THE ASIAN BULLETIN OF BIG DATA MANAGEMENT
Vol.4. Issue4 (2024)
https://doi.org/ 10.62019/abbdm.v4i4.281

ASIAN BULLETIN OF BIG DATA MANAGEMENT
http://abbdm.com/
ISSN (Print): 2959-0795
ISSN (online): 2959-0809

Adaptive Q-Learning for Fair and Dynamic Server Selection in Edge Computing: Addressing Latency Variability in Real-Time Applications

Muhammad Akhtar Nadeem, Muhammad Zunnurain Hussain*, Usama Imran, Muhammad Zulkifl Hasan, Usama Nasir, Hoor Fatima

Chronicle

Article history
Received: Oct 12, 2024
Received in the revised format: Oct 29, 2024
Accepted: Nov 11, 2024
Available online: December 20, 2024

Muhammad Akhtar Nadeem is currently affiliated with FAST Lahore, Pakistan
Email: l201171@lhr.nu.edu.pk

Muhammad Zunnurain Hussain, and Hoor Fatima are currently affiliated with Department of Computer Science, Bahria University Lahore Campus
Email: zunnurain.bulc@bahria.edu.pk
Email: hoorfatima.bulc@bahria.edu.pk

Usama Imran is currently affiliated with Trainee software engineer, Systems limited.
Email: usamaimran67@gmail.com

Muhammad Zulkifl Hasan, and Usama Nasir are currently affiliated with Department of Computer Science, Bahria University Lahore Campus
Email: zulkifl.hasan@ucp.edu.pk

Corresponding Author*

Abstract

The main purpose of edge computing is to provide real-time services, such as cloud gaming and virtual collaboration, which are closer to users thereby reducing latency. However, dynamically pairing users with appropriate edge servers becomes an increasing problem due to a changing and adaptable network environment and different latency requirements from different applications. To address this challenge, we propose a novel Adaptive Q-Learning algorithm for fair server selection while maintaining low variation in latency. The core of our approach involves enhancing the Quadruple Q-Learning model. Our model has been equipped with dynamic action suppression mechanisms that are changed by the most recent network performance indicators. Conventional Q-learning approaches typically make the error of not examining the current load on the nearest server, which can cause some users' resources to saturate and increase their latencies. With normalization of Q-values and a flexible learning rate, our algorithm adjusts better when network latencies change, packets are lost or servers become congested. We strive for more balanced traffic distribution across nodes by achieving equitable user requests spread across the network; thus preventing any one service node from becoming overwhelmed. Through simulations in a cloud gaming context, we demonstrate that our proposed Adaptive Q-Learning method outperforms existing algorithms.Our method however is not only capable of holding strictly to such latency thresholds. Besides, it is also functional in implementing fairness so all users may experience similar latency levels. The article emphasizes the necessity of adaptive and impartial server selection in edge computing environments to make the time-critical applications more user-friendly.

Keywords: Edge Computing, Adaptive Q-Learning, Server Selection, Cloud Gaming, Latency Variance, Fairness

INTRODUCTION

The For edge computing era arrived, it is no question that the demand of latency sensitive applications is the highest it has been. Significant applications presently changing our digital relationships like gaming online, virtual or hybrid augmented reality, and telecommunications natively depend on the ability of the edge infrastructures to meet the low-latency services demand. Edge servers are close to end users, and this eliminates most latency problems that are associated with cloud environment which could therefore result in user experience enhancement. This becomes of even more crucial importance considering the rather demanding services like cloud gaming, where even a small lagging may completely disrupt player performance and satisfaction.

Yet, the advent of 5G and the consequent proliferation of edge servers present a new challenge: successful edge server announcement implies optimal selection of razor-edge servers so balance can be achieved between the speed of service provision and an even distribution of the load. The currently functioning server selection tactics, tailored for cost optimization in cloud infrastructures, seem to be inefficient at the edge computational parts. User-centrism and latency-minimization are crucial in that case.

To address this lacuna, our work posits a quantum leap in server selection strategies through the development of an Adaptive Q-Learning algorithm, underscored by a "Q-Value-Normalized Action-Suppressed Quadruple Q-Learning (QQL)" framework. This novel approach not only adapts to the fluidity of network conditions but ensures a fairer distribution of latency among users, thus democratizing the gaming experience.

Reinforcement Learning (RL) proves particularly effective in dynamic environments due to its adaptability through user-defined rewards. In a prior discussion [9], we introduced AI-assisted hybrid networking for cloud gaming. Expanding on this, we present a tangible solution to the server selection challenge by framing it as an RL problem. Drawing inspiration from literature on RL in networking, we propose RL models aimed at reducing latency variance in user-server matching for edge cloudlets. Our approach incorporates fast Q-learning with bounded variance and high discount factors [11], multi-Q-table Q-learning [12], and local normalization [13]. Evaluation using real data from a cloud gaming application demonstrates that compared to conventional methods that select the server with the lowest latency, our approach not only better satisfies latency requirements but also promotes fairness by minimizing latency discrepancies among users within the same session.

The use of AI and Machine Learning techniques has been apparent in server selection approaches to edge computing, according to recent studies [24]. With the combination of AI and supply chain management in particular, the advantage of performing complex calculations to reducing lag time in virtual AI-enhanced supply chain networking has a potential to impact future business environments [25]. Smart grids have advanced forecasting tools which AI based solution can singlehandedly power and thus effectiveness in these grids suggest a very high need for AI-powered computation load balancing in edge computing settings [26]. Edge systems are able to obtain reasonable logic with machine learning methods that are able to uncover signals in vast volumes of information [27]. Using AI-based MRI techniques to scan business structures and systems will have a great value in the context of decision economics that leaders are focusing to be able to keep the principles of equity in American society rules unabated in availability of data during server proxy [28]. IoT data management architectures target vast usability requirements coming together in modern edge computing for fast executions [29]. Building lake-house architectures in the cloud further implies the value of proper resource management systems for adaptive Q-learning performance when selecting optimal servers [30]. AI enhances fairness in the fields of business and health care but directed largely to predictive analytics which considering edge proxy servers has lots of related translatability [31]. The amalgamation of Generative AI and advanced techniques demonstrate a pathway to handling variance and fairness in resource allocation algorithms [32]. Accurate weather forecasting [33] through machine learning offers lessons in adaptive model that can address latency variability in dynamic environments.

Some bullet points encapsulating our contributions:

- Our approach enhances the standard Q-Learning process by making it responsive to real-time network conditions, allowing it to adapt decision-making based on current performance metrics.
- Network Adaptation: Our algorithm considers the ever-changing nature of network conditions, using them to inform server selection with the aim of reducing latency variability and improving fairness across all users.
- We forge a new path by considering the Standard Deviation of latency in server selection, providing a fairer user experience.
- Our algorithm introduces action suppression to address the challenges of a broad action space in RL, akin to dropout in neural networks.
- The QQL model is utilized, enabling actions to be elected from four distinct Q-learning models based on the highest Q-value.
- Local Min–Max Normalization is incorporated to fairly compare Q-values from different reward functions with varying scales.

Collectively, these innovations coalesce into a robust and scalable AI-driven server selection framework that not just meets but anticipates the requirements of modern edge computing demands. We validate our solution with real-world data, and the results underscore the efficacy of our approach in delivering a more balanced and fair gaming environment relative to existing methods.

LITERATURE REVIEW

As previously noted, in cloud gaming, it's imperative not only to adhere to specific latency thresholds based on the game genre but also to minimize latency variation among players within the same gaming session to ensure fairness. Fairness is crucial because players with lower latencies gain an advantage over those with higher latencies. This advantage stems from the faster reception of game events, enabling quicker decision-making and reaction times. Consequently, players with higher latencies tend to perform worse and may even risk losing the game.

Existing methods for server selection primarily focus on minimizing delay without considering its variance. For instance, Web et al. optimized overall delay for all game players by connecting them to mirrored servers. Farlow and Trahan proposed player-server matching algorithms to maximize system capacity by redistributing players among servers during gameplay to optimize overall delay. Some approaches have addressed cost considerations in cloud gaming by integrating pricing into cloud provider selection [18].

In non-gaming contexts, Hu et al. formulated server selection for interactive video streaming as a geometric Euclidean K-median optimization problem to reduce end-to-end delay. Goel et al. suggested client-assisted content delivery network (CDN) server selection using a client-side domain name system (DNS)-proxy that shares load-balancing functionality with CDNs and selects the CDN with the lowest delay. Qin et al. introduced a model predictive control-based algorithm for routing optimization and server selection in an intelligent SDN-based CDN architecture, aiming to optimize users' response time (delay) and bandwidth. Additionally, lightweight methods have been proposed to determine network topology and select servers for multiparty video conferencing, minimizing the mean end-to-end delay between clients . Finally, Wu et al. combined genetic and simulated annealing algorithms for service selection in mobile edge computing to reduce time delay.

RL Incorporating Variance, Fairness, and Action Suppression Within RL, there exist algorithms tailored to minimize reward variance alongside expected rewards, commonly found in safe RL. Safe RL aims to learn policies maximizing rewards while ensuring system performance, reasonableness, and safety constraints, as surveyed and categorized by Garcia et al. . Two approaches include classic discounted finite and infinite horizons with a safety factor and the integration of external knowledge or risk metric guidance. While effective in their intended domains, safe RL algorithms primarily maximize long-term rewards, potentially overlooking occasional large rewards along the way and failing to avoid rare occurrences of significant negative outcomes. Hence, they may not suit our objective of reducing reward variance at each action step, which may not lead to optimal long-term variance reduction. Our work focuses on designing fair matching algorithms, akin to [34] [35], concentrating on suboptimal matching between two groups to minimize the variance of their distance function. We define fairness as reducing latency variance, leveraging geo-distance between users and edge servers as a latency indicator, as detailed in Section III .
Action space reduction, integral to our RL method, is also explored in current research. In , the action elimination network (AEN) is proposed, employing two neural networks: one approximating the Q-function and the other learning to eliminate actions. This aids in managing large action spaces, such as in NLP based generation of text, by the use of LLMs or transformers which perform actions with high probability. Inspired by this, we have adapted the concept, tailoring it for tabular scenarios while prioritizing fairness in matching problems. Unlike the AEN's neural network approach, we utilize a linear vector to indicate action availability, as demonstrated in Section IV. Our vector manages action availability with options for fixed or learned vector values.

PROBLEM DEFINITION

In simpler terms, consider the network of a gaming platform where three key components exist: players (U), individual edge servers (EN), and a central edge server known as the delegated edge node (DEN). Both ENs and the DEN form an essential part of the infrastructure provided by the gaming service, which could be leased from major cloud providers like Google, Amazon, or Microsoft, or be a proprietary setup such as the one used by Sony PlayStation Now's Gaikai .

The DEN plays a crucial role as it is the first point of contact for players. It assesses and then directs players to an appropriate edge server. While each EN has the capability to support numerous players simultaneously, their resources are not unlimited. One key aspect that the system needs to vigilantly maintain is latency – the time delay between a player and their assigned EN. For instance, in a fast-paced game like Counter-Strike, where every millisecond counts, the maximum one-way delay permitted is 50 ms, establishing a 50 ms 'zone' for each EN to operate within optimally .

Moreover, it's critical that the lag times players experience are as uniform as possible across a game session. This consistency in latency is necessary to guarantee fairness and provide an even playing field for all participants. For example, certain players might be able to choose from multiple ENs, as illustrated in the figure where players are marked in orange.

However, in reality, the situation is much more complex than this, with a potentially vast number of ENs available to choose from. This is evident in massive online games like EVE Online, which held a record-breaking battle with over 6,000 players at once. And when you factor in non-playing viewers – take for instance the League of Legends Mid-Season Invitational in 2018 that attracted millions of concurrent viewers – the scale of these gaming systems becomes immense .

Consequently, a cloud gaming system faces a myriad of choices (the ENs) and limitations (latency caps, ENs' maximum capacities), making it a multi-variable optimization puzzle that is tough to crack. This complexity escalates further when considering the fluid nature of online games, where players can come and go and service providers may add or subtract ENs dynamically. Each game, dependent on its type and speed, may have specific latency needs, which makes the server assignment process even more challenging .

In light of these challenges, a change we propose is the integration of a network monitoring system. This system continuously tracks network conditions, providing real-time data that can significantly inform and refine the server selection process. With the aid of these insights, the DEN can make more accurate decisions, ensuring that latency thresholds are maintained without overloading any single EN . This not only enhances performance and equity for current gaming sessions but also offers the agility to adapt as network dynamics shift, keeping up with the game's pace and demands.

Definition of Latency

Latency can be defined in several ways in a network. It could be the total time it takes for a signal to travel to its destination and back (round trip time), the delay in communication between two points (end-to-end delay), the total number of intermediary steps between the two points (number of hops), or the geographical distance between them. For our discussion, we're focusing on the geographical aspect – the longitude and latitude measurements – for a couple of key reasons. Firstly, in the gaming world, where quick reflexes and instant feedback are crucial, the physical distance to the gaming server is a huge determinant of a player's experience. Secondly, gauging the real-time delay for every player and every potential server (which quickly multiplies considering the number of both) would clog the network and make the process inefficient. By using geographical data, we minimize the computing work needed.

But using geographical distance to define latency brings up the question of fairness in how we connect players to servers. To ensure fairness , we aim for a server-matching system that distributes players in such a way that everyone experiences as similar a latency as possible. The goal is to minimize the difference in these geographic distances across all players.

In the conventional way assigning the same server to the next player by the one who mastered the game is no longer a deal. It also automatically benefits those who are first in line, hence those that join in later have less desirable connections as faster servers reach their limit and reducing available connection for all. We are suggesting RL to be incorporated into the server choice process and thus we are providing a more equitable remedy that isn't restricted to latency issues but also considering all players without discriminating against those that joined later in the process.

Online gaming as a dynamic and competitive space is a dense market in which the quality service is one of the necessary components nowadays. It is here that the main reason a network monitor becomes necessary. An eleven-dot monitor rates the network's health and efficiency in real-time to ensure that the toughly-defined latency requirements, essential for a steady gaming experience, are never left behind. Therefore, it establishes a controlled monitoring system for network operations that have Rayyan latency as the first priority, followed by bandwidth availability, and then overall throughput. This, in turn, is done through suggestion of the network monitoring which helps in the detection and prevention of disruptions that might come up in the process of game play and benefit the players.

Dashboards with live data which is used to measure network status are of critical importance in monitoring edge computing resources. The system converts this information to decide which server it uses and how to distribute the load. It is able to realize when the server of a node devoted at the edge is drawing close to the point of its functioning limit, and re-directs newly registered user sessions to the web servers of the alternative nodes in an effort to equalize the traffic load within the network. Conducting a precautionary task, the network monitor allows to avoid congestions and delays of the server capacity. As a result, the network monitor creates a new standard of fair play in the gaming environment that's adaptable to the constant alterations in the network characteristics. The aim is to maintain an equilibrium between user requests and servers' capacities.Our task is to address a specific operational challenge faced by Swarmio Media in their gaming infrastructure. The gaming sessions on Swarmio's platform initiate with players logging into the Swarmio portal. Here, players are grouped into teams either randomly, using an algorithm, or based on pre-established agreements. Alternatively, there are scenarios where players compete individually without forming teams.

There are two primary scenarios to consider:

- The first involves cases where an entire team's players are locally close enough to be connected to the same edge server without breaching the server's latency limit. This setup is beneficial as it enables quick and efficient communication between team members, with game states being rapidly synchronized across the edge servers, thanks to Swarmio's highly optimized low-latency platform.
- The second scenario arises when players are geographically spread out and must be assigned to different edge servers to maintain acceptable latency, regardless of their team affiliations or in individual player modes. Even in such cases, the game state is continuously synced across the edge servers, leveraging Swarmio's platform capabilities .

Additionally, we consider certain constraints in our approach:

- Our focus remains singularly on one gaming session at a time as our algorithm is designed to promote fairness among participants of the same session. This concept of fairness does not extend to players across different gaming sessions as they are not directly interacting with one another.
- The structure of the games offered by Swarmio, like Counter-Strike and League of Legends, dictates that players be present before the session kicks off. The system does not permit new players to join mid-session, particularly in tournament-style settings.
- Similarly, the system also prohibits players from switching between sessions once gameplay is underway.

- Furthermore, each server is exclusively committed to a specific gaming session, and there's no cross-utilization of servers for multiple sessions simultaneously. These decisions regarding server assignments are made in advance to ensure optimized performance for each gaming session.

Initialize $Q(s,v)$ for all $s \in S$, $v \in A(s)$, arbitrarily

Define $A_{available}$ for all possible $v \in A(s)$

Set Q(terminal state, $\cdot$)=0

Initialize NetworkMonitor

Define AdaptiveFunction to adjust learning rate based on network conditions

for each episode do

    Initialize s

    Update learning rate α using AdaptiveFunction based on NetworkMonitor

    repeat

        Choose highest v for s using policy derived from Q (e.g., ε-greedy)

        while $v \notin A_{available}$ do

            Choose next highest v using the same policy

        Take action v, observe r, s' (according to a reward function)

        if adverse network conditions detected by NetworkMonitor then

            Adjust $A_{available}$ based on current network conditions

            if limit for action suppression is reached then

                Remove v from $A_{available}$

        $Q(s,v) \leftarrow Q(s,v)+\alpha[r+\gamma\cdot m\ v\ x\ Q(s',v')-Q(s,v)]$

        $s \leftarrow s'$.

    until s is terminal and $A_{available}$ isn't empty

NetworkMonitor updates network conditions for the next episode.

Proposed QNetowrk System

In Section III, we talked about a complex problem that's always changing. To tackle it, we're turning to a type of artificial intelligence called Reinforcement Learning (RL), specifically a method called Q-learning. We've named our system QNetwork. Here's the deal: We're trying to figure out the best way to choose servers in a network. Normally, a basic Q-learning model would just pick the closest server, even if it's already too busy. So, we're getting creative. We're adding some new tricks to Q-learning to make it work better for our problem. And hey, these tricks aren't just for picking servers—they could be handy for solving other matching problems too. In our setup, we're treating users joining the network as the starting point (we call this a "state" in RL lingo), and the available servers as the options for action. Throughout this discussion, we might use "actions" to mean "picking servers" and "states" to mean "users," depending on what makes sense. The way things change from one state to another mainly depends on how many users want to join the network. Think of it like this: as soon as one user is taken care of, the next one in line becomes the focus. Now, let's dive into a cool technique we're using called "RL action suppression," and then we'll talk about our specific Q-learning models.

Suppression of Action

Why We're Doing This

Imagine a bustling buffet with a diverse array of dishes, each one vying for your attention. Now, picture your disappointment when your favourite dish suddenly runs out because the kitchen can only handle so much demand. In the realm of edge computing, our servers face similar constraints—they have finite capacities that can be quickly maxed out by user demand. When this occurs, it's imperative that our system doesn't persist in trying to assign tasks to these overloaded servers, as it could lead to performance degradation and unhappy users. This is where the concept of "action suppression" comes into play. By recognizing when servers are unavailable due to reaching their capacity limits , we can temporarily remove them from consideration, ensuring that our system operates efficiently and effectively. Inspired by similar techniques used in related fields, we're customizing this approach to suit our specific needs and challenges.

How We're Making It Work

We will nicely explore the means of action suppression now. In an event that the server reaches maximum capacity, we make updates to the network indicating that the server is down for the next server selection round. That is like replacing one of the specialities from the buffet—it's not on the menu until the chef restocks his ingredients. Nevertheless, not relying on random non-targeting but on a flexible list of available servers, we advance the system. That's why, at allotment time, our system will flawlessly skip the unavailable web servers and jump ahead to the next option that fits best. This strategy makes our system with the ability to respond in parallel to the fluctuating server capacities and to give the best performance and the users' satisfaction .

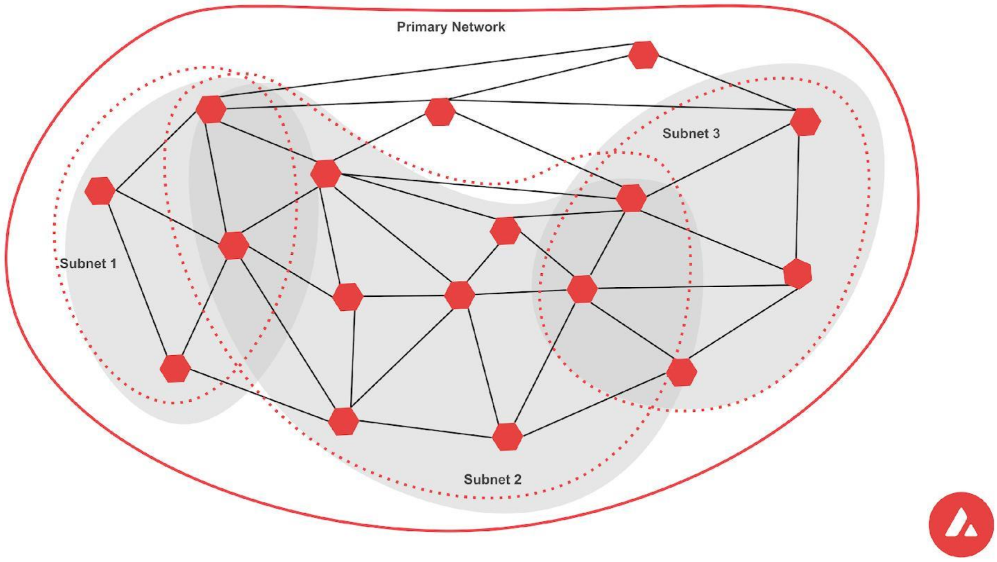

Figure 1.
A typical gaming subnetwork illustration

What It Means for Our System

Shifting from the conventional cloud paradigm to the edge computing environment requires an adaptability approach. In the same way as the customers who choose the same dish over and over might get bored, our system will need to look for ways to stop it from taking only the same pattern of actions which could impact the system efficiency. We want to make room for flexibility and innovation. Therefore, an element of randomness will ensure our decisions vary. Through a short-term process of problem solving by overloaded servers' reduction – i.e. second most popular option – we decorate our system to take a detour to another path. This approach is high in balance between policy measure implementation of known strategies and new opportunity exploration with the aim of keeping the system adaptable and responsive enough to changing dynamics.

Keeping It in Check

Let's conceive of the action-curbing metaphor represent it as a safety device which not only forestalls too much habit formation in the system but also reminds the mindfulness of the body and the ecological surroundings as well. Moreover, the experienced phenomenon behaves like the feeling of hunger craving made by reordering the buffet setting. It places into the course of unexpected and forces to the discovery. A measure of prevention makes sure the condition, where the system becomes too-specialized which affects the system adaptability and make them non-working for the new problems. The system will be given an opportunity to continuously reshuffle every once in a while thereby, it will often maintain an active and explorative search for solutions across varying environments and at the same time the system will have a consistent capacity to confidently explore the unknown in opportunities .

Looking Ahead

The knowledge acquired by the suppression of an action exceeds, by far, just the optimizations of the system per se. With the use of this tool we are able to discover dropped actions and time they are sent both of which provides us with trustworthy insights on users' behavior. Hence, we will be able to direct our efforts towards resource demand, forward planning of servers deployment and their capacities, etc. Furthermore, we will refine our behavior control system to continuously work over a myriad of cases using machine learning techniques which learn dynamically with the changed environment. This cyclical process of watching, evaluating, and evolving the system largely contributes to the system staying at the pinnacle in the edge computing innovation field— and this is achieved by constant improvements of the system in order to make it resisting, efficient and responsive to the needs of users and applications .

As a consequence, action set overriding shall remain a significant point of our approach for edge computing realizations with dynamic server selection. Servers are managed by us with high intelligence and can make immediate changes to reaction a situation. Therefore, our system reaches its highest potential while there is an optimum performance level and client satisfaction. Going forward this year we are making a promise to refine our D-Foundation and using the progress we make to increase the coverage of edge computing. We aim to do this by learning through ongoing experiments and real world deployments of edge computing and so that we can work hard on its improvement. Through action suppression being the guiding principle, we contain ourselves to realize a future where an edge computing system not only take reacting measures, but also act in a preventative way, continuously adjusting and excelling in the midst of the instability and constant change.

Proposed Q-Learning Models

Why We're Doing This: Beyond serving as a simple objective, it is equally important to respect fairness and make it an intrinsic part of each decision we take. It is our forte to divert the common perceptions about the ultimate fairness of the deal between the server and the user while they are connected via dynamic networks, especially for gamers. We have a vision of fairness which is not just matching equivalence but it is also how the connection between a user and a server is made possible in a manner many of the players world wide are satisfied. So as to work on this great task, we resort to the application of q-learning which is a worthwhile instrument of development that allows our system to become smarter and modify itself according to past things. But here's the catch: to get Q-learning to be fair, we need to formulate the reward functions that go with a fair society to be used.

Let's Dive Into the Models: Here's a deep dive into the different reward functions we've meticulously crafted:

- Model 1: This method disposes of the challenge of having a buffer between users and servers by shortening the distance. We have come up with a payout function that contrary to the miles immensely, the last miles shall get a small negative reward. We built our system in a quite simple way so as to decrease the number of times users had to quit the game just because of the slow server. An indirect result is that fairness is enhanced.

$$\sqrt{Ç - 2 + (e\text{lat} - \mu\text{lat})2}\ (1)\ d = (e\text{long}\ \mu\text{long})$$

- Model 2 ($Re\,g'\,rd = -1 \times std\ \ w'(D)$): However in this new model we will move our focus to decrease the global average distance. To this end, we formulate the reward function that gives the opposite sign to the standard deviation of the current values. Through imposing penalties for departures from the average values, our model teaches the system to bring users and servers together by parameters of distance as close to the average value as possible, which ensures fairness for everyone.

- Model 3 (de-$Re\,g'\,rd = -1 \times \Delta std\ \ w'(D)$): Here, our purpose is to minimize the variability in the standard deviation of distances. We implemented an action function that penalizes any action causing the value of standard deviation to swell. This promotes our system to settle on decisions that help with narrowing distance ranges, thus making the user to server allocation distribution more balanced and fair.

- Model 4 ($Re\,g'\,rd = -1 \times |\Delta stdw'(D)|$): This one is similar to the Model three, but with a kick. Here, we consider the absolute value of the standard deviation change, regardless of it happens to be positive or negative. The goal remains unchanged: avoidance of unpredictable distances distribution so as to maintain users-server fairness assurance.

- Model 5 (QQL): This model represents the final page of our book, effectively integrating the knowledge of the former four models that are intended to bring the "fairness" factor several steps further. Eventually our system collects experiences of all the models into one homogeneous system that understands fairness and thus is able to make user-server allocations fairly adn raises level of satisfaction of every player.

- Model 6 (Normalized QQL): Considering the magnitude that determines the variety of our reward function values, we use min-max normalization to make the comparison fair. We bring in the best approaches from deep learning with the standardization of values from within the range 0 and 1 which assists our Q-learning process, and furthermore supported us to tackle fairness issues.

What This Means for Our System

Through the introduction of these intricate reward functions into our Q-Learning models, the system gets the necessary gear to navigate the fair user-server allocations. Whether it is minimizing distances, stabilizing standard deviations or putting all models together to synthesize insights, our system is created to take informed decisions focused on fairness and hence enhance gaming experience for all players.

Keeping It Practical

Let us visualize our Q-learning models as a multifaceted toolbox where each model covers a specific fairness aspect. Through the combination of different game models, the system we will create will be flexible and fast reacting, though relevant user demands are changing and a fair and entertaining gaming environment is maintained.

Looking Ahead

As we continue to refine and optimize our Q-learning models, we're committed to pushing the boundaries of fairness in edge computing. By embracing innovation and leveraging cutting-edge techniques, we're laying the groundwork for a future. The future where fairness is not just an aspiration, but a cornerstone of every interaction in the gaming ecosystem.

$$X_{\text{norm } j} = \frac{X^{j} - X_{\min}}{X_{\max} - X_{\min}} \qquad \forall \in \text{ rows}, \ \forall j \in \text{ columns}$$

Q-Table Scalability

Q-learning is about scalability, which is a major concern. Our Q-table grows in complexity as the number of users and servers in our system increases; the Q-table being the big database where all learning happens. However, if we maintain a fixed-size Q-table, it won't manage the growing load. Therefore, we needed to come up with an answer—something that would keep our Q-table efficient and functional irrespective of how large our system became.

To deal with scalability upfront, we devised a smart scheme: approximating function. This handy gadget helps us assign new states (e.g., users) and new actions (e.g., servers) into existing Q-tables. Let's find out: when a new state or action is encountered, what this approximation function does is look at those K-nearest neighbors from within our current Q-table set of entries , takes their averages on the previous knowledge (Q-values) and adds another entry according to that average. It's like taking two near matches together and making an entry in our Q table based on their joint knowledge blending thereby getting better results for closer approximate values.

But we didn't stop there. So as to rapidly scale our efforts, we also retooled the structure of our Q-table. Instead of the old-fashioned fixed-size table, we opted for a dynamic hashmap which is just a more sophisticated term for lookup tables that are super fast and flexible. In case of hashmaps, entries associated with a particular user or server can be found, added or removed in no time at all. It's like maintaining an efficient filing system for our Q-table that enables us to handle the continuous surge of data .

We are making our system future-proof against growth-related challenges through these scalability enhancements. Our Q-learning algorithms can adjust themselves continuously as they learn while keeping up even if our number of users and servers becomes humongous. This simply means that our system must always be strong enough to endure amid changing requirements and growing needs while still delivering high performance.

As our system continues to evolve, we will keep refining and optimizing our mechanisms for scalability too. From fine-tuning an approximation function to improving dynamic hashmaps; we remain leaders in Q-learning scalability. We are therefore building on this platform by encouraging invention that surpasses what is possible today so that tomorrow's systems will never have a limitation but strength based on scalability capabilities .

Network Manager

Our Network Manager is the keyring that keeps our system architecture together, smoothly ordering users and servers as an art of rhythm. Frankly, the role of the network operations coordinator is in the essence of being the central nervous system that coordinates and manages all other aspects of this network.

The goal of the Network Manager is to use it to make a network as efficient as possible and find the best solutions. A dynamic environment where the demands of users change and, at the same time, new server capacities evolve, resource efficiency takes top priority. Network Manager faced the task effectively of dynamical allocation of users to servers by use of the dane that was availabale to him in real time and demand patterns. It does this by leveling the load such as workloads of the network among servers, and this improves the performance of the network and prevents its impediments, thus making the user experience better.

Notably, a Network Manager is the top of the list of ways to offer sure that every user is allocated their fair share of the server resources. In multiplayer gaming scenarios, where a player has only a short millisecond of a chance to succeed, fairness is no longer a luxury; it is essential. Through the use of intelligent algorithms, sophisticated heuristics and heuristics, the Network Manager seeks to match users to servers which contain the smallest amount of latency and the greatest gameplay quality. Either by cutting user-host distances to the servers or stabilizing standard deviations, the Network Manager tirelessly acts to implement fairness and give all players a level playing field to play on.

In addition to that, the Network Manager is a basic function that is used to monitor the system and optimize as well. Through recurrent watching over network performance indices, like latency, throughput, and packet loss, it can spot problems as they develop and step in to fix them before they escalate. As a result of using technologies like load balancing and congestion control in a proactive approach, the manager of the Network makes sure that the network operates at its maximum performance, delivering quick and consistent gaming experiences to subscribers.

Performance Evaluation

Data Collection

We gathered our data from the CGCSDD dataset, a treasure trove of information derived from a real cloud gaming tournament organized by Swarmio Inc. In this tournament, a total of 181 players engaged, each connecting to one of nine different servers. During gameplay, we scheduled a program each of gaming servers to capture crucial metrics like FPS, input, output, and location coordinates (longitude and latitude) .

For our demonstration, we opted to focus on the 153 players located in North America, where the largest player base resides. Placing our servers in the same region as the players not only reflects real-world conditions but also enables us to better evaluate the impact of fair selection. By concentrating on a single region, we can more accurately gauge how our algorithm performs given the limited number of users .

Next, we emulated servers within our radius nearest to the selected North region of American players. With 153 players in North America, we matched them with an equal number of simulated servers. The distribution of our collected data and the simulated edge server nodes is visualized in Fig. 4, providing insights into the geographic spread of our player-server pairs.

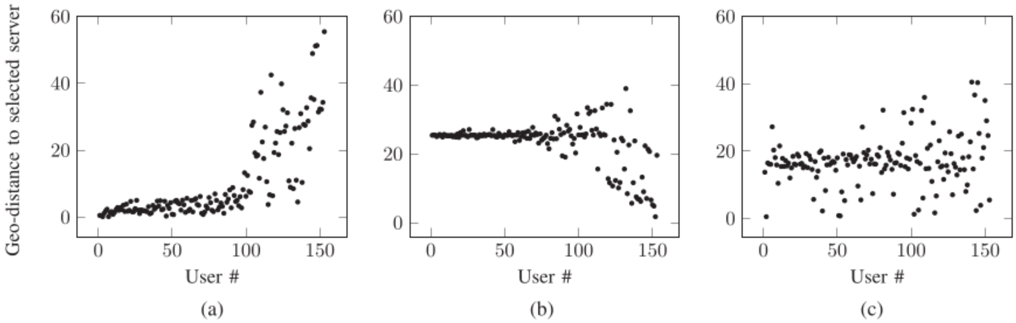

Figure 2.
Average Distance from Server

Following data collection and server simulation, we embarked on a series of experiments, both for individual gaming sessions and grouped sessions. We submit these experiments to the furnace where we try the code against different conditions that would allow us to understand how the algorithm will work as well as the areas that could be improved.

First Session Experiment

In game of our single session experiment, which stands for case 2 of Section III-B, we were challenged with massive virtual-world gaming scenario as soon as EVE Online. In front of us was the formidable problem of finding a single server from a lineup of 153 players playing in North America. The magnitude of this task, furthermore, can be seen in the immense number of possible pairings (factorial of 153 or 10 to the power of 269, twenty doubly multiplied).

To tackle this Herculean task, we fine-tuned our reinforcement learning (RL) hyperparameters through meticulous optimization. Drawing insights from similar RL problems, we subjected these hyperparameters to a rigorous grid search, ultimately settling on values that proved effective across all our models and experiments. These included a learning rate of 0.1, a reward discount factor of 0.6, an exploration factor of 0.1, and a training duration spanning 10,000 epochs.

To benchmark the performance of our models against industry standards, we formulated three anchor methods and a conventional heuristic method, each tailored to address specific aspects of latency optimization:

- **Anchor 1:** The most prevalent method in practice, prioritized matching users with the closest available server based on geographical distance and remaining capacity.
- **Anchor 2:** Adopted a similar approach to Anchor 1 but opted for the second closest server, potentially reserving the best servers for later users.

- **Anchor 3:** Employed a unique strategy, assigning the first half of users to servers within the 50th percentile of distance and the remaining half to the closest servers.

- **Conventional heuristic method:** Aimed to directly minimize latency variance. It accomplished this by first determining the latency range between a user node and any server, then selecting servers with latencies closest to the average of the range for each subsequent user node .

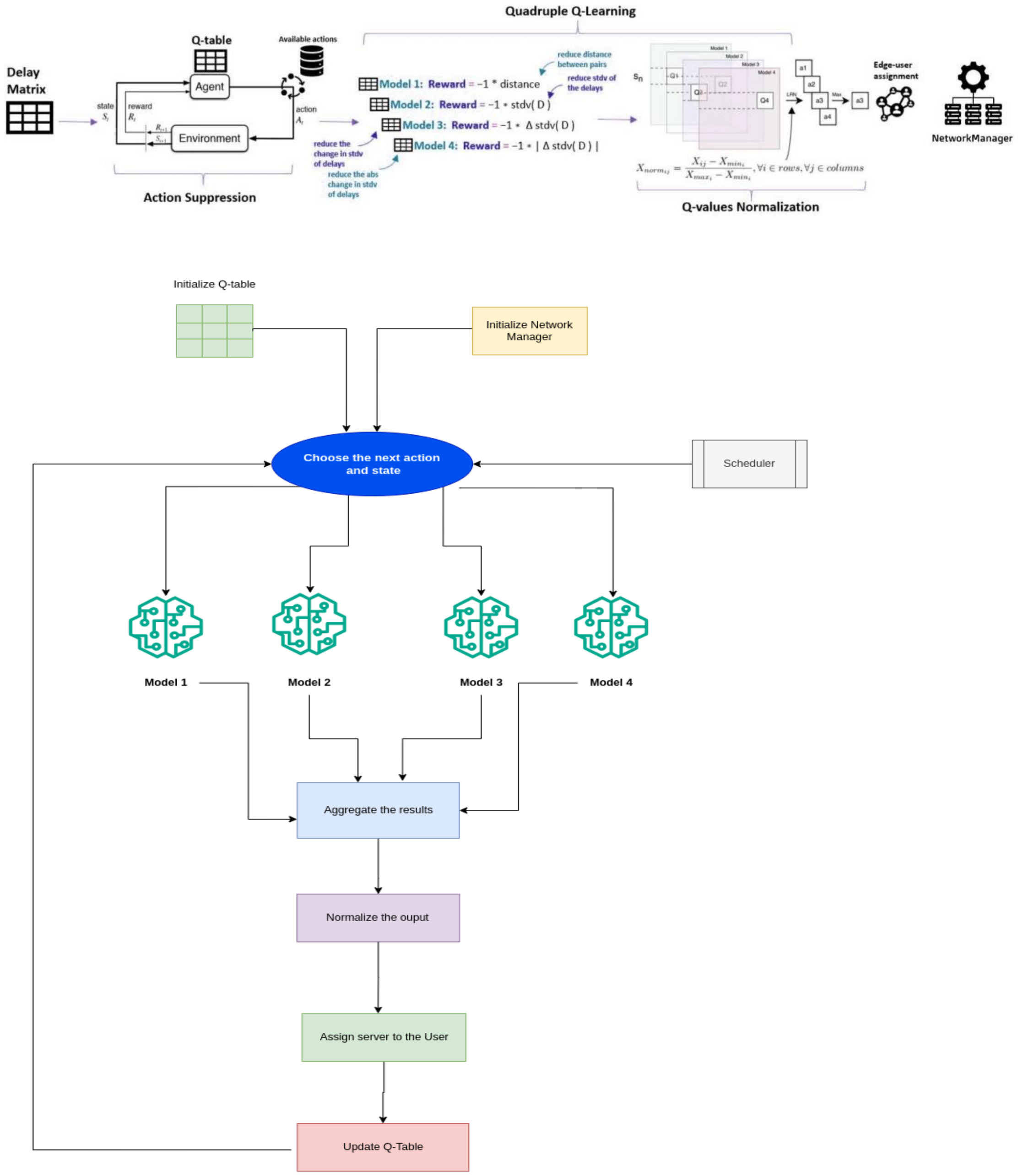

Figure3.
Flowchart diagram of our system

RESULTS

The results from our experiments, as displayed in Table I, provide valuable insights into the performance of various matching methods. Among the Anchor methods, Anchor 2 and Anchor 3 exhibit a slight reduction in deviation while changing in median, with Anchor 2 demonstrating the most favourable variance outcome. Notably, Fig. 3(a) vividly illustrates the shortcomings of existing methods, particularly Anchor 1, emphasizing the need for more efficient matching algorithms.

In contrast, the conventional algorithm's efficacy is contingent upon the initial node selection, dictating the value of dconv. Conducted across 153 iterations with diverse starting nodes, our analysis yielded results categorized based on three distinct scenarios: lowest, median, and highest cone.

Of particular interest is the performance of our proposed models, with Model 6 emerging as the standout performer by exhibiting the least variance, surpassing Anchor 2 by a substantial 35%. Notably, Model 6 not only outperforms the conventional method in terms of variance but also delivers significantly improved average latency.

However, it's important to observe that the output and delay of most of the anchor methods outpaces as similar as of Network models, though this disparity is inconsequential given that any latency below the game's threshold ensures a excellent quality of gaming experience. What truly matters is ensuring a consistent gaming experience for all players, minimizing latency variation among them. As depicted in Fig. 3, while Anchor methods struggle to achieve this goal, QNetwork consistently meets the latency threshold, guaranteeing a fairer gaming experience for all players.

Table 1.

Results of experiment of single session

Method	avg	stdv
1st Anchor	10.63	12.67
2nd Anchor	10.78	12.60
3rd Anchor	13.13	12.62
Low Convolution	**17.89**	9.30
Median Convolution	23.38	8.92
Highest Convolution	27.20	8.75
1st Model	16.76	14.33
2nd Model	19.22	11.95
3rd Model	17.19	10.92
4th Model	19.35	9.83
5th Model	18.94	8.67
6th Model	**17.40**	8.22

Moreover, the effectiveness of Model 6, an aggregation of four distinct models, is underscored by its consistent utilization of Model 3, which was predominantly employed throughout the 153 epochs. This reinforces the efficacy of Model 3, with the combined application of multiple models contributing to enhanced overall results.

Role of Network Manager

The Network Monitor continues to be crucial in the effort of implementing the analytics effectively. Just as it is a central planner of network resources, it dictates how users access servers, and taking into account that it requires maximum efficiency and equality. The Network Manager has the power to intelligibly make the best choice out of many factors such as latency, server capacity, and network traffic according to the current situation at any given moment. Through dynamically changing the server allocation depending on the changing events, the Network Manager gets access to optimized resource use and the reduction of latency between players, and thus improves the quality of gaming for everyone.

Also among his duties, the Network Manager is a central node for data collection and analysis, letting him access crucial information about user latency, server performance, and network conditions. With reference to that data, it keeps on refining the decision-making algorithms. This process goes on while the network dynamics become more sophisticated and demands of users change. Through trial and error learning and optimization, the Network Manager approaches the ultimate aim of managing the dynamic changes by eliminating the latency while tackling the efficiency issues. Through the utilization of the machine learning and predictive analytics the system generates a sound platform to deliver the finest gaming experience that is free of innervations, this in turn improves the satisfaction and loyalty of the customers.

Finally, the network manager is the central glue that binds all network components in the infrastructure and controls complicated interactions between users and the server's performances ensuring their best performance and fairness. By means of firsthand resource management, plus its adaptive in-game learning characteristics, entities are equipped to hold the forefront on top of the increasingly complicated gaming environment. Organizations that embody the Network Manager into their network design as a critical component will be able to unlock new levels of effectiveness, reliability, and engagement from their players in their gaming platforms.

CONCLUSION

On this research journey, we've ventured into the tiniest details of fine-tuning network resources predominantly in the dynamic and challenging world of online gaming. With the birth of QNetwork, an innovative approach designed to address the challenges of quick multivariate optimization in the real world, it became apparent that powerful ideas such as action suppression and adaptive Q-learning modules were critical. These tactics proposed were founded on the basis of reinforcement learning and were meant to facilitate an innovative transformation of the space selection dynamics by adjusting automatically to the user needs and capacities of servers.

We then went on to engage more in I the scalability bottlenecks which are mostly seen in such advanced models. Nevertheless, thanks to innovative methods like approximating functions and dynamic hashmaps, we were able to receive sufficiently sound solutions to deal with these technical challenges. Alongside, our research revealed the extraordinary advantages offered by QNetwork over traditional system through the reduction of time delay and fairness in selection of server respectively.

Concluding our investigation, the vital role of network adaptation for a smooth gaming is dominant. Through the utilization of the latest technologies such as reinforcement learning and dynamic resource allocation, we provide gaming platforms to fulfill a dreaming to users of the whole world. In addition, the ongoing research and progress in this area seems to guarantee the future of online gaming that gamers will love for the long time without the distress.

DECLARATIONS

Acknowledgement: We appreciate the generous support from all the supervisors and their different affiliations.
Funding: No funding body in the public, private, or nonprofit sectors provided a particular grant for this research.
Availability of data and material: In the approach, the data sources for the variables are stated.
Authors' contributions: Each author participated equally to the creation of this work.
Conflicts of Interests: The author declares that there is no conflict of interest related to this study. All research activities, data collection, and analysis were conducted with full transparency and impartiality. No financial or personal relationships that could influence the research outcomes exist. The findings and conclusions presented in this work are solely based on the data collected and the academic analysis carried out throughout the study.
Consent to Participate: Yes
Consent for publication and Ethical approval: Because this study does not include human or animal data, ethical approval is not required for publication. All authors have given their consent.

REFERENCES

[1] Reddy, B. I., & Srikanth, V. (2019). Review on wireless security protocols (WEP, WPA, WPA2 & WPA3). International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 5(4), 28-35.
[2] Halbouni, A., Ong, L. Y., & Leow, M. C. (2023). Wireless Security Protocols WPA3: A Systematic Literature Review. IEEE Access.
[3] Ambavkar, P. S., Patil, P. U., Meshram, B. B., & Swamy, P. K. (2012). Wpa exploitation in the world of wireless network. Int J Adv Res Comput Eng Technol, 1(4), 609-618.
[4] Cahyadi, D., Astuti, I. F., & Nazaruddin, N. (2023, February). Comparison of throughput and CPU usage between WPA3 and WPA2 security methods on wireless networks 802.11 n. In AIP Conference Proceedings (Vol. 2482, No. 1). AIP Publishing.
[5] Baray, E., & Ojha, N. K. (2021, April). WLAN security protocols and WPA3 security approach measurement through aircrack-ng technique. In 2021 5th International conference on computing methodologies and communication (ICCMC) (pp. 23-30). IEEE.
[6] Y. A. Khan, U. Khalil, H. Khan, A. Uddin, S. Ahmed, "Power flow control by unified power flow controller",sss Engineering, Technology & Applied Science Research., vol. 9, no. 2, pp. 3900-3904, Feb. 2019
[7] H. Khan, I. Uddin, A. Ali, M. Husain, "An Optimal DPM Based Energy-Aware Task Scheduling for Performance Enhancement in Embedded MPSoC", Computers, Materials & Continua., vol. 74, no. 1, pp. 2097-2113, Sep. 2023
[8] Khan, S., Ullah, I., Khan, H., Rahman, F. U., Rahman, M. U., Saleem, M. A., ... & Ullah, A. (2024). Green synthesis of AgNPs from leaves extract of Saliva Sclarea, their

characterization, antibacterial activity, and catalytic reduction ability. Zeitschrift für Physikalische Chemie, 238(5), 931-947.
[9] S. Khan, I. Ullah, H. Khan, F. U. Rahman, M. U. Rahman, M. A. Saleem, A. Ullah, "Green synthesis of AgNPs from leaves extract of Salvia Sclarea, their characterization, antibacterial activity, and catalytic reduction ability", Zeitschrift für Physikalische Chemie., vol. 238, no. 5, pp. 931-947, May. 2024
[10] H. Khan, M. U. Hashmi, Z. Khan, R. Ahmad, "Offline Earliest Deadline first Scheduling based Technique for Optimization of Energy using STORM in Homogeneous Multi-core Systems", IJCSNS Int. J. Comput. Sci. Netw. Secur., vol. 18, no. 12, pp. 125-130, Dec. 2018
[11] Waleed, A. Ali, S. Tariq, G. Mustafa, H. Sarwar, S. Saif, I. Uddin, "An Efficient Artificial Intelligence (AI) and Internet of Things (IoT's) Based MEAN Stack Technology Applications", Bulletin of Business and Economics (BBE)., vol. 13, no. 2, pp. 200-206, July. 2024
[12] Liang, Y., Ur Rahman, S., Shafaqat, A., Ali, A., Ali, M. S. E., & Khan, H. (2024). Assessing sustainable development in E-7 countries: technology innovation, and energy consumption drivers of green growth and environment. Scientific Reports, 14(1), 28636.
[13] Noor, A. Ilyas, Z. Javaid, H. Khan, "Framing a Knowledge Domain Visualization on Green Human Resource Management: A Bibliometric Analysis from 2008-2022", Pakistan Journal of Humanities and Social Sciences., vol. 11, no. 4, pp. 4200-4212, Aug. 2023
[14] M. Gondal, Z. Hameed, M. U. Shah, H. Khan, "Cavitation phenomenon and its effects in Francis turbines and amassed adeptness of hydel power plant", In 2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), IEEE., pp. 1-9, Mar. 2019
[15] Liang, Y., Ur Rahman, S., Shafaqat, A., Ali, A., Ali, M. S. E., & Khan, H. (2024). Assessing sustainable development in E-7 countries: technology innovation, and energy consumption drivers of green growth and environment. Scientific Reports, 14(1), 28636.
[16] U. Hashmi, S. A. ZeeshanNajam, "Thermal-Aware Real-Time Task Schedulabilty test for Energy and Power System Optimization using Homogeneous Cache Hierarchy of Multi-core Systems", Journal of Mechanics of Continua and Mathematical Sciences., vol. 14, no. 4, pp. 442-452, Mar. 2023
[17] Y. A. Khan, F. Khan, H. Khan, S. Ahmed, M. Ahmad, "Design and Analysis of Maximum Power Point Tracking (MPPT) Controller for PV System", Journal of Mechanics of Continua and Mathematical Sciences., vol. 14, no. 1, pp. 276-288, May. 2019
[18] Ali, M., Khan, H., Rana, M. T. A., Ali, A., Baig, M. Z., Rehman, S. U., & Alsaawy, Y. (2024). A Machine Learning Approach to Reduce Latency in Edge Computing for IoT Devices. Engineering, Technology & Applied Science Research, 14(5), 16751-16756.
[19] Khan, A. Yasmeen, S. Jan, U. Hashmi, "Enhanced Resource Leveling Indynamic Power Management Techniqueof Improvement In Performance For Multi-Core Processors" ,Journal of Mechanics of Continua and Mathematical Sciences., vol. 6, no. 14, pp 956-972, Sep. 2019
[20] Khan, M. U. Hashmi, Z. Khan, R. Ahmad, "Offline Earliest Deadline first Scheduling based Technique for Optimization of Energy using STORM in Homogeneous Multi-core Systems", IJCSNS Int. J. Comput. Sci. Netw. Secur., vol. 18, no. 12, pp. 125-130, Oct. 2018
[21] Akmal, I., Khan, H., Khushnood, A., Zulfiqar, F., & Shahbaz, E. (2024). An Efficient Artificial Intelligence (AI) and Blockchain-Based Security Strategies for Enhancing the Protection of Low-Power IoT Devices in 5G Networks. Spectrum of engineering sciences, 2(3), 528-586.
[22] H. Khan, M. U. Hashmi, Z. Khan, R. Ahmad, A. Saleem, "Performance Evaluation for Secure DES-Algorithm Based Authentication & Counter Measures for Internet Mobile Host Protocol", IJCSNS Int. J. Comput. Sci. Netw. Secur., vol. 18, no. 12, pp. 181-185, July. 2018
[23] Nasir, M. S., Khan, H., Qureshi, A., Rafiq, A., & Rasheed, T. (2024). Ethical Aspects In Cyber Security Maintaining Data Integrity and Protection: A Review. Spectrum of engineering sciences, 2(3), 420-454.
[24] Khan, A. Ali, S. Alshmrany, "Energy-Efficient Scheduling Based on Task Migration Policy Using DPM for Homogeneous MPSoCs", Computers, Materials & Continua., vol. 74, no. 1, pp. 965-981, Apr. 2023
[25] Fakhar, M. H., Baig, M. Z., Ali, A., Rana, M. T. A., Khan, H., Afzal, W., ... & Albouq, S. (2024). A Deep Learning-based Architecture for Diabetes Detection, Prediction, and

Classification. Engineering, Technology & Applied Science Research, 14(5), 17501-17506.
[26] Shah, S. Ahmed, K. Saeed, M. Junaid, H. Khan, "Penetration testing active reconnaissance phase–optimized port scanning with nmap tool", In 2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), IEEE., pp. 1-6, Nov. 2019
[27] Y. A. Khan, "A GSM based Resource Allocation technique to control Autonomous Robotic Glove for Spinal Cord Implant paralysed Patients using Flex Sensors", Sukkur IBA Journal of Emerging Technologies., vol. 3, no. 2, pp. 13-23, Feb. 2020
[28] Y. A. Khan, "A high state of modular transistor on a 105 kW HVPS for X-rays tomography Applications", Sukkur IBA Journal of Emerging Technologies., vol. 2, no. 2, pp. 1-6, Jun. 2019
[29] Khan, S. Ahmad, N. Saleem, M. U. Hashmi, Q. Bashir, "Scheduling Based Dynamic Power Management Technique for offline Optimization of Energy in Multi Core Processors", Int. J. Sci. Eng. Res., vol. 9, no. 12, pp. 6-10, Dec. 2018
[30] Nasir, M. S., Khan, H., Qureshi, A., Rafiq, A., & Rasheed, T. (2024). Ethical Aspects In Cyber Security Maintaining Data Integrity and Protection: A Review. Spectrum of engineering sciences, 2(3), 420-454.
[31] Y. A. Khan, "Enhancing Energy Efficiency in Temperature Controlled Dynamic Scheduling Technique for Multi Processing System on Chip", Sukkur IBA Journal of Emerging Technologies., vol. 2, no. 2, pp. 46-53, Jan. 2019
[32] Khan, K. Janjua, A. Sikandar, M. W. Qazi, Z. Hameed, "An Efficient Scheduling based cloud computing technique using virtual Machine Resource Allocation for efficient resource utilization of Servers", In 2020 International Conference on Engineering and Emerging Technologies (ICEET), IEEE., pp. 1-7, Apr. 2020
[33] Hassan, H. Khan, I. Uddin, A. Sajid, "Optimal Emerging trends of Deep Learning Technique for Detection based on Convolutional Neural Network", Bulletin of Business and Economics (BBE)., vol. 12, no. 4, pp. 264-273, Nov. 2023
[34] Khan, Q. Bashir, M. U. Hashmi, "Scheduling based energy optimization technique in multiprocessor embedded systems", In 2018 International Conference on Engineering and Emerging Technologies (ICEET), IEEE., pp. 1-8, Sep. 2018
[35] Javed, M. A., Anjum, M., Ahmed, H. A., Ali, A., Shahzad, H. M., Khan, H., & Alshahrani, A. M. (2024). Leveraging Convolutional Neural Network (CNN)-based Auto Encoders for Enhanced Anomaly Detection in High-Dimensional Datasets. Engineering, Technology & Applied Science Research, 14(6), 17894-17899.
[36] Y. A. Khan, M. Ibrahim, M. Ali, H. Khan, E. Mustafa, "Cost Benefit Based Analytical Study of Automatic Meter Reading (AMR) and Blind Meter Reading (BMR) used by PESCO (WAPDA)", In 2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), IEEE., pp. 1-7, Aug. 2020
[37] Rahman, M. U., Khan, S., Khan, H., Ali, A., & Sarwar, F. (2024). Computational chemistry unveiled: a critical analysis of theoretical coordination chemistry and nanostructured materials. Chemical Product and Process Modeling, 19(4), 473-515.
[38] Naz, H. Khan, I. Ud Din, A. Ali, and M. Husain, "An Efficient Optimization System for Early Breast Cancer Diagnosis based on Internet of Medical Things and Deep Learning", Eng. Technol. Appl. Sci. Res., vol. 14, no. 4, pp. 15957–15962, Aug. 2024
[39] Khan, I. Ullah, M. U. Rahman, H. Khan, A. B. Shah, R. H. Althomali, M. M. Rahman, "Inorganic-polymer composite electrolytes: basics, fabrications, challenges and future perspectives", Reviews in Inorganic Chemistry., vol. 44, no. 3, pp. 1-2, Jan. 2024
[40] Naveed, A., Khan, H., Imtiaz, Z., Hassan, W., & Fareed, U. (2024). Application and Ethical Aspects of Machine Learning Techniques in Networking: A Review. Spectrum of engineering sciences, 2(3), 455-501.
[41] Sarwar, H. Khan, I. Uddin, R. Waleed, S. Tariq, "An Efficient E-Commerce Web Platform Based on Deep Integration of MEAN Stack Technologies", Bulletin of Business and Economics (BBE)., vol. 12, no. 4, pp. 447-453, Jun. 2023
[42] Sebbar, A., Boulahya, S. E., Mezzour, G., & Boulmalf, M. (2016, May). An empirical study of wifi security and performance in morocco-wardriving in rabat. In 2016 International Conference on Electrical and Information Technologies (ICEIT) (pp. 362-367). IEEE.
[43] Apthorpe, N., Reisman, D. and Feamster, N., 2016. A smart home is no castle: privacy

vulnerabilities of encrypted IoT traffic, Workshop on Data and Algorithmic Transparency (DAT), New York, USA, 19 November.
[44] [48] Zhou, W. and Piramuthu, S., 2014. Security/privacy of wearable fitness tracking IoT devices, CISTI, IEEE, Barcelona, Spain, 18-21 June, pp. 1-6.
[45] [49] Perera, C., 2017. Privacy guidelines for Internet of Things: a cheat sheet, Technical report, New Castle University, UK, pp. 1-9.