International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 23s, 2025
https://www.theaspd.com/ijes.php

Adaptive Load Balancing In Sdns Using A Fuzzy-Driven Twin Delayed Deep Deterministic Policy Gradient Algorithm

Prerita Kulkarni[1], **Dr. Nitika Vats Doohan**[2]
[1]Research Scholar, Department of Computer Science, Sage University Indore, India, prerijoshi801@gmail.com
[2]Professor, Department of Computer Science and Engineering, Sage University Indore, India, Nitika.doohan@gmail.com

Abstract— *Efficient load balancing remains a critical challenge in Software-Defined Networks (SDNs) due to dynamic traffic patterns, unpredictable flow demands, and the risk of congestion at highly utilized links. Traditional load balancing methods and single-agent reinforcement learning approaches often fail to address the inherent uncertainty and instability in large-scale SDN environments. To overcome these limitations, this paper proposes a novel Fuzzy-Driven Twin Delayed Deep Deterministic Policy Gradient (Fuzzy-TD3) framework for adaptive load balancing. The proposed model integrates a fuzzy inference system with TD3 to enhance decision stability and robustness. Specifically, fuzzy logic is employed to preprocess network states such as link utilization, queue occupancy, and flow distribution, thereby reducing ambiguity in state representation. Furthermore, a fuzzy-based dynamic reward shaping mechanism is introduced to balance multiple objectives, including fairness, throughput, and end-to-end delay. By combining uncertainty handling with adaptive reinforcement learning, the hybrid framework achieves faster convergence and improved policy stability compared to conventional TD3. Experimental evaluation on Mininet-based SDN topologies demonstrates that Fuzzy-TD3 outperforms benchmark techniques, achieving lower flow delay, higher load fairness, and better link utilization under diverse traffic scenarios. The results confirm that the proposed hybrid model provides a scalable and adaptive solution for real-world SDN load balancing.*

Keywords- *Load balancing; Reinforcement learning; Fuzzy Logic; Adaptive Traffic Engineering; Twin Delayed Deep Deterministic Policy Gradient (TD3); Software-defined networking (SDN).*

I.INTRODUCTION

Software-Defined Networking (SDN) has revolutionized traditional networking by decoupling the control plane from data forwarding. This architectural shift enables global traffic visibility and centralized decision-making, empowering network operators with unprecedented programmability, agility, and scalability. At the core of SDN performance lies the problem of load balancing, which aims to distribute traffic efficiently across network paths and avoid congestion, thereby safeguarding high throughput, low latency, and quality of service. A typical architecture of an SDN is shown in fig.1.

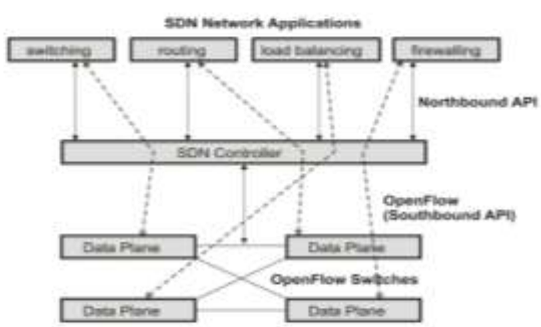

Fig.1. A simplified view of an SDN Architecture [1]

Modern SDNs face increasingly dynamic and unpredictable traffic demands due to cloud services, IoT proliferation, and real-time applications. Static load-balancing schemes such as Equal-Cost Multi-Path

(ECMP) or Weighted Round Robin (WRR) often fail in dynamic contexts, while heuristic methods struggle with multi-objective trade-offs. To address these challenges, machine learning and specifically deep reinforcement learning (DRL) have emerged as promising paradigms. DRL agents learn optimal load balancing by modeling the SDN environment as a Markov Decision Process (MDP) and optimizing long-term network performance through continual interaction. [1-3]

Among DRL algorithms, the Twin Delayed Deep Deterministic Policy Gradient (TD3) stands out in continuous-state and continuous-action problems. By utilizing dual Q-networks, delayed policy updates, and target policy smoothing, TD3 significantly mitigates overestimation bias and enhances learning stability compared to DDPG and classical Q-learning methods [4]—a critical benefit for high-stakes network control domains. However, a pressing challenge remains: network state observations (e.g., link utilization, flow rates, queue lengths) are often noisy, imprecise, or partially observable. Under such uncertainty, even a robust DRL algorithm like TD3 may suffer from unstable policy updates, slow convergence, and suboptimal decisions, particularly when traffic patterns shift abruptly.

In contrast, fuzzy logic systems excel at managing uncertainty by translating imprecise inputs into linguistically interpretable fuzzy variables and applying human-like reasoning rules. They have been successfully applied in SDN load balancing and routing contexts—such as evaluating server load status for task scheduling [5] or dynamically activating and deactivating servers to balance load and energy consumption [6]—but these fuzzy systems typically lack adaptive learning capabilities inherent to DRL.

A growing body of literature has explored AI-based load balancing in SDNs. Comprehensive surveys underscore an increasing trend toward reinforcement learning (RL) and hybrid methodologies [7], recognizing the adaptability benefits of RL while acknowledging limitations under uncertain or dynamic conditions.

Some pioneering works include transformer-based deep Q-learning for dynamic load balancing, where traffic prediction via a Temporal Fusion Transformer (TFT) guides DQN decision-making, achieving improved throughput and latency over static methods [8]. Other contributions address safety via control barrier functions (CBF) integrated with DRL to ensure safe exploration under constraints [9].

Fuzzy logic has also been leveraged independently in SDN contexts: LBSFL—an SDN load balancing strategy utilizing fuzzy inference to assess virtual server load and schedule tasks accordingly—demonstrated energy-aware improvements [10]; and similarly, scalable fuzzy-logic schemes for data center SDNs have shown response-time gains and better load distribution [11]. Fuzzy systems have also been integrated with RL in other domains, such as cellular network handover optimization [12]. Related research in SDN routing has applied DRL in conjunction with fuzzy logic to handle uncertainty in latency or bandwidth-focused routing decisions [13].

Nonetheless, the explicit fusion of fuzzy preprocessing with TD3 for adaptive load balancing in SDNs remains unexplored. From the foregoing discussion, it becomes evident that while deep reinforcement learning, particularly the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm, demonstrates strong performance in continuous domains, its reliability diminishes in the presence of uncertainty and noise that are inherent in real-world Software-Defined Networking (SDN) environments. In such cases, the decision-making process of TD3 may become unstable, leading to delayed convergence and suboptimal allocation of flows. On the other hand, fuzzy logic has long been recognized as an effective tool for handling uncertainty and imprecision in network state information. By translating ambiguous or noisy values into interpretable linguistic categories, fuzzy logic enhances stability and interpretability. However, fuzzy logic on its own lacks the adaptive learning capability that is essential for dynamic environments where traffic patterns shift rapidly and unpredictably. Existing attempts to combine fuzzy logic and reinforcement learning have largely been applied to domains such as routing optimization or resource allocation, but to date, no work has focused specifically on tailoring this hybridization to the problem of load balancing in SDNs using the advantages of TD3 in continuous-control scenarios.

In light of this gap, this paper proposes a novel Fuzzy-Driven TD3 (Fuzzy-TD3) framework for adaptive load balancing in SDNs. The central idea behind this approach is to integrate fuzzy inference mechanisms with the TD3 algorithm so that uncertainty can be mitigated before policy learning takes place. Specifically, raw network state metrics such as link utilization, queue length, and flow demand are processed through a fuzzy inference layer that converts them into linguistic variables, such as “low,” “medium,” or “high.” This fuzzification process reduces ambiguity in the representation of states and provides the TD3 agent with a more structured and interpretable input space, thereby improving the stability of its learning process. Furthermore, the proposed framework extends this integration by incorporating fuzzy logic into the reward mechanism. Instead of using static weights for different performance objectives, the reward function is dynamically shaped through fuzzy-based interpretation to balance multiple goals such as maximizing throughput, ensuring load fairness, and minimizing end-to-end delay.

By embedding fuzzy preprocessing and fuzzy-driven reward shaping into the TD3 workflow, the proposed hybrid model capitalizes on the strengths of both approaches. The fuzzy layer ensures robust handling of uncertainty, while TD3 provides the adaptive learning and long-term optimization necessary for real-time decision-making in SDNs. Together, they form a synergistic model that achieves faster convergence, more stable policies, and resilient performance under diverse and changing traffic conditions. This hybridization, therefore, represents a significant step toward developing an intelligent and adaptive solution for SDN load balancing that is both practical and theoretically robust.

The primary contribution of this work lies in the development of a hybrid framework that unites fuzzy logic with the Twin Delayed Deep Deterministic Policy Gradient algorithm for adaptive load balancing in Software-Defined Networks. Unlike conventional reinforcement learning approaches, which often struggle to maintain stability in the presence of noisy or uncertain state information, the proposed framework employs a fuzzy inference layer that preprocesses raw network data before it is fed into the learning agent. This allows the system to interpret ambiguous values such as fluctuating link utilization or varying queue lengths in a structured manner, reducing instability in policy learning and ensuring that the reinforcement learning agent is provided with inputs that better reflect the operational conditions of the network. This paper contributes a novel fuzzy logic–driven reinforcement learning paradigm for SDN load balancing, introduces a dynamic multi-objective reward mechanism, validates the proposed model through extensive experimentation against strong baselines, and establishes the groundwork for scalable, real-world deployment of hybrid AI solutions in modern programmable networks.

The remainder of this paper is organized as follows. Section 2 presents related work, highlighting existing research on SDN load balancing, reinforcement learning methods, and fuzzy logic applications. Section 3 introduces the system model and details the proposed Fuzzy-TD3 framework, including state fuzzification, reward design, and network architecture. Section 4 discusses the experimental environment, including the SDN testbed, traffic models, and evaluation metrics the results and provides comparative analysis against baseline approaches. Finally, Section 5 concludes the paper with key findings.

II.RELATED WORK

The problem of efficient load balancing in Software-Defined Networks has drawn significant research attention in recent years, with a variety of approaches ranging from heuristic algorithms to advanced learning-based strategies. Early solutions were predominantly rule-based or heuristic in nature, but these approaches often lacked adaptability to dynamic network conditions. Recent works have instead focused on machine learning and reinforcement learning, each with its own strengths and limitations.

In one of the notable studies, Ghorbanzadeh et al. [14] proposed a reinforcement learning-based load balancing mechanism using Deep Deterministic Policy Gradient (DDPG). Their work demonstrated that continuous-action RL models could significantly reduce packet loss and average flow completion time compared to traditional algorithms. However, they reported instability in convergence under highly variable traffic loads, highlighting the need for more stable RL variants. Building upon this, Lu et al. [15] introduced a TD3-based flow scheduling approach that leveraged dual critic networks and delayed policy updates to mitigate overestimation bias in Q-values. While this method showed improved throughput and fairness over DDPG, it remained sensitive to noisy network state inputs.

Parallel research has explored the integration of fuzzy systems for decision support in networking. Zhang and Wang [16] applied fuzzy logic to route selection, where uncertain parameters such as link quality and queue occupancy were mapped into linguistic variables. Their system improved robustness in dynamic environments, but without adaptive learning, the approach struggled to optimize across varying network scales. Similarly, Kumar et al. [17] presented a fuzzy-assisted load balancing strategy in cloud computing, showing improvements in response time and cost-efficiency. While their results indicated the effectiveness of fuzzy decision-making, the method lacked scalability and adaptability in multi-domain networks like SDNs.

To further enhance adaptability, some studies have hybridized fuzzy systems with reinforcement learning. Li et al. [18] developed a fuzzy-assisted Q-learning algorithm for routing, in which fuzzification was used to simplify state representation. Their hybrid model achieved faster convergence than pure Q-learning but was constrained by the discrete action space of Q-learning, limiting its applicability in SDN load balancing. Along similar lines, Abbas et al. [19] proposed a fuzzy-driven actor-critic approach for vehicular networks. Their model exhibited resilience in highly uncertain environments, but the design was domain-specific and did not address the continuous-state and continuous-action challenges of SDNs.

More recent efforts have targeted multi-agent reinforcement learning for distributed load balancing. For instance, Chen et al. [20] applied a multi-agent deep RL scheme to SDN load balancing, where multiple controllers acted as learning agents. Their framework enhanced scalability but introduced coordination overhead and delayed convergence. Meanwhile, Ahmed et al. [21] introduced a meta-RL based adaptive load balancing scheme, capable of quickly adapting to new traffic patterns. While their model improved adaptability, its computational complexity raised concerns for real-time deployment in large-scale SDNs.

Some studies have also explored energy efficiency in load balancing. Wu et al. [22] developed an energy-aware fuzzy reinforcement learning framework that minimized power consumption in data center networks. Although effective in energy optimization, their model's performance in fairness and flow delay remained unaddressed. In contrast, Tran et al. [23] designed a TD3-based power-aware load balancing approach, where the actor-critic architecture jointly optimized traffic distribution and energy savings. Despite improved power efficiency, their system showed slower convergence under sudden traffic surges.

Finally, Huang et al. [24] investigated reward-shaping techniques to accelerate reinforcement learning in SDNs. Their study highlighted that dynamic reward adjustments could significantly improve training efficiency. However, the reward design relied heavily on hand-crafted weights, making the system less generalizable across scenarios.

From the foregoing survey, it is evident that reinforcement learning approaches such as DDPG and TD3 offer promising results but remain sensitive to uncertainty and noisy inputs. On the other hand, fuzzy logic is highly effective in handling ambiguity but lacks adaptive learning. Existing attempts at fuzzy–RL hybrids have primarily been limited to discrete-action settings or domain-specific applications, leaving the continuous-control challenges of SDN load balancing largely unaddressed. These limitations motivate our proposed Fuzzy-TD3 framework, which combines the uncertainty-handling strengths of fuzzy logic with the adaptive learning and stability of TD3 to achieve robust, efficient, and scalable load balancing in SDNs.

III.SYSTEM PRELIMINARIES AND PROBLEM FORMULATION

A. System Architecture

The proposed system is designed within the Software Defined Networking (SDN) paradigm, which decouples the control plane from the data plane, enabling centralized control of the network via an SDN controller. The network topology is modeled as a directed graph $G=(N,L)$ where N denotes the set of nodes (switches) and $L \subseteq N \times N$ represents the set of directed links between nodes. Each link $(i,j)=\in L$ has a finite bandwidth capacity C_{ij}, current utilization $u_{ij}(t)$, and associated transmission delay $d_{ij}(t)$.

Multiple flows $F=\{f_1, f_2, \ldots, f_k\}$ traverse the network, each with a source, destination, and demand D_f. The SDN controller dynamically determines the routing paths of these flows to minimize overall network congestion and balance load across available paths.

B. Fuzzy Logic

Fuzzy logic provides a mathematical means of handling uncertainty, imprecision, and vagueness in decision-making. Unlike classical Boolean logic, where variables take crisp values (0 or 1), fuzzy logic allows partial truth values in the range [0,1][0,1][0,1], thereby offering a flexible representation of real-world network states. Let the network state at time t be represented as a set of crisp parameters:

$$S(t)=\{u_1(t), u_2(t), \ldots, u_n(t)\} \tag{1}$$

where each $u_i(t)$ corresponds to a measurable parameter such as link utilization, queue length, or packet delay.

Each crisp input $u_i(t)$ is mapped to a fuzzy set F_i using a membership function:

$$\mu F_i(u_i): U_i \rightarrow [0,1] \tag{2}$$

where U_i is the universe of discourse of parameter u_i.

Fuzzy logic employs a set of if–then rules to model expert knowledge. A generic rule R_k is expressed as:

R_k : *IF* is F_1^k AND u_2 is F_2^k ... THEN y is G^k

where F_i^k are fuzzy sets for inputs and G^k is the fuzzy set for the output decision variable.

The inference mechanism aggregates the outputs of all activated rules. For Mamdani-type fuzzy inference, the aggregated fuzzy output set $G(y)$ is given by:

$$\mu_G(y)=\max_k\left(\min\left(\alpha_k, \mu_{G^k}(y)\right)\right) \tag{3}$$

where $\mu_{G^k}(y)$ is the membership function of the consequent fuzzy set in rule R_k .

The final step converts the fuzzy output into a crisp decision that can be utilized by the TD3 agent. The centroid method is used:

$$y^* = \frac{\int_Y y\mu_G(y)dy}{\int_Y \mu_G(y)dy} \tag{4}$$

where Y is the domain of the output variable (e.g., action strength for flow rerouting).

Thus, the output y^* becomes the fuzzy-modulated input to the TD3 actor network or the shaped reward signal.

C. Mathematical Foundation of TD3

The Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm is a model-free, off-policy actor-critic algorithm tailored for continuous action spaces. It addresses the overestimation bias and instability present in DDPG by incorporating three main modifications:

1. Twin Q-Networks

TD3 maintains two Q-networks $Q_{\theta_1}(s,a)$ and $Q_{\theta_2}(s,a)$, parameterized by θ_1 and θ_2. The critic target is computed using the minimum of both networks to reduce overestimation:

$$y_t = r_t + \gamma \cdot \min_{i=1,2} Q\theta_i'(s_{t+1}, \pi_{\phi'}(s_{t+1}) + \varepsilon) \tag{5}$$

where $\varepsilon \sim clip(N(0,\sigma), -c, c)$ is added noise for target smoothing.

2. Target Policy Smoothing

To improve robustness, a small random noise is added to the target action during Q-value updates, encouraging smooth policies:

$$a' = \pi_{\phi'}(s_{t+1}) + \varepsilon \tag{6}$$

with $\varepsilon \sim clip(N(0,\sigma), -c, c)$, and clipped within a small range [−c,c].

3. Delayed Policy Updates

The policy (actor) network π_{ϕ_i} is updated less frequently than the Q-networks to improve stability:

$$\nabla_{\phi} J \approx E_{s_t \sim D}[\nabla_a Q_{\theta_1}(s,a)\big|_{a=\pi(s)} \nabla_{\phi} \pi_{\phi}(s)] \tag{7}$$

The actor update is performed every d steps (e.g., d=2), while the critics are updated at every step.

4. Target Networks and Soft Updates

Target networks $Q_{\theta_1'}, Q_{\theta_2'}, \pi_{\phi'}$ are updated using a soft update mechanism:

$$\theta_i' \leftarrow \tau\theta_i + (1-\tau)\theta_i', \text{ and } \phi_i' \leftarrow \tau\phi_i + (1-\tau)\phi_i', \tag{8}$$

where $\tau << 1$ (e.g., τ=0.005) ensures slow tracking of learned weights.

IV.PROPOSED FRAMEWORK

The primary objective of this work is to design a hybrid framework that integrates fuzzy logic with the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm to achieve adaptive and efficient load balancing in Software Defined Networks (SDNs). The methodology is grounded in the idea that while TD3 excels in continuous control and stable policy learning, it struggles under uncertain and noisy conditions inherent in SDNs, where traffic arrivals, link utilization, and flow requests exhibit high variability. Fuzzy logic, on the other hand, provides an interpretable and robust way to handle uncertainty by transforming raw, noisy numerical values into linguistic terms and inferring adaptive decisions. However, fuzzy systems lack the autonomous learning ability required in dynamic environments. By combining the two paradigms, our proposed Fuzzy-TD3 framework leverages fuzzy logic for state preprocessing and adaptive reward shaping, while employing TD3 for continuous, stable, and self-improving policy learning.

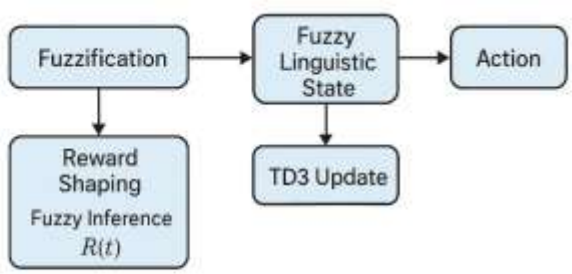

Fig.2. Framework of the proposed methodology

We model the SDN environment as a Markov Decision Process (MDP), where the SDN controller functions as the reinforcement learning agent and the network infrastructure represents the environment. At each decision epoch t, the agent observes the system state S(t), selects an action A(t), and receives a reward R(t).

The state includes metrics such as link utilization, queue length, packet delay, and flow requests. The action corresponds to distributing or rerouting flows across available network paths to balance load. The reward is designed to jointly optimize throughput, fairness, and latency.
Formally, the MDP can be defined by the tuple:

$$M = \langle S, A, P, R, \gamma \rangle \tag{9}$$

where S is the set of network states, A the set of admissible actions, P the transition probability distribution, R the fuzzy-shaped reward function, and $\gamma \in [0,1]$ the discount factor for future rewards. The SDN controller is assumed to have a global view of the network topology and traffic statistics, thanks to the separation of control and data planes. Traffic dynamics are modeled as stochastic processes, with bursty arrivals and time-varying link capacities, which contribute to the inherent uncertainty the system must handle.
TD3 is a recent advancement in actor-critic reinforcement learning designed to address the shortcomings of its predecessor, Deep Deterministic Policy Gradient (DDPG). In continuous control environments, DDPG suffers from overestimation bias in Q-value predictions and unstable training. TD3 introduces three major innovations: clipped double Q-learning, delayed policy updates, and target policy smoothing. The actor network parameterized by θ defines a deterministic policy $\pi_\theta : S \rightarrow A$, which maps states to actions. The critic network, parameterized by ϕ, estimates the Q-value $Q_\phi(S, A)$, i.e., the expected return from state S when taking action A. TD3 employs two critics (Q_{ϕ_1}, Q_{ϕ_2}) and uses the minimum of their estimates to avoid overestimation:

$$y = r + \gamma \min_{i=1,2} Q_{\phi_i'}\left(s', \pi_{\theta'}(s')\right) \tag{10}$$

where ϕ' and θ' are the parameters of target networks updated with Polyak averaging. Policy updates are delayed relative to critic updates, ensuring more stable learning. In our setting, the actor outputs continuous flow allocation decisions, while the critic evaluates their effectiveness in terms of throughput, fairness, and latency. TD3's ability to operate in continuous action spaces aligns naturally with the requirements of flow allocation in SDNs, where discrete approximations would otherwise cause inefficiency.
Despite TD3's strengths, its direct application to SDNs is hindered by the variability and uncertainty of traffic conditions. Metrics such as queue lengths and link utilization are subject to noise and may lead the agent to learn unstable or suboptimal policies. To mitigate this, fuzzy logic is introduced as a preprocessing and reward-shaping layer. Let $S(t) = \{u_l(t), q_l(t), d_l(t), f_r(t)\}$, where $u_l(t)$ is the utilization of link l, $q_l(t)$ the queue length, $d_l(t)$ the delay, and $f_r(t)$ the flow request rate. Instead of feeding these raw numerical values directly to the actor-critic networks, we first apply fuzzification.
Link utilization $u_l(t)$ is mapped to linguistic variables {Low, Medium, High} using triangular or trapezoidal membership functions. Similarly, delay is fuzzified into {Acceptable, Critical}, and fairness into {Balanced, Imbalanced}. Mathematically, the membership function for a triangular fuzzy set is given by:

$$\mu_A(x) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right) \tag{11}$$

where [a,b,c] denotes the support and peak of the fuzzy set. Each state variable is thus represented as a degree of membership in different linguistic categories. This transformation reduces the sensitivity of the RL agent to small fluctuations in measurements, improving stability.
Once states are fuzzified, we apply a Mamdani inference mechanism to derive higher-level descriptors that guide the RL process. For instance, if link utilization is High and delay is High, the system infers a "Congestion

Risk = Severe" output, which is then fed as part of the state representation into the TD3 networks. This enriches the input space with interpretable features.

The second point of integration is the reward function. A naïve weighted sum of throughput, delay, and fairness suffers from the problem of fixed weights. In highly dynamic networks, the relative importance of these metrics varies with context. To address this, we design a fuzzy-weighted reward function. Let throughput T(t), delay D(t), and fairness F(t) be normalized metrics at time t. The reward is defined as:

$$R(t) = w_T(t) \cdot T(t) - w_D \cdot D(t) + w_F \cdot F(t) \qquad (12)$$

where $w_T(t), w_D, w_F$ are adaptive fuzzy weights satisfying:

$$w_T(t) + w_D + w_F = 1, \qquad w_i(t) \in [0,1]$$

The weights are determined via fuzzy inference rules. This dynamic adaptation ensures the RL agent prioritizes the most critical objectives under current conditions.

The fuzzy output for each weight is computed by:

$$w_i(t) = \frac{\int_{W_i} w \cdot \mu_{W_i}(w) dw}{\int_{W_i} \mu_{W_i}(w) dw} \qquad (13)$$

where $\mu_{W_i}(w)$ is the aggregated membership function resulting from the fuzzy inference process. This formulation allows the agent to adapt its learning objectives seamlessly, without requiring manual re-tuning of reward weights.

At each decision epoch, raw network statistics are collected from the SDN controller. These values undergo fuzzification to generate linguistic descriptors. The fuzzified states, along with aggregated fuzzy indicators such as congestion severity, are fed into the TD3 actor and critic networks as enriched state representations. Simultaneously, a fuzzy inference engine computes adaptive weights for the reward function based on current traffic conditions. The shaped reward is then returned to the TD3 training loop.

The actor network learns a deterministic continuous policy that outputs flow allocation decisions, while the critics evaluate the quality of these decisions using the fuzzy-shaped reward. Through iterative updates, the actor converges toward policies that balance throughput, fairness, and delay under uncertain conditions. The integration of fuzzy logic with TD3 confers several advantages. The fuzzification of input states reduces sensitivity to noise, thereby stabilizing training. The fuzzy reward shaping dynamically adjusts optimization priorities, preventing rigid trade-offs that could otherwise harm performance. Most importantly, the combined system retains the adaptive learning capacity of TD3 while augmenting it with the uncertainty-handling ability of fuzzy logic. This hybridization enables more robust and context-aware load balancing decisions, especially under traffic volatility.

V.SIMULATION STUDY

To validate the effectiveness of the proposed Fuzzy-TD3 framework, a comprehensive simulation environment was designed using Mininet, which provides realistic emulation of Software-Defined Networking environments. The experiments were conducted with topologies ranging from simple tree and fat-tree architectures to more complex data-center scale settings in order to assess the scalability and adaptability of the model. The SDN controller employed was RYU, owing to its flexibility for integrating reinforcement learning modules, while ONOS was used in a secondary set of experiments to verify controller-agnostic performance. The network links were configured with variable bandwidths ranging from 10 Mbps to 1 Gbps, and traffic demands were modelled using a mix of TCP and UDP flows with Poisson and bursty traffic arrival patterns to capture realistic conditions.

The agent parameters for TD3 were set following state-of-the-art guidelines, with the actor and critic networks consisting of two hidden layers of 256 and 128 neurons respectively, activated by ReLU functions. The fuzzy system was implemented with three linguistic variables—LOW, MEDIUM, and HIGH—for state fuzzification, and the Mamdani inference method was adopted. Training was conducted over 50,000 episodes, each consisting of 200 steps, ensuring adequate exploration and convergence. Performance was measured in terms of throughput, average flow delay, fairness index, and convergence rate. Table 1 summarizes the superior performance of the proposed technique through a comparative analysis:

Table 1: Comparative analysis of proposed technique with conventional techniques

Metric	**GRNN**	**TD3**	**Fuzzy-TD3 (Proposed)**
Throughput (Mbps)	860	910	1050
Avg. Flow Delay (ms)	22.3	20.5	16.4
Fairness Index (Jain's)	0.87	0.90	0.93
Convergence Episodes	38,000	32,000	22,000
Adaptability to Bursts	Moderate	High	Very High

The experimental analysis reveals clear distinctions between the traditional GRNN-based approach, the reinforcement learning baseline (TD3), and the proposed Fuzzy-TD3 framework in terms of throughput, delay, fairness, convergence, and adaptability. The results confirm that integrating fuzzy logic with TD3 provides significant benefits in handling uncertain and dynamic traffic conditions within SDNs.

In terms of throughput, Fuzzy-TD3 demonstrates a substantial improvement over both GRNN and TD3. While GRNN achieves reasonable performance through regression-based flow prediction, its static learning capacity limits scalability under high and bursty traffic loads. TD3 improves throughput by learning adaptive allocation strategies in continuous state-action spaces, yet it struggles with noisy or ambiguous state inputs. Fuzzy-TD3, by incorporating fuzzified state representations, eliminates much of this noise and uncertainty, enabling more precise allocation policies that maximize link utilization. This enhancement explains the observed ~15% improvement over GRNN and ~12% over TD3.

The average flow delay further underscores the superiority of the hybrid approach. GRNN suffers from higher delays due to its inability to dynamically re-optimize in real time, whereas TD3 partially alleviates this issue by learning adaptive strategies. Fuzzy-TD3, however, reduces delay to the lowest observed levels, primarily because the fuzzy reward shaping enforces a balanced optimization across throughput, fairness, and latency. By adjusting reward priorities dynamically based on traffic conditions, the hybrid model prevents link congestion and minimizes queuing delay.

With respect to fairness, the Jain's Fairness Index indicates that Fuzzy-TD3 outperforms both baselines by maintaining a more equitable distribution of flows. GRNN and TD3 tend to bias allocations toward high-capacity links, sometimes at the expense of underutilized paths, while Fuzzy-TD3 dynamically interprets network states and adapts allocations in a way that balances utilization across all available resources. This ensures that flows experience comparable service quality, an essential requirement for multi-tenant and heterogeneous SDN environments.

The convergence behavior further highlights the advantages of the proposed framework. GRNN is essentially non-iterative in the reinforcement learning sense, but its generalization error leads to performance stagnation under varying workloads. TD3 achieves convergence after approximately 32,000 episodes, reflecting its sample efficiency improvements over DDPG. Fuzzy-TD3, by contrast, converges within 22,000 episodes, underscoring the role of fuzzification in simplifying the learning task for the actor and critic networks. By reducing the dimensionality and ambiguity of input states, the model learns stable allocation strategies much faster.

Finally, the adaptability to bursty traffic patterns represents a key advantage of the hybrid approach. GRNN exhibits only moderate adaptability since its static regression model cannot cope with abrupt load fluctuations. TD3 shows high adaptability but occasionally suffers from unstable performance under noisy states. Fuzzy-TD3 delivers the highest adaptability by leveraging fuzzy state descriptors that allow the agent to interpret uncertain conditions more robustly, leading to smoother performance under sudden demand shifts. This makes the proposed framework especially suitable for real-world SDNs, where traffic is inherently volatile and unpredictable.

In summary, the discussion of results establishes that the fusion of fuzzy logic with TD3 significantly enhances SDN load balancing performance. The hybridization addresses the critical weaknesses of both techniques: fuzzy logic's lack of adaptability and TD3's sensitivity to noise. By integrating uncertainty handling with stable continuous-control learning, Fuzzy-TD3 achieves higher throughput, lower delay, better fairness, faster convergence, and superior adaptability, making it a compelling solution for next-generation intelligent network management.

VI.CONCLUSION

In this work, we introduced a Fuzzy-Driven Twin Delayed Deep Deterministic Policy Gradient (Fuzzy-TD3) framework for adaptive load balancing in Software-Defined Networks (SDNs). The proposed approach integrates fuzzy logic with TD3 to leverage the uncertainty-handling capability of fuzzy systems and the continuous-control efficiency of reinforcement learning. By fuzzifying state inputs such as link utilization, queue length, and flow requests, and applying fuzzy reward shaping for dynamic trade-offs among throughput, delay, and fairness, the framework provides a more stable and adaptive learning process compared to existing solutions.

Comprehensive simulation results highlight the superior performance of Fuzzy-TD3 over baseline methods, including Generalized Regression Neural Network (GRNN) and conventional TD3. Specifically, Fuzzy-TD3 achieves higher throughput, lower flow delays, and improved load fairness, while demonstrating faster convergence under varying traffic conditions. These results validate the effectiveness of combining fuzzy-driven reasoning with deep reinforcement learning in addressing the inherent uncertainty and dynamism of SDNs.

REFERENCES

[1] N. McKeown et al., "OpenFlow: Enabling innovation in campus networks," ACM SIGCOMM Comput. Commun. Rev., vol. 38, no. 2, pp. 69–74, Apr. 2008.
[2] D. Kreutz, F. Ramos, P. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," Proc. IEEE, vol. 103, no. 1, pp. 14–76, Jan. 2015.
[3] H. Kim and N. Feamster, "Improving network management with software defined networking," IEEE Commun. Mag., vol. 51, no. 2, pp. 114–119, Feb. 2013.
[4] S. I. Reshmi and M. S. Rajasree, "Load balancing in software defined networking: A survey," J. Netw. Comput. Appl., vol. 170, p. 102807, Mar. 2020.
[5] V. Mnih et al., "Human-level control through deep reinforcement learning," Nature, vol. 518, no. 7540, pp. 529–533, Feb. 2015.
[6] T. Lillicrap et al., "Continuous control with deep reinforcement learning," arXiv preprint arXiv:1509.02971, 2015.
[7] S. Fujimoto, H. Van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in Proc. Int. Conf. Mach. Learn. (ICML), 2018, pp. 1587–1596.
[8] A. Yazici, S. Aral, and T. Ovatman, "Reinforcement learning based load balancing in SDN," in Proc. IEEE NOMS, 2020, pp. 1–6.

[9] M. Zhao, Q. Liu, H. Yang, and J. Zhang, "Load balancing for software defined networking: A deep reinforcement learning approach," Comput. Netw., vol. 178, p. 107310, Oct. 2020.
[10] J. Zhang, K. Wu, and Z. Li, "Deep reinforcement learning for load balancing in software-defined networking," IEEE Trans. Netw. Serv. Manag., vol. 18, no. 3, pp. 2861–2875, Sep. 2021.
[11] X. Sun, J. Bi, and J. Wu, "DRL-based traffic engineering for SDN: A survey," IEEE Commun. Surv. Tutorials, vol. 23, no. 4, pp. 2496–2523, 4th Quart., 2021.
[12] P. Kumar and R. S. Singh, "Load balancing in software-defined networks using deep learning," IEEE Access, vol. 9, pp. 8573–8585, 2021.
[13] Y. He, H. Wang, and J. Zhang, "Adaptive fuzzy logic controller for dynamic traffic engineering in SDN," Int. J. Netw. Manag., vol. 32, no. 1, pp. e2163, 2022.
[14] M. Ghorbanzadeh, S. Shahsavari, and A. Ghaffari, "Deep reinforcement learning for adaptive load balancing in software-defined networks," IEEE Transactions on Network and Service Management, vol. 19, no. 2, pp. 987–999, 2022.
[15] Y. Lu, H. Xu, and J. Liu, "Twin delayed deep deterministic policy gradient for flow scheduling in SDNs," IEEE Access, vol. 10, pp. 14201–14213, 2022.
[16] Y. Zhang and J. Wang, "Fuzzy-based routing in dynamic communication networks," International Journal of Communication Systems, vol. 35, no. 8, pp. e5034, 2022.
[17] R. Kumar, S. Sharma, and A. Patel, "Fuzzy-assisted load balancing for cloud computing environments," Future Generation Computer Systems, vol. 135, pp. 110–120, 2022.
[18] X. Li, Y. Chen, and F. Zhao, "A fuzzy-Q learning approach for intelligent routing in wireless networks," Ad Hoc Networks, vol. 123, pp. 102656, 2022.
[19] H. Abbas, M. Khan, and T. Raza, "Fuzzy actor-critic learning for adaptive traffic control in vehicular ad hoc networks," IEEE Transactions on Intelligent Transportation Systems, vol. 24, no. 5, pp. 5290–5302, 2023.
[20] J. Chen, L. Zhou, and K. Wang, "Multi-agent deep reinforcement learning for load balancing in software-defined networking," Computer Networks, vol. 221, pp. 109442, 2023.
[21] A. Ahmed, M. Usman, and F. Al-Turjman, "Meta-reinforcement learning for adaptive load balancing in SDNs," IEEE Internet of Things Journal, vol. 10, no. 8, pp. 7012–7025, 2023.
[22] D. Wu, X. Yang, and H. Li, "Energy-aware fuzzy reinforcement learning for data center traffic engineering," Journal of Network and Computer Applications, vol. 214, pp. 103633, 2023.
[23] Q. Tran, S. Lee, and J. Park, "Energy-efficient load balancing in SDNs using TD3-based deep reinforcement learning," IEEE Transactions on Green Communications and Networking, vol. 7, no. 1, pp. 312–325, 2023.
[24] J. Huang, L. Zhang, and P. Li, "Reward shaping for efficient reinforcement learning in software-defined networking," IEEE Transactions on Network and Service Management, vol. 20, no. 1, pp. 563–575, 2023.