

Advancing Cloud Gaming: Addressing Latency Challenges through Research and Innovative Solutions

Jahanzaib Tanveer

Computer Engineering

Information and Technology University

jahanzaib.ts.dev@gmail.com

ABSTRACT

Cloud gaming is an area of interest for both entrepreneurs and scholars yet evident by the growing body of knowledge and start-ups in this sector. However, cloud gaming has several drawbacks that are listed below and present several research opportunities in different domains like Distributed system, Virtualization, Human-computer interface, Video codecs, Quality of Experience, Optimum Assignment, and Dynamic Adaptation. These challenges can be surmounted in a way that benefits the service providers by enhancing the usage thereby increasing profitability and success. It should be noted that today's cloud gaming solutions have high costs, limited network throughput and provide a low Quality of Experience (QoE) to end-users. Further, closed-mainstream solutions remain dominant over others, and open-source systems are unavailable for experimenting widely.

It is therefore with this that a holistic approach to designing and developing solutions to latency problems and improving user-interface experiences in cloud gaming can be developed. This relates to integration frameworks, superior video codecs, GPU virtualization solutions, mobile UI mapping models, Quality of Experience (QoE) models, efficient server selection, and intelligent parameters. Frameworks for integration allow for easy integration of games into cloud platforms and minimize system dependencies, while video codecs enhancing certain graphical perspectives address encoding optimization.

Our research defines several promising approaches to improve cloud gaming experience. One such solution is Gaming Anywhere, which is a pluggable, and open source software compatible with iOS, Android, Mac, Windows, and Linux. This structure is far superior to commercial solutions as it attains measurably minimal processing latency. Another solution, Transparent-Gaming, is an accurate gaming system that enables the real-time connectivity of PC and console games on high end GPU utilizing consumer graphics cards. Thus, Transparent-Gaming succeeds in controlling the costs while making higher network usage and enhancing the quality of use. Moreover, Edge Game uses the EDGE technology in carrying out most of the computations avoiding high latency and band use in the network. In addition we have trained the machine learning models on the given set of data with emphasis on the dependent

variable which is latency. They help in achieving better results of cloud gaming and making users happier in the process.

Keywords — Cloud Gaming, Latency, Bandwidth, Video Codec, Edge Game, Gaming Anywhere, Virtualization, QoE, Distributed Systems

I. INTRODUCTION

An invention called cloud gaming, which is a new business model that has captured the imaginations of both global firms in the industry and individual gamers in recent years, has had a massively favorable impact on the growing video game industry. These cloud computing resources may be leveraged by various applications, and among them the resource-hungry computer games have been recognized as the killer application for cloud computing [1]. When it comes at mobile cloud gaming, the design of user interface plays a critical role in affecting user experience especially if the streamed games were not originally designed for mobile use.

By using remote servers and shedding the fat client tendency for delivering a faster and agiler gaming experience across gaming platforms and consoles, cloud gaming provides consumers with a new mode of engaging in games. This improved version, which has opened with better accessibility, fewer constraints regarding the hardware, and more efficient reviews of the gameplay dimensions, is slowly changing the way people play, consume and engage with video games.

Thus, the fast advancement of cloud settings and new legal aspects of the environment at the skinny customer period can be discussed with reference to the following principal innovations related to cloud gaming platforms. Currently, many game companies have developed different game platforms such as OnLive, StreamMyGame, and Gaikai that have revolutionized the cloud gaming through practicing and experimenting on different trends and strategies that meet the different demands and needs of gamers in the world. Among these challenges has been formerly addressed by these platforms such as software installation overhead, compatibility issues with hardware, and the repetitiveness in hardware

upgrades through the scalabilities of the cloud-based assets hence providing an excellent movie review to the intended audience [2].

However, it is important because there is a large number of challenges, percentages, conditions, factors that do not allow one to define the opportunities of cloud gaming. One of such is a question of managing the latency, the time that elapses between the user's action and a reaction of the interface. The level of latency existing in any experiment carried out in cloud computing evaluates the interactivity exhibited in the process and the subject under study. In addition, because of the strictly closed nature of several cloud gaming platforms and related solutions it becomes virtually next to impossible to try out many prospective cloud gaming platforms for actual-time gaming envelopes. It is suggested that possible mixed and comparative analyses of the existing cloud gaming systems could be useful in shedding further light to the enterprising targets and future manual regarding destiny investigations in this emerging discipline.

However, as the foundation of the examination for this study paper, it is important to carry on with the analysis of the structures of cloud gaming in order to better understand its characteristics, benefits, and inconveniences. Thus, these pioneering platforms in cloud gaming first launched by the spearhead firms like, OnLive, StreamMyGame, and Gaikai have introduced new novelties in approaches and strategies in cloud gaming which suits the unbounded diversification of preferences, demands and proclamations of global gamers. In the first instance, it is important to note that main goal of our research is to present rather intricate and sometimes unfavorable aspects of the existing relationships between generations and consumers, and between the latter and Industry related to the realm of cloud gaming.

The shape of this paper is as follows: We start our work by providing literature review that would further stand the study in the framework of cloud gaming research. We proceed to characterize a robust methodology for quantifying latency in closed structures, given the methodological challenges networked by means of proprietary structures. We then go further to analyze and compare various top cloud gaming systems such as OnLive and StreamMyGame which are explained here in detail about the streaming latency, total performance, and consumer pleasure of the systems. At long last, we provide suggestion and a possible perspective for another research: The continued investigations are crucial for the development of cloud gaming in the future.

II. BACKGROUND

Cloud gaming is getting increasingly popular, e.g., CloudUnion has too many subscribers compared with its current infrastructure, and an admission control algorithm was proposed [4] to alleviate the long waiting time. On-demand or

cloud gaming or game streaming is a type of technology by which without having a gaming console one can play games that are delivered online from servers that can be away from such users at very large distances. Web-based means that there are no applications installed in clients, computers, or other devices such as console or PC, instead, the powerful servers describing data centers handle all the game processes. Controlling flows are then sent to the game to produce the display of the user in real time such as the images to be displayed, the sound to be played, the signal for keyboard click or the movement of the game pad are then sent back to the server for interpretation or for a computation to be made. This configuration makes it possible for games to be played on virtually all platforms including pc even with low config, tablets, smart phones, and smart TVs since most of the computation is done on the server. This technology is exemplified by current practices such as NVIDIA GeForce NOW, Google Stadia, Microsoft Xbox Cloud Gaming, and PlayStation Now. There are a number of benefits when it comes to on-demand gaming: firstly, the costs of overall gaming is slightly reduced, and secondly, the ownership of consoles is not necessary in order to play the games on demand without needing to download or install them. It also has its disadvantages: high server response time, bandwidth I/O, and significant data usage; these factors impact the game performance and might be inconsiderate towards the players with poor connectivity or a restricted data plan. The video game industry has played a significant role in the software and entertainment sectors. For example, the global video game market is predicted to increase from 66 billion US dollars in 2010 to 81 billion US dollars in 2016 [6]. Another market research study [7] further breaks down the market growth into three categories: boxed-games, online-sold games, and cloud games.

However, it is also possible to point out certain negative consequences which can be attributed to the On-demand technology listed above, but it is also essential to consider the possibility to play video games whenever one feels like it.

III. LITERATURE REVIEW FOR LATENCY MEASUREMENT

GAMING ANYWHERE

In theory, GamingAnywhere plays a crucial role in enabling real-time cloud gaming with minimal latency, which in turn enhances the quality of the game. [5]

- **Efficient Processing:** In order to reduce the processing delay (PD) of GamingAnywhere, overall system design and various facets of the gaming pipeline are fine tuned. This includes improving the cycle time involved in the actual computation at server-side including accepting player commands, capturing video frames and encoding, and packaging of data for communication. Likewise, for clients, it enhances processes such as receiving frame data,

decoding frames, and rendering frames. In this regard, GamingAnywhere helps to optimize timing parameters so that the waiting time between the user action and the corresponding visualization is minimal on average.

- **Responsiveness Optimization:** The system is developed in accordance to the main principle of quick reaction: the commands provided by the user are accepted and translated into game actions with minimal delay. As will be explained in the following sections, GamingAnywhere can determine where latency can be eliminated or region-ed by breaking down the RD into processing delay, playout delay, and network delay. This, in turn, enables specific enhancements to be made to enhance the website's response even further.
- **Network Load Optimization:** Compared to the other systems in cloud computing, GamingAnywhere sustains lower traffic on the networks. In this way, it cuts down the uplink and downlink traffic which decreases the chance of having congested networks and a consequent packet loss than may lead to increased latency. Moreover, dependency of GamingAnywhere on various compression algorithms to reduce the network load to a minimal while providing high quality videos improves network performance.
- **Video Quality Maintenance:** Moreover, as latency optimization is one of its primary objectives, GamingAnywhere achieves high video quality due to the optimization of encoding and decoding. This affirms that users get the best of fun and an equally good experience in gaming not compromised by video quality. GamingAnywhere does make the picture quality much better than the other cloud gaming systems and at the same time reduces the latency. [14]

A MULTI-FACETED APPROACH

To tackle the latency issues of the cloud gaming and construct optimal user interface experiences, which need at least five layers to work on, including integration frameworks, interactive video codecs, graphic processing unit (GPU) virtualization technologies, mobile related mapping, QoE models, server selection optimization, and smart parameter adjusting. Solutions that enable integration of a game into cloud-platforms, helping to decrease system dependencies, are Integration frameworks, Reduction of exploration cost of

relevant video codecs, especially those suitable for various graphical styles, has led to enhanced codec encoding efficiency. Developments in GPU virtualization technologies offer benefit with regards to scalability due to virtualization guaranteeing each game instance a dedicated GPU. Autonomous pre-processing of non-mobile counterparts in Mobile Interfaces for games promotes easy access and fun on the mobile gadgets. Moreover, the proposed QoE models are able to adjust system parameters based on the detected state of networks, and optimal server choice affects latency. Applying intelligent parameter tuning methods, resource management is kept balanced, preventing excessive loads and negatively affecting the balance between the game's performance and gameplay. This has been done as shown below to address the latency issues and enhance the user interface in cloud gaming solutions.

EDGE GAME

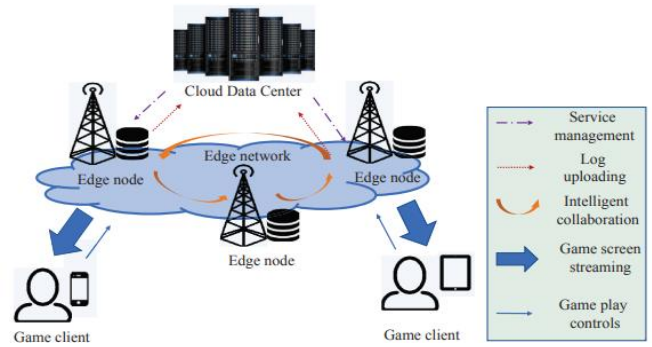


Figure 3: Framework of EdgeGame (Xu Zhang, Hao Chen, 2019).

Thus, Edge Game improves business visions and perceptions fundamentally of cloud gaming by deploying computation-intensive GPU rendering near the edge nodes, curbing latency and usage of bandwidth. It means that the game processes and graphics are managed near the user, far from latency that negatively affects the games hosted in the cloud. It takes advantage of machine learning techniques to dynamic video encoding and control of network throughput for the purpose of achieving optimum video quality on the basis of the existing physical network standards. A relay node works in direct cooperation with other relay nodes to update the synchronous status of different game modes which are very important for multiplayer games; a relay node optimizes traffic flow, making the delay time shorter. [13] [25]

The technological base of the system is virtual nodes (vNodes) with relatively high levels of game processing on edge devices, while user accounts, system diagnostics, and the configuration of game sessions are held in a cloud data center. Furthermore, Edge Game includes future expansion plans like utilizing idle user devices as the edge nodes of the network, and implementing a blockchain-based reward system for resource sharing in the network; With such concepts, Edge Game strives to envision and allow users to achieve highly efficient, scalable, and high-quality cloud gaming experiences.

In this research paper, we leverage two extensive datasets to evaluate the current state of cloud gaming infrastructure and its ability to support seamless, low-latency gaming experiences. Our analysis is based on data collected from 189 players connecting to 9 servers across four regions (North America, South America, Europe, and East Asia) and an additional dataset of 67 players interacting with 11 servers in various global locations. This multi-faceted approach provides a comprehensive overview of the latency and performance issues inherent in current cloud gaming setups. It is proven that If the cloud gaming servers are distributed across geographical locations, whenever a user attempts to log into the system and starts playing games, a server selection problem would naturally arise [3] [8] [25].

TRANSPARENT- GAMING

In this paper, a comprehensive description of the processes explored in the study and a detailed discussion of the different strategies applied towards reducing latency challenges in cloud gaming solution is outlined, particular through the use of T-Gaming platform. With this performant separation of the latency factors, the work endeavors to expound more on processing delay (PD), playout delay (OD) and the network delay (ND) which are Among the most delectable factors that causes latency to user. Out of the flow of implementation methods inclusive of GPU, CPU-software, HWA and method of benchmarking, the undertaking finds out the best possible way of avoiding the delay in processing and frame encoding. [10].

In addition, other tentative strategies like internal compression in GPUs and selective encoding schema are discussed in an effort to cut down the time used for processing even more but only if it does not result in a decreased level of videos which is imperative to the users of the videos.

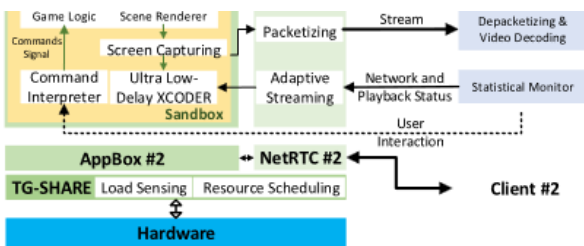


Figure 1: Game Cloud Design with Virtualized CPU/GPU Servers and Initial Performance Results (Zhao, Hwang, & Villeta, 2012).

Further, the research aims at investigating advance streaming strategies with the incorporation of the enhanced DRL algorithms to adaptively fine-tune the bitrate based on real-time network conditions. This technique developed in the course of this study known as ARS outperforms the conventional approaches of congestion control; because it has the ability of dynamically dominating bitrate based on this real-time network conditions in order to enhance the Quality of Experience (QoE) for these games. In fact, considering the experiment of the

study, ARS is trained with the help of various simulated network traces of real-world datasets and consequently it realizes that ARS offered superior average QoE and stability in different types of networks.

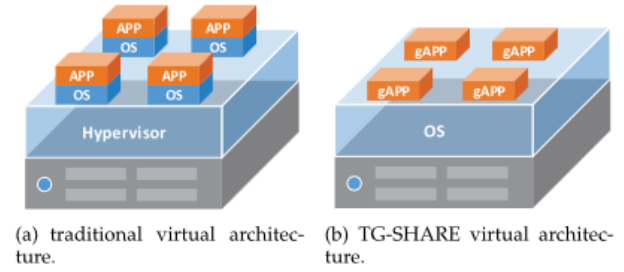


Figure 2: TG-SHARE Virtual Architecture (Zhao, Hwang, & Villeta, 2012).

Therefore, thanks to the application of demand and scientifically grounded research tactics, the unity and multifaceted of latency reduction in cloud gaming can be recognized, and efficient approaches for their resolution can be suggested. Thus, the work done within the present paper, thanks to the advanced technical groundwork of T-Gaming, paints a clear and realistic picture of how top-end video gaming can be performed in the context of the cloud while keeping latency at bay.

FOG COMPUTING

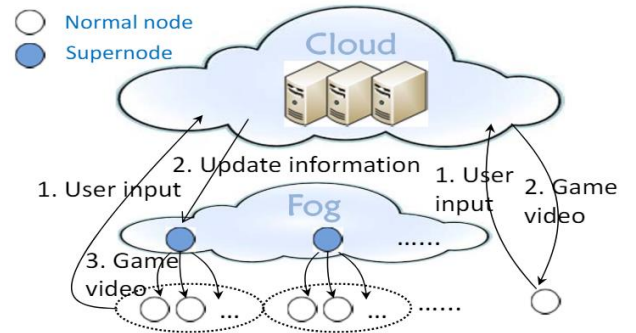


Figure 3: Architecture of FOG Computing (N. Tolia, D. Andersen, 2019)

Due to the popularity of massively multiplayer online games (MMOGs) and the growth of mobile gaming, certain drawbacks of traditional cloud gaming models—such as latency and coverage—are becoming more apparent. [16] Due to the distance between the players and the servers, traditional cloud gaming uses large server farms and fast internet connections, which can lead to latency issues. This latency is detrimental to the user's quality of experience (QoE), especially when combined with high bandwidth consumption. [17] [18] Prior studies have indicated that increasing the number of datacenters may be one way to address these issues, but doing so will come at a cost.

In order to tackle these issues, the concept of CloudFog has been proposed. Another "fog" layer with supernodes located closer to the consumers is suggested by CloudFog. While the cloud performs the calculations for the game updates, these supernodes manage the rendering of the game videos and broadcast them to the players [19] [23]. By reducing the quantity of data sent from the cloud to the end users, CloudFog reduces response time and bandwidth usage.

An assessment of CloudFog's performance on the PlanetLab testbed reveals that, in comparison to EdgeCloud and other alternatives, it not only improves playback continuity but also decreases response time [20] [21]. Even with the addition of a few powerful scattered servers to increase user coverage, EdgeCloud cannot outperform CloudFog in terms of performance because it does not distribute the computation of game states and video rendering tasks efficiently. Game videos are only streamed to the user from adjacent supernodes when utilizing CloudFog's mechanism. This reduces the amount of distance the data must travel to reach the user, improving the quality of experience [22] [16].

The results of these trials support the concept of CloudFog, which can be viewed as a potential remedy for the latency issue in cloud gaming. As such, it adds to the corpus of knowledge and establishes the groundwork for additional research and development [24].

TRANSLATING METRICS TO EXPERIENCE RATINGS

In this section, we will explore how to translate measured metrics into a performance rating that impacts the end gamer experience.

A. Input Latency Rating

The diagram below illustrates the conversion of latency readings into a color-coded rating system. For instance, a user experiencing a median latency of over 150 ms with a standard deviation exceeding 16.6ms will perceive the experience as poor quality.

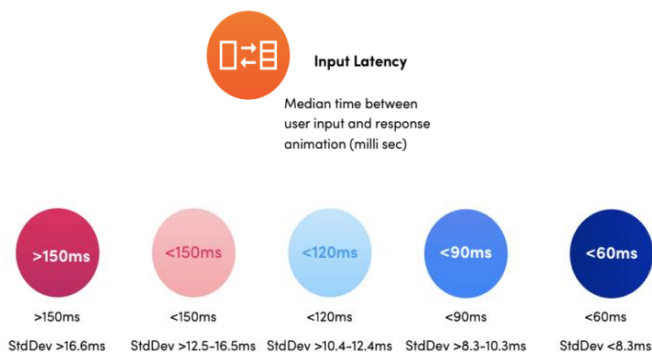


Figure 4: Architecture of FOG Computing (N. Tolia, D. Andersen, 2019)

B. FPS Ratings

The diagram below demonstrates the conversion of FPS readings, including Minimum Frame Rate (FR) and the variability of the frame rate, into a color-coded rating system. For example, a user experiencing a median frame rate of less than 20 will perceive the experience as poor.

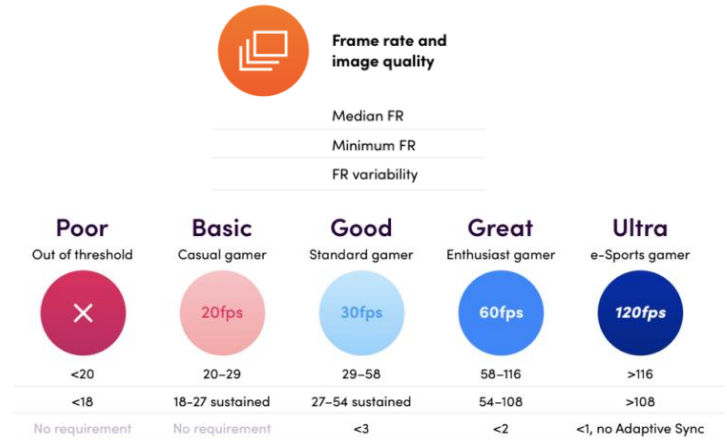


Figure 5: Architecture of FOG Computing (N. Tolia, D. Andersen, 2019)

C. Battery Ratings

The diagram below shows how Gameplay hours are converted into a battery rating. From a power consumption perspective, a device that can stream a cloud game for less than 3 hours on a full charge is considered poor.

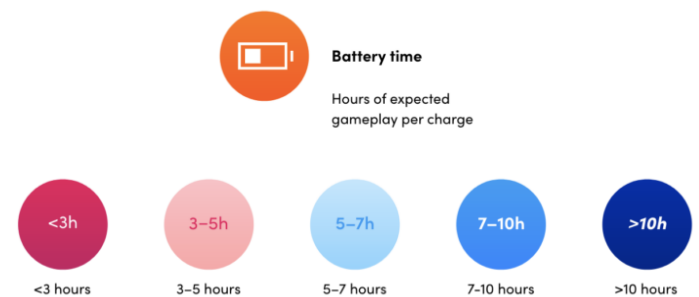


Figure 6: Architecture of FOG Computing (N. Tolia, D. Andersen, 2019)

NETWORK LATENCY FORMULATION

Network latency is the sum of all possible delays a packet can face during data transmission. We generally express network latency as round trip time (RTT) and measure in milliseconds (ms). Network delay includes processing, queuing, transmission, and propagation delays. Let's look at the formula to calculate network latency:

$$N_L = D_P + D_Q + D_T + D_{PR}$$

Transmission delay is the time to push all the available data in a transmission medium or wire. We can calculate transmission delay (in a second) by dividing the number of bits by the transmission rate:

$$D_T = \frac{N_B}{T_R}$$

Another delay a packet might face during data transmission is propagation delay. It's the time taken for a packet to cross the transmission medium. It depends on the distance (D) and speed of the packet (S). Hence, propagation delay:

$$D_{PR} = \frac{D}{S}$$

We can calculate network throughput (TP) using TCP receive window (W) and network round trip time (RTT) which is related to latency:

$$TP \leq \frac{W}{RTT}$$

IV. METHODOLOGY

The approach to research used in this study includes formulation of research questions that follow a systematic approach that deals with latency prediction in network communications. Specifying the set of parameters with a dataset obtained from IEEE Dataport, the study is concerned with the modeling of network delay for the players connected to various servers located in different geographical zones. It covers steps such as data gathering, data cleaning, selecting algorithms, building models, assessment and interpretation. Each of the steps is highly worked out in order to achieve the high degree of measurability to express latency, which is one of the key area defined for network performance assessment. It is further expected that practical data preprocessing methods, multiple regression model selection and assessment strategies will be used in the study to offer a clearer understanding on the performance of the different models of latency prediction. The various sections that follow outline the workings of the methodology in step-by-step fashion describing the processes performed in each stage of the research undertaking.

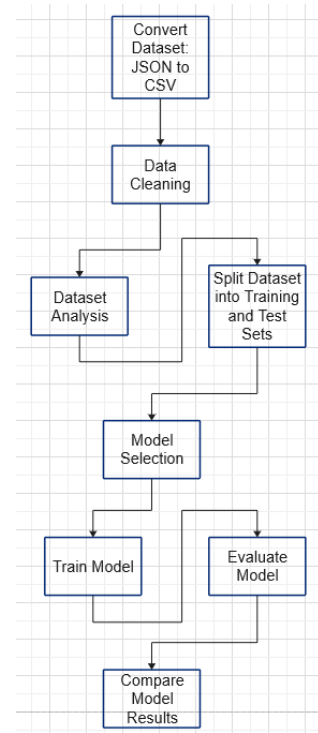


Figure 7: Machine Learning Flowchart

A. Dataset Collection

The dataset utilized in this study was sourced from IEEE Dataport, comprising network delay measurements between 189 players and 9 servers distributed across 4 regions: Producers from North America, South America, Europe and East Asian countries. The servers were in nasc of Santa Clara, and nach of Chicago, nada of Dallas, nato of Toronto, sabr of Brazil, in uk of London, in nl of Amsterdam, in hk of Hong Kong, and in sg of Singapore.

Each player's connection details were recorded, including: Each player's connection details were recorded, including:

User Identifier: The number for identification requirements a 32-character unique number that is generated for each player.

Time of Access: Stored in Unix time format, which represents the number of seconds since 1970-01-01 00:00:00 UTC.

Longitude and Latitude: Geo-location of the player often referred to as 'x-y coordinates'.

IP Address: They include: · The player's Identification details such as the IP address.

Access Support Network (ASN) or Internet Service Provider (ISP): The network provider or ISP that the player is associated with during the contest.

Median Latency/Delay (latency): The RTT latency of the median of milliseconds for the completion of one round trip.

Delay Jitter (jitter): The four corresponding std measures of latency indicate variation that remains unaccounted for by simple models of RTT.

Minimum Obtained Delay (min): Minimum latency recorded from when the measurement was started.

Maximum Obtained Delay (max): This refers to the highest figure recorded in latency measurements.

Average Obtained Delay (avr): Latency values recorded during measurement periods with average being the most commonly affected.

Dataset Conversion:

The data was provided in JSON database format and with the help of Python language it was converted to CSV format. The JSON records contained network delay measurements from each player to the servers, with additional data such as: The JSON records contained network delay measurements from each player to the servers, with additional data such as:

Code written for the conversion of dataset to csv:

```
1 # Parse the JSON data
2 records = []
3 for entry in data:
4     user_id = entry['user_id']
5     latitude = entry['latitude']
6     longitude = entry['longitude']
7     asn_org = entry['asn_org']
8     timestamp = entry['data']['timestamp']
9     for server, server_data in entry['data']['stats'].items():
10         if 'stats' in server_data and isinstance(server_data['stats'], dict):
11             server_stats = server_data['stats']
12             record = {
13                 'user_id': user_id,
14                 'latitude': latitude,
15                 'longitude': longitude,
16                 'asn_org': asn_org,
17                 'timestamp': timestamp,
18                 'server': server,
19                 'latency': float(server_stats['latency']),
20                 'jitter': float(server_stats['jitter']),
21                 'min': float(server_stats['min']),
22                 'max': float(server_stats['max']),
23                 'avr': float(server_stats['avr'])
24             }
25             records.append(record)
26
27 # Create DataFrame
28 df = pd.DataFrame(records)
29 df.to_csv("JsonToCsv.csv")
```

Figure 7: Dataset Conversion from Json to Csv

Initial Dataset format:

```
{
  "user_id": "01057c6e-8263-425e-8dfa-d7a8780b8cde",
  "longitude": "-46.6417",
  "asn_org": "CLARO S.A.",
  "latitude": "-23.5733",
  "data": {
    "timestamp": 1528123114774,
    "stats": {
      "nl": {
        "endpoints": [
          "https://nl.swarmio.gg:2222"
        ]
      }
    }
  }
}
```

Figure 8: Dataset in the form of Json

Updated Dataset format:

```
1 user_id,latitude,longitude,asn_org,timestamp,server,latency,jitter,min,max,avr
2 0,01057c6e-8263-425e-8dfa-d7a8780b8cde,-23.5733,-46.6417,CLARO S.A.,1.52812E+12,nasc,108.9,12.74,192.7,473.2,282.56
3 1,01057c6e-8263-425e-8dfa-d7a8780b8cde,-23.5733,-46.6417,CLARO S.A.,1.52812E+12,nabr,31.9,33.14,5.55,2,24.78
4 2,01057c6e-8263-425e-8dfa-d7a8780b8cde,-23.5733,-46.6417,CLARO S.A.,1.52812E+12,nada,308.4,38.99,179.7,369.9,282.92
5
```

Figure 9: Dataset in the form of Csv

To capture the latency measurements, a background JavaScript script that Swarmio's client software employs helped in achieving this. This script wanted Swarmio's portal to get a list of servers and ran above stated line for loop through each of the servers to check the RTT latency. The latency values were calculated from the averages of responses that were obtained after sending 11 packets from the player to each of the servers and then sent back to Swarmio main server.

Most interestingly, only one server, the first one identified as "nl", was used for the testing of connections and it was represented by the testing field having a value of 1. As a result, the "stats" field for the first server was an empty object that should have stored the measurements.

The generated latency prediction dataset was comprehensive, allowing the evaluation of such models.

B. Dataset Cleaning

It was important to clean and transform the data in order to acquire a sound and accurate dataset before engaging in the modeling process. Data cleaning is a vital step in data preparation that was performed on the dataset by undertaking several vital processes to fix numerous issues such as inconsistent values, missing values, and outliers.

```

1 # Drop the 'Unnamed: 0' column
2 data = data.drop(columns=['Unnamed: 0'])
3
4 # Convert 'latitude' to numeric
5 data['latitude'] = pd.to_numeric(data['latitude'], errors='coerce')
6
7 # Handle any potential NaNs after conversion
8 data = data.dropna()
9
10 # Encode categorical variables
11 label_encoders = {}
12 categorical_columns = ['user_id', 'asn_org', 'server']
13 for col in categorical_columns:
14     le = LabelEncoder()
15     data[col] = le.fit_transform(data[col])
16     label_encoders[col] = le
17

```

Figure 10: Dataset Cleaning Code Snapshot

C. Dataset Analysis

Latency and Performance Analysis

Our Analysis highlight serious shortcomings in the existing cloud gaming setup, especially with regard to network latency. Based on the player's location, the ISP they use, and the server they connect to, measurements show that latency varies significantly. Remarkably, connection testing was consistently conducted only in the United Kingdom (uk), suggesting a possible bottleneck in evaluating the entire range of infrastructure performance.

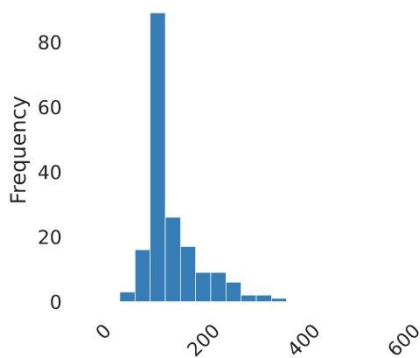


Figure 11: UK Latency

Regional Disparities

The geographical dispersion of players and servers indicates significant regional variations in the performance of cloud gaming:

- North America: Because of their close proximity and strong network infrastructure, servers in Santa Clara, Chicago, Dallas, and Toronto typically offer lower latency for players in this region..

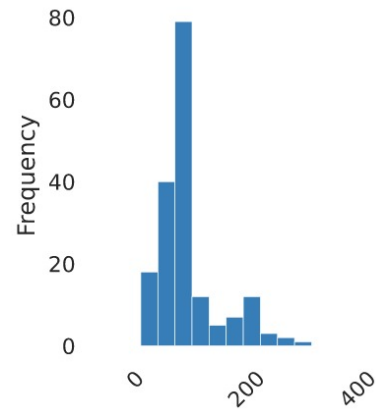


Figure 12: Santa Clara Latency

- South America: Players in this region frequently experience higher latencies due to the single server in Brazil, which highlights the need for more regionally tailored server infrastructure.

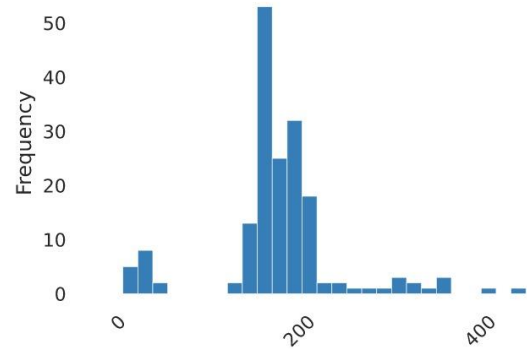


Figure 13: Brazil Latency

- Europe: Players from Europe can still benefit from slightly lower latency metrics on the London and Amsterdam servers, especially those from Eastern Europe.

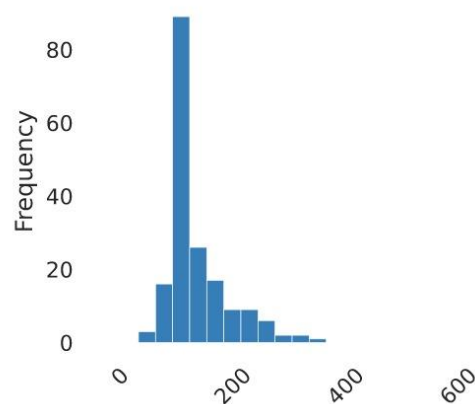


Figure 14: London Latency

- East Asia: Players in East Asia experience higher latencies on servers in Hong Kong and Singapore, which further emphasizes the need for more servers to improve coverage.

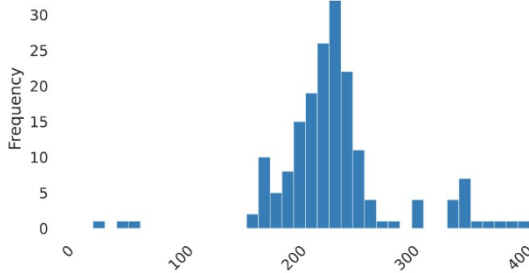


Figure 15: Hong Kong Latency

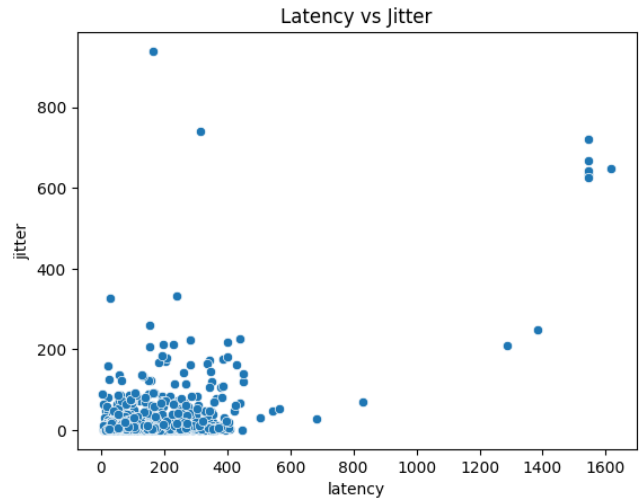


Figure 17: Latency and Jitter Comparison

LATENCY AND JITTER COMBINED ANALYSIS

X-axis: Represents the time range in milliseconds (ms).

Y-axis: Represents the count (number of events).

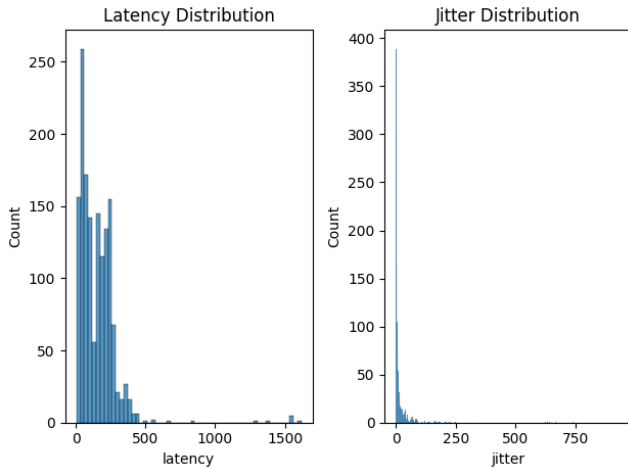


Figure 16: Latency and Jitter Distribution

Relationship between Latency and Jitter:

The scatter plot helps reveal any correlation between latency and jitter. In this specific graph, it's challenging to visually identify a strong linear correlation due to the scattered nature of the points.

Observations based on the distribution of points:

There appears to be a slight upward trend as latency increases (moving from left to right on the x-axis). However, there are also points with high jitter even at lower latency values.

Points are spread across the entire jitter range (y-axis) for most latency values (x-axis), indicating that jitter does not consistently increase or decrease with latency in a predictable manner for all data points.

Lines:

Latency Distribution: This line shows the distribution of latency values. Latency refers to the time it takes for data to travel from a source to a destination. In the graph, a higher latency value on the x-axis corresponds to a higher count of events on the y-axis, indicating more events occurred at that latency.

Jitter Distribution: This line shows the distribution of jitter values. Jitter is the variation in latency experienced over time. In the graph, a higher jitter value on the x-axis corresponds to a higher count of events on the y-axis, indicating more events occurred at that jitter level.

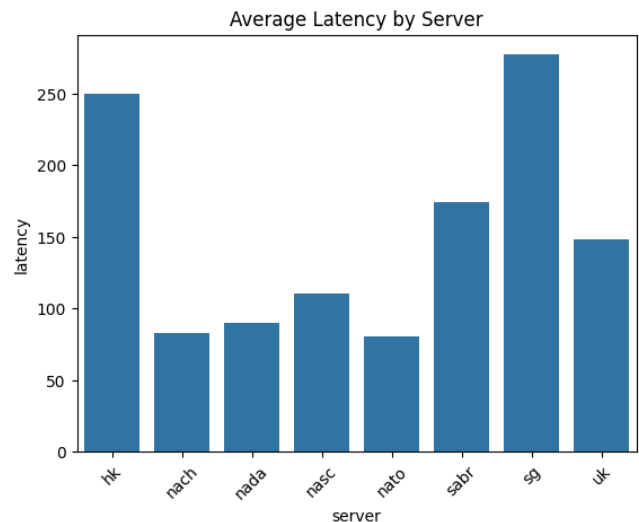


Figure 18: Average Latency based on Server

X-axis:

Represents the servers (referring to their location)

Y-axis: Represents the average latency in milliseconds (ms)

Bars: The height of each bar corresponds to the average latency experienced for that particular server. A higher bar indicates a higher average latency for that server.

Distributions by Server:

This type of map helps visualize how latency varies geographically across user locations. Areas with darker or more intense colors represent higher average latency, while lighter colors indicate lower latency.

By looking at the color distribution, you can identify regions or countries with generally higher or lower latency. This might be helpful in understanding how geographical factors influence user experience.

Circles on the map provide additional context about user distribution. Densely populated areas might have more circles clustered together.

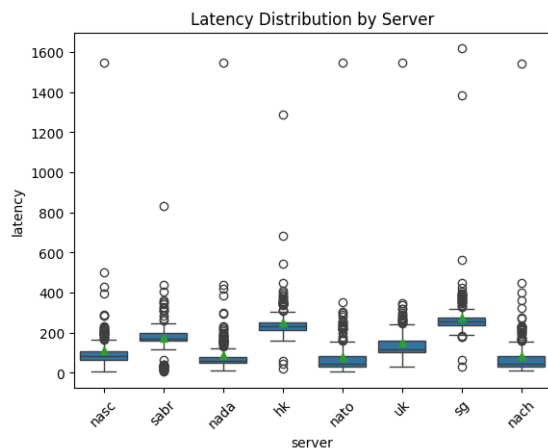


Figure 19: Latency Distribution based on Server

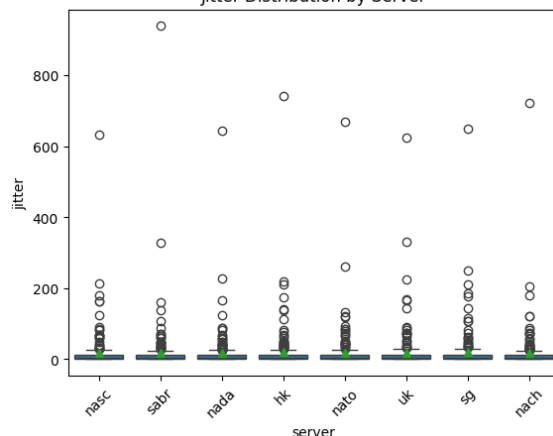


Figure 20: Jitter Distribution based on Server

Observations based on Correlation Matrix**Strong Positive Correlation:**

The correlation between latency and max is very close to 1 (dark red), indicating a strong positive correlation. This means higher overall latency values tend to coincide with higher maximum latency values, and vice versa, as expected.

The correlation between jitter and max is also positive and relatively strong (dark red), suggesting that higher jitter values are often accompanied by higher maximum latency values.

Moderate Positive Correlation:

The correlation between latency and average (avr) is positive (light red), but not as strong as the correlation with max. This suggests that higher overall latency tends to be associated with higher average latency, although some exceptions or outliers may affect the strength of this correlation.

There is a moderate positive correlation between jitter and average (avr) as well (light red), indicating that higher jitter values tend to occur with higher average latency values.

Weaker Correlations:

The correlation between latency and minimum (min) is close to zero (light color), indicating little to no linear relationship between overall latency and the minimum latency observed.

The correlation between jitter and minimum (min) is also weak (light color), suggesting that jitter does not have a strong linear relationship with the minimum latency observed.

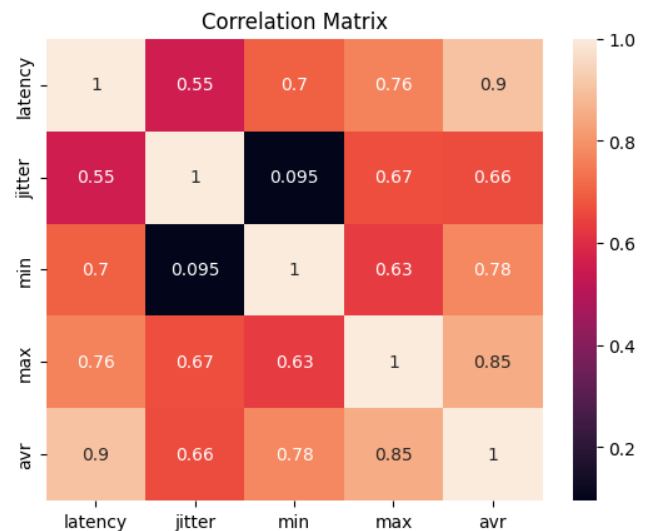


Figure 21: Correlation Matrix between variables

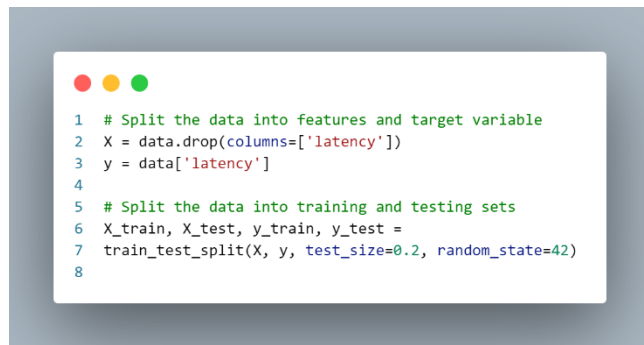
Performance Metrics

Analysis of the main dataset reveals key insights:

- **High Latency:** A considerable percentage of users encounter latencies greater than 160ms, which negatively impacts both the user experience and real-time gameplay.
- **Delay Jitter:** By resulting in uneven performance, latency variation, also known as jitter, adds even more complexity to the gaming experience.
- **Regional Inequities:** Players in under-served regions such as South America and parts of Asia face higher latencies, indicating a need for a more evenly distributed server network.

D. Dataset Splitting

We split our dataset for training and testing almost 80% of data is for training and 20% for testing. It is a basic rule in the machine learning process.



```
1 # Split the data into features and target variable
2 X = data.drop(columns=['latency'])
3 y = data['latency']
4
5 # Split the data into training and testing sets
6 X_train, X_test, y_train, y_test =
7 train_test_split(X, y, test_size=0.2, random_state=42)
8
```

Figure 22: Dataset Splitting Code Snapshot

E. Models Selections

The selection of the model was also an important process that involved the examination of the regression models that can be used in order to predict the network latency while taking into consideration the characteristics of the dataset and the purpose of the study. Three distinct regression models were considered for this study: Linear Regression, Random Forest and Gradient Boosting among others. Each model was evaluated not only by its intrinsic characteristics and the ability of the model to fit regression problems but also by its ability to fit the data characteristics of latency distribution.

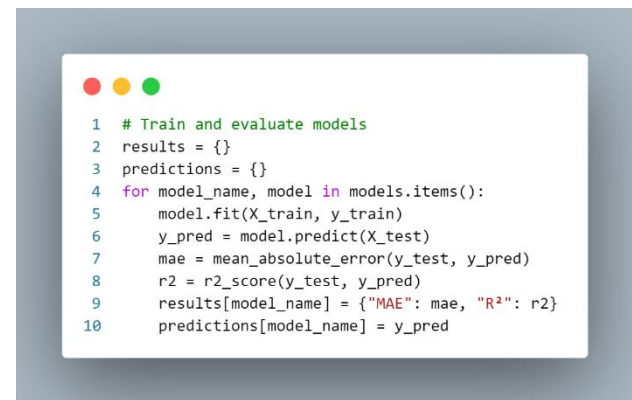


```
1 # Initialize models
2 models = {
3     "Linear Regression":
4     LinearRegression(),
5     "Random Forest":
6     RandomForestRegressor(n_estimators=100, random_state=42),
7     "Gradient Boosting":
8     GradientBoostingRegressor(n_estimators=100, random_state=42)
9 }
10
```

Figure 23: Models Selection Code Snapshot

F. Models Training and Evaluation

During model training and evaluation, we use the split-input on the dataset, where 80% of the data was utilized for training purposes and 20% of the data as the testing purpose. Lately, the training set and the test set were normalized to have the standardized distribution of values using the StandardScaler. In the next step, we trained the models of Linear Regression, Random Forest, and Gradient Boosting on the training set, tuning them to produce minimal error in predictions. To test on unseen data and avoid overfitting due to training the same data multiple times, K-fold cross-validation with K=5 was performed on the model.



```
1 # Train and evaluate models
2 results = {}
3 predictions = {}
4 for model_name, model in models.items():
5     model.fit(X_train, y_train)
6     y_pred = model.predict(X_test)
7     mae = mean_absolute_error(y_test, y_pred)
8     r2 = r2_score(y_test, y_pred)
9     results[model_name] = {"MAE": mae, "R²": r2}
10 predictions[model_name] = y_pred
```

Figure 24: Models Training and Evaluation Code Snapshot

Measurement based tool such as **Mean Absolute Error**, **Root Mean Squared Error**, and **R-squared** were used to assess performance of the constructed models. Hence performance metrics were employed to compare the models and select the one that best predicted and resembled accurate and precise network latency. In this way, the comprehensive approach helped to achieve a reliable set of latency values and provide useful parameters for assessing the dynamics of network performance.

The linear regression model was proved to be the most accurate model among all of the models as it got the lowest Mean Absolute Error (MAE). Behind the decision trees is the Random Forest model while Gradient Boosting is slightly behind the

Random Forest. However, what we must be cautious of, while interpreting this graph, is the magnitude of the y-axis. It is important to realize that if the scale is defined as very small indeed, then there may be not much difference between the models in question.

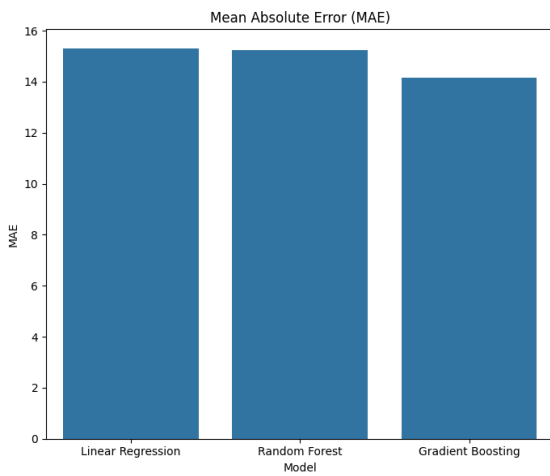


Figure 25: Models Training and Evaluation Code Snapshot

The degree to which the sample data fits the regression model can be measured by the R-squared (R^2) value, which is a statistic that determines the proportion of the variance in the dependent variable that is explained by the independent variable(s). In other words, R^2 helps determine how well the regression line fits the actual data points, the range being from 0 to 1, where 1 is considered a better fit.

Examining the results, found on the graph below, linear regression yields the highest mean R^2 value therefore suggesting it accounts for the greatest amount of variance in the data set. The second best classifier is again random, then gradient boosting comes next.

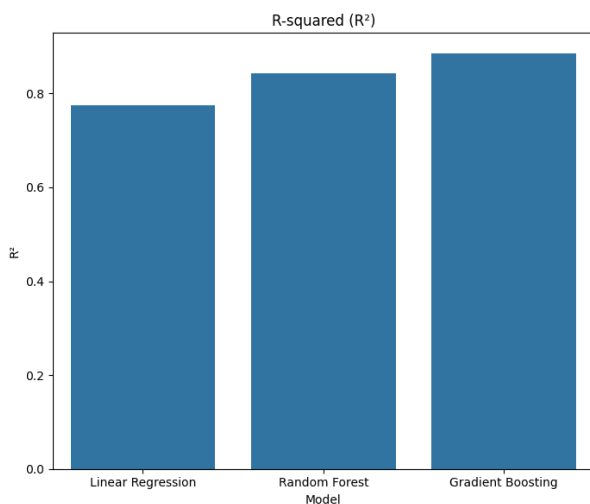


Figure 26: Models Training and Evaluation Code Snapshot

Actual vs Predicted Values Comparison

To show the relation between two variables, we use scatter plots. On these graphs, actual latency values are represented by x-axis; predicted latency values are represented by y-axis. Each data point represents a prediction. In a perfect correlation between actual and predicted values, all datapoints would line up on a 45-degree angle.

In all three of the graphs above, most of the points appear to cluster around a straight line, albeit with some scattering.

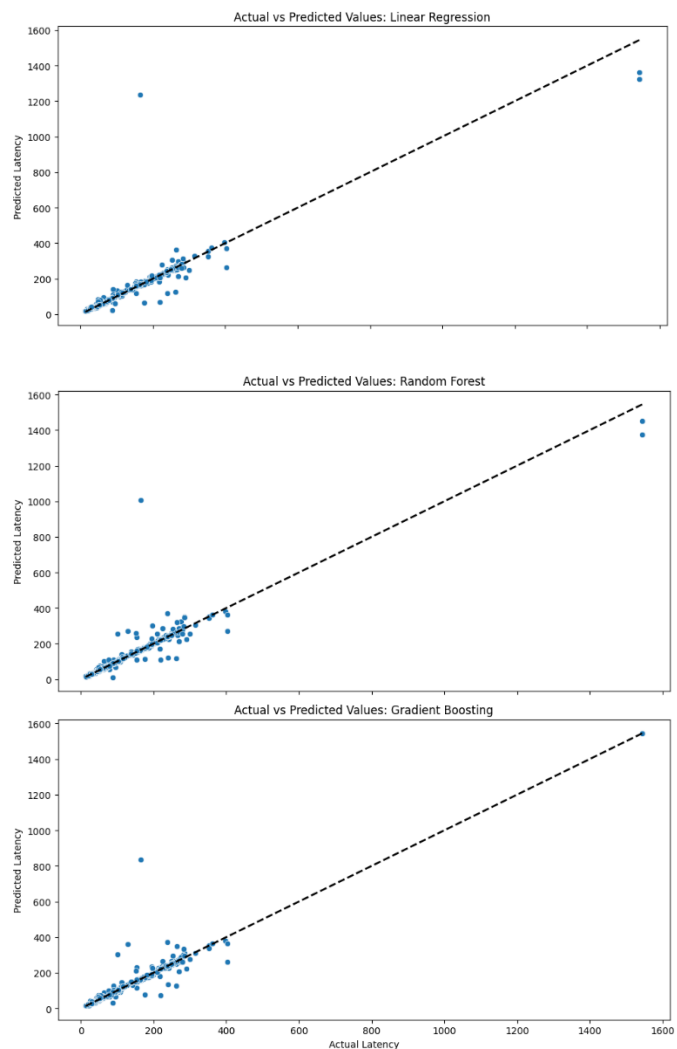


Figure 27: Actual vs Predicted Values between 3 models

This shows that for many of the data points, the models can be quite accurate in their predictions. However, it is also evident that several points lie far away from this line meaning that these models were not correct in predicting them.

V. INFRASTRUCTURE LIMITATIONS

The amount of servers currently in use is not enough to offer a worldwide user base low-latency gaming experiences. Real-time gameplay may be hampered by the high latencies and noticeable jitter seen, which would make for a frustrating user experience. These problems are especially noticeable in areas with lower network infrastructure and fewer server deployments.

VI. POTENTIAL SOLUTIONS

In order to tackle these obstacles, various approaches could be utilized, such as:

1. **Increasing Server Density:** By putting more servers in different places, you can physically close the distance between players and servers, which will cut down on latency. This strategy would entail adding more servers to underserved areas, like parts of South America and currently of Asia. EDGE GAME can be the best fit for this purpose. [11]

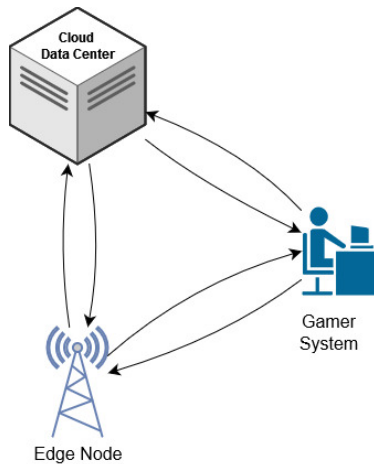


Figure 28: Infrastructure of the Edge Game

Outcomes of combining cloud gaming and edge gaming after setting up various edge nodes, which will work better than extra servers to accommodate various regions. between various regions. [12] [25].

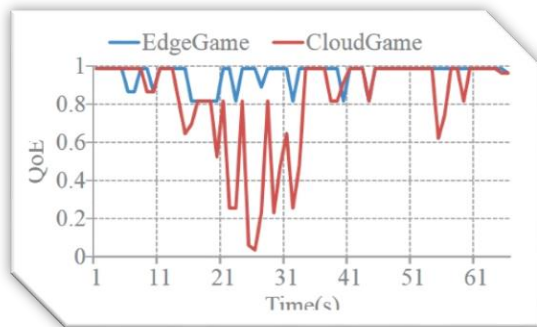


Figure 29: Results after installing edge nodes [12]

2. **Optimizing Server Location:** By carefully placing servers in relation to user density and network performance information, latency can be decreased and coverage can be improved. Using advanced algorithms to analyze player distribution and network conditions can help identify optimal server locations.
3. **Developing Network Technologies:** You can reduce latency even more by implementing state-of-the-art network technologies like edge computing and improved routing protocols. Especially with edge computing, latency can be greatly decreased by bringing computation closer to the user.
4. **Enhancing GPU Virtualization:** More concurrent game instances can be supported by enhanced GPU virtualization technologies, which will increase the performance and scalability of cloud gaming services. To guarantee exclusive access to resources, each game instance is given its own virtual GPU and GPU memory.
5. **Developing Detailed (QoE) Models:** The gaming experience can be enhanced by developing sophisticated Quality of Experience (QoE) models that dynamically adjust system parameters based on network conditions and player device capabilities. These models can use learning algorithms to adjust video coding bitrate and frame sampling rate in real-time.
6. **Smart Parameter Adaptation:** Implementing control-theoretic algorithms for smart parameter adaptation can optimize resource utilization and maintain a balance between workload and gaming experience. This approach can dynamically adjust parameters based on network conditions and player device capabilities, ensuring an optimal gaming experience. [31]
7. **Gaming Anywhere:** It significantly reduces latency in cloud gaming, enhancing performance through efficient processing and responsiveness optimization. By minimizing processing delay (PD) and optimizing server-side and client-side tasks, GamingAnywhere ensures swift user input responses. The system prioritizes responsiveness by breaking down response delay (RD) into components and targeting latency reduction. Additionally, GamingAnywhere reduces network traffic loads and uses efficient compression algorithms to optimize bandwidth usage without sacrificing video quality, maintaining high visual fidelity. Its open, extensible architecture allows for ongoing improvements and customizations, keeping GamingAnywhere at the forefront of cloud gaming performance.

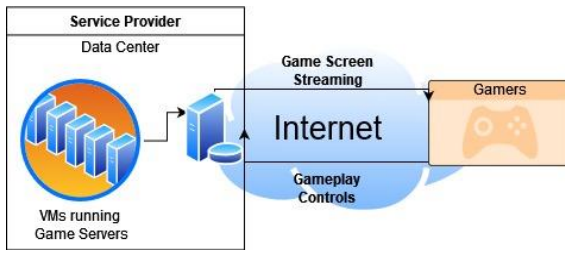


Figure 30: Flow Diagram for Cloud Gaming.

8. **NVIDIA'S GEFORCE GRID:** With the use of strong GPU servers, NVIDIA's GeForce GRID cloud gaming technology provides top-notch online gaming experiences. GeForce GRID eliminates the need for expensive gaming gear by enabling users to play visually demanding games on any device with a reliable internet connection.

With the help of NVIDIA's graphics processing know-how, GeForce GRID encrypts and streams game footage from distant servers to users' devices. By processing game inputs and rendering frames on cloud servers, this technique lowers latency and delivers the visual stream to the user's device as soon as possible. Even on low-powered devices, GeForce GRID provides responsive and fluid gaming by shifting the processing burden to the cloud.

Additionally, NVIDIA's GeForce GRID integrates cutting-edge technology such as adaptive bit-rate

VII. CONCLUSION

The rapid evolution and growing popularity of cloud gaming underscore both its potential and the challenges that lie ahead. Our exploration into cloud gaming has highlighted several critical areas for research and development. From the need for efficient resource utilization and reduced latency, as demonstrated by systems like GamingAnywhere and T-Gaming, to the integration of advanced GPU virtualization and content-dependent video codecs, there is a clear pathway for enhancing cloud gaming services.

We have seen that systems such as GamingAnywhere offer significant performance advantages, including lower processing delays and higher video quality compared to commercial platforms like OnLive and StreamMyGame. These findings emphasize the importance of open, extensible, and configurable architectures in driving innovation and performance in cloud gaming.

Moreover, the development of smart parameter adaptation and the leveraging of edge computing resources, as explored in systems like Edge Game, show promising avenues for further reducing latency and bandwidth consumption. These advancements ensure high-quality gaming experiences even under varying network conditions, enhancing the overall quality of Experience (QoE) for users also Specifically for

MMOGs, this study tackles the significant latency and bandwidth difficulties in cloud gaming. The distance between customers and datacenters causes traditional models to struggle and necessitate expensive infrastructure additions.

our research finds out that Cloud Fog, which reduces latency and data transfer by rendering and streaming game footage using neighboring super nodes. Test results indicate that Cloud Fog outperforms Edge Cloud and conventional models in terms of reaction times and playback continuity. These outcomes confirm that CloudFog is a practical, scalable, and affordable way to reduce latency in cloud gaming, improving user Quality of Experience (QoE) and emphasizing the promise of cutting-edge network designs in gaming.

The role of dedicated hardware solutions, such as NVIDIA's GeForce Grid, also highlights the potential for hardware manufacturers to contribute significantly to mitigating latency and improving rendering capabilities. This, combined with the continuous improvement in network technologies like LTE, can address some of the most pressing issues in cloud gaming, such as high interaction delays over cellular networks.

while the current state of cloud gaming represents just the tip of the iceberg, the future holds exciting possibilities. By addressing the identified research opportunities and challenges, from system-oriented game integration to human-centric QoE modeling, the research community and industry can collaboratively drive the development of high-quality, commercially viable cloud gaming platforms. The ongoing enhancements and innovations in this field will undoubtedly lead to more immersive and accessible gaming experiences for users worldwide.

ACKNOWLEDGMENT

I would like to express our sincere gratitude to Dr. Zunnurain Hussain, who taught us the Information Security course and provided invaluable guidance and supervision throughout the process of writing this paper. His expertise in the field and insightful feedback were instrumental in shaping my research and ensuring the quality of my work. I am deeply grateful for his dedication and support.

REFERENCES AND FOOTNOTES

- [1] P. Ross, "Cloud computing's killer app: Gaming," IEEE Spectrum, vol. 46, no. 3, p. 14, March 2009.
- [2] C.-Y. Huang, C.-H. Hsu, D.-Y. Chen, and K.-T. Chen, "Quantifying user satisfaction in mobile cloud games," in Proceedings of ACM Workshop on Mobile Video Delivery (MoViD'14), March 2014, pp. 4:1–4:6.
- [3] S. Choy, B. Wong, G. Simon, and C. Rosenberg, "The brewing storm in cloud gaming: A measurement study on cloud to end-user latency," in Proceedings of the IEEE/ACM NetGames 2012, October 2012.

- [4] D. Wu, Z. Xue, and J. He, "iCloudAccess: Costeffective streaming of video games from the cloud with low latency," *IEEE Transactions on Circuits and Systems for Video Technology*, January 2014, accepted to appear
- [5] C.-Y. Huang, C.-H. Hsu, Y.-C. Chang, and K.-T. Chen, "GamingAnywhere: An Open Cloud Gaming System," Department of Computer Science, National Taiwan Ocean University, Department of Computer Science, National Tsing Hua University, Institute of Information Science, Academia Sinica, Department of Electrical Engineering, National Taiwan University.
- [6] Online sales are expected to pass retail software sales in September 2011. <http://www.dfcint.com/wp/?p=311>.
- [7] Distribution and monetization strategies to increase revenues from cloud gaming, July 2012. <http://www.cgconfusa.com/report/documents/Content5minCloudGamingReportHighlights.pdf>
- [8] M. Claypool and K. Claypool. Latency and player actions in online games. *ACM*, 49:40–45, November 2006.
- [9] D. Zhang et al., "Delay-Optimal Proactive Service Framework for Block-Stream as a Service," *IEEE Wireless Commun. Lett.*, vol. 7, no. 4, Aug. 2018, pp. 598–601.
- [10] Z. Zhao, K. Hwang, and J. Villeta, "Game Cloud Design with Virtualized CPU/GPU Servers and Initial Performance Results," *Proc. 3rd Wksp. Scientific Cloud Computing Date*, 2012, pp. 23–30
- [11] H. Riiser et al., "Commute Path Bandwidth Traces from 3G Networks: Analysis and Applications," *Proc. 4th ACM Multimedia Systems Conference, MMSys '13*, New York, NY, USA, 2013, ACM, pp. 114–18.
- [12] H. Yin et al., "Edge Provisioning with Flexible Server Placement," *IEEE Trans. Parallel & Distributed Systems*, vol. 28, no. 4, Apr. 2017, pp. 1031–45.
- [13] X. Zhang et al., "Resource Provisioning in the Edge for IoT Applications with Multi-Level Services," *IEEE Internet of Things J.*, 2018, pp. 1–1.
- [14] C.-Y. Huang et al., "GamingAnywhere: The First Open Source Cloud Gaming System," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 10(1s), Jan. 2014, pp. 10:1–25.
- [15] Y. Xu et al., "A Cost-Efficient Cloud Gaming System at Scale," *IEEE Network*, vol. 32, no. 1, Jan. 2018, pp. 42–47.
- [16] N. Bilton. Video Game Industry Continues Major Growth, *Gartner Says. The New York Times*, 2011.
- [17] M. Jarschel, D. Schlosser, S. Scheuring, and T. Hobfeld. An Evaluation of QoE in Cloud Gaming Based on Subjective Tests. In *Proc. of IMIS*, 2011.
- [18] S. Choy, B. Wong, G. Simon, and C. Rosenberg. The brewing storm in cloud gaming: A measurement study on cloud to end-user latency. In *Proc. of NetGames*, 2012.
- [19] E. Carlini, M. Coppola, and L. Ricci. Integration of P2P and Clouds to support Massively Multiuser Virtual Environments. In *Proc. Of NetGames*, 2010.
- [20] C. Bezerra and C. Geyer. A load balancing scheme for massively multiplayer online games. *Multimedia Tools Appl*, 2009.
- [21] PlanetLab. <http://www.planet-lab.org/>, [Accessed in Nov 2014].
- [22] C. Huang, C. Hsu, Y. Chang, and K. Chen. GamingAnywhere: An Open Cloud Gaming System. In *Proc. of MMSys*, 2013.
- [23] S. Choy, B. Wong, G. Simon, and C. Rosenberg. EdgeCloud: A New Hybrid Platform for On-Demand Gaming. Technical Report CS-2012-19, University of Waterloo, 2012
- [24] Y. Lin and H. Shen, "Leveraging Fog to Extend Cloud Gaming for Thin-Client MMOG with High Quality of Experience," Department of Electrical and Computer Engineering, Clemson University, Clemson, SC, USA. [Online]. Email: {yuhual, [shenh](mailto:shenh@clemson.edu)}@clemson.edu.
- [25] X. Zhang, H. Chen, Y. Zhao, Z. Ma, Y. Xu, H. Huang, H. Yin, and D. O. Wu, "Improving Cloud Gaming Experience through Mobile Edge Computing".
- [26] T. Mangla, E. Halepovic, M. Ammar, and E. Zegura, "Mimic: Using passive network measurements to estimate HTTP-based adaptive video QoE metrics," in *Proc. Netw. Traffic Meas. Anal. Conf.*, 2017, pp. 1–6.
- [27] J. van der Hooft, S. Petrangeli, M. Claeys, J. Famaey, and F. De Turck, "A learning-based algorithm for improved bandwidth-awareness of adaptive streaming clients," in *Proc. IFIP/ IEEE Int. Symp. Integr. Netw. Manag.*, 2015, pp. 131–138
- [28] Y.-T. Lee, K.-T. Chen, H.-I. Su, and C.-L. Lei. Are all games equally cloud-gaming-friendly? an electromyographic approach. In *Proceedings of IEEE/ACM NetGames 2012*, Oct 2012.
- [29] A. S. Tanenbaum. *Computer Networks*. Prentice Hall Professional Technical Reference, 4th edition, 2002.

[30] N. Tolia, D. Andersen, and M. Satyanarayanan. Quantifying interactive user experience on thin clients. *IEEE Computer*, 39(3):46–52, March 2006.

[31] D. Wu, Z. Xue, and J. He, “Costeffective streaming of video games from the cloud with low latency,” *IEEE Transactions on Circuits and Systems for Video Technology*, January 2014, accepted to appear

[32] K.-W. Lee, B.-J. Ko, and S. Calo, “Adaptive server selection for large scale interactive online games,” *Computer Networks*, vol. 49, no. 1, pp. 84–102, September 2005