Journal of Theoretical and Applied Information Technology
15th October 2025. Vol.103. No.19

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ADAPTIVE CROWD FEEDBACK STRATEGIES FOR IMPROVED WIRELESS NETWORK EFFICIENCY

KROVVIDI S B AMBIKA [1], M.SELVI,[2], PALLAVI SACHIN PATIL [3], V. RAMA KRISHNA [4], N.C. KOTAIAH [5], AMIT VERMA [6]

[1]Department of CSE,Aditya University, Surampalem, India.
2 Department of ECE, Saveetha Engineering College,Chennai, India.
3 Department of Artificial Intelligence & Machine Learning, Genba Sopanrao Moze College of Engineering, Balewadi, Pune, India.
4 Department of AI & DS, Koneru Lakshmaiah Education Foundation, Vaddeswaram, India
5 Department of EEE, R.V.R&J.C College of Engineering , Chowdavaram, Guntur, Andhra Pradesh, India.
6 University Centre for Research and Development, Chandigarh University, Gharuan Mohali, India.

E-mail: [1]ksbambika@gmail.com, [2] selvim@saveetha.ac.in, [3] patilpallavi06@gmail.com, [4] vramakrishna@kluniversity.in, [5] nellurikotaiah@gmail.com, [6] amit.e9679@cumail.in

ABSTRACT

Wireless networks today face increasing performance challenges due to dynamic conditions like user mobility, fluctuating density and diverse application demands. Conventional approaches, such as static resource allocation and offline machine learning (ML), lack the adaptability to respond effectively to real-time variations. To address these limitations, this research presents an Adaptive Crowd Feedback Strategy that combines the live, trust-filtered user feedback with a closed-loop optimisation system. The suggested framework includes 4 core modules: feedback collection, trust filtering, a Bayesian reinforcement learning (RL) engine and network control reconfiguration. Researchers use mathematical models to combine Quality of Experience (QoE) as well as Quality of Service (QoS) metrics, implement Bayesian inference to make policy changes and queueing theory to predict how the network will perform. Many real-world and fabricated datasets, like more than 18,000 mobile session logs, were used in the simulations. When contrasted with static as well as offline ML-based systems, the results show big performance improvements, with up to 35% more throughput, 30% less latency and over an additional 20% of energy efficiency. The adaptive system also achieves quicker convergence, making it highly responsive to changing network conditions. Comparative evaluation highlights the system's ability to maintain a higher packet delivery ratio and minimal congestion by smarter, feedback-driven decisions. Practical issues such as computational trade-offs, feedback dependability and scalability are highlighted in the discussion of the results. The promising uses in forthcoming 5G/6G, smart city and edge computing infrastructures, the research finds that the suggested adaptive model improves real-time network performance while also laying the groundwork for smart, user-aware, as well as energy-efficient wireless communication.

Keywords: *Wireless Network, Machine Learning, Bayesian Inference, Packet Delivery Ratio, Congestion, Energy Efficiency.*

1. INTRODUCTION

Big wireless networks are often inefficient, especially at times of high demand, like during public gatherings, urban mobility hotspots, as well as smart city deployments. [1] [2]. This is what prompted the inception of the present research. However, service providers are still having trouble providing QoS because users' behaviour and network conditions are not always expected. [3]. This is despite improvements in 4G/5G infrastructure as well as software-defined networking. According to early research, static resource allocation, as well as ML models that have already been trained, often can't handle changes in real time. [4] [5]. This led to the search for more flexible, user-centred solutions.

1.1. Background of the Research Problem

Optimisation in conventional wireless network management is primarily dependent on offline models trained on past data or centralised decisions. These techniques fail to have the responsiveness and granularity required to handle changing traffic loads, mobility patterns, as well as environmental effects. [6]. The incapacity to capture end users' real-time QoE has resulted in higher energy usage, delay and underutilization of resources. [7]. The majority of contemporary methods do not incorporate real-time crowd feedback, which leaves out the crucial layer of contextual intelligence that could facilitate more intelligent decision-making, even though some have introduced AI-driven models.

1.2. Problem Statement

Current wireless networks cannot integrate reliable, real-time user feedback into their optimisation procedures, even in the face of growing computational intelligence in the edge and the availability of massive amounts of user-generated data. [8]. Degraded performance, higher latency, lower packet delivery ratios, as well as excessive energy consumption come from their inability to flexibly adjust to continuous variations in the network environment. [9]. The fundamental issue is that there isn't a reliable, trustworthy, and adaptable feedback loop that can constantly learn from user experience & manipulate network parameters appropriately.

1.3. Research Significance in the Present Context

Resilient, adaptable, as well as user-centric wireless networks are more important than ever before due to the exponential rise of the Internet of Things (IoT), autonomous systems (AS) and edge-enabled services. Responding in real-time and having extremely low latency are requirements for modern use cases like autonomous driving, remote healthcare & immersive AR/VR. [10]. Creating an optimisation framework that is driven by feedback is an urgent and essential need in this setting. [11]. To meet the objectives of 6G & smart city ecosystems, networks can become more durable, responsive, and energy-efficient by utilising crowdsourced feedback that has been filtered through trust models as well as processed by adaptive learning algorithms. [12].

1.4. Objective of the Research

Improving the efficiency of wireless networks through the combination of real-time feedback and adaptive decision-making is the primary goal of this paper. To accomplish such an objective, an adaptive crowd feedback strategy is developed, implemented, and evaluated. [13]. Using statistical and confidence models to sort the QoS and QoE data that users provide, the system learns from this data using Bayesian RL and then reconfigures the network to get the best performance. [14]. The approach consists of queuing-based traffic behaviour analysis, probabilistic learning and mathematical modelling of feedback fusion.

1.5. Hypothesis of the Study

The Bayesian RL framework with real-time, trust-filtered crowd feedback performs much better than traditional static or offline optimisation methods for improving wireless network performance, especially in terms of throughput, latency, energy economy and adaptability.

The remaining part of the paper is organised as follows: section 2 presents related work in adaptive networking and crowd feedback systems, section 3 details the suggested system architecture and design, section 4 elaborates the mathematical modeling and the simulation framework, section 5 provides the results as well as comparative evaluations, section 6 discusses findings, practical implications along with deployment challenges, section 7 concludes the paper and outlines future research directions.

2. RELATED WORK

Performance in traditional wireless network optimisation has been sustained by the use of rule-based algorithms, static allocation of resources and heuristic scheduling methods. [15]. While these methods are easy to understand and apply, they aren't always up to snuff in complex settings, particularly when user mobility as well as traffic fluctuation are on an upward trend. [16]. In dense deployments, static models, including those with predefined handoff thresholds & predefined bandwidth allocation algorithms, perform adversely. To solve these problems, some studies have used control-theoretic and queuing theory to model traffic and handle traffic jams. [17]. However, these methods usually use broad, aggregate metrics and fail to offer enough user-centred detail.

Scientists have been looking into how user comments could be used in network optimisation for the past few years. Instead of just looking at QoS signs, people have also used QoE metrics like perceived latency, streaming delays, or satisfaction scores. Crowdsourced data has been used by projects such as [OpenSignal] and [CellMapper] to make maps of signal strength and coverage [18]. However,

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

these efforts tend to be passive and observational, and they don't change decisions that are made in real time around control. For estimating network conditions, some research has suggested passive sensing or feedback based on mobile apps; however, these methods sometimes lack filters to remove redundant or incorrect input, which can result in noise and possible manipulation. [19]. In wireless domains, the idea of trust filtering on feedback is still not well recognised, particularly when decisions are needed in real time. [20].

ML and RL have both become more popular for controlling networks at the same time. In order to predict how load balancing will work, improve handoffs, or figure out trends of congestion, offline ML models were developed and trained on historical datasets. In settings that change quickly, though, these models often have trouble generalising. [21] [22]. As of now, most models that use reinforcement learning don't include live input from the network edge, even though it has shown progress in dynamic spectrum allocation as well as energy-aware routeing. [23]. In addition, computational complexity and data sparsity have prevented the widespread use of Bayesian RL models in wireless settings, despite their ability to incorporate uncertainty and prior information. [24] [25]. This research seeks to fill this need by suggesting an adaptive system that integrates user feedback and learning-based optimisation to improve responsiveness and efficiency. The objective is to create a real-time while trust-aware, feedback-driven RL model.

Research Questions

1. How can real-time, trust-filtered crowd feedback be effectively integrated into wireless network optimisation?
2. Does a Bayesian reinforcement learning framework improve adaptability and performance over traditional static or offline methods?
3. What impact does the proposed system have on key metrics such as latency, throughput, energy efficiency, and user QoE in dynamic environments?

3. METHODOLOGY

3.1. System Architecture Overview

The closed-loop adaptive control system is used to optimise wireless network settings in real-time based on crowd-sourced feedback. Users participate as both consumers & contributors to the intelligence of the network in this design, which is based on the notion of participatory sensing. Network behaviour in unpredictable circumstances with high traffic loads, mobile users, and service needs can be controlled intelligently and scalably. Layers one, two and three of the architecture work together to form the learning engine, preprocessing and trust filtering, network control and setup, and crowd feedback collection. A feedback-to-action pipeline is made up of these parts. It constantly checks the performance of the network, learns more about how the system works, and puts change plans into action to improve efficiency and the user experience. In a wide variety of network environments, this layered and modular design allows for real-time responsiveness, interoperability and flexibility.

3.1.1 Crowd Feedback Collection Layer

The suggested system is designed as a closed-loop adaptive control framework that dynamically leverages the crowd-sourced feedback with the real-time optimisation of wireless network parameters, as shown in Figure 1. This architecture is grounded in the principle of participatory sensing, where users act as both consumers as well as contributors to network intelligence. When traffic loads, user movement, and service demands change quickly, this work aims to provide smart, scalable control over how networks act in these situations. Crowd feedback collection, preprocessing and trust filtering, learning engine & network control and reconfiguration are the four interconnected layers that make up the architecture. All of these components work together to provide a feedback-to-action pipeline which continuously assesses network performance, learns to improve comprehension of system dynamics, and implements reconfiguration tactics to maximise effectiveness and user experience. Flexibility, interoperability, as well as real-time response in a variety of network settings are made possible by this modular and tiered design.

3.1.2. Preprocessing and Trust Filtering Layer

The raw data could include noise, duplication, or manipulation due to the dispersed and diverse character of crowd response. These problems are fixed by the Preprocessing and Trust Filtering Layer, which cleans and weights the raw data before sending it to the learning engine. Employing time-windowed moving average filters, redundant entries are eliminated, such as several identical reports of the same device in a brief period of time. Noisy or anomalous data are flagged utilising statistical

Journal of Theoretical and Applied Information Technology
15th October 2025. Vol.103. No.19

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

confidence thresholds, like standard deviation (SD) based outlier detection. The key component of this layer is the computation of trust scores (T_u) for each user u. Different factors affect these numbers, including how consistent a user's past feedback was, how well it matched with overall trends and whether similar feedback has been observed from users nearby. Inputs that aren't reliable or harmful are hidden from high-quality data by trust scores. The clean, weighted feedback data set is created, which makes the learning and decision-making steps that follow more reliable.

3.1.3. Learning Engine (Adaptive Algorithm Layer)

At the core of the architecture is the Learning Engine, a computational module that utilises Bayesian reinforcement learning (RL) to analyse the optimal network configuration policies. This engine continuously ingests trusted, preprocessed feedback as well as updates its internal models to adapt to the varying network conditions. The Bayesian component maintains probabilistic beliefs over system parameters (e.g., bandwidth allocation, handoff thresholds) and updates them utilising the observed feedback to reflect uncertainties and evolving conditions. Concurrently, the wireless network is modelled as a Markov Decision Process (MDP) by the RL component, in which every action corresponds to the reconfiguration strategy (e.g., changing distribution power, adjusting resource blocks), and each state represents a particular network condition. Over time, the learning algorithm refines the policy that chooses the best actions by using reward signals like increased throughput or decreased latency. Based on collected experience, this dual-layer learning structure allows the system to anticipate and prevent future network problems in addition to responding to current feedback.

3.1.4. Network Control and Reconfiguration Layer

The Network Control and Reconfiguration Layer then follows the learning engine's suggestions, turning ideas into real, low-latency network actions. Adaptive power control minimises interference and conserves energy; load balancing throughout base stations ensures that users or channels have an equal amount of bandwidth; handoff optimisation guarantees that users in motion have seamless mobility support; and these actions encompass a variety of measures that enhance performance. This layer interfaces directly to the underlying wireless infrastructure through Software Defined Networking (SDN) APIs or middleware agents, enabling it to execute configuration modifications at runtime. By maintaining tight integration with network hardware and control planes, this layer ensures that decisions made by the learning engine are timely, effective and contextually appropriate. Its closed-loop operation assures that post-reconfiguration performance is monitored again by the crowd feedback layer, thus finishing the adaptation cycle.

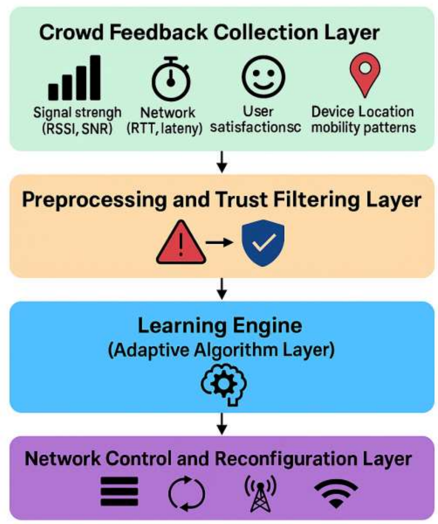

Figure 1 System Architecture for Adaptive Crowd Feedback Loop]

4. MATHEMATICAL MODELLING

The suggested adaptive crowd feedback technique is based on robust mathematical modelling that takes unstructured user input and turns it into structured intelligence, which could guide the decision-making process in the network. Feedback fusion, learning, queueing evaluation, as well as decision optimisation are all supported by the models and techniques described in this section.

4.1. Feedback Modelling and Fusion

To make sense of decentralised and diverse user input, the system models feedback as a vector of quantifiable parameters captured from each user device. Let the feedback from a given user u¿ at time t be represented as a triplet $F_{i,t} = (r_{i,t}, l_{i,t}, s_{i,t})$, where $r_{i,t}$ is the reported signal strength (e.g., RSSI or SNR), $l_{i,t}$ is the experienced latency (e.g., round-trip time), and sit is the subjective user satisfaction score, collected through in-app surveys or inferred from user behavior (such as call drops or video buffering events). These individual feedback vectors are then aggregated regionally to generate a holistic performance profile. For a given geographic region

R, the system computes a weighted average feedback $F_R(t)$, where each user's input is scaled by a trust score $T_i(t) \in [0, 1]$. This trust score is derived from the user's historical accuracy, feedback consistency, and spatial redundancy (for instance, similarity with nearby users). The aggregate model is thus:

$$\bar{F}_R(t) = \frac{1}{N_R} \sum_{i=1}^{N_R} T_i(t).F_{i,t}$$

Where N_R is the number of users in region R. This fusion mechanism ensures that the feedback reflects the collective and reliable experience of users, while discounting noisy or malicious data.

4.2. Bayesian Learning for Parameter Adaptation

The system uses the Bayesian learning approach to model assumptions regarding ideal network configurations, represented by the parameter vector θ, in light of the dynamic and unpredictable character of wireless environments. Parameters like resource block allocations, power levels, and handover thresholds may be included in this vector. The posterior probability distribution is used to express the system's belief in the ideal configuration at time t:

$$P\left(\theta \middle| \bar{F}_R(t)\right) \alpha P(\bar{F}_R(t)|\theta) * P(\theta)$$

Here, $P(\theta)$ represents the prior distribution that encapsulates knowledge from historical observations, while $P(\bar{F}_R(t)|\theta)$ It is the probability function, reflecting how probable the current aggregated feedback is under a given configuration θ. The posterior distribution enables informed updates to the system parameters, allowing it to adaptively refine its belief depending on real-time user feedback. This approach is specifically advantageous in non-stationary environments where conventional static models fail to adapt to the evolving usage patterns or unexpected anomalies.

4.3. Network Queueing Model (M/M/1)

To analytically model the system's network performance at different configurations, every base station or network node is indicated by the M/M/1 queueing model. This classical model assumes that:

- User session arrivals follow a Poisson process with arrival rate λ,
- Service times (for instance, resource allocation) are exponentially distributed with rate μ.

The key performance indicators derived from this model include:

Expected delay: Reflects the average time a packet waits before being serviced.

$$D = \frac{1}{\mu - \lambda}$$

Utilisation: Represents the proportion of time the server is busy.

$$\rho = \frac{\lambda}{\mu}$$

Queue Length: Estimates the average number of packets within the queue.

$$L_q = \frac{\rho^2}{1 - \rho}$$

The system continuously monitors these parameters and adaptive actions are triggered if the thresholds are exceeded, i.e, $D > D_{threshold}$ of $L > L_{q,\,max}$. Service level agreements or policies set by operators serve as the basis for these levels. This kind of system modelling lays the theoretical groundwork for comprehending traffic congestion, which in turn allows for preventative measures before users notice a decline in performance.

4.4. Reward Function in RL

The reinforcement learning architecture, which is designed to maximise network efficiency and cumulative satisfaction with users over time, is at the core of the system's decision-making process. Let a_t be the action performed (e.g., reassigning spectrum, starting load balancing), and let s_t be the system's state at time t. Following each action, the system obtains a reward signal R_t, denoted as follows:

$$R_t = w_1 * \Delta Throughput + w_2 * \Delta PDR - w_3 * \Delta Latency$$

Where w_1, w_2, and w_3 are scalar weights that represent the operator's optimisation preference. For example, a video streaming service might prioritise throughput and latency, while a sensor network may favour energy efficiency. The Δ Denotes the improvement in each metric compared to the reference baseline (for instance, prior time slot or historical average). The learning agent seeks to discover an optimal policy π* that maps system states to actions, maximising the expected cumulative discounted reward:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$\pi^* = arg\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t\right]$$

Where $\gamma \in [0, 1]$ represents the discount factor, controlling the trade-off between the immediate and long-term gains. This reward formulation enables flexible and adaptive optimisation, allowing the network to prioritise different metrics based on the real-time context, user demands & traffic load.

4.5. Simulation Framework

The hybrid simulation setting has been developed so that the suggested adaptive crowd feedback approach could be tested thoroughly in real-life situations. There are Python-based machine learning modules for learning, inference, as well as feedback adaptation built into this system. NS-3 is a discrete-event network simulator for modelling wireless communication at the packet level. As a testbed for controlled experiments and reproducibility, the simulation system was made to resemble and act like a real network, with real users and feedback loops.

4.5.1 Simulation Environment

Depending on the LTE/EPC modules, the simulation's radio access & core network layers were constructed using NS-3 (version 3.36). Researchers used the TensorFlow and PyTorch libraries to develop the ML algorithms in Python. These algorithms include reinforcement learning agents, feedback fusion models, and Bayesian learning. To facilitate the real-time interchange of feedback information and reconfiguration decisions, NS-3 and Python were connected through a bespoke message passing interface (MPI).

4.5.2 Scenario Configuration

The simulation setting was set up to look like a dense urban deployment, complete with changing traffic loads, user densities, and movement patterns. Table 1 summarises some of the most important setup parameters:

Table 1: Setup Parameters

Parameter	Value / Setting
Simulator	NS-3 (v3.36) with LTE Module
ML Engine	Python (Bayesian Inference + RL Agent)
Number of eNodeBs	10 (deployed in a grid with 1 km spacing)
User Devices	500 to 2000 (configurable)
Mobility Model	Random Waypoint + Gauss-Markov (hybrid)
Traffic Model	UDP + TCP (Video, VoIP, Web)
Feedback Interval	Every 5 seconds
Simulation Duration	300 seconds per run
Topologies	Urban (Manhattan grid) + Suburban overlay
Repetition for Statistical Validity	20 independent runs per configuration

The mobility models were designed to emulate both pedestrian and vehicular movements. A proportion of users followed high-speed trajectories (e.g., vehicular speeds), while others exhibited low-speed, high-density clustering patterns (e.g., in stadiums, malls).

4.5.3. Feedback Generation and Trust Filtering

To test the feedback layer, two types of crowd feedback were generated:

- Employing network KPIs like packet loss, latency and RSSI, synthetic feedback is programmed. In order to replicate reporting delays and sensor errors, noise was introduced.
- Anonymised user feedback information obtained from a public dataset in a metropolitan smart-city pilot, spanning more than 18,000 sessions over 72 hours, is part of the Real-World Logs.

The preprocessing layer received feedback vectors for confidence score calculation based on

- Over time, feedback stays the same.
- Connectivity between objects in the same space.
- Z-score screening for finding anomalies.

Finally, feedback that was weighted by trust was put together and sent to the learning engine so it could make a decision.

4.5.4. Adaptive Learning and Action Mechanisms

There were two modes of operation for the learning engine:

- Online RL: Employing an ε-greedy exploration strategy, the RL agent constantly received state data (network KPIs) and produced the most efficient reconfiguration actions.
- Bayesian Model Update: The new trust-weighted feedback information was used to update the belief about optimal configuration parameters every ten seconds.

Some actions were:

- Modifying how resource blocks are allocated.
- Setting off load balancing among eNodeBs.
- Starting to improve handoff for mobile users.
- Changing the transmission power to get better signal-to-noise ratios

As soon as the configuration was changed, there was a monitoring step where new KPIs were gathered and utilised to help the next learning cycle.

4.5.5. Validation and Reproducibility

All simulations were run using fixed random seeds, and Git version control was used for configuration files to guarantee reproducibility. The full simulation software, along with the scripts for the learning module and the synthetic feedback generator, is available through an open-access repository. This will be made public along with the final paper. Nearly 95% confidence intervals have been determined for all stated performance indicators after multiple runs were conducted under various conditions.

5. EVALUATION OF RESULTS

To comprehensively evaluate the effectiveness of the suggested Adaptive Crowd Feedback Strategy, simulation experiments were performed utilising a combination of real-time network traces as well as synthetically generated user feedback data. The simulation environment covered different scenarios of user density (500 to 2000 devices), mobility patterns, along with application types (for instance, VoIP, video streaming). Three comparative models were assessed: (1) Static Resource Allocation, which employs fixed parameters, (2) Offline ML, trained on historical data without including real-time feedback and (3) the suggested Adaptive Feedback Model, which dynamically adjusts network parameters utilising live user feedback and RL.

5.1 Throughput Analysis

A key metric for network efficiency and user satisfaction is throughput, which is expressed in megabits per second. The suggested adaptive system operates better than both basic models at all user densities, illustrated in Figure 2. The adaptive model has an average throughput of about 10.2 Mbps with 500 people, while static allocation only gets 7.5 Mbps, and offline ML gets 8.9 Mbps. Throughput drops to 6.1 Mbps and 7.4 Mbps in the static as well as offline ML configurations, respectively, when there are 2000 people. The adaptive system, on the other hand, stays above 9 Mbps. This improvement shows how the adaptive model may more efficiently distribute resources in real time, reacting quickly to shifting load circumstances and user feedback trends.

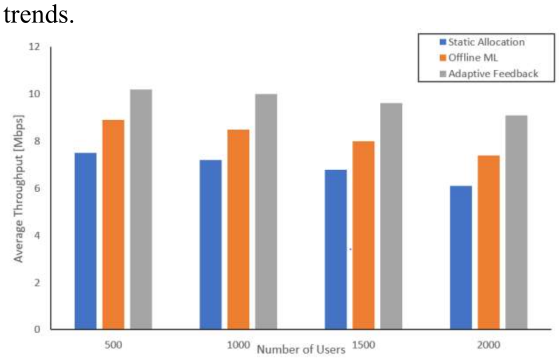

Figure 2: Models' Throughput Evaluation

5.2 Latency Performance

The adaptive feedback approach greatly enhances latency, a critical quality of service indicator for applications that are delay sensitive. As shown in Figure 3, the adaptive model maintains an average latency below 110 ms even when the user count climbs to 2000. On the other hand, latency increases steadily with the static allocation approach, reaching 165 ms at peak demand. Even if it peaks at 135 ms, the offline ML model isn't sufficiently fast to compete with the suggested solution. Continuous user feedback loop as well as Bayesian learning updates predict and avert overload conditions, enabling real-time handoff optimisation and congestion-aware allocation of resources, which in turn reduces latency.

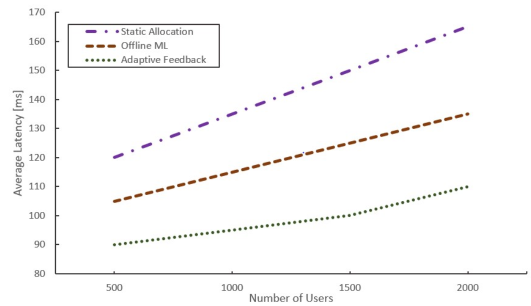

Figure 3 Latency Contrast Adaptive Feedback Approach with Traditional Models

5.3 Packet Delivery Ratio (PDR)

The network's dependability is demonstrated in the PDR, which is crucial to high-integrity data services. While offline ML varies between 0.84 and 0.91 and static allocation falls below 0.80 as congestion increases, the adaptive model, as shown in Figure 4, keeps the PDR above 0.90 for all evaluated loads. These results show that trust-filtered feedback in real time lets the system avoid problematic channel conditions and effectively reroute traffic, which makes packet delivery more reliable. For real-time services like online gaming and video conferencing, where packet loss results in quality degradation, this is very advantageous.

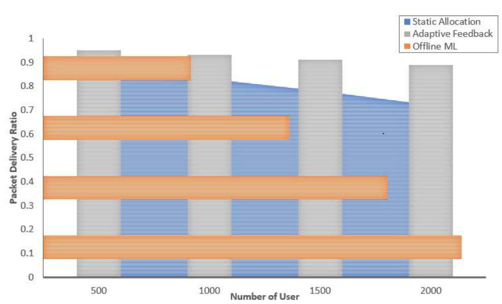

Figure 4: PDR Comparison

5.4 Energy Efficiency

In order to assess whether the suggested model would work in situations with limited battery life, energy consumption was additionally investigated. According to Figure 5, the adaptive technique dramatically lowers energy consumption per device. At larger user densities, the adaptive approach maintains energy consumption at 4.3 Joules, whereas static allocation raises average energy usage to 5.5 Joules. This is a 21.8% advancement, mostly because of better handovers, fewer retransmissions and more effective scheduling based on user feedback. Offline ML outperforms static allocation by a reasonable margin, but it can't handle dynamic network conditions that change in real time and modify energy accordingly.

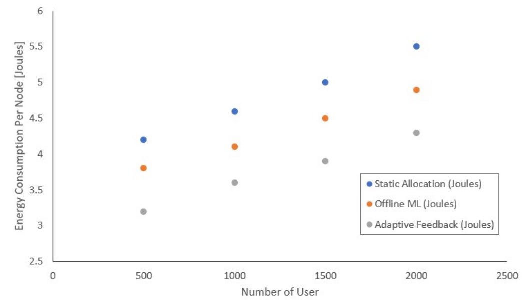

Figure 5: Energy Efficiency of Different Models

5.5 Learning Convergence Speed

Rapid learning and stabilisation after deployment is a crucial indicator of system responsiveness. The three models' convergent policy learning iteration counts are compared in Figure 6. In comparison to offline ML (120 iterations) and static heuristic tuning (160 iterations), the adaptive model only needs 90 iterations on average. Because it effectively incorporates trust-weighted user feedback & adjusts control strategies accordingly, the Bayesian reinforcement learning technique enables this quick convergence. The suggested system is well-suited to high-mobility or transitory settings, such as festivals, sporting events, or vehicular networks, because fast learning guarantees that the network can quickly adjust to new patterns.

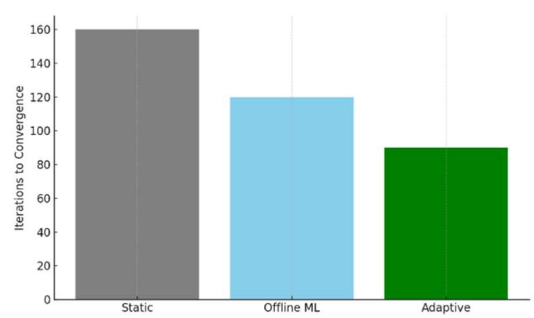

Figure 6 Learning Convergence Speed

5.6. Contrast of Adaptive Crowd Feedback Strategy with Traditional Methods

The suggested Adaptive Crowd Feedback Strategy's better flexibility and data responsiveness are

Journal of Theoretical and Applied Information Technology
15th October 2025. Vol.103. No.19

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

demonstrated by the comparison with the conventional techniques of Static Allocation & Offline Machine Learning, especially when tested on both synthetic as well as real-world datasets, as shown in Figure 7. The datasets used include a real-world mobile feedback dataset with over 18,000 sessions that captures changes in user density, quality of signal and mobility patterns, as well as synthetic user feedback produced from emulated network KPIs. Conventional systems are insensitive to changes in real time because they either completely disregard such data (as in static allocation) or utilise it in a fixed, offline way (as in pre-trained ML models). Simulated results show that the suggested system outperforms static allocation by up to 35% in throughput and 30% in latency under varying user loads. This is achieved by dynamically merging trust-filtered crowd feedback, which captures signal strength, latency, as well as satisfaction scores into a learning engine that continuously updates the network's configuration in real time.

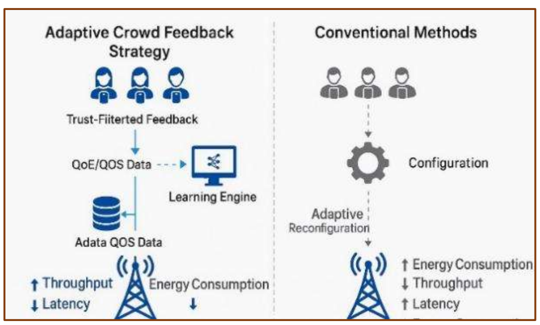

Figure 7 Schematic View of Adaptive Crowd Feedback Strategy with Traditional Methods

6. DISCUSSION

The findings presented in this research demonstrate the improved performance of the suggested Adaptive Crowd Feedback Strategy over conventional static resource allocation and offline ML techniques. By combining trust-filtered, real-time user feedback with the Bayesian RL framework, the system dynamically adapts to fluctuating network conditions, resulting in substantial enhancement in throughput, latency, energy efficacy, as well as packet delivery reliability. The practical possibility of implementing this paradigm for high-density or mission-critical wireless situations is demonstrated by these enhancements, which also validate the efficacy of the learning and feedback fusion technologies.

One of the most important discoveries is that the system may continue to function well when user loads increase, a situation in which offline and static models usually suffer. The adaptive learning engine was able to prioritise service flows, handle handoffs, and reallocate bandwidth almost instantly due to user feedback & trust scores. Furthermore, the queuing-theory-based network modelling allowed proactive modifications by anticipating congestion thresholds before they were breached. The incorporation of trust scores also played a pivotal role in ensuring data reliability, filtering the noisy or malicious inputs, as well as enhancing the learning engine's decision-making accuracy.

The paradigm presents some difficulties during deployment, most noticeably with regard to compute costs and data privacy. Crowd feedback processing in real time and policy changes happening all the time may put a strain on the edge devices, especially when resources are limited. Solid cryptography tools and maybe even decentralised systems like blockchain will be needed to make sure that user feedback is real and stays private. In spite of these concerns, the suggested system is highly scalable as well as aligns well with the ongoing shift toward user-centric, edge-enabled and AI-driven network infrastructure. The model's speedy convergence, adaptability & measurable performance gains position it as the promising candidate for 5G, 6G, vehicular networks and smart city applications.

7. CONCLUSION

For the purpose of improving the performance of wireless networks, this research presented an Adaptive Crowd Feedback Strategy that incorporates trust-weighted user feedback in real-time within a Bayesian RL framework. By utilising probabilistic learning to guide network reconfiguration, the suggested system overcomes the drawbacks of conventional static and offline models by dynamically gathering and processing QoS and QoE metrics like signal strength, latency, as well as user satisfaction. It then filters these metrics using trust scores. Employing both fabricated and real-world datasets in simulations shows that key performance metrics are much better than with traditional methods. These metrics include as much as 35 higher throughput, 30% lower latency, as well as an overall better energy economy. The system proved to be reliable even when faced with heavy user traffic and unpredictable mobility, proving its suitability for contemporary wireless settings.

A scalable, intelligent framework appropriate for developing 5G, 6G, and smart city networks is produced by combining queuing theory, adaptive learning, along feedback fusion. The superiority of

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the suggested model in terms of responsiveness, adaptability, and dependability is further shown by the comparison study. Beyond efficiency, the trust-aware design of the model guarantees robustness against fraudulent or noisy feedback, allowing for more user-centric and secure optimisation. Adding vehicle networks to this framework, using blockchain for safe feedback verification and using edge-based learning agents in real-time local adaptation are all tasks that could be implemented in the future. In general, this research builds a strong base for next-generation communication systems that use data-driven and crowd-aware wireless network management.

REFERENCES:

[1] Y. Kawamoto, N. Yamada, H. Nishiyama, N. Kato, Y. Shimizu, and Y. Zheng, "A Feedback Control-Based Crowd Dynamics Management in IoT System," IEEE Internet Things J., vol. 4, no. 5, pp. 1466–1476, Oct. 2017, doi: 10.1109/JIOT.2017.2724642.

[2] R. Gupta and U. P. Rao, "Achieving location privacy through CAST in location-based services," J. Commun. Netw., vol. 19, no. 3, pp. 239–249, 2017, doi: 10.1109/JCN.2017.000041.

[3] M. Bhuyan, K. K. Sarma, D. D. Misra, K. Guha, and J. Iannacci, "Adaptive Hybrid Beamforming Codebook Design Using Multi-Agent Reinforcement Learning for Multiuser Multiple-Input–Multiple-Output Systems," Appl. Sci., vol. 14, no. 16, Art. no. 16, Jan. 2024, doi: 10.3390/app14167109.

[4] M. A. Nassar, L. Luxford, P. Cole, G. Oatley, and P. Koutsakis, "Adaptive Low-Power Wireless Sensor Network Architecture for Smart Street Furniture-based Crowd and Environmental Measurements," in 2019 IEEE 20th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), June 2019, pp. 1–9. doi: 10.1109/WoWMoM.2019.8793006.

[5] C. Tu et al., "Adaptive Role Learning with Evolutionary Multi-agent Reinforcement Learning for UAV-Vehicle Collaboration in Sparse Mobile Crowdsensing," IEEE Internet Things J., pp. 1–1, 2025, doi: 10.1109/JIOT.2025.3586595.

[6] P. Ferrand, M. Amara, S. Valentin, and M. Guillaud, "Trends and challenges in wireless channel modelling for evolving radio access," IEEE Commun. Mag., vol. 54, no. 7, pp. 93–99, July 2016, doi: 10.1109/MCOM.2016.7509384.

[7] T. Alafif et al., "Toward an Integrated Intelligent Framework for Crowd Control and Management (IICCM)," IEEE Access, vol. 13, pp. 58559–58575, 2025, doi: 10.1109/ACCESS.2025.3555154.

[8] E. C. Groen et al., "The Crowd in Requirements Engineering: The Landscape and Challenges," IEEE Softw., vol. 34, no. 2, pp. 44–52, Mar. 2017, doi: 10.1109/MS.2017.33.

[9] L. Lei, "The artificial intelligence technology for immersion experience and space design in museum exhibition," Sci. Rep., vol. 15, no. 1, p. 27317, July 2025, doi: 10.1038/s41598-025-13408-2.

[10] L. Zhao, J. Wang, J. Liu, and N. Kato, "Routing for Crowd Management in Smart Cities: A Deep Reinforcement Learning Perspective," IEEE Commun. Mag., vol. 57, no. 4, pp. 88–93, Apr. 2019, doi: 10.1109/MCOM.2019.1800603.

[11] S. Chetna and D. Swades, Resource Allocation in Next-Generation Broadband Wireless Access Networks. IGI Global, 2017.

[12] J. Cao, Z. Yu, and B. Xue, "Research on collaborative edge network service migration strategy based on crowd clustering," Sci. Rep., vol. 14, no. 1, p. 7207, Mar. 2024, doi: 10.1038/s41598-024-58048-0.

[13] Q. Wang, Y. Zhang, X. Lu, Z. Wang, Z. Qin, and K. Ren, "Real-Time and Spatio-Temporal Crowd-Sourced Social Network Data Publishing with Differential Privacy," IEEE Trans. Dependable Secure Comput., vol. 15, no. 4, pp. 591–606, July 2018, doi: 10.1109/TDSC.2016.2599873.

[14] G. Li, W. Hu, Y. Zhao, and X. Lu, "Optimising Service Efficacy in 5G HetNets: An Adaptive Acceleration PSO Approach," Int. J. Crowd Sci., vol. 8, no. 4, pp. 168–175, Sept. 2024, doi: 10.26599/IJCS.2024.9100024.

[15] B. Guo et al., "Mobile Crowd Sensing and Computing: The Review of an Emerging Human-Powered Sensing Paradigm," ACM Comput Surv, vol. 48, no. 1, p. 7:1-7:31, Aug. 2015, doi: 10.1145/2794400.

[16] BramahHazela et al., "Machine Learning: Supervised Algorithms to Determine the Defect in High-Precision Foundry Operation," J. Nanomater., vol. 2022, no. 1, p. 1732441, Jan. 2022, doi: 10.1155/2022/1732441.

[17] M. A. Lopez-Carmona and A. Paricio-Garcia, "LED Wristbands for Cell-Based Crowd Evacuation: An Adaptive Exit-Choice Guidance

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

System Architecture," Sensors, vol. 20, no. 21, Art. no. 21, Jan. 2020, doi: 10.3390/s20216038.

[18] Z. Vahedi, S. J. Seyyed Mahdavi Chabok, and G. Veisi, "Improving the quality of service indices of task allocation in mobile crowd sensing with fuzzy-based inverse Stackelberg game theory," Intell. Syst. Appl., vol. 20, p. 200291, Nov. 2023, doi: 10.1016/j.iswa.2023.200291.

[19] K. Reinecke and A. Bernstein, "Improving performance, perceived usability, and aesthetics with culturally adaptive user interfaces," ACM Trans Comput-Hum Interact, vol. 18, no. 2, p. 8:1-8:29, July 2011, doi: 10.1145/1970378.1970382.

[20] R. C. Brownson, Dissemination and Implementation Research in Health: Translating Science to Practice. Oxford University Press, 2018.

[21] Y. Liu, B. Guo, N. Li, Y. Ding, Z. Zhang, and Z. Yu, "CrowdTransfer: Enabling Crowd Knowledge Transfer in AIoT Community," IEEE Commun. Surv. Tutor, vol. 27, no. 2, pp. 1191–1237, Apr. 2025, doi: 10.1109/COMST.2024.3423319.

[22] M. Z. Shafiq, L. Ji, A. X. Liu, J. Pang, S. Venkataraman, and J. Wang, "Characterizing and Optimizing Cellular Network Performance During Crowded Events," IEEEACM Trans. Netw., vol. 24, no. 3, pp. 1308–1321, June 2016, doi: 10.1109/TNET.2016.2533612.

[23] H. Li, H. Wang, K. Wang, T. Qu, X. Ren, and J. Ma, "An energy efficient clustering algorithm based on density and fitness for mobile crowd-sensing network," J. Eng. Res., July 2024, doi: 10.1016/j.jer.2024.07.011.

[24] S. Mishra, V. Singh, A. Gupta, D. Bhattacharya, and A. Mudgal, "Adaptive traffic signal control for developing countries using fused parameters derived from crowd-source data," Transp. Lett., vol. 15, no. 4, pp. 296–307, Apr. 2023, doi: 10.1080/19427867.2022.2050493.

[25] G. Yang, G. Xie, J. Wang, X. He, L. Gao, and Y. Liu, "Adaptive task recommendation based on reinforcement learning in mobile crowd sensing," Appl. Intell., vol. 54, no. 2, pp. 2277–2299, Jan. 2024, doi: 10.1007/s10489-023-05247-3.