Video Streaming Traffic Prediction Using Machine Learning Techniques and Content Delivery Networks

Hamza Majid Hamid
Department of Computer Science, Informatics Institute for Postgraduate Studies, Information Technology & Communication University
Baghdad, Iraq
ms202320747@iips.edu.iq

Rana Fareed Ghani
Department of Computer Science, University of Technology
Baghdad, Iraq
Rana.f.ghani@uotechnology.edu.iq

***Abstract*—The Internet has undergone significant transformations in recent years, leading to the emergence of numerous innovative applications and services centered on digital content consumption. To support this evolution, advanced frameworks integrating content delivery networks (CDNs) and artificial intelligence (AI) have been developed to optimize content delivery and enhance user experiences. These frameworks leverage machine learning models to improve traffic prediction, bandwidth management, and network optimization for video streaming platforms. The proposed system preprocesses video streaming data and employs AI to analyze traffic patterns during peak hours. By identifying underutilized channels and redirecting traffic to less congested routes, the system ensures faster content delivery, reduces delays, and prevents bottlenecks during high-demand periods. The results demonstrate significant improvements in prediction accuracy and reduced latency. The performance metrics for different models are as follows: in machine learning models, including Decision Tree (DT), Random Forest (RF), and Naïve Bayes (NB), for traffic prediction in CDN networks. DT achieved the highest accuracy of 99.36% with a testing time of 0.5907 seconds and a decision time of 0.018 milliseconds in P2P, proving its effectiveness in optimizing traffic distribution and enhancing network performance. These findings highlight the system's ability to address video streaming challenges, particularly during peak traffic periods, providing.**

Keywords—* *Content Delivery Networks (CDN), Machine Learning (ML), Video Streaming Traffic Prediction, Quality of Experience

I. INTRODUCTION

Nowadays, content delivery networks (CDNs) and artificial intelligence (AI) are being integrated to meet growing data traffic demands. This integration significantly enhances user experience, particularly for latency-sensitive applications like video streaming and gaming, by ensuring faster and more reliable data routing. As a result, content loads more quickly, leading to improved user satisfaction. Recent advancements demonstrate how AI enhances content caching, load balancing, and data routing to reduce traffic bottlenecks by directing traffic to the nearest available bandwidth.

The video-on-demand market is projected to reach USD 127.87 billion in 2025 and USD 212.18 billion by 2030, with a CAGR of 10.66%. Digital video is expected to account for approximately 82% of internet traffic, driven by the shift toward on-demand viewing. Platforms are increasingly adopting hybrid revenue models that combine subscriptions and ad-supported content. According to Cisco's projections, online digital video will constitute nearly 82% of all internet traffic [1], while investments in localized and exclusive offerings continue to boost engagement.

A content delivery network (CDN) is a distributed infrastructure designed to enhance content delivery by reducing latency and improving performance. It relies on caching and intelligent data distribution, making it scalable to meet evolving performance and efficiency demands. Video delivery networks face challenges such as latency, bandwidth consumption, and high resource demands due to the increasing need for high-quality content. With the rapid expansion of internet usage and streaming services, CDNs are expected to grow significantly in the coming years to meet these rising demands. CDNs address the increasing demand for video-on-demand services by enhancing the Quality of Experience (QoE) for end users.

Many researchers have explored CDNs as a solution to the challenges posed by large volumes of video-on-demand traffic. Their work focuses on improving user access to information while considering Quality of Service (QoS) and QoE variability. AI plays a crucial role in advancing various fields, including CDN technology, by enhancing content delivery efficiency, optimizing resource management, and improving user experience through intelligent automation.

Enhancing QoE in video streaming has never been more critical, especially with advancements in 4G and 5G technologies, enabling seamless streaming across dynamic networks. Dynamic Adaptive Streaming over HTTP (DASH) has emerged as a leading method due to its flexibility and efficiency, serving as a foundation for improved user experiences [2]. Incremental statistical analyses of CDN logs provide lightweight evaluations of user experiences, allowing for more effective content delivery strategies [3].

Additionally, network traffic classification plays a pivotal role in improving QoE by accurately identifying traffic patterns, enabling better optimization. Machine learning (ML) frameworks address the complexities of encrypted video streaming and adaptive protocols such as MPEG-DASH, providing accurate estimates of key QoE metrics such as video resolution and playback interruptions. Tree-based models such as decision trees and random forests offer precise predictions, adapt to evolving protocols, and address session delimitation issues [4].

Furthermore, GRU-based bandwidth prediction systems dynamically select optimal bitrates, balancing video quality and stability to minimize buffering and enhance user satisfaction. Mobile live streaming services leverage extensive data analysis, predictive models, and adaptive bitrate (ABR) techniques to reduce latency and improve video stability [5]. The BANQUET algorithm transforms QoE by intelligently selecting bitrates based on real-time network conditions [6]. Meanwhile, NetScrapper, an AI-driven classifier, surpasses traditional traffic classification methods by seamlessly integrating with ML models for accurate, real-time assessments [7].

The DASH framework's ability to detect stalling events through transport layer data analysis further aids in identifying and resolving packet interruptions [8]. ML continuously enhances adaptive video broadcasting by analyzing user behavior and interposition times, leading to improved overall streaming performance [9],[10]. Fog computing serves as a crucial intermediary between cloud services and end users, reducing latency and maximizing bandwidth utilization for superior live streaming experiences [11].

A newly proposed video QoE estimation metric, leveraging pixel-based and network variables, effectively addresses packet loss and delay issues, delivering precise evaluations without requiring original video data [12]. Systems such as SENSEI set new standards by integrating ABR algorithms with crowdsourcing, leading to substantial QoE improvements [13],[14].

Furthermore, a novel database cataloging various stalling patterns and user evaluations is crucial for developing accurate QoE prediction models and enhancing network management strategies [15],[16]. AI-driven predictive analytics are reshaping network performance and content delivery, laying the groundwork for future trends in adaptive video streaming [17]. Hybrid CDN-P2P frameworks leverage ML technology for optimized peer selection, reducing dependency on ISPs and geographical methods.

Predictive models, particularly those derived from the LIVE-Netflix QoE Database, demonstrate significant advantages over traditional metrics, enabling the adoption of perceptually driven network strategies for unparalleled video quality [18]. Through ML-based predictive prefetching in MEC-enabled networks, cache efficiency is enhanced, and access delays are minimized by anticipating segment requests. The analysis of segment fetch times and throughput prediction techniques, incorporating player-specific features, is paving the way for robust strategies that ensure a seamless video streaming experience [19],[20].

In this work, ML in traffic management effectively predicts and manages traffic surges from platforms like YouTube, Netflix, and gaming services. A new CDN model addresses sudden demand spikes while maintaining broadcast quality and reducing latency. The Optimizing Content Delivery AI identifies efficient content delivery routes and balances demand across CDN servers to prevent bottlenecks. The results demonstrate significant improvements in streaming quality.

This research is structured as follows: Section II reviews related work, Section III presents the work proposal, Section IV details the results, Section V discusses the findings, and Section VI concludes the study.

I. RELATED WORK

Evolution of Content Delivery Networks Using Artificial Intelligence: Previous Works.

A. *Evolution of Content Delivery Networks Using Artificial Intelligence: Previous Works*

Several researchers have examined the evolution of CDNs using AI, and below, we list a number of previous works. Live streaming services have leveraged AI and CDN technologies to analyze and optimize traffic behavior, ensuring seamless delivery and minimal latency. ML models have been used to predict traffic patterns, enabling efficient resource allocation and enhancing the user experience for live video streaming.

In [21], ML and deep learning techniques have been applied to optimize data distribution in CDN networks, minimizing buffering and improving traffic flow management. A deep learning-based model integrates advanced neural architectures to forecast network congestion, achieving high prediction accuracy but facing challenges related to data dependency and computational demands. Additionally, ML-based predictive video storage using the RF model (78.1% accuracy) reduced data traffic by 37.25%, enhancing transmission efficiency while maintaining QoS (4.31) and QoE (4.38). However, both approaches encounter limitations in handling dynamic traffic conditions and high data demand in CDNs, highlighting the need for further research to improve model generalization and efficiency in real-world deployment.

In [22], encrypted live-streaming channels by analyzing network traffic using packet filtering, dynamic time warping, and deep learning models based on comment patterns and timestamps. The methodology examines traffic sequences and comment rates using real-time streaming data from multiple encrypted channels. The proposed approach achieves 93.2% accuracy, demonstrating robustness even under challenging network conditions. However, it faces scalability limitations with large real-time data volumes and reduced effectiveness in low-comment scenarios. Additionally, the dataset lacks scenario diversity, limiting its generalizability to broader streaming environments.

In [23], they manage traffic in CDNs more effectively, use machine learning (ML) techniques for caching in next-generation edge networks, categorize methods into supervised, unsupervised, reinforcement learning, neural networks, and transfer learning. They utilized datasets involving user mobility, preferences, and content popularity, achieving up to 95% accuracy in predicting content popularity. ML methods addressed caching issues by predicting content demand, optimizing cache placement, and improving cache hit ratios by up to 20%. Their analysis highlighted proactive caching strategies that significantly reduced latency by up to 60%. Challenges for future work include privacy preservation and federated learning in intelligent caching decisions. However, the survey predominantly focused on theoretical aspects, lacking extensive practical validation and real-world implementation details.

In [24], ReCLive, a real-time machine learning system, was developed to distinguish live from VoD streams and infer quality of experience (QoE) using network traffic features. Their methodology used datasets of approximately 23,000 video streams from YouTube and Twitch, employing LSTM neural networks achieving over 95% accuracy in live stream classification. They predicted video resolution and buffer stalls with Random Forest classifiers and statistical models, obtaining accuracies of 93% and 90%, respectively. ReCLive significantly aided ISPs by providing real-time QoE monitoring without relying on Deep Packet Inspection. However, the system was unable to infer QoE for YouTube live streams delivered via QUIC due to difficulties in accurately identifying video chunks. Moreover, its evaluation relied primarily on controlled experiments, limiting insights into real-world operational complexity and generalizability. In [25], valuated video streaming quality and user engagement for YouTube, Netflix, and Amazon Prime in Germany using browser extensions. Their dataset comprised over 400,000 video playbacks from 2,000 users, analyzing factors such as loading delays, stalling, and viewer behavior. They observed significant differences in engagement, video completion rates, and QoE across platforms and ISPs, emphasizing a strong relationship between QoE and viewer engagement. However, the study's limitation was its exclusive reliance on desktop browser data, excluding mobile streaming scenarios, thus limiting the generalizability of the findings to diverse device contexts.

B. *Content Delivery Networks*

CDNs use distributed servers to deliver content efficiently by routing users to the nearest node, reducing the load on the origin server and enhancing performance. Users are directed to the closest server based on location and server load to ensure speed and efficiency. CDNs enable fast response times and effectively manage traffic surges by dynamically distributing network load [26].

As a critical component of modern web architectures, CDNs improve performance by caching frequently accessed content across globally distributed edge nodes (PoPs). Key functionalities include content caching, managed by time-to-live settings to determine how long resources remain cached; purging, which updates cached content from origin servers; multiple origins, including cloud storage and dedicated servers; and access restrictions, which regulate content availability based on domains, regions, or IP groups.

By incorporating AI, CDNs can further enhance caching efficiency, reduce latency, increase availability, personalize content delivery, and improve user experience, making content faster, more reliable, and highly efficient [27].

II. METHODOLOGY

CDN performance and enhance video streaming quality. By analyzing historical traffic patterns and user behavior, ML model predicts network conditions, enabling proactive resource allocation and streaming parameter adjustments. Real-time monitoring further supports adaptive decision-making to optimize content placement and mitigate latency issues during peak traffic periods. This ML-driven approach ensures efficient bandwidth utilization, smooth video playback, and consistent streaming quality. methodology aims to improve user experience while maximizing CDN resource efficiency.

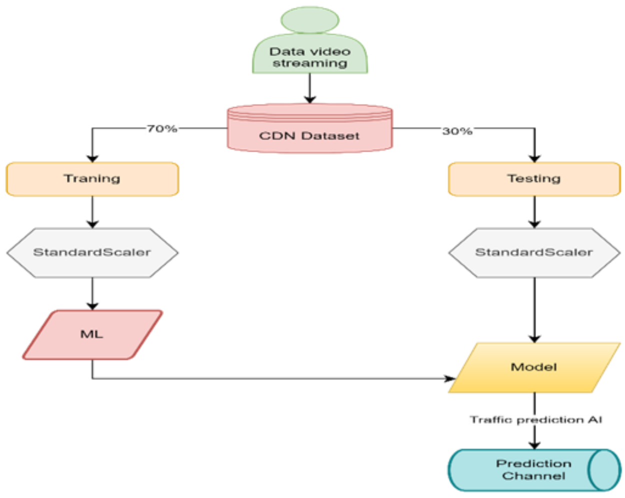

Fig. 1. Traffic Prediction System for Video Streaming Using AI and Machine Learning.

A. *Dataset*

In [28], a comprehensive dataset comprising 4.5 million entries was gathered from 45 days of continuous video streaming via the original YouTube mobile app. It included 11,142 measurements across 171 bandwidth entries and 80 diverse network conditions, totaling 332GB of video traffic over TCP and UDP/QUIC protocols. Covering extensive real-world mobile streaming scenarios, this dataset enables accurate modeling and prediction of modern network behaviors. Data was systematically divided into 70% for training and 30% for testing, facilitating robust machine learning model development and evaluation. The dataset classifies network behavior as "filling", "depletion," and "stalling".

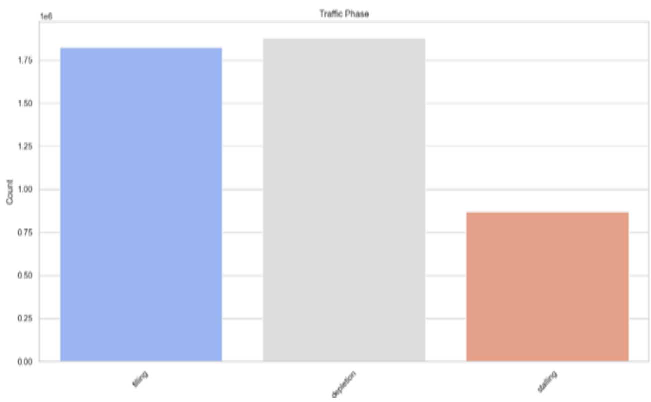

Fig. 2. Distribution of Network Traffic States.

The figure shows the distribution of network traffic states, with "depletion" being the most frequent, followed by "filling," and "stalling" being the least common.

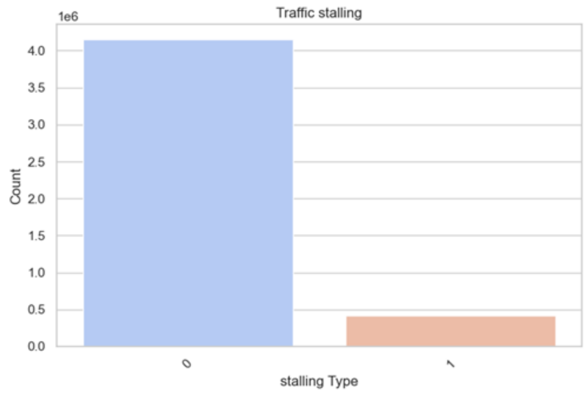

Fig. 3. Distribution of Traffic Stalling Types.

The figure illustrates the distribution of stalling in network traffic, highlighting that type "0" significantly dominates over type "1."

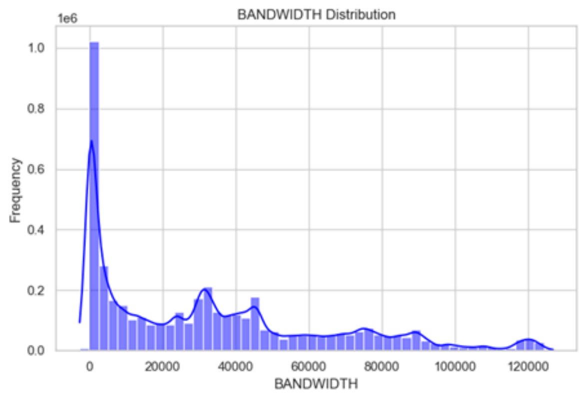

Fig. 4. Bandwidth Distribution

The figure shows the distribution of bandwidth values, where low bandwidth ranges (below 10,000) dominate the dataset, with frequency decreasing as bandwidth increases.

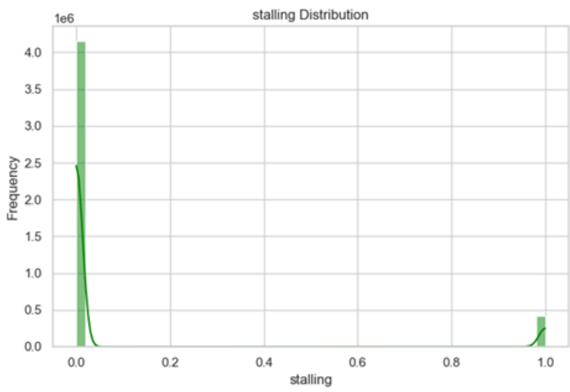

Fig. 5. Stalling Value Distribution

The figure reveals that most stalling values are concentrated near 0, indicating minimal interruptions, while a smaller peak near 1 suggests fewer instances of complete stalling.

B. Standard Scaler

In [28], the Standard Scaler is a normalization technique that transforms a dataset to have a mean of zero and a standard deviation of one, ensuring that features contribute equally to machine learning models. Given a dataset X, each feature xi is standardized using the equation:

$$x_i' = \frac{x_i - \mu}{\sigma}$$

where: x_i' is the standardized value of feature x_i; μ is the mean of the feature values; σ is the standard deviation of the feature values.

This transformation is particularly effective for models that assume normally distributed data, such as logistic regression and support vector machines, improving numerical stability and convergence speed in optimization algorithms.

C. Machine learning

Machine learning algorithms— in [29], Decision Tree (DT), Random Forest (RF), and Naïve Bayes (NB)—are employed to predict traffic patterns in video streaming over CDNs. These models are selected for their complementary strengths in accuracy, robustness, and computational efficiency. Preprocessed streaming data is used to classify network states into "depletion," "filling," and "stalling" phases. The models are trained and tested using a structured dataset to evaluate their performance under real-world streaming conditions. Comparative results guide the selection of the most suitable model for optimizing CDN resource allocation and ensuring consistent Quality of Experience (QoE).

The decision tree (DT) A Decision Tree is a supervised learning algorithm used for classification and regression tasks, structured as a tree-like model of decisions and their possible consequences. The model recursively splits the data into subsets based on feature values, forming a hierarchical structure of decision nodes and leaf nodes.

Mathematically, a decision tree partitions the feature space by selecting the optimal split s∗ at each node, which minimizes the impurity function I(s), often measured using Gini impurity or entropy:

1) *Gini Impurity (for classification)*

$$G = 1 - \sum_{i=1}^{C} p_i^2 \qquad (1)$$

Equation (1) represents the Gini impurity,"where " p_i^2" is the probability of class " i" in a given node".

2) *Entropy (Information Gain)*

$$E = -\sum_{i=1}^{C} p_i \log_2 p_i \qquad (2)$$

Equation (2) represents the Entropy, where higher entropy indicates more uncertainty in classification.

For regression tasks, the decision tree minimizes the variance reduction at each split:

$$R = \sum_{i=1}^{N} (y_i - \bar{y})^2 \qquad (3)$$

Equation (3) represents the Entropy, "where " y_i" are the target values and " $\bar{y}$" is the mean response." As in the following equation.

Decision trees are widely used due to their interpretability, but they are prone to overfitting, which can be mitigated using pruning techniques or ensemble methods such as Random Forests and Gradient Boosting Trees.

In [30], The Random Forest (RF) Random Forest (RF) is a robust ensemble learning algorithm widely employed for classification tasks due to its high predictive accuracy, resilience to overfitting, and capacity to model complex, non-linear relationships. It operates by constructing a multitude of decision trees during training and outputting the mode of the classes (classification) as the final prediction.

Let $D = \{(x_i, y_i)\}_{i=1}^{N}$ denote a training dataset, $x_i \in R^d$ where represents

a d- dimensional input

Feature vector, and $y_i \in \mathcal{Y}$ is the corresponding class label. A Random Forest classifier

$f_{\mathrm{RF}}(x)$ comprises T individual decision trees $\{h_t(x)\}_{t=1}^{T}$, each trained on a bootstrap

sample of D. For a given input x, the final classification output is determined via

majority voting:

$$f_{\mathrm{RF}}(x) = arg\max_{y \in Y} \sum_{t=1}^{T} I(h_t(x) = y) \_\_\_\_\_\_\_\_\_\_\_\_ (4)$$

Equation (4) represents the decision function for a Random Forest classifier, where $I(\cdot)$is the indicator function that returns 1 if the condition is true and 0 As in the following equation (4) otherwise.

Each tree in the forest is trained independently and incorporates a random subset of features at each split node, which introduces diversity among the base learners and mitigates the risk of overfitting. The aggregation of multiple diverse trees enhances generalization and robustness, particularly in high-dimensional and noisy classification problems.

In [31], The naïve Bayes (NB) Naive Bayes (NB) is a probabilistic classification algorithm rooted in Bayes' Theorem. It is particularly effective in high-dimensional spaces and is widely used for tasks such as text classification, spam detection, and medical diagnosis due to its simplicity, scalability, and competitive performance.

The core assumption of Naive Bayes is the conditional independence of features given the class label. x= (x_1, x_2…, x_d) denote a feature vector and y ∈Y be a class label from the set of possible classes Y\mathcal{Y}Y. Using Bayes' Theorem, the posterior probability of class yyy given the input x\mathbf{x}x is expressed as:

$$P(y \mid x) = \frac{P(y)\prod_{j=1}^{d} P(x_j \mid y)}{P(x)} \_\_\_\_\_\_\_\_\_\_\_\_ (5)$$

Equation (5) represents the Bayes' theorem in the context of a Naïve Bayes classifier,In practice, the denominator. $P(x)$ is constant across all classes and can be omitted during classification. Therefore, the Naive Bayes classifier predicts the class y* that maximizes the posterior probability:

$$y^* = arg\max_{y \in Y} P(y) \prod_{j=1}^{d} P(x_j \mid y) \_\_\_\_\_\_\_\_\_\_\_\_ (6)$$

Equation (6) describes the decision rule for a Naïve Bayes classifier, Applications, particularly when the dimensionality of the data is high and the class-conditional distributions are well-separated. The algorithm's computational efficiency and interpretability make it an attractive choice for baseline models and large-scale classification tasks.

However, experimental results in this study revealed that NB performance was inferior compared to other classification algorithms, indicating limitations when the independence assumption is violated or when complex feature interactions are present.

III. Experimental Results

In this study, we evaluate the performance of machine learning models for intelligent traffic prediction using a real-world CDN YouTube Dataset on Mobile

Streaming. We implement three widely used classification algorithms: Random Forest (RF), Decision Tree (DT), and Naïve Bayes (NB). The models are assessed using key performance metrics: Precision [32], Recall [33], F1-score [34], Accuracy [35], and Execution Time.

1. The evaluation metrics are defined as follows:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Where TP is True Positives and FP is False Positives.

2. Recall (measuring the ability to identify actual positive instances):

$$\text{Recall} = \frac{TP}{TP + FN}$$

Where FN is False Negatives.

3. F1-Score (harmonic mean of Precision and Recall):

$$F_1 = 2 \times \frac{\text{Precision } \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

4. Accuracy (overall correctness of the model):

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Where *TN* is True Negatives.

Performance Analysis

The models are trained and tested using a train-test split approach, ensuring a balanced evaluation. Execution time is recorded to assess computational efficiency. The results indicate that RF outperforms DT and NB in terms of accuracy and F1-score, while NB demonstrates faster execution time due to its probabilistic nature. The trade-off between accuracy and computational efficiency is discussed to determine the most suitable model for real-time CDN traffic prediction.

IV. Model Performance (Decision Tree)

Classification:
Execution Time: 14.95 seconds
Testing Time: 0.5906 seconds
Average Time per User Request: 0.018 MS
Overall Accuracy: 99.36%

TABLE I. Machine Learning Decision Tree

Class	**Decision Tree**		
	Precision	**Recall**	**F1-Score**
Depletion	99%	100%	99%
Filling	100%	100%	100%

Class	Decision Tree		
	Precision	**Recall**	**F1-Score**
Stalling	100%	97%	98%
Accuracy			99.37
Weighted Avg.	99%	99%	99%

TABLE II. DECISION TREE CONFUSION MATRIX

Confusion Matrix	Decision Tree		
	Predicted: Depletion	**Predicted: Filling**	**Predicted: Stalling**
Actual: Depletion	556,157	0	0
Actual: Filling	0	544,868	0
Actual: Stalling	6,499	2,179	260,820

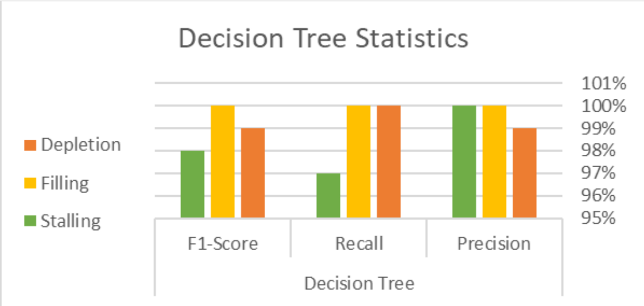

Fig. 6. Decision Tree Statistics

The table presents a summary of the performance of the Decision Tree model in classifying three categories: Depletion, Filling, and Stalling. The high values in Precision, Recall, and F1-Score for each category indicate that the model performs excellently in accurately classifying the data, with an overall accuracy of the model reaching 99.37%. This information is detailed in Table I. Additionally, Table II shows the Confusion Matrix, which provides detailed insight into the number of correct and incorrect predictions for each class. It shows almost perfect classification for Depletion and Filling with no misclassifications, while Stalling shows some misclassifications.

A. Model Performance (Random Forest)

Execution Time: 539.085 seconds
Testing Time: 0.8025 seconds
Average Time per User Request: 0.018 MS
Overall Accuracy: 99.36%

TABLE III. MACHINE LEARNING RANDOM FOREST

Class	Random Forest		
	Precision	**Recall**	**F1-Score**
Depletion	99	100	99
Filling	100	100	100
Stalling	100	97	98
Accuracy			99.37%
Weighted Avg.	99%	99%	99%

TABLE IV. RANDOM FOREST CONFUSION MATRIX

Confusion Matrix	Random Forest		
	Predicted: Depletion	**Predicted: Filling**	**Predicted: Stalling**
Actual: Depletion	556,157	0	0
Actual: Filling	0	544,868	0
Actual: Stalling	6,499	2,179	260,820

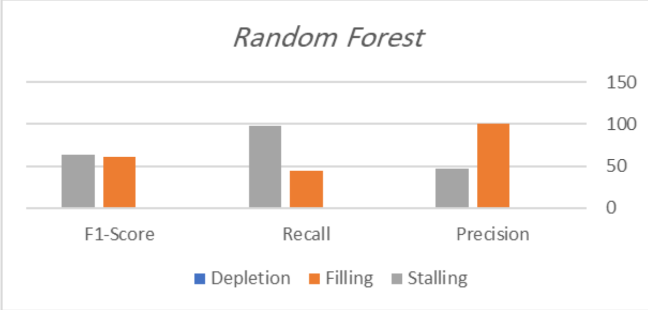

Fig. 7. Random Forest

The RF model demonstrated high accuracy in classification, achieving near-perfect results, as shown in Table III. Misclassifications were noted in one category, indicating room for improvement. Overall, the model achieved over 99% in precision, recall, and F1-Score, underscoring its effectiveness and reliability, as detailed in Table IV.

B. Model Performance (Naïve Bayes)

Classification Report:
Execution Time: 5.791 seconds
Testing Time: 0.6084 seconds
Average Time per User Request: 0.018 MS
Overall Accuracy: 48.92%

TABLE V. MACHINE LEARNING NAÏVE BAYES

Class	Naïve Bayes		
	Precision	**Recall**	**F1-Score**
Depletion	0.00	0.00	0.00
Filling	100	44	61
Stalling	47	98	64
Accuracy			48.92%
Weighted Avg	95%	49%	61%

TABLE VI. NAÏVE BAYES CONFUSION MATRIX

Confusion Matrix	Naïve Bayes		
	Predicted: Depletion	**Predicted: Filling**	**Predicted: Stalling**
Actual: Depletion	0	0	0
Actual: Filling	559,770	547,047	137,349
Actual: Stalling	2,886	0	123,471

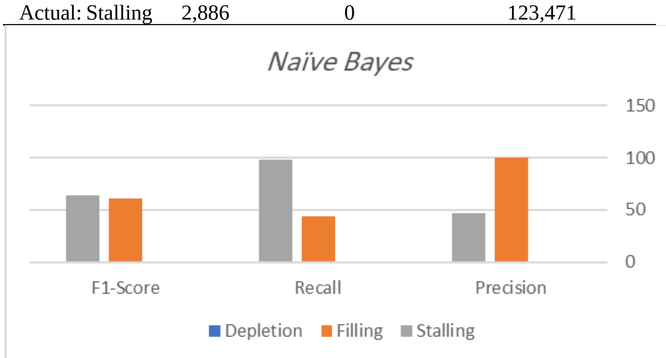

Fig. 8. *Naïve Bayes*

The NB model faced significant challenges in this classification task, as shown in Table V. Significant misclassifications were observed, impacting precision, recall, and F1-score values. Despite its fast execution time, the model's performance remains suboptimal, indicating a need for alternative models or further refinements to enhance classification accuracy. These issues and overall metrics are detailed in Table VI.

1) Comparative Accuracy Evaluation of Selected Classification Algorithms

TABLE VII. ACCURACY COMPARISON OF CLASSIFICATION ALGORITHMS

Accuracy	Algorithm
99.36	Decision Tree
99.36	Random Forest
48.92	Naïve Bayes

A comparative evaluation was conducted among three classification algorithms—Decision Tree, Random Forest, and Naïve Bayes—based on their accuracy in predicting the target class. As shown in Table VII. Accuracy Comparison of Classification Algorithms, both Decision Tree and Random Forest achieved a remarkably high classification accuracy of 99.36%, indicating their strong capability in learning complex decision boundaries and handling feature interactions effectively. In contrast, the Naïve Bayes classifier yielded a significantly lower accuracy of 48.92%, highlighting its limitations in scenarios where the assumption of feature independence is violated. This stark performance gap underscores the importance of selecting algorithms aligned with the underlying data characteristics. While Naïve Bayes remains computationally efficient and interpretable, its predictive power in this case was notably inferior, making tree-based ensemble methods a more reliable choice for the given classification task.

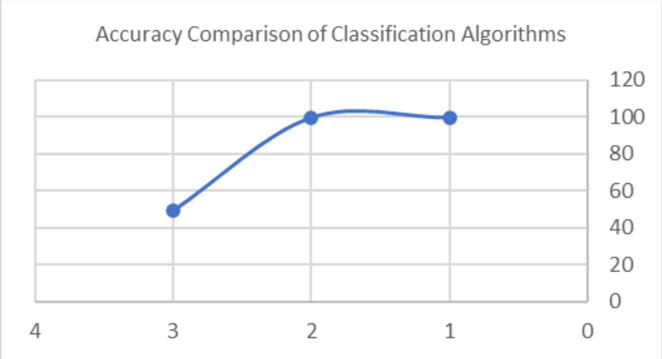

Fig. 9. Accuracy Comparison of Classification Algorithms

2) Table B. Execution Time During Testing Phase for Different Classification Algorithms

TABLE VIII. TESTING TIME PERFORMANCE OF CLASSIFICATION ALGORITHMS

Time Testing	Algorithm
0.5906	Decision Tree
0.8025	Random Forest
0.6084	Naïve Bayes

In addition to classification accuracy, the computational efficiency of each algorithm was assessed based on their testing time, As shown in Table VIII. The Decision Tree algorithm demonstrated the fastest inference time at 0.5906 seconds, making it well-suited for real-time or latency-sensitive applications. Naïve Bayes followed closely with a testing time of 0.6084 seconds, reflecting its lightweight and straightforward computational nature. Conversely, the Random Forest algorithm exhibited the longest testing time of 0.8025 seconds, attributable to the ensemble nature of the model, which involves aggregating predictions across multiple trees. While Random Forest and Decision Tree offer superior accuracy (as previously shown in Table A), the slight trade-off in execution time should be considered in scenarios where computational speed is critical.

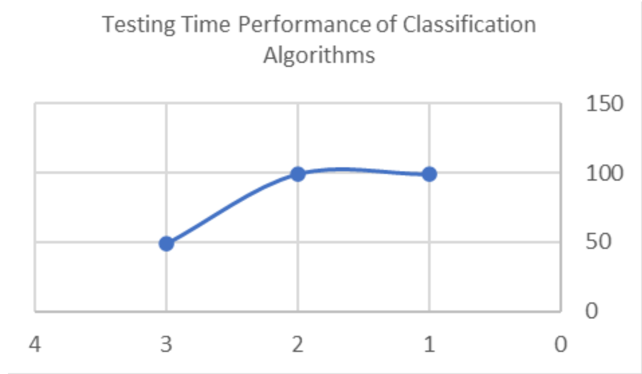

Fig. 10. Testing Time Performance of Classification Algorithms

V. DISCUSSION

In this work, several machine learning algorithms were evaluated for intelligent data distribution prediction and reducing video streaming delay. The algorithms used included Decision Tree (DT), Random Forest (RF), and Naïve Bayes (NB), with DT achieving the best performance—an accuracy of 99.36% and a low-test time of 0.5906 seconds. As shown in Table IX. Compared to previous works such as ReCLive and Fast-LTS, our approach demonstrated superiority in terms of both accuracy and time. Based on these results, the DT algorithm will be adopted as the primary choice for balancing efficiency and accuracy.

TABLE IX. COMPARISON OF MACHINE LEARNING METHODS FOR NETWORK PERFORMANCE

Reference	Method	Precision (%)	Recall (%)	F1-score (%)	Accuracy (%)	Test Time (s)
This work	Decision Tree (DT)	99	99	99	99.36	0.5906
This work	Random Forest (RF)	99	99	99	99.36	0.8025
This work	Naïve Bayes (NB)	95	49	61	48.92	0.6084
[22]	Stalling Event Prediction (LSTM)	–	–	92.3	–	–
[23]	ML Caching in Edge Networks	–	–	–	95	–
[24]	ReCLive (LSTM + RF)	93	–	90	95	–

Reference	Method	Precision (%)	Recall (%)	F1-score (%)	Accuracy (%)	Test Time (s)
[25]	QoE Assessment (YouTube, Netflix, Prime)	–	–	–	92	–
[31]	Batali	95	95	95	95	150
[36]	Fast-LTS (Imitation Learning)	–	–	–	–	Reduced ×7–10
[37]	LiveNAS (Neural Super-Resolution)	–	–	–	–	QoE ↑12–69%
[38]	Broadcaster behavior clustering using K-Means	0.85	0.96	0.90	–	QoE ↑29.5%

VI. Conclusions

The analysis of mobile streaming services highlights the critical roles of CDNs and AI in enhancing performance and improving user experience. By effectively managing response time and video quality, these technologies play an essential role in maintaining high levels of user satisfaction. Through the application of machine learning techniques, we employed normalization and Standard Scaler preprocessing methods on a range of algorithms, including Decision Tree (DT), Random Forest (RF), and Naive Bayes (NB). Among these, DT provided the most favorable results, demonstrating significant improvements over prior approaches. Overall, the findings were highly satisfactory, underscoring the effectiveness of these methodologies in advancing mobile streaming performance.

This study suggests that AI-driven solutions can proactively address potential downtime events, ensuring smooth operation and significantly improving overall service quality. The classification model used in this research demonstrated strong performance, particularly in the “filling” and “depletion” categories, where it achieved high accuracy and recall rates. However, challenges remain, especially in the “stalling” category, where classification errors were observed, highlighting the need for further enhancements in AI algorithms.

While the methodology proved robust, it exhibited limitations in adaptability and efficiency, suggesting that improvements in scalability are necessary for real-time applications. Overall, the results emphasize the need for continuous advancements in CDN and AI technologies to elevate user experience and support high-quality video delivery standards.

This study lays the groundwork for future research aimed at developing more dynamic and efficient streaming solutions, particularly in the domain of deep learning algorithms, that fully leverage AI and CDN capabilities.

References

[1] “Video-on-Demand Market Analysis | Industry Trends, Size & Growth Outlook,” accessed Mar. 7, 2025. [Online]. Available: https://www.mordorintelligence.com/industry-reports/video-on-demand-market.

[2] A. Hodroj, M. Ibrahim, and Y. Hadjadj-Aoul, “A survey on video streaming in multipath and multihomed overlay networks,” IEEE Access, vol. 9, pp. 66816–66828, 2021, doi: 10.1109/ACCESS.2021.3076464.

[3] H. Girault, L. Bonniot, and C. Neumann, “Inferring video streaming quality of experience at scale using incremental statistics from CDN logs,” in Proc. 3rd Mile-High Video Conf. (MHV), Denver, CO, USA, Feb. 2024, pp. 34–40, doi: 10.1145/3638036.3640803.

[4] A. Azab, M. Khasawneh, S. Alrabaee, K.-K. R. Choo, and M. Sarsour, “Network traffic classification: Techniques, datasets, and challenges,” Digit. Commun. Netw., 2024, doi: 10.1016/j.dcan.2022.09.009.

[5] I. Orsolic and L. Skorin-Kapov, “A framework for in-network QoE monitoring of encrypted video streaming,” IEEE Access, vol. 8, pp. 74691–74706, 2020, doi: 10.1109/ACCESS.2020.2988735.

[6] T. V. Huu, S. Van Pham, T. N. T. Huong, and H. C. Le, “QoE Aware Video Streaming Scheme Utilizing GRU-Based Bandwidth Prediction and Adaptive Bitrate Selection for Heterogeneous Mobile Networks,” IEEE Access, vol. 12, pp. 45785–45795, 2024, doi: 10.1109/ACCESS.2024.3382155.

[7] Z. Li, J. Li, Q. Wu, G. Tyson, and G. Xie, “A large-scale measurement and optimization of mobile live streaming services,” IEEE Trans. Mobile Comput., vol. 20 September 2022. doi: 10.1109/TMC.2022.3208094.

[8] T. Kimura, A. Matsumoto, and K. Yamagishi, “Balancing quality of experience and traffic volume in adaptive bitrate streaming,” IEEE Access, vol. 9, pp. 15530–15547, 2021, doi: 10.1109/ACCESS.2021.3052552.

[9] A. A. Ahmed and G. Agunsoye, “A real-time network traffic classifier for online applications using machine learning,” Algorithms, vol. 14, no. 8, Aug. 2021, doi: 10.3390/a14080250.

[10] F. Loh, F. Poignée, F. Wamser, F. Leidinger, and T. Hoßfeld, “Uplink vs. downlink: Machine learning-based quality prediction for HTTP adaptive video streaming,” Sensors, vol. 21, no. 12, Jun. 2021, doi: 10.3390/s21124172.

[11] R. Farahani, Z. Azimi, C. Timmerer, and R. Prodan, “Towards AI-assisted sustainable adaptive video streaming systems: Tutorial and survey,” Jun. 2024, [Online]. Available: http://arxiv.org/abs/2406.02302.

[12] S. F. Hassan and R. FareedGhani, “PWRR algorithm for video streaming process using fog computing,” Baghdad Sci. J., vol. 16, no. 3, pp. 667–676, 2019, doi: 10.21123/bsj.2019.16.3.0667.

[13] R. F. Ghani and A. S. Ajrash, “Quality of experience metric of streaming video: A survey,” Iraqi J. Sci., vol. 59, no. 3B, Aug. 30, 2018, doi: 10.24996/IJS.2018.59.3B.19.

[14] A. S. Ajrash, R. F. Ghani, and L. Al-Jobouri, “Quality of experience (QoE) measurement algorithm for transmitted video,” in Proc. 11th Comput. Sci. Electron. Eng. Conf. (CEEC), Colchester, UK, Sep. 2019, pp. xx–xx, doi: [Missing DOI].

[15] IEEE, Proc. 11th Comput. Sci. Electron. Eng. Conf. (CEEC), Colchester, UK, 2019, DOI: 10.1109/CEEC47804.2019.8974336.

[16] X. Zhang, Y. Ou, S. Sen, and J. Jiang, “SenSei: Aligning video streaming quality with dynamic user sensitivity,” in Proc. 18th USENIX Symp. Netw. Syst. Des. Implement. (NSDI), Boston, MA, USA, Apr. 2021. [Online]. Available: https://www.usenix.org/conference/nsdi21/presentation/zhang-xu.

[17] D. Ghadiyaram, J. Pan, and A. C. Bovik, “A subjective and objective study of stalling events in mobile streaming videos,” IEEE Trans. Circuits Syst. Video Technol., vol. 29, no. 1, pp. 183–197, Jan. 2019, doi: 10.1109/TCSVT.2017.2768542.

[18] K. Khan, “Enhancing adaptive video streaming through AI-driven predictive analytics for network conditions: A comprehensive review,” Int. Trans. Electr. Eng. Comput. Sci., vol. 3, no. 1, pp. 57–68, Mar. 2024, doi: 10.62760/iteecs.3.1.2024.67.

[19] Z. Ma, S. Roubia, F. Giroire, and G. Urvoy-Keller, “When locality is not enough: Boosting peer selection of hybrid CDN-P2P live streaming systems using machine learning,” accessed Submitted on 22 Nov 2021. [Online]. Available: https://inria.hal.science/hal-03441537v1.

[20] J. Uriol, I. Yeregui, A. Gabilondo, R. Viola, P. Angueira, and J. Montalban, “Context-aware adaptive prefetching for DASH streaming over 5G networks,” in Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast. (BMSB), 2023, doi: 10.1109/BMSB58369.2023.10211275.

[21] M. Wang, X. Tang, F. Chen, and Q. Lu, “Encrypted live streaming channel identification with time-sync comments,” IEEE Access, vol. 10, pp. 27630–27642, 2022, doi: 10.1109/ACCESS.2022.3157716.

[22] J. Shuja, K. Bilal, W. Alasmary, H. Sinky, and E. Alanazi, “Applying machine learning techniques for caching in next-generation edge networks: A comprehensive survey,” J. Netw. Comput. Appl., vol. 177, p. 102983, May 2021, doi: 10.1016/j.jnca.2021.102983.

[23] S. C. Madanapalli, A. Mathai, H. H. Gharakheili, and V. Sivaraman, “ReCLive: Real-Time Classification and QoE Inference of Live Video Streaming Services,” in IEEE/ACM International Symposium on Quality of Service (IWQOS), Jun. 2021, pp. 1–10, doi: 10.1109/IWQOS52092.2021.9521288.

[24] W. Robitza, A. M. Dethof, S. Goring, A. Raake, A. Beyer, and T. Polzehl, “Are you still watching? Streaming video quality and engagement assessment in the crowd,” in Proc. 12th Int. Conf. Quality Multimedia Exp. (QoMEX), 2020, doi: 10.1109/QoMEX48832.2020.9123148.

[25] A. S. Tanenbaum and D. J. Wetherall, Computer Networks, 5th ed. Pearson, 2010. [Online]. Available: https://csc-knu.github.io/sys-prog/books/Andrew%20S.%20Tanenbaum%20-%20Computer%20Networks.pdf.

[26] X. N. Nabijanovich, “The role of AI in content delivery networks,” in Proc. ICARHSE Int. Conf. Adv. Res. Humanities, Sci. Educ., 2022. [Online]. Available: https://confrencea.org/index.php/confrenceas/article/view/749/760.

[27] F. Loh, F. Wamser, F. Poignée, S. Geißler, and T. Hoßfeld, “YouTube dataset on mobile streaming for internet traffic modeling and streaming analysis,” Sci. Data, vol. 9, no. 1, Dec. 2022, doi: 10.1038/s41597-022-01418-y.

[28] J. Ko, S. Paw, S. Ameen, and A. Alkhayyat, “Deep Learning for Robust Iris Recognition: Introducing Synchronized Spatiotemporal Linear Discriminant Model-Iris Saif Kadhim,” 2025. [Online]. Available: https://www.oajaiml.com/uploads/archivepdf/102751197.pdf

[29] S. T. Abd Al-Latief, S. Yussof, A. Ahmad, S. M. Khadim, and R. A. Abdulhasan, “Instant Sign Language Recognition by WAR Strategy Algorithm Based Tuned Machine Learning,” International Journal of Networked and Distributed Computing, Dec. 2024, doi: 10.1007/s44227-024-00039-8.

[30] Kadhim, Saif Mohanad, Johnny Koh Siaw Paw, Yaw Chong Tak, and Shahad Thamear Abd Al-Latief. "Instant Fingerprint Recognition Using Optimized Machine Learning Models by Corona Virus Optimization Algorithm." In 2024 International Conference on Intelligent Computing and Next Generation Networks (ICNGN), pp. 01-07. IEEE, 2024.

[31] S. Kadhim, J. K. Siaw Paw, Yaw. C. Tak, S. Ameen, and A. Alkhayyat, “An Optimized Machine Learning Models by Metaheuristic Corona Virus Optimization Algorithm for Precise Iris Recognition,” Advances in Artificial Intelligence and Machine Learning, vol. 05, no. 01, pp. 3389–3408, 2025, doi: 10.54364/AAIML.2025.51194.

[32] Abdulhasan, Raed Abdulkareem, Shahad Thamear Abd Al-latief, and Saif Mohanad Kadhim. "Instant learning based on deep neural network with linear discriminant analysis features extraction for accurate iris recognition system." Multimedia Tools and Applications 83, no. 11 (2024): 32099-32122.

[33] Kadhim, Saif Mohanad, Johnny Koh Siaw Paw, Yaw Chong Tak, and Shahad Ameen. "Deep Learning Models for Biometric Recognition based on Face, Finger vein, Fingerprint, and Iris: A Survey." Journal of Smart Internet of Things (JSIoT) 2024, no. 01 (2024): 117-157.

[34] Abd Al-Latief, Shahad Thamear, Salman Yussof, Azhana Ahmad, and Saif Khadim. "Deep Learning for Sign Language Recognition: A Comparative Review." Journal of Smart Internet of Things (JSIoT) 2024, no. 01 (2024): 77-116.

[35] Kadhim, Saif Mohanad, Johnny Koh Siaw Paw, Yaw Chong Tak, Shahad Thamear Abd Al-Latief, Ahmed Alkhayyat, and Deepak Gupta. "Lionfish Search Algorithm: A Novel Nature-Inspired Metaheuristic." Expert Systems 42, no. 4 (2025): e70016.

[36] R. X. Zhang et al., “A practical learning-based approach for viewer scheduling in the crowdsourced live streaming,” ACM Trans. Multimedia Comput. Commun. Appl., vol. 16, no. 2s, Jul. 2020, doi: 10.1145/3397226.

[37] J. Kim, Y. Jung, H. Yeo, J. Ye, and D. Han, “Neural-enhanced live streaming: Improving live video ingest via online learning,” in Proc. SIGCOMM 2020 - ACM Special Interest Group Data Commun. Conf., 2020, pp. 107–125, doi: 10.1145/3387514.3405856.

[38] J. Li, Z. Li, Q. Wu, and G. Tyson, “On Uploading Behavior and Optimizations of a Mobile Live Streaming Service,” in Proceedings of the IEEE INFOCOM 2022 Conference on Computer Communications, 2022. [Online]. Available: https://www.eecs.qmul.ac.uk/~tysong/files/INFOCOM22-MLS.pdf.