discarded. Next, the combined dataset is sorted by "item ID" and "timestamp". The "user ID" and "item ID" columns are then removed as they do not provide relevant information. Categorical features such as gender, age, and occupation are encoded using a one-hot encoder. The timestamp column is discretized with a resolution of one day to ensure edge device storage is updated during off-peak times, and the new column is called "day". Finally, a new column called "label" is added to the dataset to indicate the content's popularity as either popular or unpopular, which is used in the fine-tuning phase to evaluate the performance of the proposed CoPo framework.

To predict the popularity of multimedia content in the future, we define an observational window for each content c_m with a length of T_τ^m at time τ. This window allows us to study the request pattern of contents within this range and predict their popularity within the time window $[\tau, \tau + T_s]$, where T_s represents the study window. We follow the approach in Reference [24] by considering the same number of requests, N_o, in the observational window for all contents, with zero padding for those with fewer requests. We assume that the study window has a length of $T_s = 1$ since the storage capacity of edge devices is updated daily. We create a dataset, $\mathcal{D} = \{d_\tau^m\}_{m=1}^M$, comprising time-series observational data for M multimedia contents and U users, where $d_\tau^m = \{(x_k^m, t_k^m, y_k^m)\}_{k=1}^{N_o}$. The

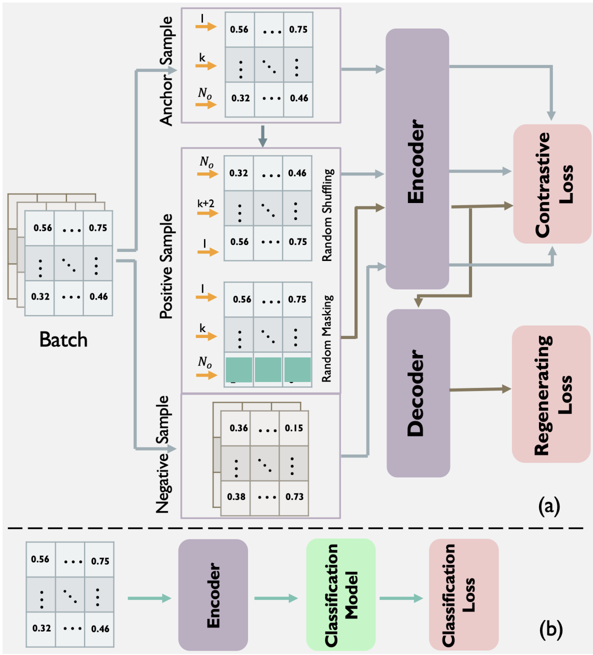

Fig. 2. The overall perspective of the CoPo architecture.

contextual information of users requesting content c_m and the rating they gave to content c_m are included in $\{x_k^m\}_{k=1}^{N_o}$ (see Fig. 1(b)). The rating time is represented by $\{t_k^m\}_{k=1}^{N_o}$, and $y_k^m \in \{0,1\}$ indicates content popularity. Specifically, y_k^m is set to 1 if content c_m becomes popular during the study window T_s; otherwise, $y_k^m = 0$.

III. Proposed CoPo Framework

This section outlines the constituent parts of the CoPo framework. The CoPo architecture, depicted in Fig.2(a), is built upon the CL network, which is used to learn the latent representation in which similar samples are positioned close together, while dissimilar samples are positioned far apart. In each batch, one input sample is selected as the anchor sample, and data augmentation is used to create a positive pair with the anchor sample, while the other samples in the batch are considered negative samples. In this study, we employ two data augmentation methods to produce positive samples (wherever possible, we remove the subscript τ to simplify the notation): (i) *Masking:* To create a positive sample for content c_m, which is represented by the longitudinal data d_m, several users' information $\{x_k^m\}_{k=1}^{N_o}$ is randomly masked. This results in a new positive sample, denoted by $d_m^{(MA)}$, and (ii) *Shuffling:* This data augmentation method involves generating a positive sample for content c_m by randomly rearranging the time order of the users' information that requested the content. The resulting shuffled sample is denoted by $d_m^{(SH)}$. Then, the augmented and anchor samples are processed by a shared encoder. To preserve the temporal correlation of request patterns, the encoder is built upon the LSTM architecture, where the latent representation of x_k^m at time t_k^m, denoted by $h_k^m = LSTM(x_k^m, h_{k-1}^m)$ where $k = \{1, \ldots, N_o\}$.

Similarly, the encoded shuffled and masked samples are denoted as $h_k^{m,(SH)}$ and $h_k^{m,(MA)}$, respectively. To complete the process, we apply the masked and shuffled CL loss functions, which are denoted as $L_{cl}^{(MA)}$ and $L_{cl}^{(SH)}$, respectively. In these loss functions, the shuffled/masked learned representation is considered the positive sample, while the other representations in the batch are regarded as negative samples. The loss functions are expressed as follows

$$L_{cl}^{(MA)} = -\sum_{m=1}^{M} \log \frac{\exp\left(h_{N_o}^m (h_{N_o}^{m,(MA)})^T\right)}{\sum\limits_{j=1, j\neq m}^{M} \exp\left(h_{N_o}^m (h_{N_o}^{j,(MA)})^T\right)}, \quad (1)$$

$$L_{cl}^{(SH)} = -\sum_{m=1}^{M} \log \frac{\exp\left(h_{N_o}^m (h_{N_o}^{m,(SH)})^T\right)}{\sum\limits_{j=1, j\neq m}^{M} \exp\left(h_{N_o}^m (h_{N_o}^{j,(SH)})^T\right)}, \quad (2)$$

where $(.)^T$ is the transpose function, and the total CL loss is $L_{cl} = L_{cl}^{(MA)} + L_{cl}^{(SH)}$. Furthermore, the Time-LSTM2 [25] is used as the decoder to recreate the masked version of x_k^m, where the output, denoted by $\overline{x}k^m$, is given by $\overline{x}_k^m = Time-LSTM2(H_k^m)$, where $H_k^m = [(h_1^{m,(MA)}, t_2^m - t_1^m), (h_2^{m,(MA)}, t_3^m - t_2^m), \ldots, (h_k^{m,(MA)}, t_{k+1}^m - t_k^m)]$ with $k \in \{1, \ldots, N_o\}$, where each pair consists of a masked encoded sample $h_k^{m,(MA)}$ at time t_k^m and the time difference between two consecutive requests of content m, which is $t_k^m - t_{k-1}^m$. The regenerating loss, denoted by L_{re}, is given by

$$L_{re} = \sum_{k=1}^{N_o} ||\overline{x}_k^m - x_k^m||^2, \quad (3)$$

where the goal is to minimize L_{re}. Note that the self-supervised CoPo framework is used for pre-training, where the shared encoder is taught to map input samples into meaningful latent representations. Following the training of the CL model, a supervised classifier is implemented for fine-tuning. In Fig. 2(b), the encoder that has been trained during the pre-training phase is employed to transform input samples into latent representation. To demonstrate the efficiency of the proposed CoPo framework, a logistic regression model is utilized during the fine-tuning process, where the labels $\{y_k^m\}_{m=1}^{M}$ are used as the true values.

IV. Simulation Results

A UAV-aided MEC network was studied, comprising four ground-based caching nodes and two aerial ones, and has a total of 943 users and 1682 multimedia items. Based on the typical assumption, the storage capacity of each caching node is 10% of the entire multimedia collection, with all items having identical sizes. The model was trained using a five-fold cross-validation approach, with 80% of the samples being used for training and 20% for testing. The Adam optimizer was employed during training, with betas set at $(0.9, 0.999)$ and weight decay at $1e-7$. To avoid overfitting, l_2 regularization was set at $1e-6$. The encoder in this case is implemented using an LSTM network, which utilizes a Rectified Linear Unit (ReLU) and sigmoid activation functions. The output

TABLE I
ACCURACY, PRECISION, RECALL, AND F1-SCORE FOR TWO CLASSES (I.E., POPULAR (CLASS 1) AND UNPOPULAR (CLASS 0)) USING 5 FOLD CROSS-VALIDATION.

	Class	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average
Accuracy	-	95.19 ± 1.05	92.17 ± 0.71	91.62 ± 0.23	93.54 ± 0.45	92.44 ± 0.23	92.99 ± 1.44
Precision	**0**	97.96 ± 1.10	94.76 ± 1.41	92.68 ± 0.58	92.72 ± 0.95	98.43 ± 0.60	95.31 ± 2.68
	1	92.74 ± 1.76	89.87 ± 0.88	90.61 ± 0.45	94.39 ± 0.06	87.77 ± 0.46	91.08 ± 2.49
Recall	**0**	92.30 ± 2.00	89.28 ± 1.05	90.38 ± 0.54	94.50 ± 0.01	86.26 ± 0.63	90.54 ± 3.01
	1	98.07 ± 1.05	95.05 ± 1.41	92.85 ± 0.63	92.58 ± 1.05	98.62 ± 0.54	95.43 ± 2.74
F1-score	**0**	95.04 ± 1.28	91.93 ± 0.83	91.51 ± 0.27	93.60 ± 0.48	91.94 ± 0.30	92.81 ± 1.51
	1	95.33 ± 1.16	92.38 ± 0.82	91.72 ± 0.28	93.47 ± 0.56	92.88 ± 0.25	93.16 ± 1.40

TABLE II
COMPARISON WITH STATE-OF-THE-ART BASED ON THE CLASSIFICATION ACCURACY.

Model	Self-Supervised/Unsupervised Learning				Supervised Learning			
	CoPo	AGNN [27]	ANN + Modified K-Means [27]	SCARF [28]	ViT [8]	MTEC [12]	ViT-CAT [9]	DLCC [29]
Accuracy	92.99%	81%	74%	87.17%	93.72%	94.13%	94.84%	92.81%

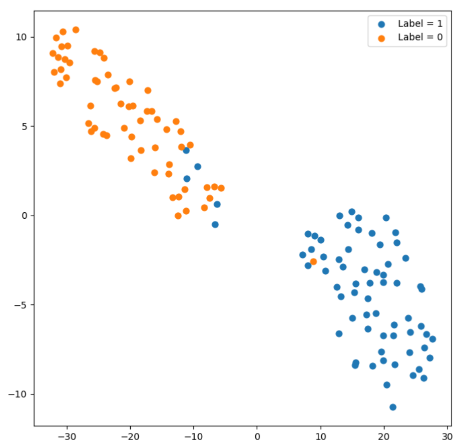

Fig. 3. A typical embedded space for the latent representation of popular and unpopular content, using The TSNE method.

size of this block is indicated by D_{E}. Finally, the decoder performs based on the Time-LSTM2, with an output size of D_{D}, where the activation functions are sigmoid and Tanh. By using a process of trial and error, the best version of the CoPo framework was determined to have the following features: D_{E} and D_{D} were both set to 512, the batch size, the learning rate, and N_o were established at $1e-3$, 128, and 20, respectively. Table I illustrates the accuracy, precision, recall, and f1-score for different 5 folds.

Furthermore, we leverage the T-distributed Stochastic Neighbor Embedding (TSNE) technique [26] to assess the effectiveness of the CL block in producing latent representations that can discriminate between popular and unpopular content. To demonstrate this, we present Fig. 3, which illustrates the embedded space of a test set obtained from one of the five-fold cross-validation experiments, which is clearly separable for popular and unpopular contents. Finally, we compare the proposed CoPo framework in terms of classification accuracy with several self-supervised, unsupervised, and supervised learning models. As shown in Table II, the proposed CoPo architecture outperforms other unsupervised baselines, i.e., Adaptive Genetic Neural Network (AGNN) [27] and Artificial Neural Network (ANN) with modified K-Means [27], and Self-Supervised Contrastive Learning using Random Feature Corruption (SCARF) [28]. Moreover, we compare the proposed CoPo framework with several supervised learning models, such as Vision Transformer [8], Multiple-model Transformer-based Edge Caching (MTEC) [12], Vision Transformers with Cross Attention (ViT-CAT) [9], and Deep Learning-based Content Caching (DLCC) [29] frameworks. Table II demonstrates that the classification accuracy is comparable to that of supervised learning models while eliminating the need for manual labeling of datasets, which saves time.

V. CONCLUSION

In this paper, we presented the Contrastive learning Popularity (CoPo) prediction framework, a self-supervised learning algorithm for predicting content popularity in Mobile Edge Caching (MEC) networks. By leveraging the CL paradigm and utilizing the LSTM model, the CoPo framework captured temporal and spatial information of historical requests of contents without the need for manual labeling. Our simulation results demonstrate that the proposed CoPo framework outperforms unsupervised state-of-the-art methods while maintaining comparable classification accuracy to supervised learning popularity prediction models.

References

[1] M. Sheraz, M. Ahmed, X. Hou, Y. Li, D. Jin, Z. Han, T. Jiang, "Artificial Intelligence for Wireless Caching: Schemes, Performance, and Challenges," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 631-661, First quarter 2021.

[2] Z. HajiAkhondi-Meybodi, J. Abouei, M. Jaseemuddin and A. Mohammadi, "Mobility-Aware Femtocaching Algorithm in D2D Networks Based on Handover," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 10188-10201, June 2020.

[3] L. Ale, N. Zhang, H. Wu, D. Chen and T. Han, "Online Proactive Caching in Mobile Edge Computing Using Bidirectional Deep Recurrent Neural Network," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5520-5530, June 2019.

[4] Y. Lin, C. Yen and J. Wang, "Video Popularity Prediction: An Autoencoder Approach With Clustering," *IEEE Access*, vol. 8, pp. 129285-129299, 2020.

[5] C. Zhong, M. C. Gursoy and S. Velipasalar, "Deep Reinforcement Learning-Based Edge Caching in Wireless Networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 1, pp. 48-61, March 2020.

[6] P. Wu, J. Li, L. Shi, M. Ding, K. Cai and F. Yang, "Dynamic Content Update for Wireless Edge Caching via Deep Reinforcement Learning," *IEEE Communications Letters*, vol. 23, no. 10, pp. 1773-1777, Oct. 2019.

[7] Y. Wang, Y. Li, T. Lan and V. Aggarwal, "DeepChunk: Deep Q-Learning for Chunk-Based Caching in Wireless Data Processing Networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 4, pp. 1034-1045, Dec. 2019.

[8] Z. Hajiakhondi-Meybodi, A. Mohammadi, E. Rahimian, S. Heidarian, J. Abouei, and K. N. Plataniotis, "TEDGE-Caching: Transformer-based Edge Caching Towards 6G Networks," *IEEE International Conference on Communications (ICC)*, Apr. 2022.

[9] Z. HajiAkhondi-Meybodi, A. Mohammadi, M. Hou, J. Abouei, and K. N. Plataniotis. "ViT-CAT: Parallel Vision Transformers with Cross Attention Fusion for Popularity Prediction in MEC Networks," *arXiv preprint arXiv:2210.15125*, 2022.

[10] H. Mou, Y. Liu and L. Wang, "LSTM for Mobility Based Content Popularity Prediction in Wireless Caching Networks," *IEEE Globecom Workshops*, 2019, pp. 1-6.

[11] C. Zhang et al., "Toward Edge-Assisted Video Content Intelligent Caching With Long Short-Term Memory Learning," *IEEE Access*, vol. 7, pp. 152832-152846, 2019.

[12] Z. HajiAkhondi-Meybodi, *et al.*, "Multi-Content Time-Series Popularity Prediction with Multiple-Model Transformers in MEC Networks," *arXiv preprint arXiv:2210.05874*, Oct. 2022.

[13] Q. Fan, X. Li, J. Li, Q. He, K. Wang and J. Wen, "PA-Cache: Evolving Learning-Based Popularity- Aware Content Caching in Edge Networks," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1746-1757, June 2021.

[14] S. Rathore, J. H. Ryu, P. K. Sharma and J. H. Park, "DeepCachNet: A Proactive Caching Framework Based on Deep Learning in Cellular Networks," *IEEE Network*, vol. 33, no. 3, pp. 130-138, May/June 2019.

[15] K. C. Tsai, L. Wang, and Z. Han, "Mobile Social Media Networks Caching with Convolutional Neural Network," *IEEE Wireless Communications and Networking Conference Workshops*, 2018, pp. 83-88.

[16] K. N. Doan, T. Van Nguyen, T. Q. S. Quek, and H. Shin, "Content-Aware Proactive Caching for Backhaul Offloading in Cellular Network," *IEEE Transactions on Wireless Communications*, vol. 17, no. 5, pp. 3128-3140, May 2018.

[17] A. Ndikumana, N. H. Tran, D. H. Kim, K. T. Kim and C. S. Hong, "Deep Learning Based Caching for Self-Driving Cars in Multi-Access Edge Computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 5, pp. 2862-2877, May 2021.

[18] J. Tang, H. Tang, X. Zhang, K. Cumanan, G. Chen, K.-K. Wong, and J. A. Chambers, "Energy Minimization in D2D-Assisted Cache-Enabled Internet of Things: A Deep Reinforcement Learning Approach," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5412-5423, Aug. 2020.

[19] A. Sadeghi, G. Wang and G. B. Giannakis, "Deep Reinforcement Learning for Adaptive Caching in Hierarchical Content Delivery Networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 4, pp. 1024-1033, Dec. 2019.

[20] Z. Zhang and M. Tao, "Deep Learning for Wireless Coded Caching With Unknown and Time-Variant Content Popularity," *IEEE Transactions on Wireless Communications*, vol. 20, no. 2, pp. 1152-1163, Feb. 2021.

[21] M. Montazerin, E. Rahimian, F. Naderkhani, S. F. Atashzar, S. Yanushkevich, and A. Mohammadi, "Transformer-based Hand Gesture Recognition via High-Density EMG Signals: From Instantaneous Recognition to Fusion of Motor Unit Spike Trains," *arXiv preprint arXiv:2212.00743*, 2022.

[22] S. Khademi, S. Heidarian, P. Afshar, F. Naderkhani, A. Oikonomou, K. N. Plataniotis, and A. Mohammadi, , "Spatio-Temporal Hybrid Fusion of CAE and SWIn Transformers for Lung Cancer Malignancy Prediction," *arXiv preprint arXiv:2210.15297*, 2022.

[23] F. M. Harper, and J. A. Konstan, "The Movielens Datasets: History and Context," *Acm transactions on interactive intelligent systems*, vol. 5, no. 4, pp. 1-19, 2015.

[24] C. Hong, F. Yi and Z. Huang, "Deep-CSA: Deep Contrastive Learning for Dynamic Survival Analysis With Competing Risks," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 8, pp. 4248-4257, Aug. 2022.

[25] Y. Zhu, H. Li, Y. Liao, B. Wang, Z. Guan, H. Liu, and D. Cai, "What to Do Next: Modeling User Behaviors by Time-LSTM," *IJCAI*, vol. 17, pp. 3602-3608, 2017.

[26] M. C. Cieslak, A. M. Castelfranco, V. Roncalli, P. H. Lenz, and D.K. Hartline, "T-Distributed Stochastic Neighbor Embedding (T-SNE): A Tool for Eco-Physiological Transcriptomic Analysis," *Marine genomics*, vol. 51, p. 100723, 2020.

[27] C. Selvi, and E. Sivasankar. "A novel Adaptive Genetic Neural Network (AGNN) model for recommender systems using modified k-means clustering approach," *Multimedia Tools and Applications*, vol. 78, pp. 14303-14330, June 2019.

[28] D. Bahri, H. Jiang, Y. Tay, D. Metzler, "Scarf: Self-Supervised Contrastive Learning using Random Feature Corruption," *arXiv preprint arXiv:2106.15147*, 2022.

[29] S. Bhandari, N. Ranjan, P. Khan, H. Kim, Y. S. Hong, "Deep Learning-based Content Caching in the Fog Access Points," *Electronics*, vol. 10, no. 4, pp. 512, 2021.