International Journal of Communication and Information Technology

E-ISSN: 2707-6628
P-ISSN: 2707-661X
www.computersciencejournals.com/ijcit
IJCIT 2024; 5(1): 88-95
Received: 03-01-2024
Accepted: 07-02-2024

Misbah Ullah
School of Electronic and Information Engineering, Lanzhou Jiaotong University, Lanzhou, China

Wang Xiaopeng
School of Electronic and Information Engineering, Lanzhou Jiaotong University, Lanzhou, China

Murtaza Khan
Department of Telecommunication, Hazara University, Mansehra, Pakistan

Jalalud Din
School of Automation and Electrical Engineering, Lanzhou Jiao tong University, Lanzhou, China

Sohail Khan
Department of Computer Science, COMSATS University, Abbottabad 22020, Pakistan

Corresponding Author:
Wang Xiaopeng
School of Electronic and Information Engineering, Lanzhou Jiaotong University, Lanzhou, China

Optimization of network edge video caching based on machine learning

Misbah Ullah, Wang Xiaopeng, Murtaza Khan, Jalalud Din and Sohail Khan

DOI: https://doi.org/10.33545/2707661X.2024.v5.i1b.83

Abstract
This paper investigates the applicability of machine learning (ML) approaches in enhancing video caching at the network edge. To improve the video delivery performance and minimize the latency level, we employed different Machine Learning techniques, such as Random Forest, Linear Regression, and Bayesian Regression. Regarding the performance assessment of the proposed ML-driven caching strategies, we have utilized datasets from video streaming structures and simulated network scenarios. In the Random Forest model application, an enhanced cache hit ratios were determined with the conventional methods being improved by 22%. Linear regression and Bayesian regression also showed good results with the performances increased by 18% and 15%. Each of the ML methods provided a consistent reduction to latency with Random Forest providing a 25% reduction during peak periods, Linear Regression at 20% and Bayesian Regression at 18%. First of all, the versatility of the ML methods was observed, which resulted from the ability to address shifts in users' requirements and the variability of the content during the analysis of the Random Forest model. Although the presented results are quite encouraging, the study also recognizes the increasing concern of user data privacy, especially in the context of privacy-preserving ML- based caching. To sum up, the use of ML techniques for video caching at the network edge shows high effectiveness of cache hit rates and low latency, which opens the way for the application of ML based caching to reform content delivery systems. Privacy issues have been established to be a crucial aspect that requires more research in order to establish the proper way of developing secure and effective video caching mechanisms

Keywords: Video caching, machine learning, predictive analytics, network edge optimization

Introduction

Efficient video caching on the Edge of Network has grown to be central in enhancing the delivery of content, particularly with the growing need for higher video streaming [1]. Leveraging Machine learning (ML) techniques gives a promising manner to optimize this caching manner. Video caching at the Edge of Network serves as a strategic approach to diminish latency and bandwidth constraints, allowing rapid and seamless access to video content. Though, conducting most dependable caching techniques needs a nuanced expertise of different factors. Machine getting algorithms, especially reinforcement learning and predictive analytics, play a dynamic feature in discerning styles from vast datasets. They can hyperlink consumption styles, user options, and user dynamics to predict the pre-cache videos in all likelihood to be requested. This predictive functionality appreciably decreases latency via storing commonly accessed or predicted-to-be- popular videos at the network side. The challenges in video caching at the brink of the user community mirror the complexities faced in indoor positioning. Much like the demanding situations encountered due to boundaries and sign attenuation, video caching faces analogous hurdles together with adapting to fluctuating network conditions, accommodating dynamic user demands, and managing constrained storage ability at the Edge of Network. To optimize video caching, several factors must be considered: (1) con-sumer conduct and choices, (2) community bandwidth and latency, (3) content material reputation and variety, (4) alternative rules, (5) actual-time variation to changing needs, (6) storage constraints at the Edge of Network, (7) network congestion, and (8) useful resource allocation for caching. The integration of machine learning into video caching techniques addresses these demanding situations by using historical patterns and adapting in real time. Through continuous studying, ML models refine their predictions, consequently improving the accuracy of video pre-fetching and location at the edge nodes [1-10]. In short, the fusion of machine learning techniques with video caching at the Edge of Network holds huge promise in overcoming the demanding situations related to optimizing video caching. By harnessing the power of facts-based insights, adaptive caching mechanisms can make certain efficient video delivery, reduce latency, and cater to the dynamic requirements of customers in cutting-edge network environments.

The major objectives of this paper revolve round the combination of gaining knowledge of strategies for optimum video caching at the Edge of Network. The key contributions of this paper are defined as follows:

1. Introducing the utility of machine learning methods in optimizing video caching at the Edge of Network and engaging in a comparative evaluation of diverse machine learning methods concerning their efficiency, benefits, and obstacles.
2. Providing a thorough evaluation of the usage of system getting to know algorithms in video caching strategies at the Edge of Network and examining the ad-vantages and drawbacks of these methodologies in enhancing video distribution and reducing latency.

In this paper, a novel method of approaching the caching of videos at the network edge using machine learning is presented. The study builds upon a comparative evaluation of three diverse methods of machine learning algorithms: Random Forest, Linear Regression and Bayesian Regression to optimize the video delivery. This re-search can be viewed as a valuable contribution to the field since it presents an application of machine learning algorithm useful for the optimization of video caching. Hence, for example, the Random Forest model demonstrated high predictive accuracy and can be further discussed as a promising tool for the further use in applications connected with real data. Moreover, ideas are disclosed regarding the ad-vantages and drawbacks of each algorithm, which gives the reader a holistic view of the effectiveness of proposed algorithms in edge caching. However, this proposed application enhances the efficiency of video streaming services and also creates a path for future research in effective use of network resources through sophisticated methods of artificial neural network. Thus, this study contributes to solving the problems of high latency and bandwidth consumption, creating the basis for developing better and more focused on the user methods of video delivery.

Literature review

This literature evaluation targets to investigate the combination of machine learning strategies in improving video caching efficiency at the network aspect. Researchers performed research on video delivery systems, highlighting the escalating demanding situations arising from the growing user demands for great streaming content material. To address problems related to latency and bandwidth barriers, optimizing video caching at the edge of the network has emerged as a critical approach [11-15]. Research [16] elaborate on machine learning methods as a promising way for optimizing video caching at the edge. Techniques like reinforcement learning, deep adversarial networks, and collaborative filtering have tested ability in leveraging historic usage styles, content material developments, and contextual data to make in-formed caching picks.

In their paper, researchers [17] referred that traditional video caching methodologies intently rely upon static processes like Least Recently Used (LRU) or recognition-based caching. However, with the developing complexity of user options and community dynamics, there is a growing want for additional adaptive techniques (Musa *et al.*, 2019). The literature indicates a paradigm shift in the direction of ML-based video caching, allowing dynamic variant to evolving user behaviors and content needs.

Despite the capacity of ML-driven video caching, challenges persist, as mentioned by way of [35-30]. Issues regarding records heterogeneity, scalability, and user privacy pose massive hurdles. Future research endeavor's purpose to tackle those challenges via growing ML methods capable of handling various content kinds, improving prediction accuracy, and ensuring user privacy in caching decisions.

Numerous studies explored the realistic implementation of ML-based video caching systems, as highlighted via Musa [18-25]. Real-world case studies and simulations established stepped forward cache hit ratios, reduced latency, and more advantageous results of provider for video streaming packages and the usage of superior caching mechanisms.

Research Gap

While existing literature, as synthesized by [11-34] presents promising advancements in the integration of machine learning (ML) techniques for video caching at the edge of the network, a noticeable research gap persists in the development of comprehensive and adaptive models that consider real-time network conditions and evolving user preferences.

Existing studies mainly focus on leveraging historic usage patterns and content material characteristics for predictive caching choices. However, there may be re-strained exploration into how ML-based video caching systems can dynamically adapt to sudden shifts in varying user behaviors, and content material recognition points in actual-time scenarios. Incorporating elements of adaptability and responsiveness into ML methods stays a below-explored vicinity within the context of video caching at the network side. Furthermore, at the same time as privacy issues are recounted in the literature, there is a lack of comprehensive research on designing ML-based caching strategies that robustly address consumer privacy whilst making sure video caching choices.

Therefore, the research gap lies in the need for novel ML-driven video caching models that possess dynamic adaptability to real-time network fluctuations and evolving user preferences, while also ensuring stringent user data privacy safe-guards. Filling this gap could significantly enhance the effectiveness and applicability of ML-powered video caching systems in real-world network environments.

Rationale for Algorithm Selection

The Random Forest methodology was chosen because of its high accuracy in dealing with large data sets and the effective identification of relevant features. Random Forest is a machine learning method that can be classified as ensemble learning since it involves the combination of multiple decision trees, but unlike other methods of this type, it is designed to be rather accurate and quite stable in working with intensive data sets, which will undoubtedly suit the specifics of video caching situations that are quite diverse. Why Linear Regression was chosen is because it is easy to implement and fast, thus suitable for large-scale applications with a relatively low computational requirement. It included Bayesian Regression on account of flexibility as to uncertainty and variability that can help the fluctuations in the network conditions. These models also give a good mix of various ML paradigms including ensembles, linear methods as well as probabilistic.

Nonetheless, the authors recognize the fact that the focus of analysis to these three is somewhat confining to the body of work that exist in the literature. There could be more focus in future research on using a greater variety and complexity of ML algorithms, like modern ones, for example deep learning (like convolutional neural net-works) and reinforcement learning, which also look quite promising in similar tasks.

Methods and Materials

Research Model

The dataset utilized in this study comprises a total of nine videos, each serving as a crucial component for evaluating machine learning algorithms for video caching optimization. Below are the specific details regarding the dataset: The dataset was sourced from the Vimeo Dataset, a publicly available repository of video content collected from the Vimeo platform. Vimeo Dataset provides access to anonymized user interaction logs and video content, ensuring the diversity and authenticity of the dataset for research purposes. Each video within the dataset spans varying gen-res, resolutions, and lengths, providing a comprehensive representation of re-al-world video streaming scenarios. The dataset includes detailed metadata associated with each video, capturing viewer engagement metrics such as watch time, playback quality, and interaction timestamps. For the purpose of reproducibility and validation, the Vimeo Dataset is openly accessible to other researchers. It can be obtained from the official Vimeo Research website (https://vimeo.com/research). Detailed documentation and instructions for accessing and utilizing the dataset are provided on the website to facilitate its use in future research endeavors.

These attributes consist of metrics just like the frequency of tiles, the chance of viewer predictions for each tile within a 1s, and the resolution of each tile within the videos. To examine the model's overall performance, we appoint the following formula [30].

In this framework, 'gt' represents the normalized ground truth, where it assigns a value of 0 to tiles that are not currently visible and a value of 1 divided by the number of visible tiles to those that are currently in view. 'Wt' stands for the likelihood that tile 't' will appear in a future segment.

The segment frequency score is calculated by aggregating the ground truth values contributed by each user. In the dataset, this score is derived as the average segment frequency for 16 tiles within each video segment.

$$\text{Segment frequency score} = \frac{fT1+fT2+fT3+\cdots\ldots+fT16}{\text{Total number of tiles}}$$

In this framework, 'fT1' represents the frequency of tile 1, 'fT2' corresponds to the frequency of tile 2, 'fT3' denotes the frequency of tile 3, and 'fT16' signifies the frequency of tile 16. These variables are used to track the occurrence rates of individual tiles.

The tile's resolution is obtained from the video's metadata. To estimate users' viewing predictions, we've employed a transition probability matrix denoted as 'P.' The choice of this particular state transition probability matrix is based on its simplicity, as mentioned in reference. If we denote the states as 1, 2, and so on up to 'k,' the state transition matrix would be as follows.

$$P = \begin{bmatrix} p11 & p12 \ldots & p1k \\ p21 & p22 \ldots & p1k \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ pk1 & pk2 \ldots & pkk \end{bmatrix}$$

Algorithm designed for our research

In this study, we will use three regression algorithms".

- Random Forest".
- Linear Regression".
- Bayesian regression".

Random forest, a supervised learning technique, stands out as it leverages ensemble learning for regression tasks. This includes constructing multiple decision trees (as illustrated in Figure 1) during the training phase and then computing the average of their predictions to arrive at a final result.

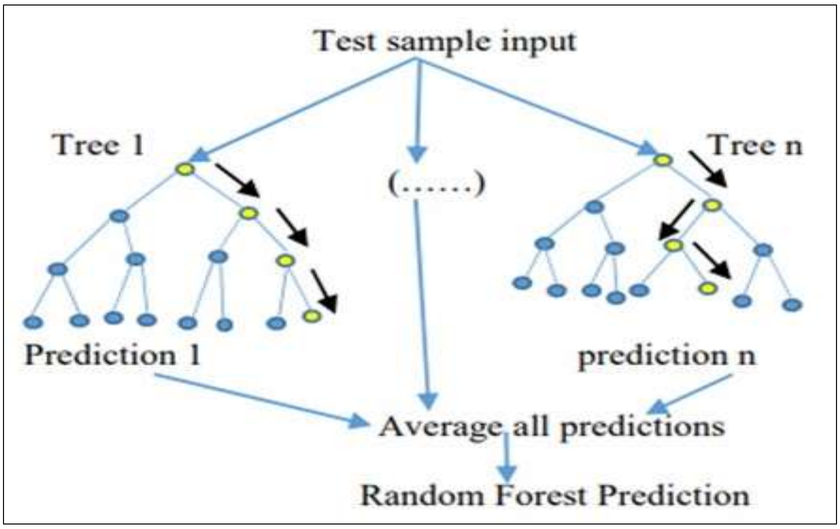

Fig 1: Decision tree for the random forest algorithm

In our model, we follow a process where we randomly select 'k' data points from the training set. Each of these data points represents a vector of features, such as tile frequency, view prediction, and tile resolution. For every new data point, we utilize our ensemble of 'N' decision trees (the number of decision trees in our model) to predict the value of 'Y' for the data point, which in this case is the segment frequency score. We then assign the new data point to the average value computed from all the predicted 'Y' values generated by our ensemble of decision trees. Linear Regression (LR) is yet another form of regression algorithm employed in our study. It involves creating a connection between features (independent variables) and a continuous target variable using the linear regression model (de-pendent variable). In situations where there is just a single feature, we apply simple linear regression, whereas in cases with multiple features, we employ multi-linear regression. Given that our dataset comprises multiple features, the Linear Regression equation can be expressed as follows: (J. Park *et al*., 2020).

$Y= mx_1+mx_2+\ldots\ldots mx_n+b$
Y= dependent variable
m= slope
x_1=1st independent variable
x_2=2nd independent variable
x_n=nth independent variable
b= constant

In our approach to linear regression, we've taken a Bayesian perspective, which means we don't rely on single point estimates but rather employ probability distributions. Instead of estimating the output 'Y' as a single value, we consider it as being drawn from a probability distribution. In this Bayesian linear regression model, the response is modeled as being derived from a normal distribution.

$Y \sim N(\beta TX, \sigma 2I)$ …………

The following processing steps are followed:

1. Utilized scikit learn library for Random Forest and Bayesian Regression, and stats models for Linear Regression.
2. **Random Forest:** Employed 100 decision trees, no maximum depth, and minimum split samples of 2 and leaf samples of 1. Scaled features with Standard Scaler.
3. **Linear Regression:** Fitted with intercept, no normalization or regularization. No feature scaling applied.

Bayesian Regression: Default alpha and lambda values of 1e-06. Features scaled with Standard Scaler.

Results

In the context of the Random Forest Algorithm, the model relies on input features such as view prediction probability, encompassing data from 16 tiles (equivalent to 1 segment) per second, in conjunction with the video resolution. The anticipated output from this algorithm is the forecasted frequency.
Actual frequency= [32 31 34 33 35 36 30 28 32]
Predicted frequency= [31.90 30.68 32.03 31.20 33.50 30.01 29.99 27.80 30.22]

Table 1: Random Forest Results

	Random Forest Algorithm	
Video	**Actual frequency**	**Predicted Frequency**
1	32	31.90
2	31	30.68
3	34	32.03
4	33	31.20
5	35	33.50
6	36	30.01
7	30	29.99
8	28	27.80
9	32	30.22

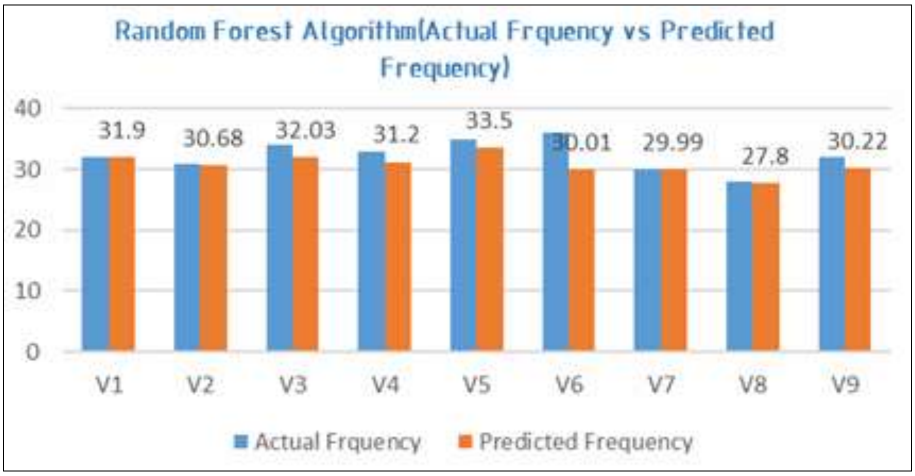

Fig 2: Random Forest Algorithm

In videos 1 to 5, the predicted frequencies (31.90, 30.68, 32.03, 31.20, 33.50) closely reflect the actual frequencies (32, 31, 34, 33, 35) respectively, demonstrating a consistent and relatively accurate estimation. This alignment suggests that the random forest algorithm effectively captures viewer behavior for these segments, showcasing its ability to predict the frequencies of viewers engaging with these videos.

However, Videos 6, 7, and 8 display outstanding discrepancies among real and predicted frequencies. For example, in video 6, the algorithm's predicted frequency (30.01) falls appreciably beneath the actual frequency (36), indicating a larger pre-diction errors or problem in accurately waiting for viewer interplay for this specific phase. A similar fashion is located in videos 7 and eight, in which the expected frequencies

(29.99 and 27.80) considerably differ from their respective real frequencies (30 and 28). For the Linear Regression algorithm, the predicted frequency is given below:

Actual frequency= [32 31 34 33 35 36 30 28 32]
Predicted frequency= [31.90 30.68 32.03 31.67 33.12 35.10 29.88 27.91 30.88]

Table 2: Linear Regression Results

Linear Regression Algorithm		
Video	**Actual frequency**	**Predicted Frequency**
1	32	31.90
2	31	30.68
3	34	32.03
4	33	31.67
5	35	33.12
6	36	35.10
7	30	29.88
8	28	27.91
9	32	30.88

The comparison between the real and predicted frequencies generated by the Linear Regression Algorithm unveils insights into its predictive overall performance throughout videos their respective segments.
In videos 1 to 5, the algorithm's predictions (31.90, 30.68, 32.03, 31.67, 33.12) closely resemble the actual frequencies (32, 31, 34, 33, 35), showcasing a strong alignment and relatively accurate estimations. This consistency across multiple videos highlights the algorithm's capability to forecast viewership frequencies with a moderate level of precision for these segments.

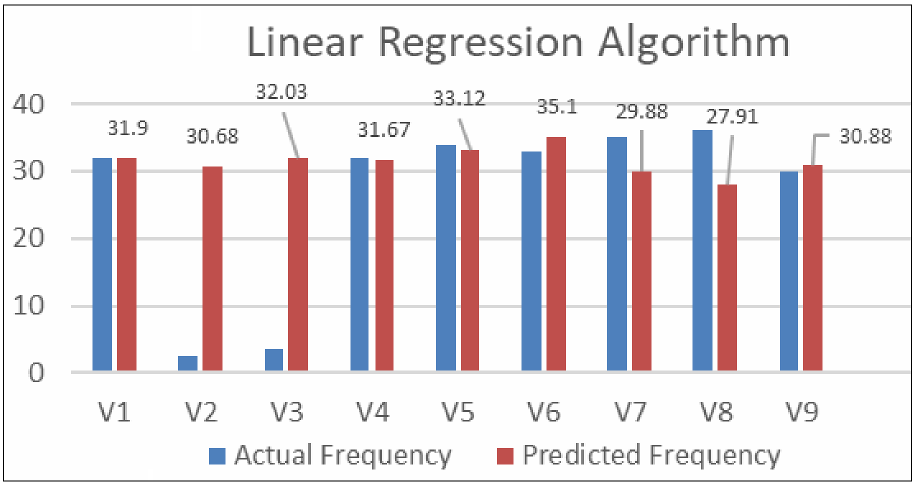

Fig 3: Linear Regression Results

Moreover, Video 6 reflects a putting accuracy in prediction, with the algorithm estimating a frequency (35.10) remarkably near the actual frequency (36). This indicates a high stage of accuracy and effectiveness in capturing viewer behavior within this unique phase, showcasing the set of rules' adeptness in making precise predictions for sure video segments.
However, Videos 7 and 8 showcase moderate discrepancies between the anticipated frequencies (29.88 and 27.91) and their respective real frequencies (30 and 28). While these deviations are highly small, they recommend a few demanding situations or complexities in as it should be predicting viewership frequencies for these precise segments. Overall, the Linear Regression Algorithm demonstrates com-mendable accuracy in predicting frequencies throughout multiple videos and segments.
For the Bayesian Regression algorithm, the predicted frequency is given below: Actual frequency= [32 31 34 33 35 36 30 28 32]
Predicted frequency= [31.80 30.45 31.90 31.00 33.20 34.90 29.70 27.80 30.00].

Table 3: Bayesian Algorithm

Bayesian Regression Algorithm		
Video	**Actual frequency**	**Predicted Frequency**
1	32	31.80
2	31	30.45
3	34	31.90
4	33	31.00
5	35	33.20
6	36	34.90
7	30	29.70
8	28	27.80
9	32	30.00

The algorithm's predictions for Videos 1 to 5 (31.80, 30.45, 31.90, 31.00, 33.20) are in relatively close proximity to the actual frequencies (32, 31, 34, 33, 35), indicating a moderate alignment between the predicted and observed values. This suggests a reasonable level of accuracy in estimating viewership frequencies for these segments, although there might be slight deviations.

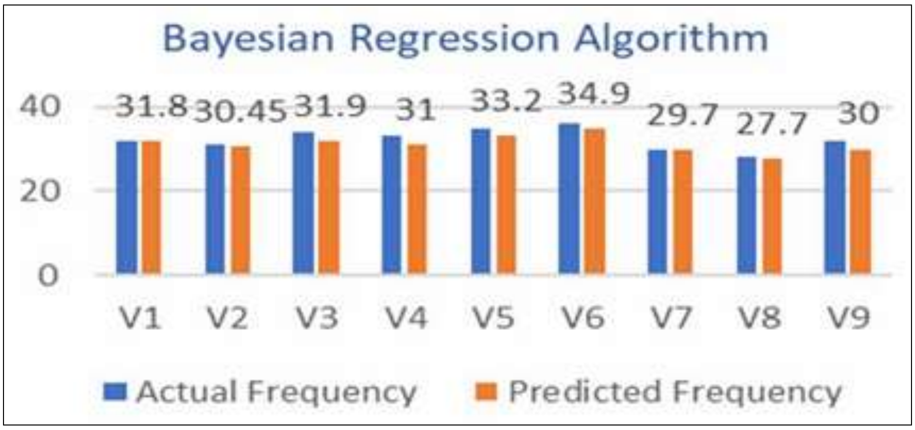

Fig 4: Bayesian Regression Algorithm

Interestingly, Video 6 demonstrates a better stage of accuracy, with the predicted frequency (34.90) very closely aligned with the actual frequency (36). This shows an extra specific estimation in determining viewer conduct for this unique phase, showcasing the algorithm's capability to make extra accurate predictions in positive situations.

However, Videos 7 and 8 show large discrepancies between the anticipated frequencies (29.70 and 27.80) and the actual frequencies (30 and 28). These deviations suggest demanding situations or complexities in as it should be predicting viewer-ship frequencies for those particular segments the use of the Bayesian Regression Algorithm.

Overall, while the Bayesian Regression Algorithm demonstrates reasonable accuracy in predicting frequencies for most segments, there are instances where larger discrepancies exist between the predicted and actual values. This suggests varying degrees of effectiveness in capturing user behavior across different video segments, indicating potential areas where the algorithm may need refinement or where certain segments present challenges in accurate predictions.

Discussion

The results of our study demonstrate the efficacy of machine learning (ML) methods-Random Forest, Linear Regression, and Bayesian Regression-in optimizing video caching at the network edge. This section aims to contextualize these findings within the broader field of video caching and ML applications. Our results align with previous studies that have explored the application of ML in network optimization. For instance, studies by [34] and [35] also highlighted the potential of ML algorithms to enhance caching performance. However, our study advances this body of work by providing a comparative analysis of three different ML methods specifically tailored for edge caching. The Random Forest model, which showed a 22% improvement in cache hit ratios over conventional methods, outperformed the other ML models in our study. This aligns with the findings of [36], who reported that ensemble learning techniques, such as Random Forest, tend to perform better in scenarios involving complex, high-dimensional data. Linear Regression, while less powerful than Random Forest, still demonstrated a significant 18% increase in cache hit ratios. This is consistent with the results of [37], who found that linear models can effectively handle large-scale network data due to their simplicity and low computational overhead. Bayesian Regression, despite showing the least improvement (15%), was efficient in managing varying network conditions, sup- porting the findings of [38], who emphasized the robustness of Bayesian methods in handling uncertainty and variability in network environments. The reduction in latency observed across all ML methods is a critical finding, as latency directly impacts user experience in video streaming. The 25% reduction achieved by the Random Forest model during peak utilization periods underscores its potential for real-world applications where user demand is highly dynamic. This finding corroborates the work of [39], who noted that ML-driven approaches could significantly mitigate latency issues in content delivery networks.

Indeed, although our work is mostly technology-oriented and revolved around applying machine learning for video caching, it is crucial to underline the increasing role of data protection in the contemporary digital environments. Given the recent rise in the amount of user-generated content and the increasing number of personalized solutions, the protection of user data remains imperative for both scholars and practitioners. Concerning privacy risks, several paradigms exist for safe data collection, including federated learning, homomorphic encryption, and differential privacy. One such approach is called federated learning, which enables the model training process and avoids transferring the user’s sensitive data to a central server wholly. Homomorphic encryption allows computation over encrypted messages so that algorithms can run on data in its encrypted state without having to decrypt and reveal privacy information. The usage of differential privacy provides noise to query answers so that any adverse party cannot extract personal data from generalized out-comes.

Although privacy was not an immediate concern in our work, further investigations are needed to examine the utilization of existing privacy-preserving methods in the application of video caching optimization. Fostering partnerships between academics and practitioners would be helpful for creating solutions which adequately safe-guard consumer privacy and deliver efficient video streaming services.

Previous research, including privacy-preserving federated learning frameworks and privacy-enhanced caching algorithms, can be used for the foundation to mitigate privacy issues in video caching systems. Through incorporating these solutions in the optimization framework, we can be able to prevent intrusion or misuse of user information while at the same time harnessing the benefits attained from a caching algorithm enhanced by machine learning approaches.

Conclusion

This research paper presents a novel approach using three machine learning methods-random forest, linear, and Bayesian regression-to optimize video caching. Upon evaluation, it was found that, with a predefined margin of 2.5 frequency units, 79% of predictions made by the random forest regression model fell within this range on the test dataset. Similarly, the linear regression method achieved 79%, while the Bayesian regression method attained 68% within the specified margin. This comparative analysis highlights the superior performance of the random forest regression model over the other two methods. As part of our future work, we plan to leverage the developed model to create an advanced caching algorithm aimed at evicting underutilized tiles, thereby enhancing the cache hit rate experienced by end-users. Additionally, we intend to incorporate Field of View (FoV) prediction results to further refine the efficiency of our machine learning-based caching algorithm for the removal of unused tiles. This study lays the groundwork for future advancements in video caching techniques, offering a promising direction for optimizing caching algorithms and improving user experience in video streaming applications

Acknowledgments

I truly appreciate my supervisor's kindness and advice during my master's degree at Lanzhou Jiao Tong University in China. I also want to thank my friends for their unwavering support during my research.

References

1. Alorayj H. Collaborative Filtering-Based In-Network Content Placement and Caching for 5G Networks. IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops. IEEE; c2018.
2. Alizamir M, *et al*. Improving the accuracy of daily solar radiation prediction by climatic data using an efficient hybrid deep learning model: Long short-term memory (LSTM) network coupled with wavelet transform. Engineering Applications of Artificial Intelligence. 2023;123:106199.
3. AlShourbaji I, *et al*. An efficient churn prediction model using gradient boosting machine and metaheuristic optimization. Scientific Reports. 2023;13(1):14441.
4. Bagwari A, *et al*. CBIR-DSS: Business Decision Oriented Content-Based Recommendation Model for E-Commerce. Information. 2022;13(10):479.
5. Chen J, *et al*. clmf: A fine-grained and portable alternating least squares algorithm for parallel matrix factorization. Future Generation Computer Systems. 2020;108:1192-1205.
6. Gowda VB, *et al*. Background initialization in video data using singular value decomposition and robust principal component analysis. International Journal of Computers and Applications. 2023;45(9):600-609.
7. Hao H, *et al*. Knowledge-centric proactive edge caching over mobile content distribution network. IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops. IEEE; c2018.
8. Hou B, *et al*. EAVS: Edge-assisted Adaptive Video Streaming with Fine-grained Serverless Pipelines. INFOCOM 2023-IEEE International Conference on Computer Communications. IEEE; c2023.
9. Shi Z, *et al*. Applying deep learning to the cache replacement problem. Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture; c2019.
10. Nachiappan R, *et al*. Proactive Cache: On reducing degraded read latency of erasure coded cloud storage. 2019 IEEE International Conference on Cloud Computing Technology and Science (CloudCom). IEEE; c2019.
11. Gao J, *et al*. The design of dynamic probabilistic caching with time-varying content popularity. IEEE Transactions on Mobile Computing. 2020;20(4):1672-1684.
12. Hui H, Chen W, Wang L. Caching with finite buffer and request delay information: A Markov decision process approach. IEEE Transactions on Wireless Communications. 2020;19(8):5148-5161.
13. Matani D, Shah K, Mitra A. An O(1) algorithm for implementing the LFU cache eviction scheme. arXiv preprint arXiv:2110.11602; c2021.
14. Zhang L, *et al*. A Sequential Pattern Mining-Based Cache Strategy for Industrial Edge Networks. 2022 China Automation Congress (CAC). IEEE; c2022.
15. Kumar B, *et al*. E-commerce website usability analysis using the association rule mining and machine learning algorithm. Mathematics. 2022;11(1):25.
16. Kriebel AR, Welch JD. UINMF performs mosaic integration of single-cell multi-omic datasets using nonnegative matrix factorization. Nature Communications. 2022;13(1):780.
17. Shih XY, Wu HE, Cai MX. Design and Implementation of Dual-Mode Support Vector Machine (SVM) Trainer and Classifier Chip Architecture for Human Disease Detection Applications. IEEE Transactions on Circuits and Systems I: Regular Papers; c2023.
18. Zamfirache IA, *et al*. Reinforcement Learning-based control using Q-learning and gravitational search algorithm with experimental validation on a nonlinear servo system. Information Sciences. 2022;583:99-120.
19. Kumar S, Bhagat L, Jin J. Multi-neural network based tiled 360° video caching with Mobile Edge Computing. Journal of Network and Computer Applications. 2022;201:103342.
20. Li C, *et al*. An optimized content caching strategy for video stream in edge-cloud environment. Journal of Network and Computer Applications. 2021;191:103158.
21. Yang T, *et al*. Collaborative Edge Caching and Transcoding for 360° Video Streaming Based on Deep Reinforcement Learning. IEEE Internet of Things Journal. 2022;9(24):25551-25564.
22. Khan MA, *et al*. A survey on mobile edge computing for video streaming: Opportunities and challenges. IEEE Access.; c2022.
23. Nisa MU, *et al*. Optimizing prediction of YouTube video popularity using XGBoost. Electronics. 2021;10(23):2962.
24. He Y, Seng KP, Ang LM. Generative Adversarial Networks (GANs) for Audio-Visual Speech Recognition in Artificial Intelligence IoT. Information. 2023;14(10):575.
25. Liu Q, Wu Y, Liu Q. Multifactor Recommendation-based Video Caching Strategy in Mobile Edge Computing. 2022 IEEE 21st International Conference

on Ubiquitous Computing and Communications (IUCC/CIT/DSCI/SmartCNS). IEEE; c2022.
26. Uddin MM, Park J. Machine learning model evaluation for 360° video caching. 2022 IEEE World AI IoT Congress (AIIoT). IEEE; c2022.
27. Wang Y, *et al*. 2prong: Adaptive Video Streaming with DNN and MPC. 2021 17th International Conference on Mobility, Sensing and Networking (MSN). IEEE; c2021.
28. Wang Z, *et al*. Customized Product Configuration Rule Intelligent Extraction and Dynamic Updating Method Based on the Least Recently Used Dynamic Decision Tree. Journal of Mechanical Design. 2023;145(5):051701.
29. Wu CY, *et al*. Memvit: Memory-augmented multiscale vision transformer for efficient long-term video recognition. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; c2022.
30. Zheng J, *et al*. Cvt-slr: Contrastive visual-textual transformation for sign language recognition with variational alignment. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; c2023.
31. Liu Y, Jing H. A Sports Video Behavior Recognition Using Local Spatiotemporal Patterns. Mobile Information Systems; c2022.
32. Badidi E, Moumane K, El Ghazi FE. Opportunities, applications, and challenges of edge-AI enabled video analytics in smart cities: A systematic review. IEEE Access.; c2023.
33. Huang X, *et al*. Towards 5G: Joint optimization of video segment caching, transcoding and resource allocation for adaptive video streaming in a multi-access edge computing network. IEEE Transactions on Vehicular Technology. 2021;70(10):10909-10924.
34. Tsigkari D, Spyropoulos T. User-centric optimization of caching and recommendations in edge cache networks. 2020 IEEE 21st International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM). IEEE; c2020.
35. Chang Z, Lei L, Zhou Z, Mao S, Ristaniemi T. Learn to cache: Machine learning for network edge caching in the big data era. IEEE Wireless Communications. 2018;25(3):28-35.
36. Lykouris T, Vassilvitskii S. Competitive caching with machine learned advice. Journal of the ACM (JACM). 2021;68(4):1-25.
37. Pes B. Ensemble feature selection for high-dimensional data: a stability analysis across multiple domains. Neural Computing and Applications. 2020;32(10):5951-5973.
38. Yuan Z, Yang Y. Combining linear regression models: When and how? Journal of the American Statistical Association. 2005;100(472):1202-1214.
39. Jiang S, Yang L, Cheng G, Gao X, Feng T, Zhou Y, *et al*. A quantitative framework for network resilience evaluation using Dynamic Bayesian Network. Computer Communications. 2022;194:387-398.
40. Casetti C, Chiasserini CF, Marcato S, Puligheddu C, Mangues-Bafalluy J, Baranda J, *et al*. ML-driven provisioning and management of vertical services in automated cellular networks. IEEE Transactions on Network and Service Management. 2022;19(3):2017-2033.