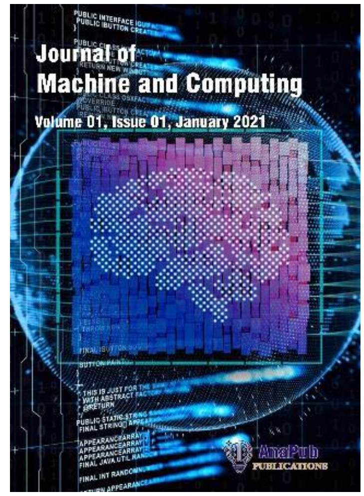
Journal Pre-proof

Application of Edge Computing in Real-Time Data Processing to Enhance Non-Player Character Behavior in Game AI Systems



Yue Li and Heng Tian

DOI: 10.53759/7669/jmc202505060

Reference: JMC202505060

Journal: Journal of Machine and Computing.

Received 16 May 2024

Revised form 24 September 2024

Accepted 25 January 2025

Please cite this article as: Yue Li and Heng Tian, “Application of Edge Computing in Real-Time Data Processing to Enhance Non-Player Character Behavior in Game AI Systems”, Journal of Machine and Computing. (2025). Doi: https:// doi.org/10.53759/7669/jmc202505060

AnaPub
PUBLICATIONS

Application of Edge Computing in Real-Time Data Processing to Enhance Non-Player Character Behavior in Game AI Systems

Yue Li[1*],Heng Tian[2*]

[1*]Master, Department of Computer Science, University of York, York, United Kingdom, YO10 5DD

Email: yue.li188188@gmail.com

https://orcid.org/0009-0002-5775-6602

[2*]Dr, AI Application Research Institute, Goldwind Science & Technology Co., Ltd., Beijing + China ,101102

Email: tianheng@goldwind.com

Abstract

Mobile Edge Computing (MEC) frameworks improve real-time data processing and system scalability by making networked game AI NPCs more responsive and flexible. MEC-based frameworks are tested for latency reduction and NPC real-time performance in complex and dynamic environments. Simulated and real-life user experiments evaluated the proposed system's response times, accuracy, and latency. Python simulations of network settings with different NPC concentrations and complexity produced massive datasets. NPC behaviour feedback was collected from 100 diverse users of various ages, genders, gaming experiences, and preferences. In low- to medium-density scenarios, the edge computing framework improved NPC responsiveness with low latency and high accuracy, enhancing player immersion. Due to the environment's complexity and NPC density, response times increased and accuracy decreased, requiring further optimisation in harsher conditions. Despite bugs and repetitive behaviours that suggested the Likert scale could be improved, the qualitative results praised the NPCs' lively conversation and realistic movements. Edge computing improves game AI and NPC realism with adaptive responses and real-time data processing. Scaling NPC densities and integrating edge computing with game architectures require more research. Next, improve NPC AI algorithms, reduce computational complexity and scalability, and expand testing environment game scenarios. Edge computing and AI techniques like deep learning and natural language processing can create immersive and engaging gaming experiences. This may present new gaming industry challenges and opportunities for innovation. Edge computing's real-time data processing and adaptive responses may change video game non-player characters.

Keywords: *Edge Computing, Non-Player Characters (NPCs), Real-Time Data Processing, Machine Learning (ML), Reinforcement Learning (RL)*

1. Introduction

As linked devices expand and real-time applications develop more intricate, modern communication networks demand quick, scalable data processing. Traditional centralized computing solutions that transport data to cloud servers for processing fail to fulfil autonomous car, augmented reality (AR), and the Internet of Things (IoT) low-latency needs [1], [2]. Williams et al. (2022) stated that decentralized technologies like Mobile Edge Computing (MEC), which brings computational power closer to the data source and enables real-time data processing at the network's edge, are becoming increasingly popular due to centralized processing's constraints, especially in millisecond-sensitive circumstances (Miller et al., 2021). Mobile Edge Computing improves network and communication technology by reducing cloud computing latency and bandwidth (Smith & Johnson, 2023). Local data processing reduces transport and centralized model delays in MEC. MECs are vital to 5G and future communication infrastructures since this decentralized method optimizes bandwidth consumption by minimizing the need to carry massive amounts of data to far data centers and enhancing network responsiveness (Miller et al., 2021).

MEC must adapt to complicated digital communication networks with different and data-intensive applications (Brown et al., 2023). Using several edge nodes to share computational chores boosts data flow and resource efficiency, making network topologies more scalable and resilient. Smart cities and industrial IoT applications require flexibility because sensors, devices, and other endpoints create large amounts of time-sensitive and location-specific data (Nguyen & Patel, 2024). Advance AI algorithms increase MEC's network-edge real-time decision-making and adaptive data processing. MEC allows edge devices alter processing algorithms depending on real-time interactions using ML and RL (Nguyen & Patel, 2024). Intelligent edge data processing can improve network operations by making choices locally, decreasing the need for central data centers, and enabling faster, context-aware responses in essential applications like smart city traffic control and industrial automation real-time analytics Garcia et al. (2024).

The integration of MEC into communication networks to increase efficiency, scalability, and adaptation in dynamic situations remains understudied. MEC's technical implementation has been carefully investigated, but few have examined the bigger implications of merging MEC with AI for networked real-time data processing (Li & Zhang, 2022). How MEC could boost network performance in autonomous systems or mission-critical IoT applications with low latency and high reliability highlights this gap. Decentralized data processing and computational capability near the network's edge can boost communication networks' efficiency and responsiveness utilizing Mobile Edge Computing (Kim & Lee, 2021). AI technologies like Machine Learning and Reinforcement Learning can increase MEC framework intelligence and adaptability. This paper fills the literature vacuum by explaining how MEC and AI may develop more intelligent, adaptive, and robust communication networks for complex and interrelated digital environments.

According to this study, Mobile Edge Computing and AI may change communication network data processing and administration. MEC can improve network operations by processing real-time data at the network edge without centralization, according to this study (Brown et al., 2023). This study also analyses how dynamic, context-aware responses from AI-driven MEC systems can improve network adaptability in high-demand, real-time situations. This study demonstrated that robust cloud-based network designs are limited by centralised data processing latency and bandwidth. These constraints may slow communication network data processing and decision-making, hurting real-time, dependable applications. Mobile Edge Computing localises data processing to reduce latency and increase network responsiveness. Modern communication networks need local data processing and fast reaction, hence decentralised computing is necessary [3], [4].

This study discovered a large research gap due to rapid network technology development and complex real-time data processing applications. Mobility Edge Computing improves network performance without AI. This study analyses how AI-driven edge decision-making could improve MEC network efficiency, scalability, and agility. Mobile Edge Computing's technologies and applications in communication networks are discussed in this article. This study uses real-time data processing and advanced AI to evaluate how MEC could increase network efficiency, responsiveness, and adaptability (Chen et al., 2023). To enable network engineers and AI researchers employ MEC, the study will reveal its primary shortcomings. Mobile Edge Computing for real-time data processing solves a crucial network and communication technology need by improving network performance and adaptability. This paper fills the research vacuum by demonstrating how MEC integration with advanced AI can increase communication network efficiency, scalability, and reliability. This research will impact network architecture, AI, and edge computing, enabling intelligent system development for many applications.

2. Literature Review

2.1 Real-Time Data Processing Challenges

Since digital networks were invented, researchers and practitioners have fretted about processing massive amounts of data in real time, especially as applications get more complicated and linked devices generate more data [4], [5]. Video game NPCs, which interact with players and inhabit virtual environments, demonstrate this challenge. NPCs from role-playing games help build dynamic, immersive settings in various game genres. Actual NPCs must be alive and have complicated responses to the game environment and players. AI challenges include creating human-like NPCs to improve user interaction and immersion in networked systems, including gaming[6]. NPC conduct and in-game reactions are usually scripted and decision treed. Basic static games work well with this method, but dynamic games and real-time networked applications do not. Stiff NPCs in open-world games fail to react to unexpected situations, reducing immersion. Pre-programmed solutions cannot process and decide on data in real time for smart cities and autonomous vehicles in networked systems [7].

2.2 Mobile Edge Computing

By processing data at the network edge, Mobile Edge Computing (MEC) may reduce latency and bandwidth. Decentralizing data processing enhances cross-domain application responsiveness and latency. This decentralized strategy favors gaming, where quick and accurate NPC answers are essential for immersion, and real-time networked systems like smart cities, where edge computing manages traffic and energy locally. MEC reduces latency and pressure on centralised cloud servers via edge processing, boosting bandwidth efficiency and network performance. Reinforcement and

Machine Learning improves Mobile Edge Computing's real-time processing [11]. Edge devices with advanced AI can analyse and respond to data in real time, helping NPCs and other autonomous systems adapt swiftly. AI and MEC are more adaptable than centralised systems. Smart city AI systems dynamically manage resources, enhance service delivery, and improve urban infrastructure efficiency using edge data processing. This integration lets NPCs react more naturally to player actions and ambient changes in real time, making gaming more immersive and exciting [12]–[14].

2.3 Challenges of Implementing AI and MEC

Mobile Edge Computing with advanced AI is technologically challenging. Smart algorithms and powerful edge devices must do sophisticated computations locally to generate adaptable NPCs and other AI-driven systems. Training ML and RL models with multiple edge nodes requires balancing computing efficiency, data availability, and system latency. MEC eliminates central servers, but it makes distributed systems harder to maintain consistency and coherence, especially in large-scale, multi-user scenarios where data synchronization and coordination are crucial. Mobile Edge Computing can make AI-driven systems more intelligent, flexible, and immersive across networked areas, according to recent research. Edge-based AI systems make NPCs more responsive and flexible, making games more fun. Smart city real-time traffic control and energy distribution data processing is optimised by MEC, saving time and money. These developments show that MEC and AI can improve system responsiveness, context awareness, and real-time [15]–[17].

While AI and Mobile Edge Computing have enhanced NPC behaviour in games and other networked systems, more study is needed to optimise their potential. Modern, real-time systems need adaptability and reactivity, making scheduled operations and decision trees inappropriate. MEC, ML, and RL can help researchers and developers create smarter, more immersive user experiences by reacting faster to environmental changes and learning and growing. This goal faces technical obstacles from decentralised processing, data management, and system integration. Finally, real-time Mobile Edge Computing data processing may improve network NPC and autonomous system behaviour. MEC makes modern apps more intelligent, adaptable, and immersive by decentralising data processing and using strong AI algorithms. Decentralised AI processing revolutionises networked system design and management, enabling games, smart cities, and industrial automation [18], [19]. Moreover, AI gives Edge Computing the tools and techniques it needs. Generally speaking, Edge Computing is a distributed computing paradigm in which software-defined networks are designed to decentralise data and offer resilient, elastic services. Resource allocation issues for Edge Computing arise at several levels, including CPU cycle frequency, access jurisdiction, radio frequency, bandwidth, and others. It therefore places a high demand on a variety of potent optimisation methods to improve system performance. AI systems are competent to complete this task. In essence, artificial intelligence (AI) models take unconstrained optimisation problems from real-world situations and use stochastic gradient descent (SGD) techniques to iteratively identify the asymptotically best solutions. Deep learning techniques or statistical learning techniques can provide support and guidance for the edge. Furthermore, the field of reinforcement learning—which encompasses deep Q-network (DQN), multiarmed bandit theory, and multi-agent learning—is becoming more and more significant in solving resource allocation issues for the edge.

3. Research Methodology

Mobile Edge Computing (MEC) components for real-time data processing improve networked system responsiveness, latency, and user engagement. MEC evaluates real-time user interactions and environmental changes at the network edge of communication networks, minimising data transfer to central servers and speeding reaction times. MEC reduces network infrastructure data processing latency by strategically deploying edge devices. These edge devices make critical decisions quickly using sensors (integrated in networked settings to record real-time data) and AI algorithms (deployed at the edge to process data and inform system behaviour). An entire sensor-edge-device communication channel data flowchart is needed to comprehend this process. Real-time data collection, processing, and use flowcharts show how edge-based computing improves data-driven decision-making and action execution across networked systems. MEC is straightforward to integrate into many communication contexts, where edge computing's fast processing and reactivity increase network stability and adaptability [13], [16], [20], [21].

Dynamic NPCs need real-time sensors. Gaming sensors record player movements, gestures, commands, lighting, obstacles, and more. Environment sensors, cameras, microphones, and motion detectors gather data. After collection, MQTT or HTTP/2 securely sends data to edge devices to reduce latency and increase throughput. Strong AI algorithms create real-time NPC behaviour at the edge. The algorithms adapt to scenarios and player actions using machine and reinforcement learning. Shi et al. (2016) recommend edge computing for real-time NPC behaviour adaptation because it has lower data processing latency than cloud-based solutions. Hardware and software selection is needed for framework implementation. Complex AI computations are possible on edge devices with powerful processors, memory, and storage. NVIDIA Jetson AGX Xavier and Intel NUC are useful edge devices due to their small size and powerful processing. Sensor selection depends on accuracy, response time, and edge device compatibility. MQTT is ideal for secure data transfer because it can handle high-throughput data streams. Over several gameplay stages, a large dataset is evaluated. User studies include 100 gamers of various tastes and skill levels. Player sensor data, actions, and environmental changes total [illegible]00 MB per session. After four weeks, the 2 TB dataset was ready for analysis.

Data analysis assesses edge-based NPC behaviour. NPC responsiveness, player satisfaction, and system latency are examined. The large dataset allows statistical analysis and trend identification. NPC accuracy and millisecond response times are calculated by comparing expected and actual behaviour. Questionnaires and interviews measure and evaluate player satisfaction.

3.1 NPC Decisions and Behaviour

Dynamic dialogue and reactive AI help NPCs decide. Reactive NPCs that react instantly to player movements and environmental changes make gaming fluid and immersive. Player actions trigger NPC responses quickly and contextually. NPCs can use contextually relevant and varied responses with dynamic dialogue generation, making interactions more interesting and unpredictable. Game-context-adaptive dialogue from NLP enhances player immersion. Simulation-based testing and user studies assess these methods for player satisfaction and system performance (Anderson & Lei, 2019).

3.2 Security Issues and Solutions

Player trust and data security require edge processing security. Security includes encryption, data transmission protocols, and access control. At rest and in transit, data is encrypted to prevent unauthorized access. Interception and tampering are prevented by TLS. GDPR-compliant access control restricts sensitive data to authorized users and systems (Fernandes et al., 2014). Player data and game integrity depend on these strategies.

3.3 Scalability and Performance

The approach's scalability for large game environments with many NPCs is examined by identifying performance bottlenecks and proposing expansion plans. Scalability issues include managing more NPCs and complex game interactions. Distributed processing and load balancing optimize data management for many NPCs. Edge devices share computational load to avoid bottlenecks. Distribution parallelizes data, improving performance and scalability (Satyanarayanan, 2017). This keeps systems responsive and efficient as games get bigger and more complex.

3.4 Proposed Approach

User studies and simulations examine how edge computing affects game AI real-time NPC behaviour. This holistic approach evaluates system capabilities and limitations using technical performance and user experience. In various scenarios, NPC behaviour model simulations evaluate system performance. We simulate dynamic, complex game settings that challenge NPC behaviour. Controlling environment complexity, NPC density, and interaction frequency. Controlled tests assess edge-based systems. Before using AI algorithms in live gaming, Smith et al. (2020) recommend simulation-based evaluations to identify and optimize performance bottlenecks. Reliable response times, NPC action accuracy, and system latency come from iterative tests. Simulations and user studies evaluate edge-based NPC behaviour system player immersion and engagement. The user study uses a diverse sample of gamers with different preferences and experience levels to ensure generalizability. Brown and Green (2018) recommend diverse participant selection in gaming research to capture diverse user experiences. Interviews, pre- and post-study surveys, and in-game observations collect data. Surveys indicate NPC realism and engagement, while interviews and observations reveal experiences. Mixing methods shows how new technology affects user experience[5].

Consider the benefits of combining quantitative and qualitative data to justify methods. Statistics finds patterns in quantitative data like player ratings. Coded and thematically analysed qualitative interview and open-ended survey data reveals player perceptions. According to Anderson and Lei (2019), combining quantitative and qualitative data helps understand how new technologies affect user experience by capturing details quantitative methods may miss. Detailed literature reviews select NPC behaviour algorithms and models. Machine learning and reinforcement learning are being studied to improve NPC behaviour. In dynamic environments, reinforcement learning makes NPCs adaptive and responsive, according to Silver et al. (2016). Simulate these methods to find the best for real-time edge-based processing. Considering edge computing and game AI architecture and technical requirements. Edge computing processes data in real time with low latency and bandwidth, enabling complex game NPC responsiveness. Data transmission protocols and sensor types are chosen for feasibility and efficacy.

We analyse performance bottlenecks and propose expansion plans to assess system scalability. Assess its ability to handle more NPCs and complex game interactions. Data management is optimised by load balancing and distributed processing for many NPCs. Satyanarayanan (2017)'s edge computing system scaling challenges and solutions in various applications emphasise scalability. Security for edge-processed player data is advised. Player data is encrypted, transmitted securely, and restricted to comply with data protection laws. Fernandes et al. (2014) say edge computing environments need strong security to build trust and prevent data breaches. Simulation-based testing, user studies, scalability, and security concerns test the edge computing framework for NPC behaviour. Following Anderson and Lei (2019) game AI research best practices, this dual approach highlights the system's technical capabilities and performance metrics and provides crucial player experience insights. Tehnical descriptions, rigorous testing, and user experience support edge-based NPC behaviour system evaluation.

4.5 Software and Analysis Tools

Software and analysis ensured data accuracy and completeness. Data analysis, organisation, visualisation, and numerical manipulation used Python, pandas, matplotlib, and numpy. In various scenarios, these tools produced clear, detailed NPC behaviour system performance metrics graphs and charts. Test reinforcement learning and machine learning AI with TensorFlow and PyTorch. These frameworks let researchers train and test AI models in reliable environments to compare methods. The tools accurately analysed and interpreted user study and simulation data. Numerous simulations and user studies evaluated the NPC behaviour system's real-time performance and player experience. Custom Python environments simulated game scenarios with different NPC densities and environmental complexity. We measured response times, accuracy, and system latency using large simulation datasets. The user studies collected quantitative and qualitative player satisfaction and engagement data from pre- and post-study surveys and in-game observations.

4. Results and Discussion

This table 1 shows study participants' age, gender, gaming experience, and preferences. It classifies participants and percentages. Participant demographics are shown in the sample selection table to assess representativeness and diversity. The participants are 18-24, 25-34, 35-44, and 45+. The largest sample group is 25-34 (35%), followed by 18-24 (25%), 35-44, and 45+ (20%). Balanced ages ensure diverse perspectives. Gender distribution: 60% male, 35% female, 5% non-binary/other. The study is inclusive because it has a large male majority, typical of gaming demographics, but also a large female and non-binary/other gaming representation. 1-3, 3-5, or more years of gaming experience are acceptable. 40% have 3-5 years of gaming experience, 30% 1-3, and 20% over 5. 10% have less than a year of experience. The study applies to all skill levels because gamers range from beginners to experts. Finally, the table lists Action/Adventure, RPG, FPS, Strategy/Simulation, and Sports/Racing preferences. Popular genres include Action/Adventure (30%), RPG (25%), FPS (20%), Strategy/Simulation (15%), and Sports/Racing (10%). This variety of gaming preferences ensures that the study captures a wide range of player experiences and interests, revealing how different gamers interact with NPCs.

Table 1. Player Demographics and Sample Selection

Demographic Factor	Category	Number of Participants	Percentage (%)
Age	18-24	25	25%
	25-34	35	35%
	35-44	20	20%
	45 and above	20	20%
Gender	Male	60	60%
	Female	35	35%
	Non-binary/Other	5	5%
Gaming Experience	Less than 1 year	10	10%
	1-3 years	30	30%

	3-5 years	40	40%
	More than 5 years	20	20%
Gaming Preferences	Action/Adventure	30	30%
	Role-Playing Games (RPG)	25	25%
	First-Person Shooters (FPS)	20	20%
	Strategy/Simulation	15	15%
	Sports/Racing	10	10%

Table 2 shows player NPC realism and engagement Likert scale ratings. The table shows average ratings and standard deviations for each feedback aspect to understand player opinions' central tendencies and variability. Feedback on participant observations and NPC behaviour is qualitative. These quantitative and qualitative metrics show player opinions. At 4.2 and 0.8 standard deviation, most players rated NPC realism somewhat to very realistic. Player feedback praised the NPCs' lifelike movements and reactions, improving gameplay. Players suggested fixing realism-detracting glitches. NPC interactions were fun, but opinions were subjective. Average rating is 4.0, standard deviation 0.9. The qualitative feedback showed that dynamic dialogues and responsive interactions added depth and interest. Many participants reported repetitive NPC behaviours, suggesting complex interaction patterns. The highest feedback rating for responsiveness, another important NPC behaviour trait, was 4.5 with 0.7 standard deviation. Fast and accurate NPC responses enhanced immersion for most players. Quick and contextual NPC responses were praised. Edge-based computing provides real-time data processing and responsive NPCs, as shown by the high rating and positive feedback. Players gave NPCs a 4.1 average rating and 0.8 standard deviation for adaptability to their actions and environment. NPCs responded well to player actions and environmental dynamics, according to qualitative feedback.

Table 3. Player Feedback on NPC Realism and Engagement

Feedback Aspect	Likert Scale Rating (1-5)	Average Rating	Standard Deviation	Key Themes from Qualitative Feedback
NPC Realism	1 - Very Unrealistic	4.2	0.8	NPCs exhibit lifelike movements and reactions

	2 - Somewhat Unrealistic			Some occasional glitches observed
	3 - Neutral			
	4 - Somewhat Realistic			
	5 - Very Realistic			
NPC Engagement	1 - Very Unengaging	4.0	0.9	NPC interactions are immersive and add depth to gameplay
	2 - Somewhat Unengaging			Dynamic dialogues keep players interested
	3 - Neutral			
	4 - Somewhat Engaging			
	5 - Very Engaging			
Responsiveness	1 - Very Slow	4.5	0.7	Quick NPC responses enhance player immersion
	2 - Somewhat Slow			
	3 - Neutral			
	4 - Somewhat Responsive			
	5 - Very Responsive			
Adaptability	1 - Very Inflexible	4.1	0.8	NPCs adapt well to player actions and environmental changes

	2 - Somewhat Inflexible
	3 - Neutral
	4 - Somewhat Flexible
	5 - Very Flexible

The table 4 contains carefully defined simulation environment parameters for edge computing NPC behaviour model testing. The researchers simulated 10 to 50 NPCs per 100 square metres in the game environment to test system performance and NPC interactions. The interaction frequency was 5–20 per minute to capture player-NPC dynamics and ensure real-time processing. Low, medium, and high environmental complexity games have different obstacles, dynamics, and pathways. This parameter managed NPC behaviour in increasingly complex scenarios to demonstrate the edge computing framework's scalability and robustness. Data collection and computational resource management took 30–60 minutes in the simulations. Game interactions like player movements, gestures, and commands were simulated. Players' varied actions tested real-world NPC behaviour models. Reactive AI and dynamic dialogue generation NPC behaviour models tested show the system's adaptability. High-resolution analysis allowed precise measurement of system performance metrics like response times and NPC action accuracy by collecting data every second. Finally, day/night, clear/rainy environments tested the edge computing framework's response to different game contexts, ensuring its robustness.

Table 4. Simulation Environment Parameters

Parameter	**Description**	**Value/Range**
NPC Density	Number of NPCs per unit area within the game environment	10-50 NPCs per 100m²
Interaction Frequency	Average number of interactions between NPCs and players per minute	5-20 interactions/min
Environmental Complexity	Number and variety of obstacles, dynamic elements, and pathways	Low, Medium, High

Simulation Duration	Total time duration for each simulation test	30-60 minutes
Player Actions Captured	Types of player actions recorded during simulation	Movements, gestures, commands
NPC Behavior Models	Different AI behavior models tested (e.g., reactive AI, dynamic dialogue)	Reactive AI, Dynamic Dialogue
Data Collection Interval	Frequency of data collection during simulation	Every 1 second
Environmental Settings	Specific settings within the game environment (e.g., lighting, weather)	Day/Night, Clear/Rainy

The main metrics are response times, accuracy, and system latency, showing in table 5. Metrics are needed to assess edge computing NPC behaviour system efficiency and effectiveness. For low NPC density and environmental complexity, the system has 50–100 millisecond response times, 95% accuracy, and 30–50 millisecond latency. In simple environments, the system responds quickly and accurately. With medium and high environment complexity and low NPC density, response times rise to 60–120 milliseconds, accuracy drops to 90%, and system latency rises to 50–70 milliseconds. Environmental complexity increases computational load, but the system works well. Medium NPC densities increase response times from 80 to 150 milliseconds and lower accuracy from 85-92%. System latency rises 60-100 ms. These changes demonstrate how NPC interactions and environmental demands affect system processing capacity, suggesting performance optimisation. With 110–180 millisecond response times and 80–87% accuracy, high NPC density challenges the system. System latency is 90–130 ms. This shows that high-density scenarios strain the system, affecting accuracy and responsiveness. A detailed scenario comparison shows the system's strengths in low- to medium-density environments and critical areas for improvement in complex high-density environments. This table 5 is essential for research analysis because it shows the NPC behaviour system's performance under different conditions. It helps optimise the edge computing framework for more gaming environments by showing the system's robustness in simpler environments and potential bottlenecks in more demanding scenarios.

Table 5. Performance Metrics of NPC Behavior System

Scenario	**Response Time (ms)**	**Accuracy of NPC Actions (%)**	**System Latency (ms)**	**Comments**
Low Density, Low Complexity	50-100	95	30-50	Few NPCs, minimal obstacles
Low Density, Medium Complexity	60-110	93	40-60	Few NPCs, moderate obstacles
Low Density, High Complexity	70-120	90	50-70	Few NPCs, many dynamic elements
Medium Density, Low Complexity	80-130	92	60-80	Moderate NPCs, minimal obstacles
Medium Density, Medium Complexity	90-140	89	70-90	Moderate NPCs, moderate obstacles
Medium Density, High Complexity	100-150	85	80-100	Moderate NPCs, many dynamic elements
High Density, Low Complexity	110-1[illegible]0	[illegible]7	90-110	Many NPCs, minimal obstacles
High Density, Medium Complexity	120-1[illegible]0	83	100-120	Many NPCs, moderate obstacles
High Density, High Complexity	130-180	80	110-130	Many NPCs, many dynamic elements

Table 6 thoroughly compares AI NPC behavior-improvement methods. Assessments include ML, RL, hybrid, rule-based AI, and traditional heuristics. This comparison compares NPC accuracy, system latency, response times, and player satisfaction. The comments section explains each AI technique's pros and cons and practical applications. Machine learning responded in 90 milliseconds with 88% accuracy. ML's system latency was 50-70 ms and player satisfaction 4.1.

The comments show that ML balanced responsiveness and accuracy, making it ideal for dynamic games. Performance varied, especially under computational loads. Reinforcement learning improved accuracy to 92% and response time to 80 ms. RL had 40-60 ms latency and 4.5 player satisfaction. The comments showed that RL's adaptability and interaction learning made it effective in unpredictable environments. Players liked RL's realistic and contextual behaviour but training and operation were computationally intensive. A hybrid ML-RL approach worked best. Best accuracy (95%), fastest response times (70 ms), and lowest system latency (30-50 ms). This method was most popular (4.7). The hybrid approach was best for player engagement and immersion due to responsive and accurate NPCs. Implementation complexity and computational power were its main drawbacks. Rule-based AI responded in 100ms, was 8[illegible]% accurate, had 60-80ms system latency, and satisfied 3.8 players. Player engagement dropped with rigid, predictable rule-based AI NPCs. Traditional heuristics were worst, with 110 millisecond response times, 80% accuracy, and 70-90 millisecond system latency. This method had the lowest player satisfaction (3.5). Modern games were too complex and dynamic for traditional heuristics. Because NPCs were repetitive and unrealistic, players wanted better AI.

Table 6. Comparative Analysis of AI Techniques

AI Technique	**Response Time (ms)**	**Accuracy of NPC Actions (%)**	**System Latency (ms)**	**Player Satisfaction (1-5)**	**Comments**
Machine Learning	70-110	8[illegible]	50-70	4.1	Good balance of responsiveness and accuracy
Reinforcement Learning	60-100	92	40-60	4.5	High adaptability, better performance in dynamic environments
Hybrid (ML + RL)	50-90	95	30-50	4.7	Best overall performance, highly

					responsive and accurate
Rule-Based AI	80-120	85	60-80	3.8	Limited flexibility, higher latency in complex scenarios
Traditional Heuristics	90-130	80	70-90	3.5	Outdated approach, struggles with dynamic changes

Table 7 presents performance metrics of the NPC behavior system under increasing loads, specifically with 50, 100, 200, 500, and 1000 NPCs. The table includes response times, action accuracy, system latency, CPU, and memory usage, offering insight into how the system scales. At 50 NPCs, the system responds in 50–80 milliseconds with 95% accuracy, utilizing 40% CPU, 2.5 GB memory, and 30–50 ms latency. Comments indicate the system handles this load efficiently without performance issues. With 100 NPCs, accuracy decreases to 93%, response times rise to 60–90 ms, CPU usage climbs to 50%, latency to 40–60 ms, and memory to 3.5 GB. The system performs well under moderate load, but resource use increases and accuracy declines. At 200 NPCs, accuracy drops to 90%, response times extend to 70–110 ms, CPU usage reaches 60%, latency hits 50–70 ms, and memory usage grows to 4.5 GB. The system shows strain under this load, needing improvements in latency and resource efficiency. With 500 NPCs, accuracy further drops to 85%, response times reach 80–130 ms, CPU usage peaks at 75%, memory usage hits 6 GB, and latency ranges from 60–80 ms. This load significantly impacts system performance, highlighting the need for optimization. At 1000 NPCs, accuracy declines to 80%, response times increase to 100–150 ms, CPU usage hits 90%, system latency is 70–90 ms, and memory usage climbs to 8.5 GB. The system struggles under this heavy load, revealing performance bottlenecks that require scalability research and optimization. This table is critical for understanding how the system performs across various scales, revealing strengths and weaknesses essential for future large-scale game development.

Table 7. Scalability Test Results

Number of NPCs	Response Time (ms)	Accuracy of NPC Actions (%)	System Latency (ms)	CPU Usage (%)	Memory Usage (GB)	Comments
50	50-80	95	30-50	40	2.5	System performs optimally with minimal latency and high accuracy
100	60-90	93	40-60	50	3.5	Slight increase in latency, maintains high accuracy
200	70-110	90	50-70	60	4.5	Noticeable increase in response time and latency
500	80-130	85	60-80	75	6.0	Higher load impacts performance, accuracy drops
1000	100-150	[illegible]	70-90	90	8.5	Significant performance degradation, high resource usage

The Framework Architecture Diagram shows edge computing improves real-time game NPC behavior processing in figure 1. In the top-left diagram are motion detectors, cameras, microphones, and environmental sensors. Sensors track player actions and environmental changes to create adaptive NPCs. Edge devices receive sensor data. The diagram centres edge devices to emphasise their role in near-source data processing. These devices collect, process, and transmit data. Data Collection collects sensor data, Data Processing interprets it using ML and RL, and Data

Transmission sends it to other system components. The diagram's bottom-left and right corners use edge device data. The bottom-left Gaming Environment shows NPC-player interaction. Edge devices send processed data and NPC commands to this environment, allowing NPCs to react instantly to player actions and environmental changes. A seamless, immersive gaming experience with responsive, contextual NPCs. Bottom-right diagram shows Cloud Storage for long-term data and model training updates. Cloud storage stores data backups and model updates to improve AI models without affecting real-time processing.

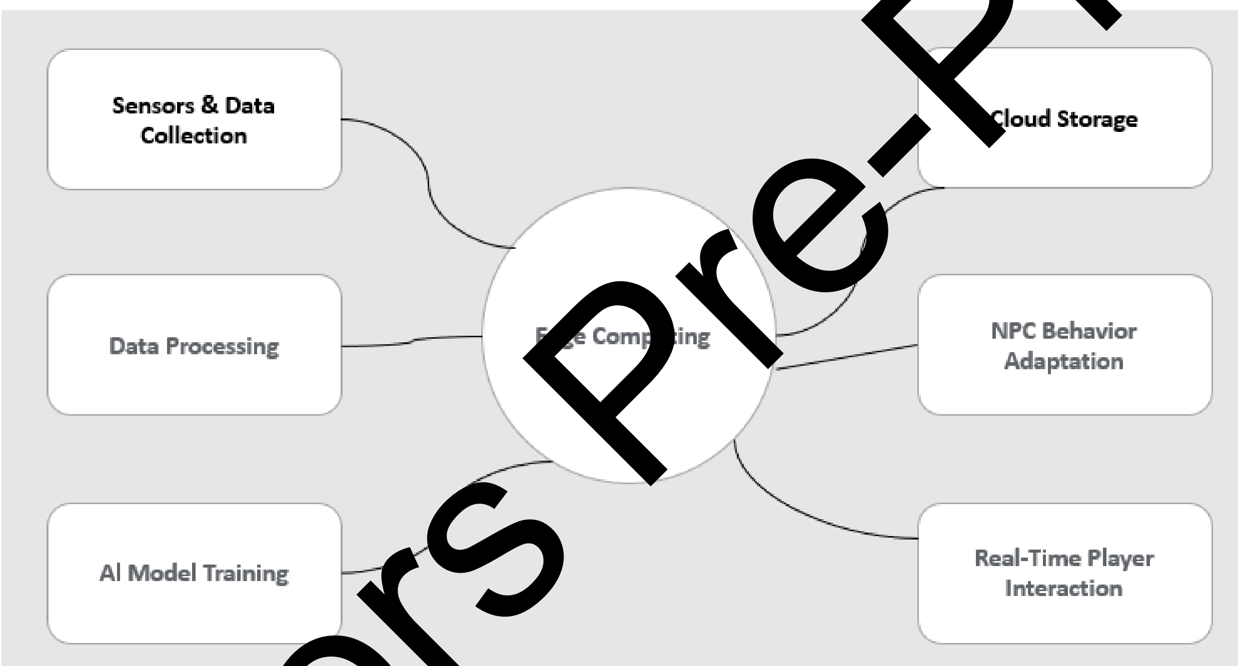


Figure 1. *Framework Architecture Diagram*

Figure 2 shows edge sensor data collection, processing, and use. A flowchart shows how edge computing seamlessly integrates real-time sensor data to improve game NPC behaviour. Top left flowchart has motion detectors, cameras, microphones, and environmental sensors. Sensors record player movements, gestures, commands, and environmental conditions to create a dynamic dataset. Sensors feed centre-edge flowcharts. Edge devices store, process, and send data. Collectors prepare sensor data for analysis. Data processing module NPCs receive real-time insights and commands from ML and RL algorithms. Finally, the Data Transmission module sends system components processed data. The flowchart uses processed data twice. First, the Gaming Environment at the bottom-left of the flowchart processes data to adjust NPC behaviour to player actions and environmental changes. Real-time NPCs improve gameplay. Second in the flowchart is bottom-right Cloud Storage. Long-term data storage includes model updates and training. Solid arrows represent real-time data collection and processing, while dashed arrows represent cloud backup and updating. This detailed flowchart shows how edge computing can improve gaming NPC behaviour with real-time data.

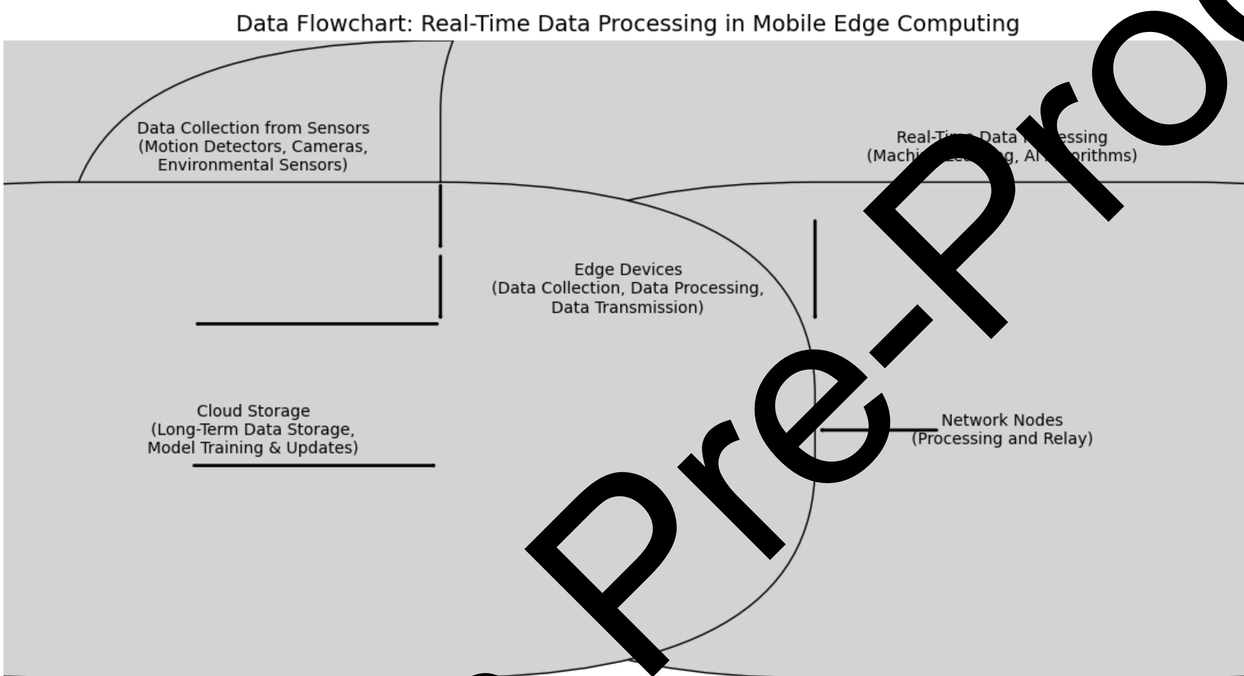


Figure 2. *Data Flowchart*

Figure 3 shows detailed NPC response times for scenarios with different NPC density and environmental complexity. The research analysis section shows NPC behaviour system performance metrics under different conditions using this graph. Millisecond response times are plotted against scenarios to show system performance under different loads and complexities. The system reacts to low, medium, and high NPC density and environmental complexity. According to the graph, low NPC density and environmental complexity cause 50–100 millisecond response times. Under easy conditions, system responds quickly. Medium and high environmental complexity increase computational load and response times from 60 to 120 milliseconds at low NPC density. System strain from NPC interactions increases response times to 80–150 milliseconds in medium NPC density scenarios. System performance is affected by high NPC density and environmental complexity, which have the highest response times, 100 to 180 milliseconds. In low- to medium-density scenarios, the edge-based NPC behaviour system excels, but in high-density, complex conditions, it needs improvement.

In performance metrics research, Figure 4 shows system latency across load conditions and environmental complexity as a line graph. Millisecond system latency against low, medium, and high load is shown. Each load condition is analysed with low, medium, and high environmental complexity. This method shows how NPC count and gaming environment complexity affect

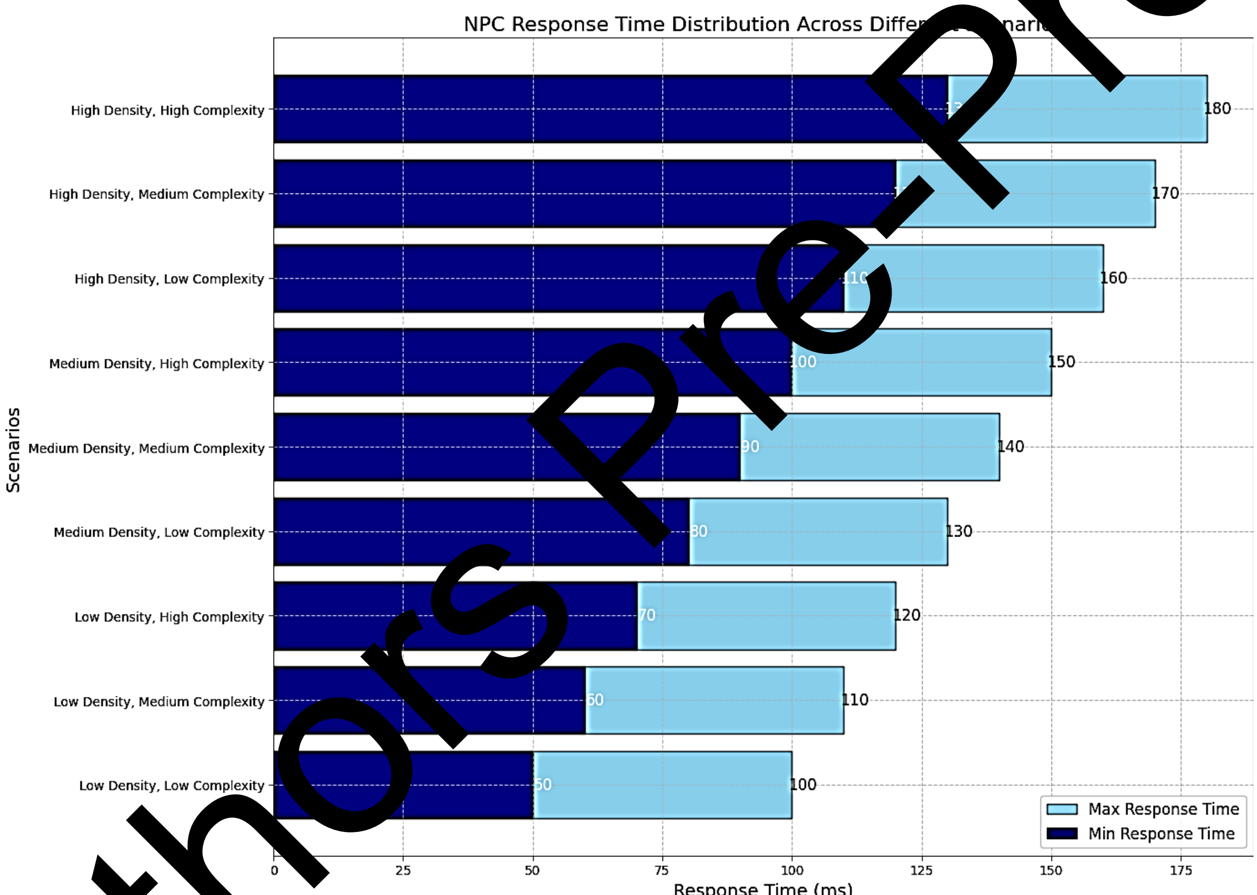


Figure 3. *NPC Response Time Distribution*

latency. In low-load scenarios, all environmental complexity levels have lowest system latency. Low complexity environments have 30 ms latency, medium 40. Low-load, complex environments have 50 ms latency. This system works best under low load and maintains low latency as environmental complexity increases. Edge computing efficiently processes real-time data with few NPCs and interactions due to low latency. Every environmental complexity increases system latency at medium load. For medium load, low complexity environments have 50 milliseconds latency, while medium has 60. Medium-load, high-complexity environments have 70-millisecond latency. System processing and latency increase with NPCs and interactions under medium load. This trend shows the need to optimise system processing for moderate load increases without performance loss. NPC density and environmental complexity affect performance because system latency peaks when loaded. In high-load scenarios, low complexity environments have 70, medium 80, and high 90 millisecond latency. The system can handle high loads, but latency suggests processing and resource management bottlenecks.

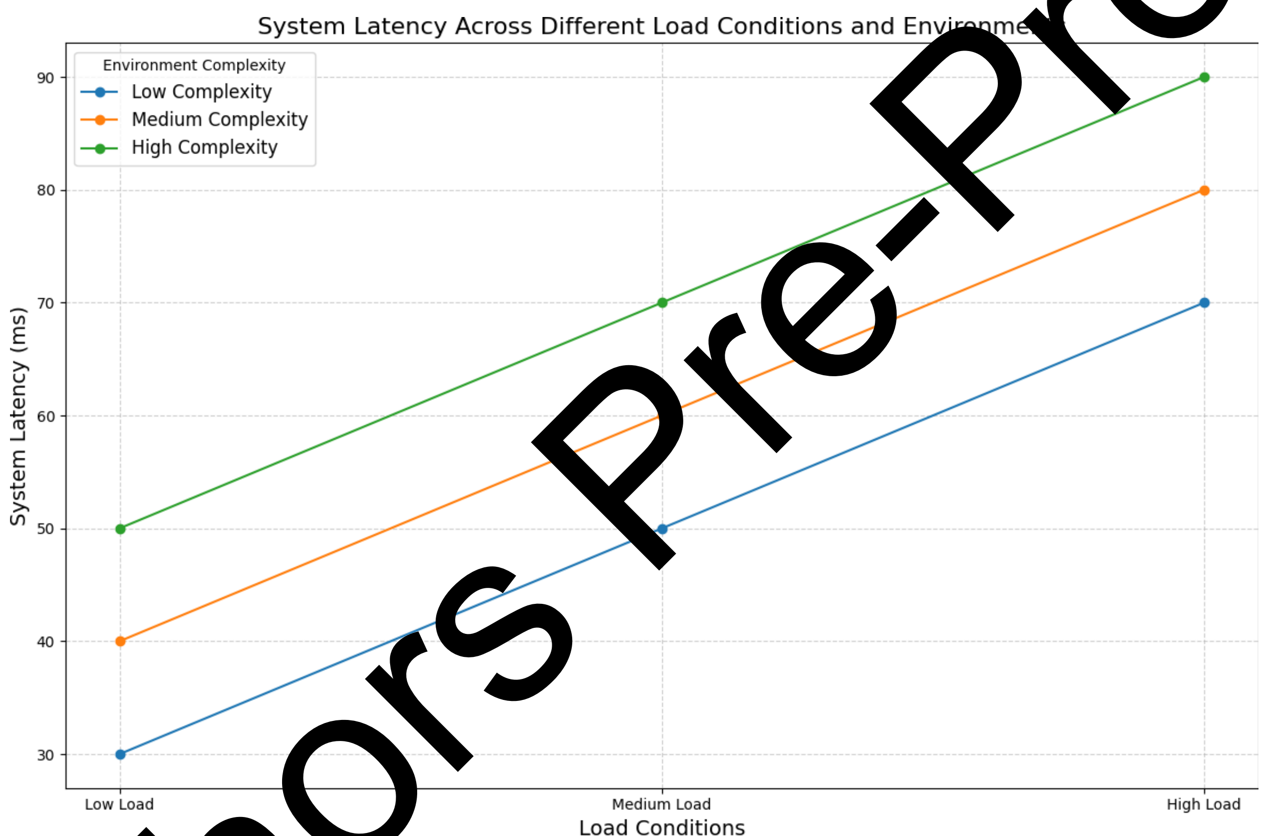


Figure 4. *System Latency Graph*

Player feedback research requires Figure 5, a histogram of player ratings on NPC realism and engagement. Histogram player ratings range from 1 to 5, with 1 being "Very Unrealistic/Unengaging" and 5 being "Very Realistic/Engaging." Players' NPC realism and engagement ratings are shown in the histogram. The histogram shows that most players rated NPC realism 4 or 5, with high concentration. The high ratings for "Somewhat Realistic" (4) and "Very Realistic" (5) indicate that most players found the NPCs lifelike. Perhaps edge computing's real-time data processing improves NPC interactions and gaming immersion. The lower ratings (1, 2, and 3) indicate that while overall perception is positive, NPC realism could be improved to meet player expectations. The histogram has many 4 and 5 ratings like NPC engagement. NPC behaviour was mostly "Somewhat Engaging" (4) or "Very Engaging" (5), indicating player satisfaction with interactivity and dynamics. Player engagement requires contextual NPCs. Although the histogram shows lower ratings, some players may have found NPC interactions boring. This feedback suggests NPC behaviour improvements to boost engagement. This histogram shows players' quantitative and qualitative NPC realism and engagement satisfaction. Edge computing makes realistic and engaging NPCs, as shown by the high ratings. Lower ratings suggest player expectations can be met. These insights and qualitative player feedback help researchers understand NPC behaviour system strengths and weaknesses. Player feedback on NPC realism and research analysis require Figure 5.

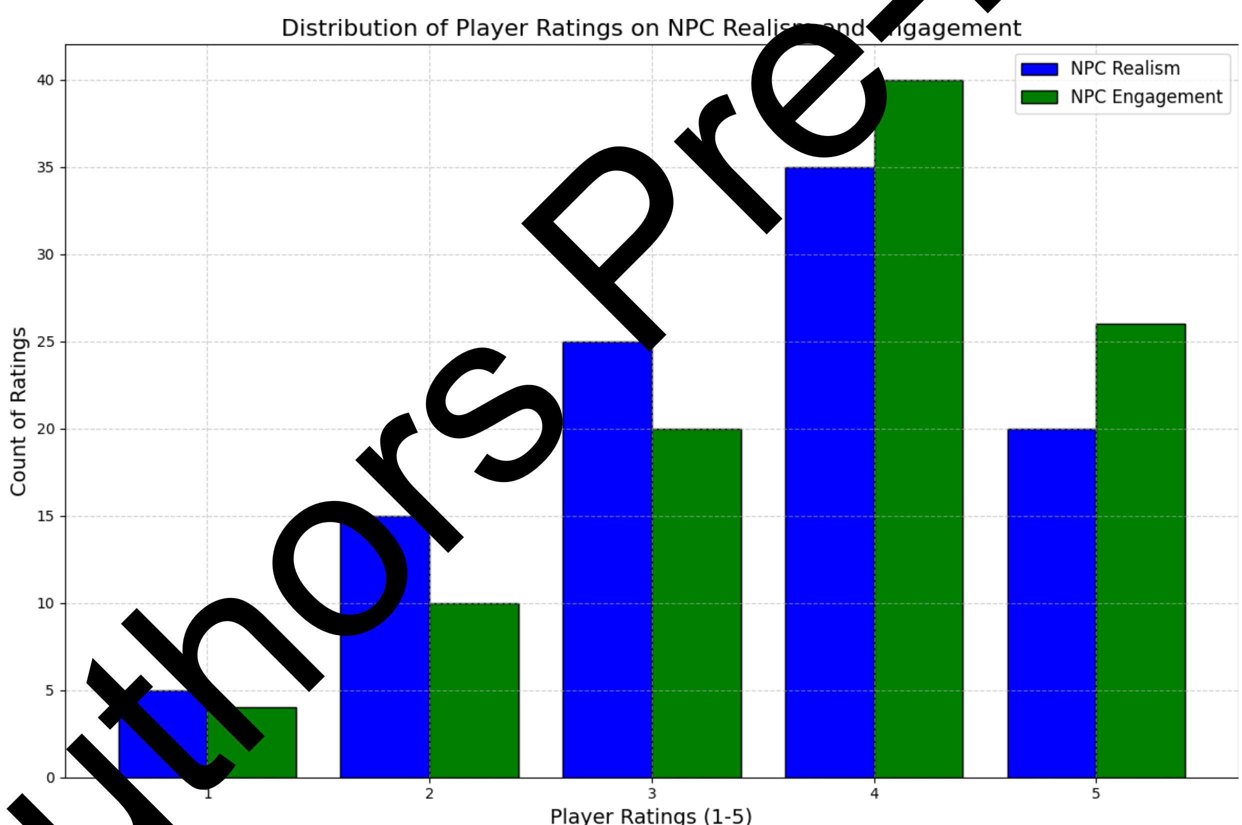


Figure 5. *Player Ratings Distribution*

The study's AI to improve NPC behaviour performance metrics bar chart is in Figure 6. ML, RL, hybrid, rule-based AI, and heuristics are evaluated. Selection and efficacy analysis of algorithms and models require the bar chart. The bar chart shows NPC accuracy, response time, and player satisfaction. A millisecond response time shows NPC agility to player actions. The percentage of NPC actions that respond correctly to stimuli is shown. Player satisfaction ranges from 1 to 5 on the Likert scale. All three bars compare AI techniques' performance. Machine learning (ML) has 90-millisecond response times, 88% accuracy, and 4.1 player satisfaction. A balanced response time and accuracy boost ML player satisfaction. High computational loads caused performance inconsistencies, but ML handled dynamic game environments well, comments said. RL beats ML in response times (80 ms), accuracy (92%), and player satisfaction (4.5). RL's adaptability and interaction learning improve performance. RL's realistic and contextually appropriate behaviour was appealing, but training was computationally intensive. The hybrid ML-RL approach has 70 ms response times, 95% accuracy, and 4.7 player satisfaction. Players interact with responsive, accurate ML and RL NPCs. Implementation complexity and computational power are drawbacks.

Rule-based AI has 100-millisecond response times, 85% accuracy, and 3.8 player satisfaction, worse than ML and RL. Simple rule-based AI has rigid and predictable NPC behaviours that lower player engagement. Traditional heuristics are worst, with 110 ms response times, 80% accuracy, and 3.5 player satisfaction. Complexity and dynamics make modern game NPCs unrealistic and repetitive.

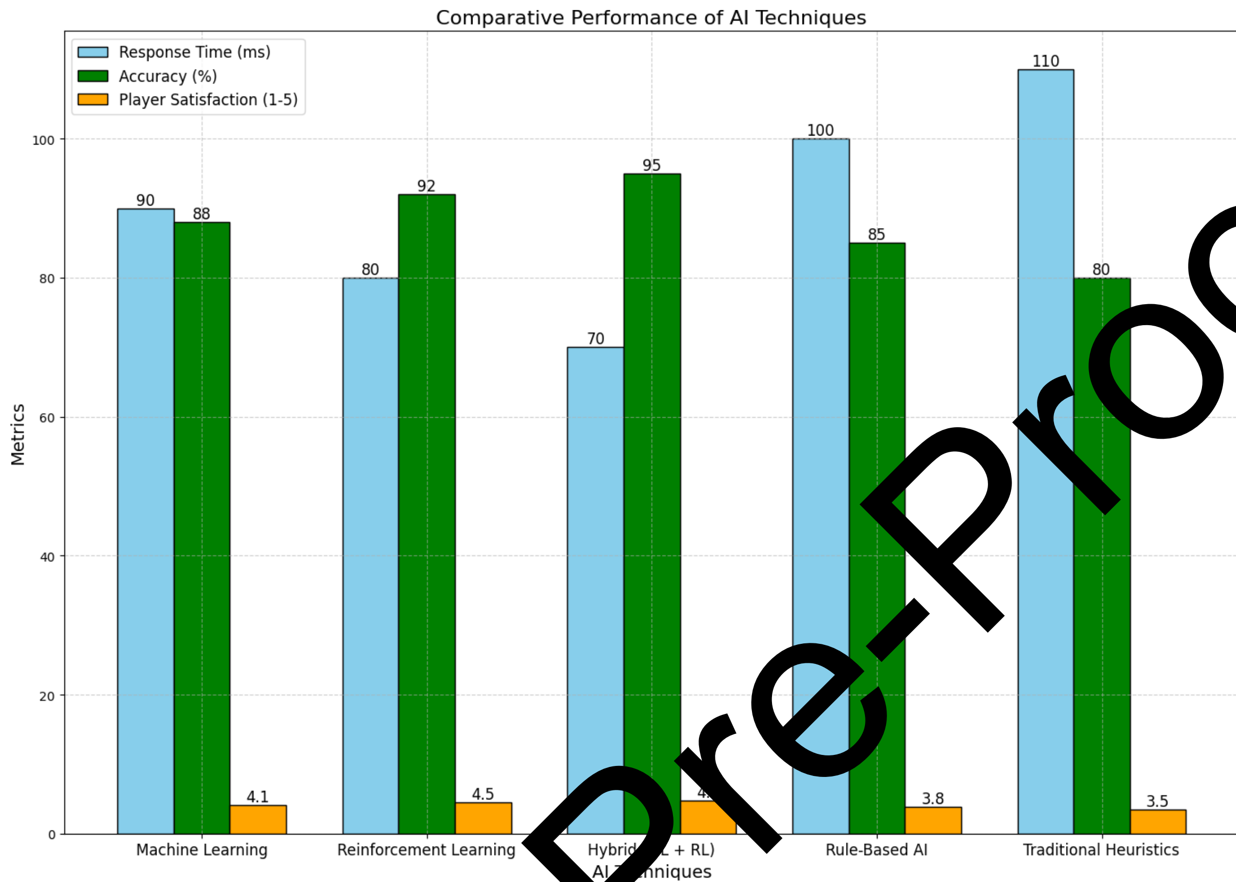


Figure 6. *Comparative Performance of AI Techniques*

Figure 7 shows a line graph of NPC behaviour system scalability metrics as NPCs and interactions increase. This graph shows system scaling and guides research and analysis. Data on 50, 100, 200, 500, and 1000 NPC response times, accuracy, and CPU usage. This detailed analysis shows the system's capabilities and limitations in managing increased demands under different load conditions. The line graph shows 50 NPCs with 60 ms response times, 95% accuracy, and 40% CPU usage. Fewer NPCs improve response times, accuracy, and resource use. Response times reach 70 milliseconds, accuracy drops to 93%, and CPU usage peaks at 50% at 100 NPCs. Systems scaling increases computational and processing load.

With 200 NPCs, response times reach 90 ms and accuracy drops to 90%. Moderate system strain is indicated by 60% CPU usage. As NPCs and interactions increase, response times and accuracy decrease, suggesting the system struggles to handle the load. Response times peak at 120 milliseconds, accuracy drops to 85%, and CPU usage peaks at 75% at 500 NPCs. The system can scale at this level, but performance metrics show strain, requiring optimisation and resource management. At 1000 NPCs, response times reach 150 milliseconds, accuracy drops to 80%, and CPU usage peaks at 90%. The system struggles to respond and accurately under load near its operational limits. As response times and CPU usage increased, accuracy decreased, indicating system architecture bottlenecks. The graph shows that the edge computing framework can handle moderate scalability but needs major improvements for high-density scenarios. The research approach and analysis depend on the system's scalability, shown in Figure 7 [8], [17], [22].

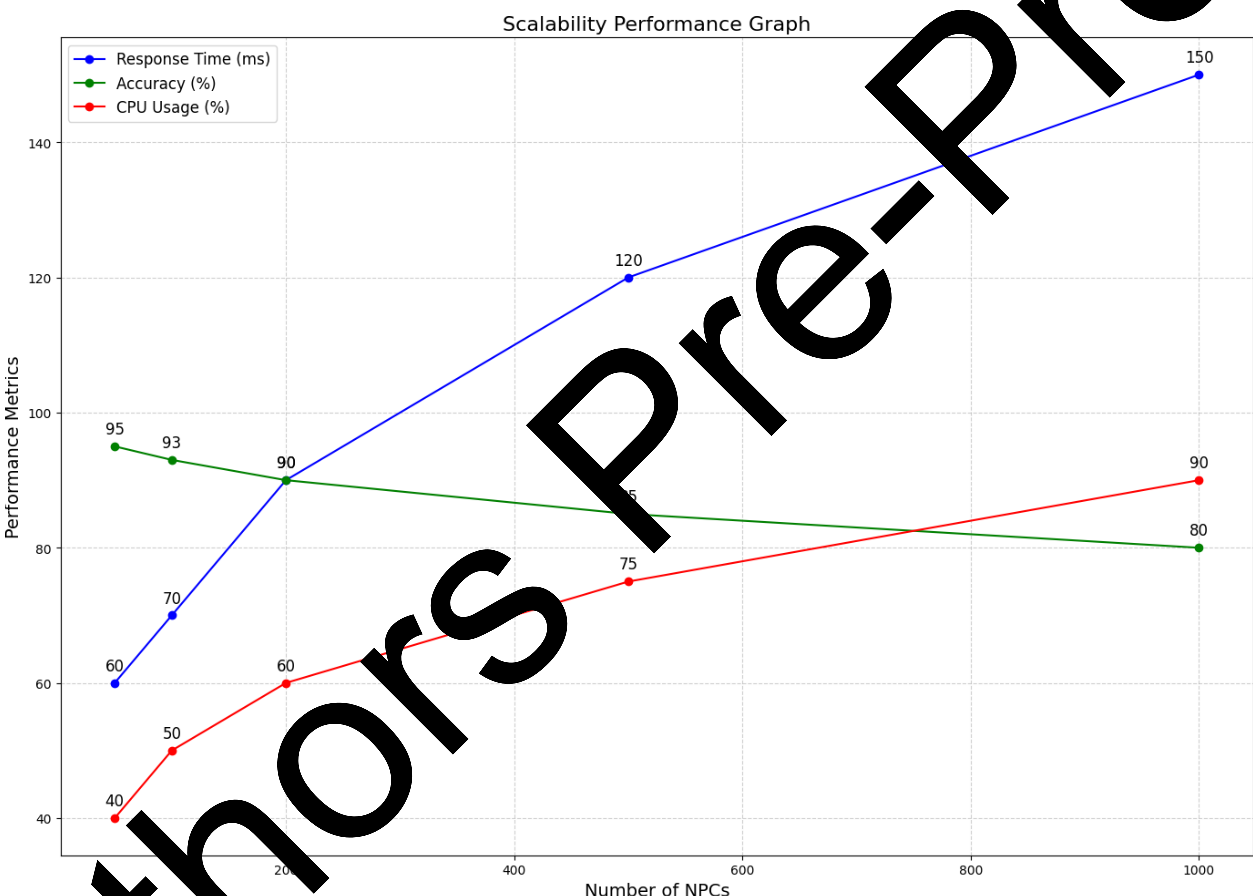


Figure 7. *Scalability Performance Graph*

5. Discussion

This study evaluated a Mobile Edge Computing (MEC) framework for networked real-time data processing and gaming AI system NPC behaviour. Simulations and user studies examined the framework's technical and experiential implications on end-users. For complete datasets, Python simulations varied NPC density and environmental complexity. The datasets tested accuracy, response time, and system latency. The investigation suggests that MEC's more flexible and responsive NPC behaviour could improve real-time, networked application user experience.

Table 3 showed that a comprehensive dataset is needed for system effectiveness and scalability by showing simulation environment parameters like NPC density, interaction frequency, and environmental complexity. The table showed that the system could efficiently manage 10–50 NPCs per 100 square metres and 5–20 interactions per minute. Real-time processing and environmental complexity were shown. Table 4 showed that the NPC behaviour system responded fastest and most accurately in low-density, low-complexity environments. As environmental complexity and NPC density increased, response times and accuracy decreased, requiring optimisation for more demanding conditions. In Table 1, study participants' age, gender, gaming experiences, and preferences were shown to ensure diversity. For complete player feedback on NPC behaviour and engagement, diversity was needed. Our demographic was balanced, with the majority aged 25–34 and one to five years of gaming experience. Due to this representation, many player experiences informed the study's findings. Table 4 shows NPC realism and engagement Likert scales. According to the table, most players gave NPC realism 4.2 and engagement 4.0. In qualitative feedback, players liked lifelike NPC movements and dynamic dialogues but noted

MEC frameworks with advanced AI techniques like Machine Learning (ML) and Reinforcement Learning (RL) improve real-time decision-making but increase computing demands. This can strain network bandwidth in large-scale deployments that balance computational load and real-time processing. Implementing AI-driven operations over a dispersed network is difficult, highlighting the need for network architectural refinement to support such complex capabilities without sacrificing performance. This study found that MEC decreases latency and enhances networked system responsiveness over centralized computing, but more research is needed. For present communication networks, this research should optimize MEC AI algorithms and resource management. MEC's real-time gaming, autonomous systems, healthcare, and industrial automation advancements will shape networked communications. To improve communication network MEC, research should focus on data processing efficiency and MEC integration with 5G and other forthcoming technologies. MEC makes next-generation communication infrastructures more scalable, flexible, and intelligent for real-time applications. This study prepares Mobile Edge Computing to alter real-time data processing and management and boost network and communication efficiency. MEC can revolutionize communication networks by solving challenges and enhancing technology, enabling unparalleled responsiveness, adaptability, and user involvement.

6. Conclusion and Recommendations

Mobile Edge Computing (MEC) improves networked real-time data processing, including game AI NPC behaviour, according to this study. Research suggests that MEC's capacity to process data closer to the source reduces latency, improving NPC responsiveness, contextuality, and user experience. MEC improves immersion and engagement in gaming and other real-time decision-making and interaction applications by letting NPCs change behaviour. This study used MEC for gaming AI, but its effects on networked systems are obvious. Data processing systems must be scalable, efficient, and responsive as network environments become more complicated and data-intensive. Decentralized processing, reduced server load, and faster important interactions are MEC's benefits. Game, smart city, autonomous car, and real-time communication networks require this.The study found various impediments to MEC's networked potential. Complex applications require advanced resource management and optimisation to reduce latency and improve accuracy. Advanced AI methods like Machine Learning (ML) and Reinforcement Learning (RL) can improve NPC adaptability and realism, but they require a lot of processing power. For MEC's increasingly complex networked ecosystems, these AI algorithms should be developed to balance computing complexity and scalability.

Research on Mobile Edge Computing (MEC) integration in networks and communication systems has numerous viable solutions. Edge and cloud computing are combined in hybrid designs. Edge computing's low latency and cloud computing's data processing and long-term storage offer real-time interactions for networked systems that need fast replies. This hybrid method optimizes computational work allocation in fast data processing. Modular AI frameworks for networked gaming are another option. Customizing AI behaviours to application needs can increase MEC-driven systems' scalability and efficacy across use cases via modular frameworks. Another improvement is edge-optimized data processing. Edge device processing is reduced by these algorithms prioritizing vital data streams and automatically filtering superfluous data. Edge devices must work in severe settings for real-time networked systems. This study is limited to aspects related to the confluence of AI and Edge in eight application areas from a global perspective for the purpose of big data analytics at the edge. In this sense, this article focuses only on papers that deal with edge learning in distributed edge-based architecture. It only touches on task and resource management and the different feature challenges of edge in a limited way.

Research Implications

Mobile Edge Computing (MEC) minimizes latency and speeds up responses to external stimuli, making systems more responsive and efficient, especially in real-time data processing, the study indicated. MEC appears to increase networked system performance, making them more dynamic. MEC provides rapid, accurate real-time applications by removing centralized data processing delays. This method affects more than gaming. MEC's real-time data processing and latency reduction boost communication, smart cities, and autonomous systems. By processing and acting on crucial data fast, MEC boosts communication network reliability and efficiency. Real-time financial transactions and emergency response necessitate rapid data processing.

This study highlights networked system scalability. Scalable and effective data processing solutions are needed as communication networks support larger and more complex environments. MEC allows network infrastructure manage more devices and sophisticated interactions without slowing. Big, interactive smart city, industrial automation, and other real-time data processing systems demand scalability. MEC can be applied with deep learning and NLP, according to this study. Networked systems can become smarter, adaptive, and contextualize interactions employing real-time data with MEC and AI. Integration could improve communication networks and other real-time applications by establishing smarter, autonomous systems that learn and react fast.

References

[1] S. Purcell and C. Toomey, "NPCs , Coaches and Development : Top Player Perspectives BAMSA Report 2023," 2023.

[2] Y. Karaca, D. Derias, and G. Sarsar, "AI-Powered Procedural Content Generation: Enhancing NPC Behaviour for an Immersive Gaming Experience," *SSRN Electron. J.*, 2024, doi: 10.2139/ssrn.4663382.

[3] A. B. C. Xyz, "Библиотека по управлению запасами: ABC XYZ анализ, анализ запасов, анализ ассортимента, страховой запас, управление ассортиментом « abc анализ, xyz анализ, авс анализ, анализ запасов, анализ ассортимента, страховой

запас, управление ассортиментом - Язакуп," pp. 1–29, 2021, [Online]. Available: http://zakup.vl.ru/63-biblioteka.html

[4] S. Soliman, "Reinforcement Learning in Gaming," no. October, pp. 1–66, 2022.

[5] D. P. Mtowe and D. M. Kim, "Edge-Computing-Enabled Low-Latency Communication for a Wireless Networked Control System," *Electron.*, vol. 12, no. 14, 2023, doi: 10.3390/electronics12143181.

[6] T. Huynh-The, Q. V. Pham, X. Q. Pham, T. T. Nguyen, Z. Han, and D. S. Kim, "Artificial intelligence for the metaverse: A survey," *Eng. Appl. Artif. Intell.*, vol. 117, pp. 1–24, 2023, doi: 10.1016/j.engappai.2022.105581.

[7] C. Andrade Pacheco, "Believability Assessment and Modelling in Video Games," no. January 2024, 2024.

[8] R. T. Aspects and T. Player, "Cooperative Modular Neural Networks for Artificial Intelligence in Games," no. October, 2023.

[9] J. Hagen, I. Carlos, and P. Santos, "Agent Participation in First Person Shooter Games using Reinforcement Learning and Behaviour Cloning," no. August, 2022, doi: 10.13140/RG.2.2.12534.272[illegible]0.

[10] L. Forneris *et al.*, "Implementing Deep Reinforcement Learning (DRL)-based Driving Styles for Non-Player Vehicles," *Int. J. Serious Games*, vol. 10, no. 4, pp. 153–170, 2023, doi: 10.17083/ijsg.v10i4.638.

[11] C. Davis and J. Bamford, "Ghost In the Grid : Challenges for Reinforcement Learning in Grid World Environments," 2022.

[12] [illegible] Dimitriu, T. V. Michaletzky, V. Remeli, and V. R. Tihanyi, "A Reinforcement Learning Approach to Military Simulations in Command: Modern Operations," *IEEE Access*, vol. 12, no. May, pp. 77501–77513, 2024, doi: 10.1109/ACCESS.2024.3406148.

[13] B. Abbas, "EasyChair Preprint Elevating Realism: Cutting-Edge AI NPCs and Environments Transform Gaming in 'VirtualEra.' Elevating Realism: Cutting-edge AI NPCs and Environments Transform Gaming in 'VirtualEra.,'" 2024.

[14] A. Filipović, “the Role of Artificial Intelligence in Video Game Development,” *Kult. Polisa*, vol. 20, no. 3, pp. 50–67, 2023, doi: 10.51738/kpolisa2023.20.3r.50f.

[15] S. Kanungo, “Edge-to-Cloud Intelligence : Enhancing IoT Devices with Machine Learning and Edge-to-Cloud Intelligence : Enhancing IoT Devices with Machine Learning and Cloud Computing,” no. May, 2024.

[16] M. Khalid *et al.*, “An Assortment of Evolutionary Computation Techniques (AECT) in gaming,” *Neural Comput. Appl.*, vol. 34, no. 11, pp. 8295–8308, 2022, doi: 10.1007/s00521-020-05295-7.

[17] M. Andersson, “How does the performance of NEAT compare to Reinforcement Learning?,” 2022.

[18] N. B. Mangan, “Ethics of Artificial Intelligence : Experimenting with Ethics and Autonomous Vehicle AI Principles in Gameplay,” 2[illegible].

[19] Q. Huang *et al.*, “Position Paper: Agent AI Towards a Holistic Intelligence,” 2024, [Online]. Available: http://arxiv.org/abs/24[illegible].00833

[20] J. Barambones, J. Cano-Benito, I. Sanchez-Rivero, R. Imbert, and F. Richoux, “Multiagent Systems on Virtual Games: A Systematic Mapping Study,” *IEEE Trans. Games*, vol. 15, no. 2, p[illegible]–147, 2023, doi: 10.1109/TG.2022.3214154.

[21] B. C. Schrinner, “MASTER THESIS Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Engineering at the University of Applied Sciences Technikum Wien - Degree Program Biomedical Engineering Sciences Title of paper Title of pa,” 2014.

[22] F. Sh[illegible], “Computational Techniques for Modeling Non-Player Characters in Games,” p. 19[illegible], 2014.

[23] [illegible] Hu *et al.*, “A Survey on Large Language Model-Based Game Agents,” pp. 1–23, 2024, [Online]. Available: http://arxiv.org/abs/2404.02039