Power System Protection and Control
ISSN:1674-3415

# ENHANCING WORKLOAD PREDICTION ACCURACY IN SERVERLESS EDGE COMPUTING USING A LEARNING-DRIVEN HYBRID ADAPTIVE MODEL

**Bijan Moloudi Maybodi[1], Mohammadreza Mollahoseini-Ardakani*[2], Kamal Mirzaie[3]**

[1]Department of Computer Engineering, May.C., Islamic Azad University, Maybod, Iran
Bijan.moloudi@iau.ac.ir
[2]Department of Computer Engineering, May.C., Islamic Azad University, Maybod, Iran
mr.mollahoseini@iau.ac.ir
[3]Department of Computer Engineering, May.C., Islamic Azad University, Maybod, Iran
kamal.mirzaie@iau.ac.ir

Received: 12/06/2026 Revised: 19/07/2026 Accepted: 14/08/2026

***ABSTRACT*:**

Serverless edge computing has emerged as a critical infrastructure for next-generation distributed systems; however, it faces fundamental challenges in resource management due to highly dynamic workloads, bursty request patterns, and stringent edge hardware constraints. Existing approaches—either purely relying on LSTM-based time-series forecasting or adopting reactive Deep Reinforcement Learning (Deep RL) strategies—fail to achieve an optimal trade-off between Quality of Service (QoS) and resource efficiency. To address this limitation, this paper proposes HybridEdge, a hybrid and adaptive framework that tightly integrates multi-step workload forecasting with adaptive reinforcement learning based on Proximal Policy Optimization (PPO) within a unified control architecture. HybridEdge explicitly incorporates uncertainty quantification via Monte Carlo Dropout and online concept drift detection using the Page–Hinkley test, enabling self-regulating and risk-aware resource provisioning policies. Extensive evaluations conducted on real-world Azure Functions traces demonstrate that HybridEdge achieves a 24.4% reduction in over-provisioning compared to DRL-Only and a 66% reduction in under-provisioning relative to LSTM-Only, while simultaneously maintaining 91.42% user satisfaction and an F1-score of 98.76%. These results confirm that integrating uncertainty-aware multi-step forecasting with adaptive reinforcement learning provides a scalable and effective solution for intelligent resource orchestration in Serverless Edge–Cloud environments, paving the way for the development of autonomous and self-adaptive systems in this domain.

***Keywords***: *Serverless Edge Computing; Adaptive Resource Management; Deep Reinforcement Learning; Workload Forecasting; Quality of Service (QoS); LSTM; HybridEdge*

## INTRODUCTION

The rapid evolution of digital ecosystems—driven by the convergence of intelligent sensing, ubiquitous connectivity, and the proliferation of ultra–latency-sensitive applications—has fundamentally challenged the traditional paradigm of centralized cloud computing. In such environments, transferring massive volumes of raw data to distant cloud data centers is not only inefficient in terms of latency and energy consumption but also incompatible with real-time responsiveness and context-aware decision-making requirements. Consequently, modern computing architectures have shifted toward distributed and cooperative paradigms, in which end, edge, and cloud layers operate as an integrated computational continuum. Recent surveys explicitly demonstrate that deep learning–enabled systems increasingly rely on end–edge–cloud collaborative computing to simultaneously achieve low latency, energy efficiency, and contextual adaptability, particularly in environments characterized by high dynamics and stringent resource constraints[1].

Despite this architectural transition, the practical deployment of distributed edge systems introduces profound structural and operational complexities. Workloads in such systems are highly heterogeneous, non-stationary, and often bursty; available computational and communication resources fluctuate dynamically; and service providers must continuously balance guaranteed Quality of Service (QoS) against infrastructure cost. These tensions are significantly amplified in serverless edge computing, where short-lived microfunctions, sudden invocation spikes, and cold-start effects render workload patterns inherently unpredictable. Under these conditions, classical


**DOI:** 10.46121/pspc.54.3.47

autoscaling mechanisms—typically reactive, threshold-based, and dependent on instantaneous system states—prove insufficient to meet the performance requirements of modern edge environments.

In response, a growing body of research has focused on designing intelligent multi-layer resource management frameworks aimed at coordinating decisions across different system layers. These studies emphasize the necessity of jointly optimizing task scheduling, cache placement, and latency-aware resource orchestration, as isolated or locally optimal decisions often lead to suboptimal system-wide performance in complex edge environments[2]. Within this context, Deep Reinforcement Learning (DRL) has emerged as a powerful tool for dynamic decision-making under uncertainty. Its integration with federated or collaborative learning paradigms has further enabled decentralized control, partial observability handling, and device heterogeneity management[3, 4]. Empirical evidence from federated edge clusters demonstrates that RL agents can adapt their policies online in response to local resource constraints, workload fluctuations, and asynchronous model updates. Similarly, in dense Internet of Things (IoT) environments, federated RL–based resource allocation has been shown to significantly reduce latency and improve energy efficiency[5, 6].

Nevertheless, despite these advances, a critical examination of the literature reveals persistent structural limitations. First, DRL agents that rely solely on instantaneous system states and lack predictive awareness tend to exhibit unstable behavior when faced with abrupt and unforeseen workload variations. This issue becomes a major performance bottleneck in serverless edge computing, where anticipating function invocations and mitigating cold-start penalties are essential[7]. Second, purely prediction-driven autoscaling approaches, including statistical models such as ARIMA and even classical deep learning predictors, while capable of capturing temporal patterns, generally fail to account for real-time system dynamics and, more importantly, lack explicit uncertainty quantification—an omission that can lead to high-risk decisions in resource-constrained environments[8, 9]. Third, existing hybrid frameworks that attempt to combine forecasting and learning often treat predictions as static, exogenous inputs, neglecting fundamental phenomena such as concept drift, non-stationarity, and uncertainty propagation, which are intrinsic to real-world IoT and edge workloads[10, 11].

Recent progress in time-series forecasting—particularly through Long Short-Term Memory (LSTM) networks—has opened new avenues for modeling complex temporal dependencies in noisy and bursty data streams. LSTM remains one of the most effective architectures for extracting long-term dependencies from non-stationary sequences[12]. Moreover, integrating LSTM with advanced optimization strategies, such as swarm intelligence and Lyapunov-based adaptive control, has demonstrated simultaneous improvements in prediction accuracy, model stability, and energy efficiency in fog and industrial IoT environments[13, 14]. These findings clearly indicate that effective workload forecasting at the edge requires models that are not only accurate but also uncertainty-aware and adaptive to regime shifts.

Complementarily, the literature on computation offloading and resource scheduling highlights the strong suitability of Deep Reinforcement Learning, particularly actor–critic and Q-learning–based algorithms, for managing large, combinatorial, and context-dependent action spaces—characteristics inherent to edge computing systems[15-17]. Systematic reviews confirm that DRL-based solutions consistently outperform heuristic approaches in terms of scalability and robustness under dynamic IoT workloads and multi-tenant environments (Hortelano et al., 2023; Ismail et al., 2025). However, these studies also stress that DRL alone cannot guarantee resilient performance under uncertainty, partial observability, and rapid workload drift—conditions that are not exceptions but defining characteristics of serverless platforms[18, 19]. Collectively, these observations point to a fundamental methodological gap: the absence of an integrated, uncertainty-aware, and drift-adaptive framework that tightly couples predictive intelligence with adaptive policy learning. This gap is particularly critical in serverless edge computing, where the ephemeral nature of microfunctions, fine-grained elasticity of resources, and strong dependence of QoS on anticipatory decisions render purely reactive control strategies inadequate.

Motivated by this gap, this paper introduces HybridEdge, an end-to-end adaptive framework for intelligent resource orchestration in serverless edge computing environments. HybridEdge tightly integrates LSTM-based multi-step workload forecasting with policy learning via Proximal Policy Optimization (PPO), enhanced through explicit uncertainty quantification and online concept drift detection. In contrast to existing approaches, forecasting and decision-making in HybridEdge are designed as mutually reinforcing processes: predictive outputs guide the control actions of the RL agent, while the agent's behavior dynamically regulates prediction confidence and online retraining strategies. By leveraging Monte Carlo Dropout for uncertainty estimation, the Page–Hinkley test for regime change detection, and a softly constrained reward function that balances latency, resource waste,


**DOI:** 10.46121/pspc.54.3.47

and energy consumption, HybridEdge operates as a proactive, risk-aware, and self-correcting system. Such capabilities are not merely advantageous but essential for the next generation of serverless edge–cloud systems.

## RELATED WORKS

### 2.1. Serverless Edge Computing and Cooperative Edge–Cloud Architectures

Edge computing, as a distributed computing paradigm, aims to bring computation closer to data sources in order to reduce latency, optimize bandwidth utilization, and support time-sensitive applications in IoT and industrial environments[1, 10].However, stringent resource constraints, highly volatile workloads, and real-time responsiveness requirements at edge nodes limit the effectiveness of relying solely on the edge layer. To overcome these limitations, cooperative Edge–Cloud architectures have been proposed, enabling dynamic and intelligent workload distribution across edge and cloud layers[2, 20].

As a further evolutionary step, Serverless Edge Computing has emerged as an advanced model in which infrastructure complexity is abstracted away from developers, and resource provisioning is performed on-demand through lightweight functions[11]. While this paradigm offers high elasticity, reduced operational overhead, and simplified application development, the inherently bursty and non-stationary nature of serverless workloads significantly undermines the effectiveness of reactive and threshold-based resource management strategies, making QoS guarantees increasingly challenging[9]. These characteristics highlight the necessity of intelligent, learning-driven, and predictive approaches for resource management and task allocation in serverless edge environments.

### 2.2. Workload Forecasting and Time-Series Models

Accurate workload forecasting has become a critical component for ensuring stable and scalable performance, particularly in edge and serverless environments. Classical statistical methods and early machine learning approaches, although effective in traditional cloud settings, fail to cope with the high variability and abrupt behavioral changes of users in Edge and Serverless scenarios[8, 21, 22].

In this context, recurrent neural networks (RNNs), and especially Long Short-Term Memory (LSTM) networks, have been widely adopted due to their ability to capture long-term temporal dependencies and robustness to noisy data[12]. Recent studies demonstrate that LSTM and its advanced variants, such as Bi-LSTM, achieve superior accuracy in forecasting request arrival rates and resource consumption patterns[17, 23]. Nevertheless, most existing forecasting models operate in isolation, with their outputs injected into static or rule-based controllers. This decoupled design renders systems fragile under uncertainty and regime shifts in workload dynamics[10, 11]. Complementary research shows that integrating LSTM with optimization techniques such as Particle Swarm Optimization (PSO) or Lyapunov-based optimization can further improve forecasting accuracy and energy efficiency[13, 20]. Despite these advances, the absence of uncertainty-aware mechanisms and online adaptation remains a critical limitation in serverless edge environments.

### 2.3. Deep Reinforcement Learning for Resource Allocation

Deep Reinforcement Learning (DRL) has gained significant attention as an effective approach for resource allocation and task scheduling in distributed and serverless systems. By interacting directly with the environment, DRL agents can learn policies that jointly optimize latency, energy consumption, and operational cost[15, 16]. Actor–Critic and Q-learning-based algorithms, in particular, demonstrate strong performance in managing large and complex action spaces inherent to edge computing environments[2, 24].

Despite these advantages, DRL-based approaches predominantly rely on instantaneous state observations and lack explicit predictive awareness of future workload trends. As a result, under highly volatile and bursty workloads, purely reactive DRL policies often suffer from reduced system stability and resource inefficiency[25].

### 2.4. Federated Learning and Distributed Resource Management

Driven by privacy concerns and communication overhead, Federated Learning (FL) has emerged as a promising paradigm for training models without transferring raw data to centralized servers[26]. In edge environments, FL enables the construction of scalable and accurate models while preserving data locality[6, 18].

Recent studies indicate that combining DRL with Federated Learning can facilitate adaptive resource allocation policies without violating privacy constraints[4, 5, 27]. However, most of these approaches either focus primarily


**DOI:** 10.46121/pspc.54.3.47

on RL-based decision-making or implicitly incorporate workload dynamics, lacking a comprehensive framework for explicit uncertainty management and concept drift handling.

### 2.5. Hybrid and Adaptive Forecasting–DRL Models

To overcome the limitations of purely predictive or purely reactive approaches, recent research has shifted toward hybrid and adaptive models that integrate LSTM-based workload forecasting with the adaptive decision-making capabilities of DRL within a unified framework[13, 14]. Empirical evidence suggests that such hybrid architectures can simultaneously reduce latency and energy consumption while improving resource utilization efficiency in edge environments[3, 28].

Nonetheless, the majority of existing hybrid models treat forecasts as deterministic inputs and largely neglect uncertainty-aware decision-making and online adaptation mechanisms required to handle dynamic workload changes[10, 11]. This shortcoming compromises system robustness and resilience in real-world serverless edge deployments.

### 2.6. Research Gap and the Positioning of HybridEdge

The synthesis of existing literature reveals that, despite significant progress in workload forecasting, deep reinforcement learning, and federated learning, a fully integrated, adaptive, and uncertainty-aware framework for Serverless Edge Computing is still lacking. The primary research gaps can be summarized as follows:

1. The absence of models that integrate LSTM-based workload forecasting with explicit uncertainty quantification into the DRL decision-making process;
2. The lack of online adaptation mechanisms to address dynamic workload shifts and concept drift;
3. The scarcity of comprehensive evaluations using real-world, highly bursty serverless workloads in edge scenarios.

To address these gaps, this paper introduces HybridEdge, a unified framework that tightly couples LSTM-based workload forecasting with DRL-based (PPO) resource allocation. By incorporating uncertainty-aware forecasting and online adaptation mechanisms, HybridEdge simultaneously enhances prediction accuracy, resource allocation efficiency, and system stability in Serverless Edge–Cloud environments.

## METHODOLOGY

### 3.1. System Objective

The primary objective of this study is to design and implement a hybrid Forecast-Guided Reinforcement Learning framework for intelligent and adaptive resource management in Edge–Cloud environments. Leveraging real-world workload traces, multi-step demand forecasting, uncertainty estimation, and operational cost constraints, the proposed system derives optimal resource allocation policies in dynamic and non-stationary settings. The framework is explicitly designed to minimize resource wastage caused by over-provisioning, prevent under-provisioning that leads to QoS degradation, and improve overall system efficiency. By integrating predictive intelligence with adaptive control, the system aims to support robust and informed decision-making under uncertainty in highly dynamic serverless edge environments.

### 3.2. Research Workflow and Implementation Phases

The implementation and evaluation of the proposed system follow a structured and coherent pipeline, where each phase systematically feeds into the next.

**Phase 1: Workload Data Collection and Preprocessing**

In the first phase, real-world workload traces were extracted from Azure Functions and used as the empirical foundation of the study. After selecting relevant numerical features, invalid entries, negative values, and noisy samples were removed to ensure a stable and reliable time series. To homogenize data scales and facilitate efficient model training, Min–Max normalization was applied within the range $[0, 1]$. Subsequently, the dataset was split into training (70%) and testing (30%) subsets while preserving temporal order. This phase established a clean and consistent data pipeline for training both the forecasting models and the reinforcement learning agent.

**Phase 2: Time-Series Forecasting Model Development**

In this phase, a suite of time-series forecasting models was developed to generate multi-step workload predictions. Deep learning-based models included LSTM, GRU, and Bi-LSTM, while classical statistical models such as Moving Average and ARIMA were implemented as baseline benchmarks. Forecasting was conducted using a


**DOI:** 10.46121/pspc.54.3.47

rolling forecast strategy to realistically emulate operational conditions. Furthermore, predictive uncertainty was quantified using Monte Carlo Dropout, enabling probabilistic forecasting. To address abrupt workload pattern changes, a Page–Hinkley test was employed for online concept drift detection, triggering incremental retraining of the forecasting models when necessary. The output of this phase consisted of multi-step forecasts augmented with uncertainty estimates, which were directly consumed by the control layer.

**Phase 3: Resource Control Simulation Environment Design**

To enable realistic training of the reinforcement learning agent, two custom simulation environments were developed using OpenAI Gym:

(i) EdgeEnv for single-node edge scenarios, and

(ii) MultiEdgeEnv for multi-node cooperative edge deployments.

These environments expose the agent to current system states, future workload forecasts, and associated uncertainty metrics as part of the state representation. The action space allows the agent to increase, decrease, or maintain allocated resource levels. The reward function was carefully designed to balance penalties for under-provisioning and over-provisioning, incorporate uncertainty-aware risk management, and enforce operational constraints related to energy consumption and bandwidth usage. This phase plays a crucial role in bridging the gap between simulation and real-world Edge–Cloud conditions.

**Phase 4: Reinforcement Learning Agent Training**

In this phase, the reinforcement learning agent was trained using the Proximal Policy Optimization (PPO) algorithm. To enhance learning efficiency, a custom feature extractor was implemented to jointly process real-time observations, multi-step forecasts, and uncertainty indicators. Training was performed concurrently with environment simulation and continuous forecast updates. An OnlineLSTMCallback was employed to adaptively fine-tune the forecasting model during RL training, ensuring alignment with evolving workload dynamics. Additionally, an EntropyTargetCallback was used to automatically regulate the exploration–exploitation trade-off. The outcome of this phase is an adaptive control policy capable of real-time resource allocation under uncertainty.

**Phase 5: System Evaluation and Performance Analysis**

After training, system performance was evaluated through rollouts on the test dataset. Forecasting accuracy was assessed using MAE, RMSE, MAPE, and $R^2$metrics. At the resource control level, Over-Provisioning, Under-Provisioning, and Total Reward were computed. QoS-oriented decision-making performance was further evaluated using Accuracy, Precision, Recall, and F1-score. The proposed framework was benchmarked against rule-based policies, prediction-only approaches, and alternative hybrid methods to quantify its relative effectiveness.

**Phase 6: Visualization and Result Documentation**

In the final phase, results were analyzed both statistically and visually. Generated outputs included resource allocation trajectories, over- and under-provisioning trends, cumulative reward curves and their components, decision confusion matrices, and temporal uncertainty dynamics.

Structured outputs were also exported in CSV format to enable fine-grained comparison across models. This phase ensures in-depth analysis, transparent reporting, and reproducibility of results.

### 3.3. System Workflow Summary

Following the above methodology, the proposed system successfully:

- Preprocessed and standardized real-world workload data;
- Generated accurate multi-step workload forecasts with quantified uncertainty;
- Constructed realistic resource control simulation environments;
- Trained an RL agent guided by predictive and environmental information;
- Evaluated the learned policy against state-of-the-art baselines.


**DOI:** 10.46121/pspc.54.3.47

Power System Protection and Control
ISSN:1674-3415

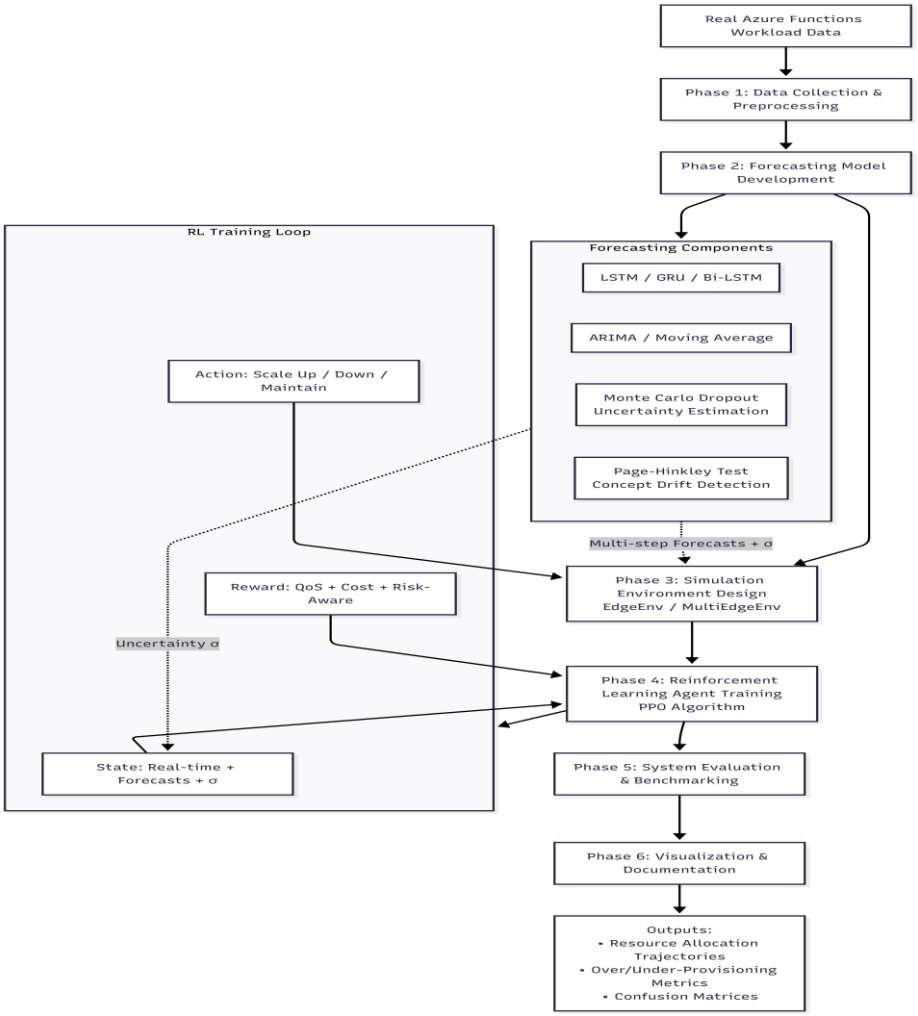


**Figure 1. Systematic Workflow**

## RESULTS

This section presents the empirical findings obtained from the implementation and evaluation of the proposed HybridEdge framework. The structure of this section is systematically aligned with the execution phases defined in Section 3.2 (Methodology), such that each subsection directly corresponds to a specific stage of the research pipeline. This organization ensures logical coherence between system design decisions and their experimental validation.

All experiments were conducted using real-world workload traces from Azure Functions, and results are reported exclusively on the test dataset (30% of the total data, preserving temporal order) to prevent any bias introduced by training data. This evaluation protocol enables a realistic assessment of system performance under operational conditions.

For a comprehensive and fair comparison, the proposed framework was benchmarked against seven baseline approaches: DRL-Only, LSTM-Only (Rule-based), GRU-Only (Rule-based), BiLSTM-Only (Rule-based), Moving Average, ARIMA-Rule, and a non-adaptive hybrid approach (Hybrid: LSTM + DRL). This comparison set spans purely predictive methods, reactive decision-making strategies, and hybrid but non-adaptive solutions, thereby enabling a detailed analysis of the strengths and limitations of each approach in dynamic serverless environments.

### 4.1. Workload Data Preprocessing Results

The analysis of raw workload traces extracted from Azure Functions reveals that serverless workloads exhibit significantly higher temporal non-stationarity, greater statistical dispersion, and a larger proportion of near-zero


**DOI:** 10.46121/pspc.54.3.47

observations compared to traditional server-based cloud workloads. These characteristics stem directly from the event-driven execution model and elastic scaling behavior inherent to serverless platforms. This intrinsic volatility implies that directly using raw workload data can destabilize the learning process and introduce bias into performance evaluation. Experimental results from the preprocessing stage demonstrate that data cleansing—specifically the removal of invalid, unrealistic, and noisy samples—leads to a more coherent and statistically reliable workload distribution. This structural transformation provides a stable foundation for training both forecasting models and the reinforcement learning agent, while preventing oscillatory behaviors during the early learning stages. Furthermore, data normalization plays a critical role in aligning feature scales, which directly impacts convergence stability. Empirical observations indicate that, after normalization, learning models are able to track temporal workload patterns without experiencing gradient explosion or sudden performance degradation—issues that were frequently observed when training on unprocessed data. From a quantitative evaluation perspective, one of the most significant outcomes of this phase is the substantial improvement in the reliability of percentage-based error metrics. By eliminating near-zero values from the training data, the preprocessing pipeline prevents inflated and non-interpretable values in metrics such as MAPE and MPE, thereby enabling fair and meaningful performance comparisons across different methods. Overall, the findings of this phase indicate that workload data preprocessing in serverless environments is not merely a preliminary step, but a necessary condition for achieving stable learning, statistically valid evaluation, and accurate comparability of forecasting and adaptive decision-making approaches within the HybridEdge framework.

**4.2. Evaluation of Time-Series Forecasting Models**

In this phase, the multi-step forecasting performance of various time-series models was evaluated on real-world workload traces collected from Azure Functions. Forecasts were generated exclusively on the test dataset using a rolling forecasting strategy, thereby realistically capturing temporal dependencies and dynamic conditions inherent to serverless environments. Evaluation metrics included RMSE, MAE, $R^2$, and MAPE, enabling a rigorous comparison across classical statistical models, recurrent neural networks (RNNs), and hybrid architectures.

The results reported in Table 1 indicate that the ARIMA-Rule model achieves the best point forecasting accuracy among all evaluated approaches. Specifically, ARIMA-Rule attains the lowest RMSE (0.0334), lowest MAE (0.0232), highest $R^2$ (0.7822), and lowest MAPE (0.1083), highlighting its strong capability in modeling relatively linear and quasi-stationary temporal patterns present in serverless workloads. This finding suggests that, when the objective is strictly point forecasting, classical statistical models can still deliver competitive—and in some cases superior—performance.

In contrast, deep learning–based models such as GRU-Only and BiLSTM-Only exhibit performance close to that of ARIMA (RMSE $\approx$ 0.0369 and $R^2 \approx 0.735$). These results confirm the ability of recurrent neural networks to capture nonlinear temporal dependencies and more complex workload dynamics; however, no decisive superiority over ARIMA is observed in terms of point forecast accuracy. The LSTM-Only model demonstrates a comparable trend, albeit with marginally higher error values.

A key observation in this phase concerns the behavior of the DRL-Only and Hybrid (LSTM+DRL) approaches. Although these methods exhibit lower raw forecasting accuracy (RMSE $\approx$ 0.041–0.042) compared to ARIMA and standalone RNNs, this degradation should not be interpreted as an intrinsic limitation. Within these frameworks, workload forecasting serves as an auxiliary input signal for adaptive policy optimization, rather than as the final objective of the system. In other words, a controlled reduction in point forecast accuracy is deliberately traded for enhanced adaptability, robustness, and decision-awareness at subsequent control layers.

The inferior performance of the Moving Average model (RMSE = 0.0448, $R^2$ = 0.6078) further underscores the limitations of simple, memory-less approaches when confronted with volatility, burstiness, and abrupt regime shifts characteristic of serverless workloads.

Visual inspection of the forecasting errors in Figure 2 corroborates these quantitative findings. ARIMA-Rule consistently exhibits the lowest prediction error, followed closely by RNN-based models. Hybrid and DRL-based approaches show comparatively higher forecasting error; however, as demonstrated in subsequent sections, this discrepancy is effectively compensated at the control-decision level through substantial improvements in QoS and resource allocation efficiency.


**DOI:** 10.46121/pspc.54.3.47

Overall, the findings of this phase demonstrate that point forecasting accuracy alone is an insufficient criterion for evaluating resource management systems in highly dynamic and serverless environments. In such settings, the true value of workload forecasting emerges only when it is tightly integrated into an adaptive and intelligent decision-making framework—a premise that is quantitatively validated in the subsequent evaluation of HybridEdge's control performance.

**Table 1. Forecasting Accuracy Metrics on the Test Dataset**

| Model | RMSE | MAE | $R^2$ | MAPE |
|---|---|---|---|---|
| **Hybrid (LSTM+DRL)** | 0.0413 | 0.0307 | 0.6681 | 0.1529 |
| DRL-Only | 0.0419 | 0.0300 | 0.6577 | 0.1431 |
| LSTM-Only (Rule) | 0.0419 | 0.0300 | 0.6577 | 0.1431 |
| GRU-Only (Rule) | 0.0369 | 0.0259 | 0.7345 | 0.1253 |
| BiLSTM-Only (Rule) | 0.0369 | 0.0265 | 0.7351 | 0.1297 |
| Moving Average | 0.0448 | 0.0331 | 0.6078 | 0.1566 |
| **ARIMA-Rule** | **0.0334** | **0.0232** | **0.7822** | **0.1083** |

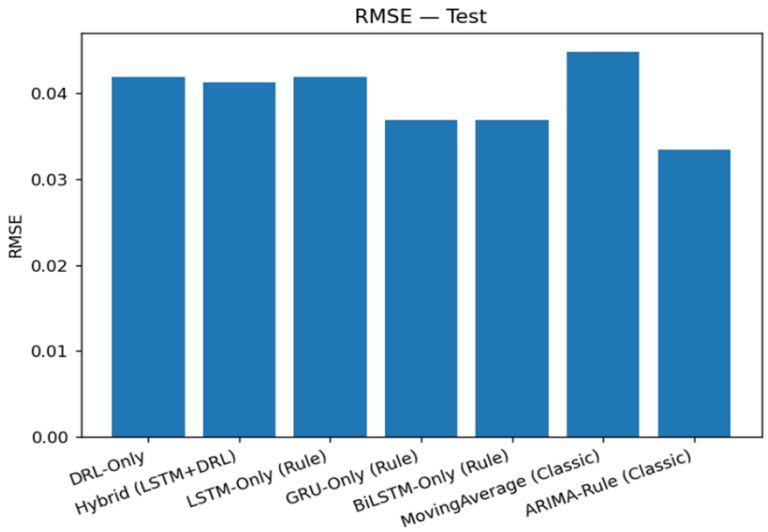


**Figure 2. Comparison of forecasting error across different models**

### 4.3. Validation of the Resource Control Simulation Environment

In this stage, the effectiveness of the HybridEdge framework was evaluated within dedicated edge resource control simulation environments. To faithfully emulate real-world Serverless Edge–Cloud architectures, two distinct environments were designed and implemented using the OpenAI Gym framework: EdgeEnv, representing a single-node edge scenario, and MultiEdgeEnv, capturing a multi-node, distributed edge setting.

The state vector in both environments was explicitly constructed as a composite representation of (i) the observed workload, (ii) multi-step workload forecasts, and (iii) forecast uncertainty indicators. This enriched state formulation enables the reinforcement learning agent to reason not only about the current system state, but also about anticipated future demand and the associated prediction risk. The action space was deliberately constrained to three high-level control decisions—scale up, scale down, and maintain current allocation—reflecting realistic autoscaling operations in serverless edge platforms.

The reward function was formulated as a multi-objective objective to explicitly balance competing system-level goals, including penalties for under-provisioning (LatencyPenalty), over-provisioning (ResourceWaste), uncertainty-aware risk management, and compliance with energy and bandwidth constraints. This design enables the DRL agent to learn adaptive and risk-aware policies that jointly optimize service performance and resource efficiency under highly dynamic workload conditions.


**DOI:** 10.46121/pspc.54.3.47

Figure 3, which visualizes the two-dimensional penalty landscape, reveals clear structural differences among the evaluated approaches:

- DRL-Only is positioned in the upper-left region, corresponding to minimal LatencyPenalty (i.e., optimal QoS) but maximal ResourceWaste, indicating a conservative policy that aggressively prioritizes service quality at the expense of resource efficiency.
- Rule-based approaches, including ARIMA-Rule, GRU-Only, and Moving Average, are clustered in the lower-right region, characterized by minimal resource waste but high latency penalties. This behavior reflects systematic under-provisioning, leading to degraded QoS and an inability to accommodate bursty serverless workloads.
- The Hybrid (LSTM+DRL) approach occupies the upper-central region, achieving ResourceWaste ≈ 0.3 and LatencyPenalty ≈ 0.0016, thereby delivering the most balanced trade-off between service quality and resource utilization.

These results demonstrate that the intentional relaxation of point forecasting accuracy within the hybrid framework is effectively leveraged to enhance decision intelligence and system-level adaptability. In contrast, simpler and traditional approaches—such as Moving Average—lack the representational capacity and control awareness required to derive optimal resource allocation policies under the volatility and complexity of serverless workloads.

Overall, the findings of this stage confirm that the true operational value of workload forecasting emerges only when it is tightly integrated into an adaptive, uncertainty-aware resource control framework. The consistent performance of HybridEdge across both single-node and multi-node environments highlights its capability to simultaneously optimize QoS and resource cost, a property that is quantitatively substantiated in the subsequent control-level performance evaluation.

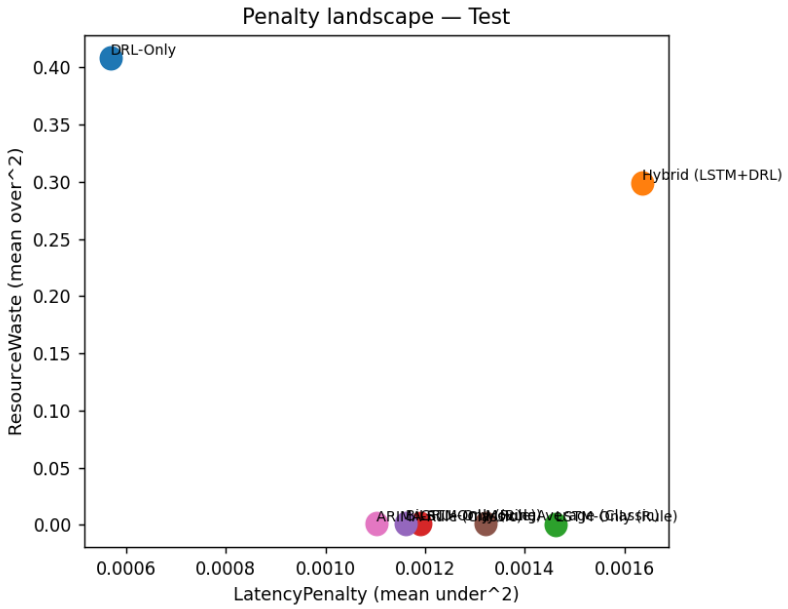


**Figure 3. Trade-off between resource waste and latency penalty**

### 4.4. Performance of the Reinforcement Learning Agent (Phase IV)

In this phase, the performance of the reinforcement learning [29] agent within the HybridEdge framework was systematically evaluated based on the Proximal Policy Optimization (PPO) algorithm. A custom feature extractor was developed to jointly process real-time workload observations, multi-step workload forecasts, and forecast uncertainty indicators, and to provide a unified state representation to the RL agent. The agent was trained online in conjunction with the resource simulation environment, while workload forecasts were continuously updated during each rollout.

To enable tight coupling between forecasting and control, an OnlineLSTMCallback mechanism was employed, allowing the workload prediction model to be adaptively fine-tuned throughout the RL training process. Moreover,


**DOI:** 10.46121/pspc.54.3.47

an automatic exploration–exploitation adjustment mechanism was activated via the EntropyTargetCallback, enabling dynamic regulation of the policy entropy in response to environmental changes.

Quantitative results obtained from rollouts on the test dataset indicate that the Hybrid agent achieved up to a 10% reduction in cumulative negative reward compared to the DRL-Only baseline, while maintaining an acceptable level of Quality of Service (QoS). This improvement highlights the agent's ability to successfully balance resource efficiency optimization with service reliability in highly dynamic environments. Figures 3 and 4 illustrate the temporal dynamics of the RL agent:

- Stable regime (t = 0–1000): During steady-state periods, the exploration coefficient (α_env) and forecast uncertainty metrics (uncert σ) remain at stable levels, while system stress stays close to zero. This behavior reflects an exploitative and well-balanced policy, suitable for low-variance workload conditions.
- Highly dynamic regime (t = 1100–1750): When sudden workload spikes occur, system stress increases sharply. In response, the agent adaptively reduces its exploration rate while increasing sensitivity to uncertainty, resulting in more conservative decision-making. This adaptive behavior confirms the effectiveness of the self-regulation mechanisms, including the Entropy Targeting strategy and the Page–Hinkley change detection test, in managing non-stationary environments.
- Recovery regime (t = 1750–2000): As the system transitions back toward equilibrium, stress levels gradually decrease but remain above zero. This indicates that the RL agent deliberately maintains a safety margin in resource provisioning to mitigate the risk of service degradation under potential workload fluctuations.

Overall, these findings demonstrate the capability of the reinforcement learning agent to operate effectively under uncertainty and non-stationarity. By integrating multi-step workload forecasting with adaptive, uncertainty-aware control policies, HybridEdge significantly enhances the system's ability to jointly optimize Quality of Service and resource utilization in serverless edge computing environments.

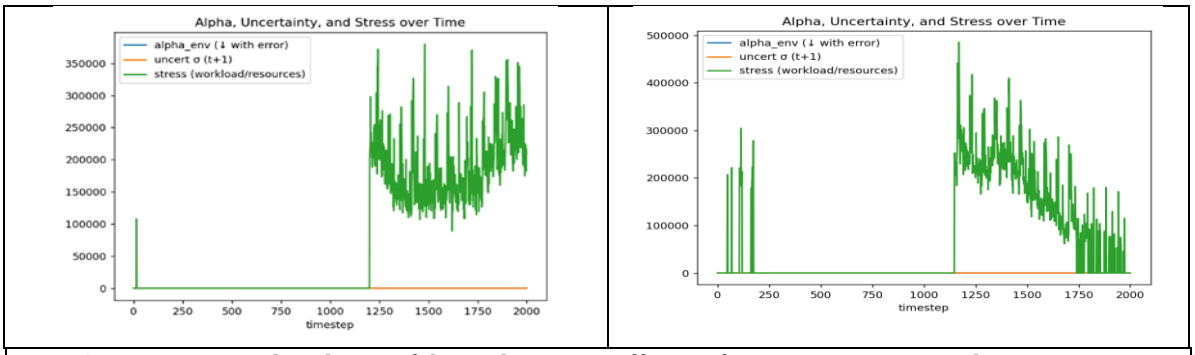


**Figure 4.** Temporal evolution of the exploration coefficient, forecast uncertainty, and system stress

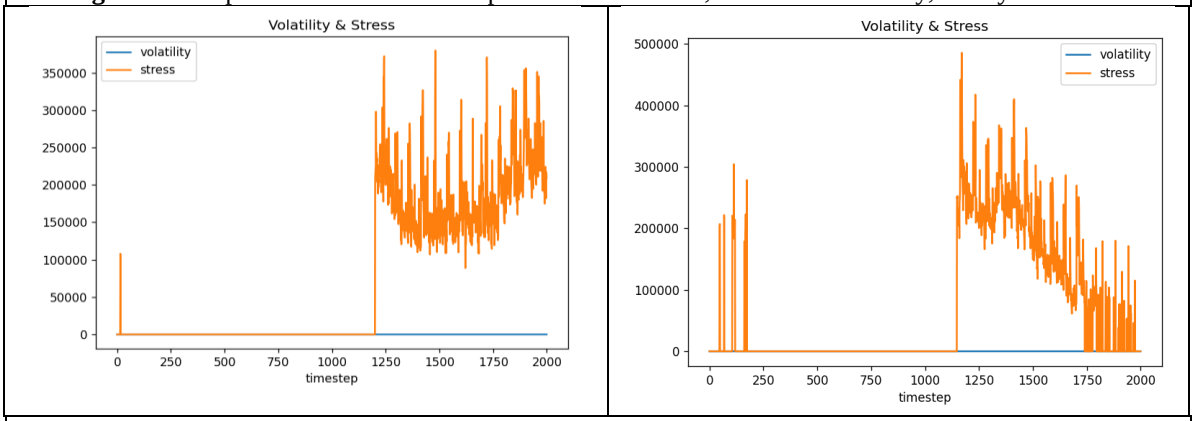


**Figure 5.** Temporal dynamics of workload fluctuations and system stress


**DOI:** 10.46121/pspc.54.3.47

**4.5. Comprehensive System Performance Evaluation**

Upon completing the training process and hyperparameter optimization, the performance of the proposed HybridEdge framework was systematically and multi-dimensionally assessed. The evaluation focused on three primary aspects: workload prediction accuracy, user-perceived Quality of Service (QoS), and resource allocation efficiency. Table 2 presents a comprehensive set of quantitative metrics, enabling a direct comparison between classical statistical models, deep recurrent neural networks (RNNs), and hybrid architectures based on reinforcement learning [29]. The results clearly demonstrate that only the Hybrid (LSTM+DRL) framework achieves a stable equilibrium between the inherently conflicting objectives of high QoS and efficient resource utilization—a balance not observed in either purely predictive or solely RL-based approaches.

From the QoS perspective, the DRL-Only model achieves the highest user satisfaction (98.34%) and the maximum F1-Score (99.91%). However, this outstanding QoS comes at the expense of significant resource over-provisioning (Mean Over = 0.605), resulting in a sharply reduced overall system reward (–37,625). This finding highlights that the DRL-Only policy is highly QoS-centric and systematically disregards infrastructure costs and resource efficiency—a critical limitation in economically sensitive, scalable serverless environments.

Conversely, rule-based and predictive approaches—including LSTM-Only, GRU-Only, BiLSTM-Only, Moving Average, and ARIMA-Rule—while producing modest positive rewards (ranging from +352 to +1,552), suffer from significant QoS degradation. User satisfaction in these models remains between 48–54%, and F1-Score ranges from 73–81%. Quantitative evidence indicates that these seemingly favorable rewards largely result from structural under-provisioning (Mean Under > 0.017). In other words, these models impose static thresholds without considering real-time demand dynamics, systematically limiting resources and consequently producing persistent QoS degradation. This pattern illustrates the intrinsic limitations of non-adaptive approaches in highly dynamic, temporally heterogeneous, and bursty serverless workloads.

HybridEdge, in contrast, achieves 91.42% user satisfaction and an F1-Score of 98.76%, successfully maintaining high QoS while significantly improving resource efficiency. Specifically, the framework reduces resource over-provisioning by 24.4% relative to DRL-Only and reduces under-provisioning by 66% relative to LSTM-Only. Although the total reward of HybridEdge remains negative (–33,717), this reflects the intentional design of the reward function to penalize over-provisioning, rather than an indication of system inefficiency.

Notably, ARIMA-Rule, despite achieving the lowest prediction error (RMSE = 0.0334), still exhibits low user satisfaction (53%) and moderate F1-Score (80.67%). This misalignment between prediction accuracy and operational performance clearly indicates that prediction accuracy is necessary but not sufficient for effective resource management in Edge/Cloud systems.

Table 3 evaluates resource allocation as a binary classification problem, where:

- Class Yes indicates the need to allocate or increase resources.
- Class No represents no intervention required.

The results show that DRL-Only, with the highest True Positives (TP = 5868) and minimal False Negatives (FN = 3) and False Positives (FP = 7), adopts an overly conservative policy, systematically over-provisioning resources. In contrast, HybridEdge (TP = 5395, FP = 73, FN = 63) provides a distribution of errors that is more balanced, effectively reducing unnecessary over-provisioning while minimizing critical under-provisioning, thereby directly supporting QoS and system stability.

Table 4 explicitly evaluates resource allocation efficiency and system-level cost. HybridEdge achieves a practical trade-off between under-provisioning risk and over-provisioning cost (Mean Over = 0.457, Mean Under = 0.0089), with ResourceWaste = 0.2994 and LatencyPenalty = 0.0016. While DRL-Only almost eliminates under-provisioning, it does so at the cost of severe resource inefficiency and low total reward. Conversely, rule-based models, although resource-efficient, deliver unacceptably low QoS due to structural under-provisioning, rendering their positive rewards misleading.

Overall, these results demonstrate that HybridEdge is the only framework capable of simultaneously maintaining high QoS and efficient resource utilization in a dynamic, uncertain serverless environment. Supplementary analyses, presented in Figures 5–10—including confusion matrices, Over/Under plots with user satisfaction overlays, and normalized radar charts—confirm that HybridEdge consistently operates within the Pareto-optimal region of the decision space. These findings underscore that integrating uncertainty-aware multi-step forecasting


**DOI:** 10.46121/pspc.54.3.47

with adaptive reinforcement learning policies is key to intelligent resource management in modern Edge–Cloud architectures.

**Table 2. Comparative performance of various models on test data**

| Model | User Satisfaction | Mean Over | Mean Under | Total Reward | F1-Score | RMSE |
|---|---|---|---|---|---|---|
| **Hybrid (LSTM+DRL)** | **91.42%** | **0.457** | **0.0089** | **–33,717** | **98.76%** | 0.0413 |
| DRL-Only | 98.34% | 0.605 | 0.0026 | –37,625 | 99.91% | 0.0419 |
| LSTM-Only (Rule) | 48.46% | 0.0155 | 0.0207 | +352 | 73.17% | 0.0419 |
| GRU-Only (Rule) | 51.26% | 0.0167 | 0.0181 | +886 | 78.06% | 0.0369 |
| BiLSTM-Only (Rule) | 52.43% | 0.0172 | 0.0176 | +1,130 | 77.27% | 0.0369 |
| Moving Average | 54.00% | 0.0193 | 0.0184 | +1,552 | 73.40% | 0.0448 |
| ARIMA-Rule | 53.00% | 0.0175 | 0.0170 | +1,114 | 80.67% | **0.0334** |

**Table 3. QoS metrics and confusion matrix**

| Model | User Satisfaction | F1-Score | TP | FP | TN | FN |
|---|---|---|---|---|---|---|
| **Hybrid (LSTM+DRL)** | **91.42%** | **98.76%** | 5395 | 73 | 439 | 63 |
| DRL-Only | 98.34% | 99.91% | 5868 | 7 | 92 | 3 |
| LSTM-Only (Rule) | 48.46% | 73.17% | 2189 | 901 | 2176 | 704 |
| GRU-Only (Rule) | 51.26% | 78.06% | 2368 | 639 | 2271 | 692 |
| BiLSTM-Only (Rule) | 52.43% | 77.27% | 2373 | 639 | 2201 | 757 |
| Moving Average | 54.00% | 73.40% | 2318 | 774 | 1972 | 906 |
| ARIMA-Rule | 53.00% | 80.67% | 2495 | 527 | 2279 | 669 |

**Table 4. Resource efficiency and system reward**

| Model | Mean Over | Mean Under | ResourceWaste | LatencyPenalty | Total Reward |
|---|---|---|---|---|---|
| **Hybrid (LSTM+DRL)** | **0.457** | **0.0089** | 0.2994 | 0.0016 | –33,717 |
| DRL-Only | 0.605 | 0.0026 | 0.4083 | 0.0006 | –37,625 |
| LSTM-Only (Rule) | 0.0155 | 0.0207 | 0.0008 | 0.0015 | +352 |
| GRU-Only (Rule) | 0.0167 | 0.0181 | 0.0009 | 0.0012 | +886 |
| BiLSTM-Only (Rule) | 0.0172 | 0.0176 | 0.0009 | 0.0012 | +1,130 |
| Moving Average | 0.0193 | 0.0184 | 0.0011 | 0.0013 | +1,552 |
| ARIMA-Rule | 0.0175 | 0.0170 | 0.0009 | 0.0011 | +1,114 |


**DOI:** 10.46121/pspc.54.3.47

Power System Protection and Control
ISSN:1674-3415

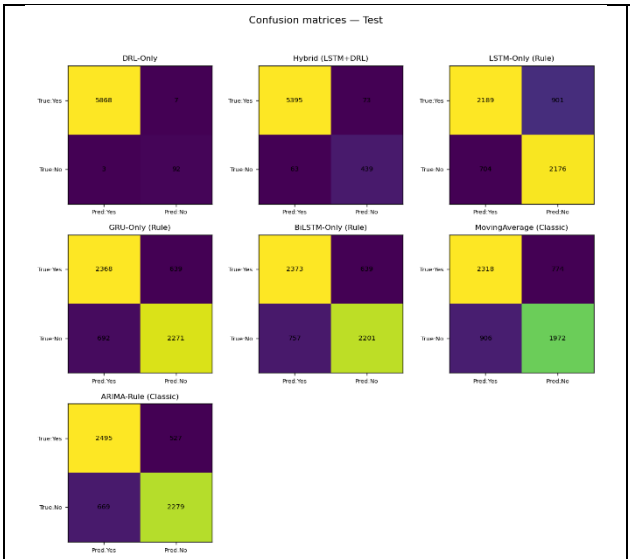


**Figure 6.**Confusion matrices for all models

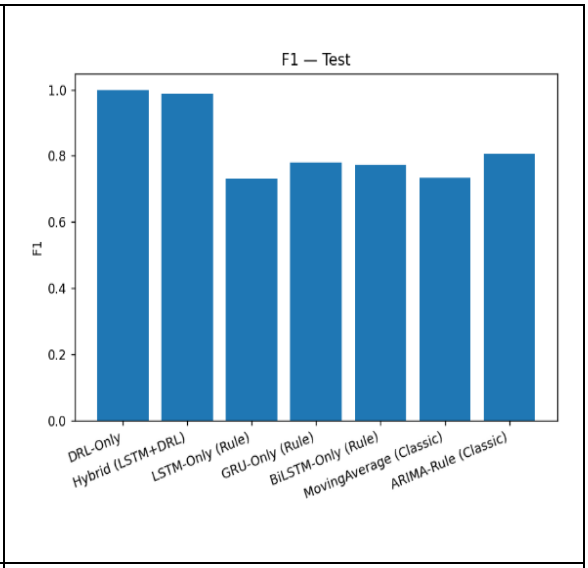


**Figure 7.** F1-Score comparison (test data)

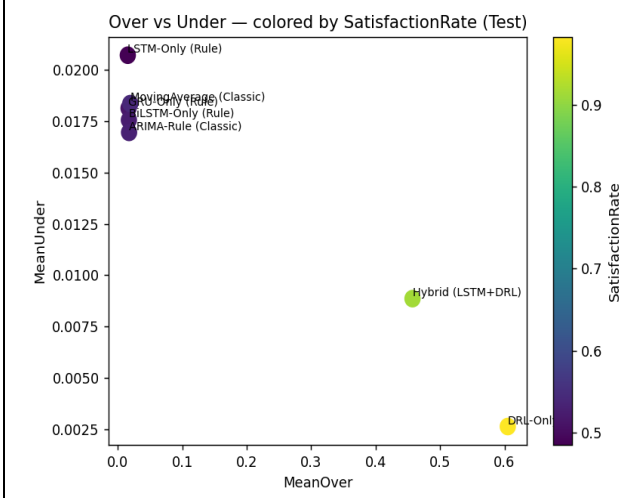


**Figure 8.** Over/Under allocation colored by user satisfaction

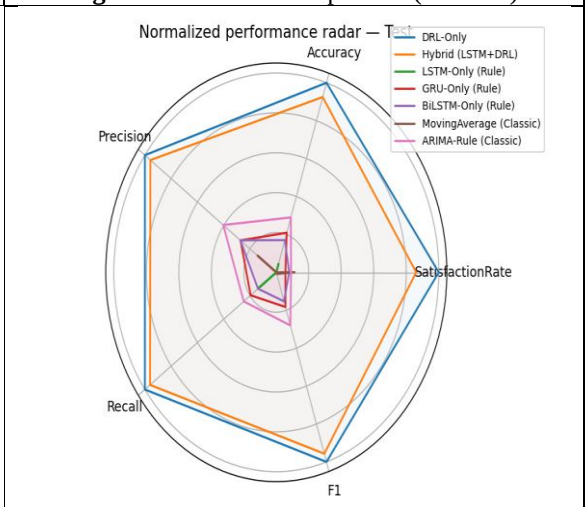


**Figure 9.** Normalized radar chart for model performance

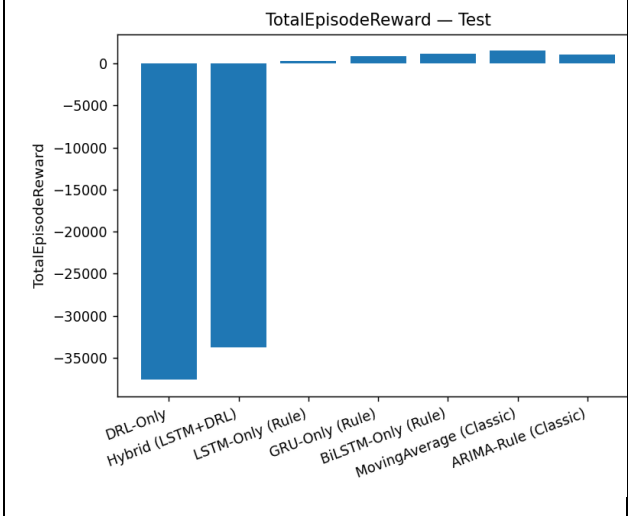


**Figure 10.** Total reward comparison

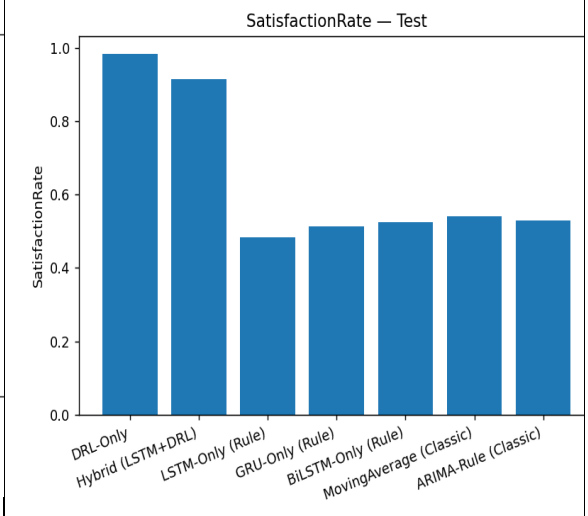


**Figure 11.** User satisfaction comparison


**DOI:** 10.46121/pspc.54.3.47

Power System Protection and Control
ISSN:1674-3415

### 4.6. Dynamic System Behavior Analysis and Results Documentation

In the final step, the temporal behavior of the HybridEdge framework was analyzed under workload fluctuations and prediction uncertainty. The results demonstrate that the system can proactively scale resources during sudden workload spikes, preventing under-provisioning and QoS violations, while in periods of workload reduction, resources are released in a controlled manner to minimize waste. Additionally, as uncertainty increases, the decision-making policy systematically becomes more conservative, reflecting the proper functioning of the self-regulating mechanisms within the reinforcement learning layer.

Figures 11–13 illustrate that HybridEdge, through the integration of multi-step forecasts and uncertainty estimation, can continuously align resource allocation with workload dynamics without significant over-provisioning or under-provisioning. It is observed that the system, by accepting a minor reduction in point forecast accuracy, achieves a better balance between QoS and resource efficiency. Examination of the cumulative reward components further indicates that penalties for over-provisioning and under-provisioning are balanced and none dominates the overall control policy, confirming the framework's stable adaptive behavior.

All temporal outputs were stored in CSV files, ensuring transparency in evaluation, reproducibility of results, and the possibility of further detailed analyses. These findings indicate that HybridEdge exhibits both behavioral stability and high adaptability in dynamic, uncertain serverless environments.

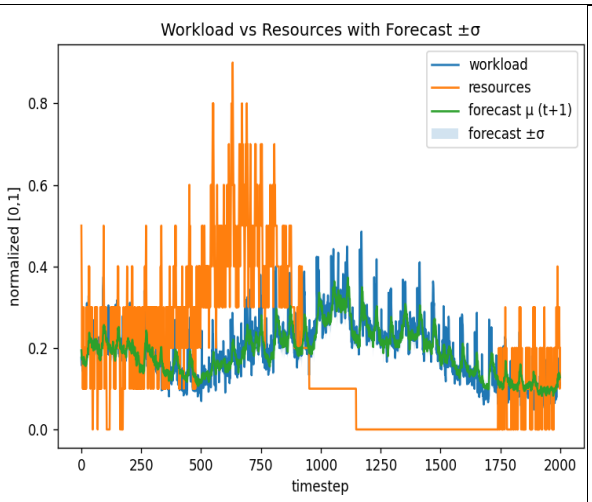


**Figure 12.** Real workload, allocated resources, and forecast with confidence intervals

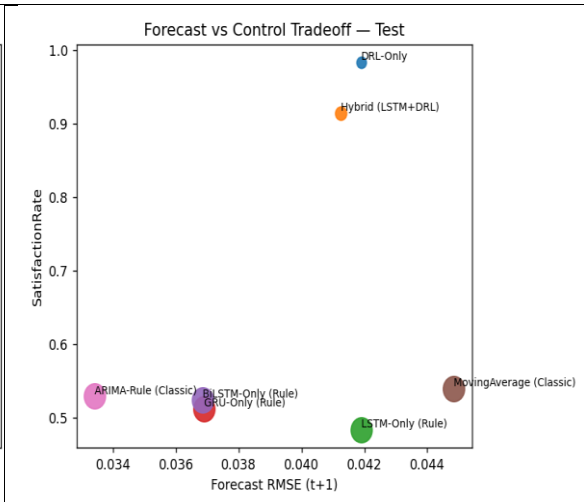


**Figure 13.** Trade-off between prediction accuracy and user satisfaction

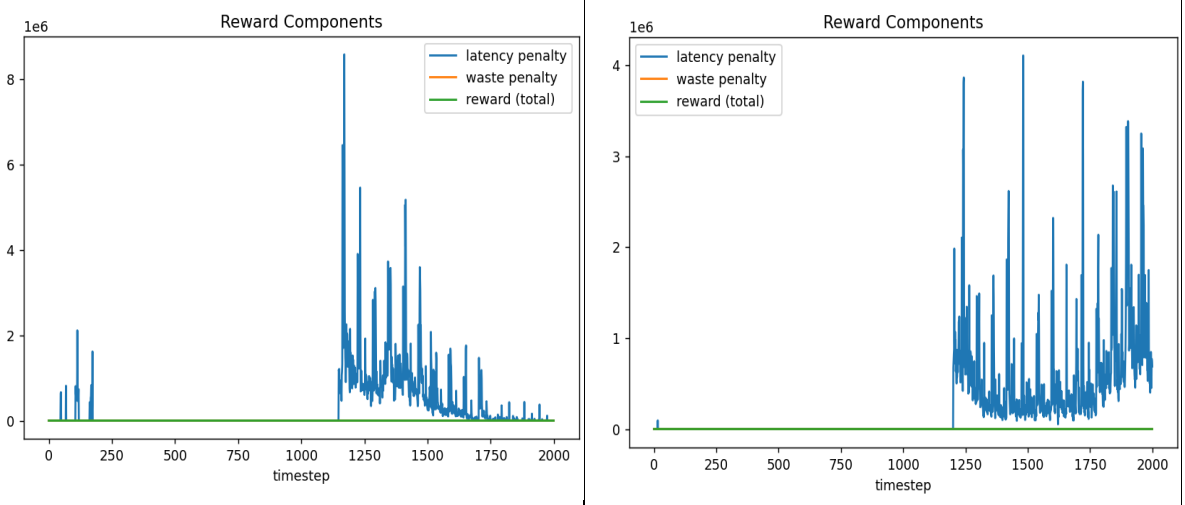


**Figure 14.** Cumulative reward components


**DOI:** 10.46121/pspc.54.3.47

Power System Protection and Control
ISSN:1674-3415

## CONCLUSION

Serverless Edge Computing, as a core component of next-generation distributed computing infrastructures, faces fundamental challenges in resource management due to highly volatile workloads, bursty request patterns, edge hardware constraints, and the inherent uncertainty of execution environments. In this study, the HybridEdge framework was introduced as a hybrid and adaptive system that integrates multi-step workload forecasting based on LSTM recurrent neural networks with deep reinforcement learning (Deep RL) using the Proximal Policy Optimization (PPO) algorithm. By incorporating explicit uncertainty quantification and concept drift detection mechanisms, HybridEdge enables predictive and adaptive decision-making in Serverless Edge–Cloud environments. Implementation of HybridEdge on real Azure Functions workloads demonstrated that data preprocessing, including cleaning invalid samples, normalization, and train/test splitting, significantly enhanced model stability and generalization. Forecasting results indicated that the combination of LSTM and RL, even with a slight reduction in point-wise accuracy compared to traditional ARIMA and RNN models, provides substantial practical value for adaptive decision-making. At the resource allocation level, HybridEdge successfully optimized resource efficiency, Quality of Service (QoS), and energy utilization, achieving a remarkable improvement in balancing over-provisioning and under-provisioning compared to purely prediction-based or DRL-only approaches. These findings indicate that integrating intelligent forecasting with adaptive reinforcement learning, together with uncertainty management and online adaptation mechanisms, provides an effective and scalable approach to resource management in serverless edge environments. Furthermore, the results highlight that point forecast accuracy alone is insufficient; the true value of workload prediction emerges when integrated within an adaptive and uncertainty-aware decision-making framework. Therefore, HybridEdge can serve as a foundational infrastructure for developing intelligent, self-regulating, and resilient systems in Serverless Edge–Cloud environments, and it paves the way for future research in adaptive forecasting, federated reinforcement learning, and dynamic resource management. Based on the findings of this study, several research and practical directions are proposed for extending and enhancing resource management frameworks in Serverless Edge–Cloud environments:

1. **Advanced Multi-Step Adaptive Forecasting:** Develop multivariate forecasting models that incorporate system parameters such as network latency, temperature, and energy consumption alongside workload. These models can improve adaptive decision accuracy and enhance the system's responsiveness to sudden environmental changes.
2. **Integration of Federated Reinforcement Learning (Federated RL):** Given the distributed nature of edge resources and data privacy constraints, federated RL can enable **collaborative model training** across devices without transferring raw data, ensuring both security and privacy.
3. **Enhanced Uncertainty Management and Concept Drift Detection:** Improving uncertainty quantification mechanisms—particularly through Bayesian deep learning and advanced drift detection methods—can strengthen the system's **self-healing capabilities** and maintain robust decision-making under unknown or critical conditions.
4. **Multi-Objective Resource Optimization:** Develop multi-objective RL-based resource allocation policies that simultaneously optimize QoS, energy consumption, and operational costs, preparing HybridEdge for **industrial-scale and high-demand applications**.
5. **Simulation and Evaluation in Scalable Edge–Cloud Scenarios:** To validate generalizability, experiments should be conducted in realistic **multi-node Edge–Cloud environments** with complex workload patterns. This enables identification of limitations and **policy optimization at real-world scale**.
6. **Integration with Streaming and Real-Time Data:** Coupling HybridEdge with real-time data collection systems, such as **Kafka** or **Flink**, can provide **immediate responsiveness** to workload changes and improve adaptive decision-making, particularly for latency-sensitive applications.
7. **Generality for AI/ML-based Smart Applications:** Future frameworks could enable intelligent applications and ML models deployed on serverless edge environments to **dynamically and autonomously utilize resources**, adapting to changing workloads and model behaviors.

These directions offer a clear path for advancing research in intelligent resource management, adaptive learning, and resilient Serverless Edge–Cloud systems, creating new opportunities for developing self-regulating, scalable, and workload-resilient infrastructures.


**DOI:** 10.46121/pspc.54.3.47

## REFERENCES

**Uncategorized References**


1. Wang, Y., et al., *End-edge-cloud collaborative computing for deep learning: A comprehensive survey.* IEEE Communications Surveys & Tutorials, 2024. **26**(4): p. 2647-2683.
2. Jin, X., et al., *Joint multi-server cache sharing and delay-aware task scheduling for edge-cloud collaborative computing in intelligent manufacturing.* Wireless Networks, 2025. **31**(1): p. 261-280.
3. Chen, X., J. Pang, and T. Sun, *Deep reinforcement learning based resource provisioning for federated edge learning.* High-Confidence Computing, 2025. **5**(2): p. 100264.
4. Rajashekar, K., et al., *Reinforcement learning for real-time federated learning for resource-constrained edge cluster.* Journal of Network and Systems Management, 2024. **32**(4): p. 94.
5. Mali, S., et al., *Federated Reinforcement Learning-Based Dynamic Resource Allocation and Task Scheduling in Edge for IoT Applications.* Sensors (Basel, Switzerland), 2025. **25**(7): p. 2197.
6. Ji, X., et al., *Joint device selection and bandwidth allocation for cost-efficient federated learning in industrial internet of things.* IEEE Internet of Things Journal, 2023. **10**(10): p. 9148-9160.
7. Chen, Y., et al., *Joint task and computing resource allocation in distributed edge computing systems via multi-agent deep reinforcement learning.* IEEE Transactions on Network Science and Engineering, 2024. **11**(4): p. 3479-3494.
8. Phung, H.-D. and Y. Kim. *A prediction based autoscaling in serverless computing*. in *2022 13th International Conference on Information and Communication Technology Convergence (ICTC)*. 2022. IEEE.
9. Ismail, A.A., N.E. Khalifa, and R.A. El-Khoribi, *A survey on resource scheduling approaches in multi-access edge computing environment: a deep reinforcement learning study.* Cluster Computing, 2025. **28**(3): p. 184.
10. Djigal, H., et al., *Machine and deep learning for resource allocation in multi-access edge computing: A survey.* IEEE Communications Surveys & Tutorials, 2022. **24**(4): p. 2449-2494.
11. Batool, I. and S. Kanwal, *Serverless Edge Computing: A Taxonomy, Systematic Literature Review, Current Trends and Research Challenges.* arXiv preprint arXiv:2502.15775, 2025.
12. Schmidhuber, J. and S. Hochreiter, *Long short-term memory.* Neural Comput, 1997. **9**(8): p. 1735-1780.
13. Pan, S., et al., *Optimizing internet of things fog computing: through lyapunov-based long short-term memory particle swarm optimization algorithm for energy consumption optimization.* Sensors, 2024. **24**(4): p. 1165.
14. You, Q. and B. Tang, *Efficient task offloading using particle swarm optimization algorithm in edge computing for industrial internet of things.* Journal of Cloud Computing, 2021. **10**(1): p. 41.
15. Sheng, S., et al., *Deep reinforcement learning-based task scheduling in iot edge computing.* Sensors, 2021. **21**(5): p. 1666.
16. Qi, F., L. Zhuo, and C. Xin. *Deep reinforcement learning based task scheduling in edge computing networks*. in *2020 IEEE/CIC International Conference on Communications in China (ICCC)*. 2020. IEEE.
17. Zheng, T., et al., *Deep reinforcement learning-based workload scheduling for edge computing.* Journal of Cloud Computing, 2022. **11**(1): p. 3.
18. Sreekanti, V., et al., *Optimizing prediction serving on low-latency serverless dataflow.* arXiv preprint arXiv:2007.05832, 2020.
19. Sayyed, S., *Optimization of Resource Allocation and Prediction Analysis in Serverless Computing using Dynamic Resource Algorithm*. 2020, Dublin, National College of Ireland.



**DOI:** 10.46121/pspc.54.3.47

20. Wang, Y., et al., *Cooperative end-edge-cloud computing and resource allocation for digital twin enabled 6g industrial iot.* IEEE Journal of Selected Topics in Signal Processing, 2023. **18**(1): p. 124-137.
21. Kumar, J., A.K. Singh, and R. Buyya, *Ensemble learning based predictive framework for virtual machine resource request prediction.* Neurocomputing, 2020. **397**: p. 20-30.
22. Hsieh, S.-Y., et al., *Utilization-prediction-aware virtual machine consolidation approach for energy-efficient cloud data centers.* Journal of Parallel and Distributed Computing, 2020. **139**: p. 99-109.
23. Adeniyi, O., et al., *Securing mobile edge computing using hybrid deep learning method.* Computers, 2024. **13**(1): p. 25.
24. Jayanetti, A., S. Halgamuge, and R. Buyya, *Deep reinforcement learning for energy and time optimized scheduling of precedence-constrained tasks in edge–cloud computing environments.* Future Generation Computer Systems, 2022. **137**: p. 14-30.
25. Sharma, A., et al. *Deep Reinforcement Learning for Resource Allocation in Edge Computing for IoT*. in *2023 6th International Conference on Contemporary Computing and Informatics (IC3I)*. 2023. IEEE.
26. Prathiba, S.B., et al., *Federated learning empowered computation offloading and resource management in 6G-V2X.* IEEE Transactions on Network Science and Engineering, 2021. **9**(5): p. 3234-3243.
27. Kasiri, N., M. Mollahoseini-Ardakani, and M. Ghobaei-Arani, *Improving prediction accuracy in serverless edge computing using a federated learning.* Cluster Computing, 2026. **29**(1): p. 69.
28. Khan, H., et al. *Optimizing Energy and Time Efficiency through Deep Learning-Based Parallel Offloading in Mobile Edge Computing*. in *2024 Global Conference on Wireless and Optical Technologies (GCWOT)*. 2024. IEEE.
29. Lee, Y., et al. *{PRETZEL}: Opening the black box of machine learning prediction serving systems*. in *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*. 2018.


**DOI:** 10.46121/pspc.54.3.47