ADDER: Service-Specific Adaptive Data-Driven Radio Resource Control for Cellular-IoT

Yingjing Wu
University of Utah
Salt Lake City, UT
wyj1993@cs.utah.edu

Ahmed Elmokashfi
Simula Metropolitan CDE
Oslo, Norway
ahmed@simula.no

Foivos Michelinakis
Simula Metropolitan CDE
Oslo, Norway
foivos@simula.no

Jacobus Van der Merwe
University of Utah
Salt Lake City, UT
kobus@cs.utah.edu

Shandian Zhe
University of Utah
Salt Lake City, UT
zhe@cs.utah.edu

Abstract— **Energy-saving methods like Discontinuous Reception (DRX) and Power Save Mode (PSM) are commonly used in Internet of Thing (IoT) applications, allowing for sleep and awake cycle adjustments to save energy. However, understanding and configuring these parameters on devices, especially actuator-type devices, is challenging for IoT service providers. Unlike sensor types, these devices must complete their sleep cycle before responding to infrequent downlink commands, making efficient parameter selection and traffic prediction vital for energy efficiency and command reception.**

To address this, we present ADDER, a network-side solution leveraging a context-aware traffic predictor. This predictor forecasts downlink arrival probabilities, guiding a deep deterministic policy gradient (DDPG) policymaker that selects energy-saving parameters based on thresholds defined by the IoT service providers. ADDER, leverages contextual information like day of week, hour, weather, holidays, and events, shifting the focus from individual device histories (often erratic) to analyzing broader service traffic patterns. This data-driven strategy enables ADDER to adjust energy-saving settings for the best balance between energy efficiency and latency, customizing to the unique requirements of each service and removing the burden of configuring complex network settings. We observed that ADDER meets latency needs while achieving a 5.9% reduction in energy consumption for services requiring rapid responses. For applications prioritizing energy conservation, such as irrigation systems and city lighting, ADDER achieves a significant 32.7% reduction in energy consumption with a slight increase (9%) in messages might not meet the strictest latency requirements. To evaluate the consequences of prediction inaccuracies from our predictor, we utilized a real-world shared mobility dataset provided by Austin's Transportation Department for a case study.

Index Terms—**NB-IoT, LTE-M, energy saving, context bandit, deep reinforcement learning**

I. Introduction

In 3GPP MTC, protocols such as NB-IoT and LTE-M utilize energy-saving features like DRX [1], [2] and PSM [3], [4], enhancing battery life. While sensor-based devices quickly reconnect to send mobile-originated (MO) messages, resulting in delays primarily due to reconnection and latency, actuator-centric applications like intelligent irrigation systems and smart lockers mainly rely on less frequent downlink communication for operations, leading to potential delays in response to urgent downlink messages due to the inactive connection state of the devices.

While applications like farm irrigation tolerate delays, unlocking a bike requires quicker action, and managing traffic signs demands real-time responsiveness to prevent jams. This creates a balance challenge between energy efficiency and response speed that varies by application. Developers must be familiar with complex cellular network energy-saving configurations and develop adaptable strategies based on each application's traffic needs. This necessitates exploring how to dynamically learn and adjust these mechanisms to meet specific energy and response time requirements.

We propose ADDER, a novel network-side solution that goes beyond device-specific approaches to achieve optimal energy savings while meeting latency constraints for specific actuator-focused services. ADDER leverages a context-aware traffic predictor that combines service provider insights with additional data like weather, traffic, and holidays to forecast traffic patterns accurately. These predictions then guide the modified DDPG-based [5] policy maker in selecting the most suitable DRX and PSM parameters. The network controller acknowledges these settings to devices, allowing them to adapt their communication accordingly.

ADDER enhances energy efficiency for IoT actuators with infrequent and unpredictable traffic, focusing on optimal DRX settings and underutilized PSM potential. Unlike prior research [6], [7] that either overlooked PSM or did not distinguish its unique energy characteristics, ADDER adapts to varying energy profiles. It considers the wake-up energy cost during the Tracking Area Update process for more accurate energy consumption analysis. Furthermore, while majority previous studies (see Section VII) relied on sparse or aggregated device data, ADDER incorporates contextual information like weather, traffic, and holidays from service providers for precise, application-specific traffic predictions. This enables ADDER to balance energy savings with service demands, optimizing network settings for peak to low demand scenarios.

ADDER's implementation faces two key challenges: the complexity of its reinforcement learning (RL) framework, which must navigate a vast state space influenced by contextual factors and a varied action space for energy-saving settings, and the impracticality of online learning due to latency issues. To tackle these, ADDER uses a simulator for safe exploration and data collection, the DDPG framework for managing the high-dimensional action space, and a context-

This material is based upon work supported by the National Science Foundation under Grant Number 1827940.

aware predictor to simplify the state space by estimating traffic patterns from contextual data. This approach enables synthetic data generation for various traffic scenarios, facilitating practical RL model training with limited real-world data and addressing the data scarcity challenge in IoT applications.

In evaluating ADDER's performance, we compared it with two baselines: EDRX, set to the longest acceptable delay, and AC-DRX, an actor-critic approach used for a similar problem in a previous study [6]. ADDER showed superior energy efficiency, particularly in low-traffic scenarios. It also introduces a traffic arrival probability threshold as a practical metric for service providers to balance energy use against their needs, simplifying the adjustment process. A case study in bike-sharing highlighted ADDER's real-world applicability, focusing on how prediction inaccuracies affect performance. This study underscores ADDER's potential for operational enhancements and predictive model refinement.

This paper is organized as follows: Section II introduces the background by explaining Radio Resource Control (RRC) energy-saving configurations and parameters. Next, Section III describes ADDER's architecture, detailing its role in parameter configuration, the reinforcement learning problem definition and solution. The paper concludes with Section IV, which presents an empirical assessment of our methodology.

II. BACKGROUND

In this section, we provide the necessary background. We first explain the energy-saving mechanisms, encompassing a variety of energy-saving parameters, as well as the procedures for reconfiguring these parameters. Particular attention is paid to pinpointing the specific parameters that ADDER can optimize for actuator applications. Following this, we provide an overview of several IoT actuator use cases, alongside an analysis of their respective traffic patterns, which will serve to formulate the problem in the subsequent sections.

A. Energy Saving Mechanisms and Related Parameters

The RRC layer is the central controller, managing radio communications between the mobile device and the network. Regarding energy management, the RRC layer plays a critical role by negotiating with mobile devices to establish an agreement on the devices' communication modules' sleep and awake time intervals. Following the schedule, mobile devices shift between different RRC states, which governs their radioactivity. This process is crucial for balancing energy efficiency with the need for consistent and reliable communication.

DRX and PSM Mechanism. DRX [1], [2] allows devices to enter a low-power state while periodically checking for incoming traffic, necessitating maintenance of network synchronization and readiness for immediate wake-up and data reception. On the other hand, PSM [3], [4] puts the device into a deeper sleep, shutting down more functionalities, completely disconnecting from the network and achieving more significant energy savings.

Energy Saving Parameters. Fig. 1 illustrates the relationship between radio activity, RRC states, and parameters for DRX and PSM. To save energy during the RRC inactivity timer, a DRX Mechanism in an RRC-connected state called cDRX was introduced, and its configurations mainly include the OnDurationTimer and cDRX cycle.

In the RRC Idle state and PSM, critical parameters such as Paging Frame (PF), Paging Occasion (PO), Paging Time Window (PTW), eDRX cycle, and T3324 play vital roles in balancing device availability and energy conservation. PF determines the specific periods for the device to wake up and check for paging messages, while PO specifies the exact instances within these frames, enhancing energy efficiency. The PTW defines a sub-duration within the DRX cycle for receiving paging messages, offering additional power savings. The eDRX cycle extends these wake-up intervals, allowing for prolonged low-power states and substantial energy conservation. Lastly, T3324 manages the transition duration from active to idle states, preventing unnecessary state changes and further contributing to power efficiency. The time following the expiration of Timer T3324 up until Timer T3412 runs out is referred to as PSM. At the end of this period, as marked by Timer T3412's expiration, UEs send TAU requests to reconnect to the network.

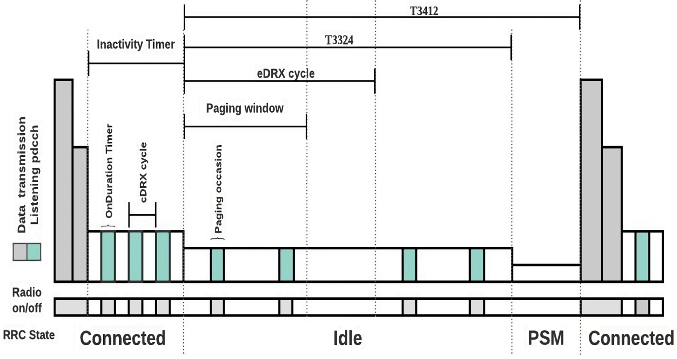

Fig. 1. RRC states, radio activity and timers for NB-IoT and LTE-M. (While there are small differences between the RRC parameter options and NB-IoT has unique and mandatory EPS optimization, the diagram applies to both protocols within the context of ADDER.)

Traditionally, RRC parameter values are initially determined during the first RRC setup, followed by negotiations between UEs and MMEs to refine these settings. These negotiations employ specific Information Element Identifiers (IEIs) for the parameters above during RRC Connection Reconfiguration and TAU procedures. However, Early Data Transmission (EDT) in 3GPP Release 15 [8] has eliminated the need for conventional RRC Connection Reconfiguration, favoring an event-triggered TAU. This approach offers significant advantages in specific scenarios, such as low battery situations or unique IoT application requirements. In Section III, the workflow depicts how ADDER leverages event-triggered and periodic TAUs for timely reconfiguration of energy-saving parameters.

B. *Targeted Use Cases and Traffic Patterns*

Earlier work performed measurements with IoT devices over commercial networks in Norway to assess how traffic load and RCC parameter settings affect battery life expectancy [9]. As expected, message frequency is the dominant factor determining lifetime, followed by the setting of DRX, signal quality, and finally packet size. These findings further *motivate* ADDER*'s approach to adapting energy saving parameters based on traffic behavior.*

TABLE I
TYPICAL TRAFFIC PATTERNS IN USE CASES OF INTEREST.

Usecases	***MT Message Frequency***	***Delay Tolerance***
Lighting control	irregular	15 seconds
Smart Irrigation	irregular , infrequent	1 minute
Smart appliances	irregular, infrequent	3 seconds
Bike Sharing	irregular	10 seconds

Table I presents various IoT actuator applications that predominantly wait for irregular mobile-terminated (MT) messages, along with their associated delay tolerance as referenced in [10]. It is worth noting that these applications typically perform periodic updates of their state or send heartbeat signals via MO messages. However, our primary focus in this context is on optimizing energy consumption for MT messages. Delivering timely and accurate commands to IoT devices through MT messages is vital, but may come at the cost of increased energy consumption.

III. METHODOLOGY

A. *Problem Formulation*

Conserving energy in an IoT actuator needs accurate prediction of packet (paging request) arrival times within a time window to ensure responsiveness to anticipated requests while transitioning to sleep mode during periods of inactivity. However, due to the unpredictable nature of burst traffic, the actuator must carefully balance entering sleep mode for energy savings with frequent monitoring to avoid missing packets. This challenge can be framed into a bandit problem, where decisions are made without full knowledge of the environment, akin to choosing between sleep and monitoring without knowing the exact timing of packet arrivals. Although precise request times are unattainable, estimating the probability of a packet arriving within a given time window is feasible. With such a probability estimation, one can set a threshold to omit infrequent requests with low arrival probabilities. When the request probability is moderate, reduced monitoring suffices; however, frequent checks or ensuring delay tolerance are inevitable for high-probability scenarios. Historical data are proved valuable in forecasting these probabilities, with time series analysis revealing busy and quiet periods. Apart from time-centric factors, external conditions also play a role. The influence of days of the week, holidays, and local events must be considered when shaping these patterns for more accurate prediction. For instance, fields may not require irrigation following prolonged rain, and areas experiencing shared-bike shortages often see increased demand.

B. ADDER *Design*

ADDER addresses the energy parameter setting challenge, which is framed as a contextual bandit problem, by employing a modified DDPG algorithm. DDPG stands out due to its deterministic policy, which directly selects the optimal action for each state. Moreover, its ”deep” neural network architecture effectively handles the high-dimensional action spaces that involve numerous energy-saving parameters. This sets it apart from AC-DRX [6], another actor-critic framework that relies on tabular learning and stochastic policy gradients. While AC-DRX reduces the action space dimension to a single parameter through redefinition, it still necessitates sampling from a distribution encompassing all discrete actions and additional search efforts to find optimal actions. To train the critic and actor networks, DDPG needs training data, which can be challenging for IoT actuators receiving infrequent commands. ADDER decouples the DDPG model from specific application contexts to address this data scarcity. It accomplishes this by employing a simulator to generate training samples encompassing varying packet arrival probabilities. When configuring parameters for a specific application, a context-aware predictor estimates the packet arrival probability based on the context. Subsequently, the DDPG model utilizes this estimated probability to make informed action selections. Using synthetic datasets enables DDPG to undergo efficient training without the reliance on an extensive volume of real-world data.

C. ADDER *Architecture*

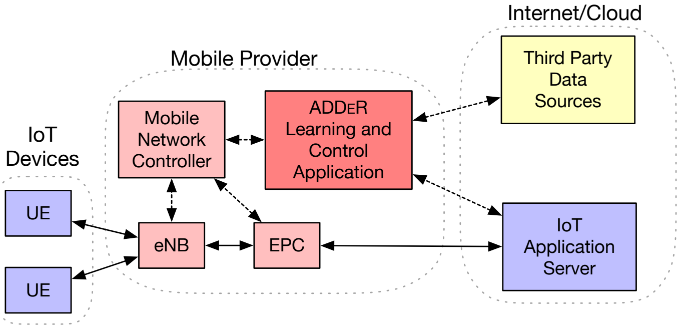

Fig. 2. ADDER Architecture

Fig. 2 shows how ADDER works and how it fits into the standard data transmission process between base stations and IoT actuators. ADDER's Learning and Control Application consists of two key components: a context-aware predictor and a DDPG-based policy maker. The predictor, using information from IoT servers and external sources, estimates the likelihood of a data packet arriving from the server in each time window. Meanwhile, a simulator generates training data for the policy maker by simulating traffic with varying arrival probabilities within a time window. This simulation records the resulting latency and energy consumption given the device power profile, DRX and PSM parameters, and service latency requirements. Following the model's training, the policy maker utilizes the predicted probability in conjunction with a predefined threshold specific to the service to determine optimal parameter settings. These newly determined parameter configurations are then transmitted to the controller, where a parameter update process is initiated to ensure that both the network and end devices are effectively prepared for the impending changes. The updated configurations are included in a TAU response. When the network receives a TAU request from the end devices, it sends them the updated settings. This allows the devices to adjust their local parameters accordingly. End devices can start the TAU request through periodic updates or utilize the on-demand TAU. While reconfiguring parameters through TAU introduces overhead in control signaling, the infrequent downlink packets (a quarter hour to serval hours) characteristic of IoT actuator-centric applications makes the overall signaling overhead manageable.

D. Context-aware Predictor

Accurately forecasting application traffic remains the primary challenge in setting power-saving parameters. Time series analysis, a common technique employed in previous studies, typically concentrates on the traffic patterns of individual devices. Even when applied to forecast the traffic pattern for a service, it may not fully capture the infrequent, unpredictable, and intermittent nature of IoT actuator applications. Consider a bike-sharing service, where usage patterns can fluctuate significantly based on the time of day, day of the week, weather conditions, and specific events. Even with a consistent usage pattern, the traffic pattern of each bike is likely to differ. While primary time series data might be adequate for tracking daily peaks, identifying variations across weekdays versus weekends or comprehending seasonal and annual shifts—such as those observed during pandemic years—requires more training data, which may be impractical. By integrating more "context" from the environment, ADDER can craft more precise traffic forecasts. This enables finer power-saving strategies and bolsters the efficiency of IoT actuator applications. In specific scenarios detailed in Section IV, our case study with a bike-sharing service revealed that ADDER surpassed conventional time series forecasting models like ARIMA [11] and LSTM [12] in predicting user demand patterns.

E. Modified DDPG

We consider the problem under the scenario that an agent operates within an independent and identically distributed (i.i.d) contextual bandit framework that allows for continuous action choices. At each time step, denoted as t, and gathers a state vector $x_t = [x_{t1}, x_{t2}]$ from the state space $\mathcal{X} = [0,1]^2$. The completion of each time step is marked by a TAU update. At the beginning of each step, the new energy-saving parameters for the step set up, and the timer for the periodic TAU is reset. If no requests are received before the timer runs out, the step terminates with the periodic TAU. If a request arrives before the timer reaches zero, the UE triggers an on-demand TAU, which completes the current step. x_{t1} is the estimated packet arrival probability. x_{t2} stands for a threshold determined by the IoT service provider. When x_{t1} falls below this threshold, energy considerations become paramount; conversely, if x_{t1} surpasses the threshold, latency takes precedence. Following this, the agent selects an action a_t from action space $\mathcal{A}$, where $\mathcal{A} = [-1,1]^N$ and receives reward $r_t = \mathcal{R}(x_t, a_t)$, where $\mathcal{R} : \mathcal{X} \times \mathcal{A}$. To simplify the problem, we leave out the settings for cDRX and PTW, only focus on the configurations of T_{3324}, T_{eDRX} (eDRX cycle), and T_{3412}, which results in $N = 3$. We modify the values of T_{3324} and T_{3412} in terms of TTI units (equivalent to the duration of a subframe), T_{eDRX} is expressed in units of 10 TTI (equivalent to the duration of a radio frame). These values can vary from 0 up to the maximum TTI in the time window of a single step and are mapped onto the action space respectively. These values were originally set as fixed values in accordance with the 3GPP protocol.

The primary objective for ADDER is to prioritize meeting delay tolerance requirements during service peak hours while concurrently aiming to maximize energy conservation during off-peak periods. Thus we define the reward function $\mathcal{R}$ as follows:

$$\mathcal{R} = \begin{cases} D_{a_t} + L_{a_t} & \text{if } x_{t1} > x_{t2},\, S > d \\ (1 - D_{a_t}) + E_{a_t} + L_{a_t} & \text{if } x_{t1} > x_{t2},\, S <= d \\ \alpha D_{a_t} + E_{a_t} + L_{a_t} & \text{if } x_{t1} <= x_{t2} \end{cases} \quad (1)$$

where d represents the delay tolerance, a variable contingent on the specific requirements of the IoT application and S is the maximum delay achievable by the selected action set. E_{a_t} and L_{a_t}, representing the normalized energy cost and normalized actual latency cost, respectively. D_{a_t} denotes the normalized distance between the maximum delay of the selected action set and the delay tolerance, i.e. $d-S$. The coefficient α is a control parameter that can be used to specify the relative importance of the maximum delay cost over energy cost depending on the IoT application requirements.

D_{a_t} serves as a mechanism for constraining action selection during peak hours. It imposes a penalty in the form of a negative reward when an action set chosen results in a waiting time exceeding the specified delay tolerance. This penalty is essential for preventing the model from overly prioritizing energy savings in scenarios where the probability of a request arrival is low but still categorized as peak hours according to the IoT application's requirements. It ensures that the model maintains latency within the delay tolerance. Conversely, D_{a_t} grants positive rewards for actions when the maximum delay incurred by the action set closely approaches but remains below the delay tolerance. This reward system discourages the model from becoming excessively focused on minimizing latency costs to the detriment of other factors. The coefficient α is a control parameter that can be used to specify the relative importance of the maximum delay cost over energy cost depending on the IoT application requirements. This approach may result in longer wait times during off-peak hours, potentially leading to the loss of some customers.

Businesses have the flexibility to adjust their threshold x_{t2} to strike a balance between device maintenance and customer profitability, striving for maximum overall gain.

The process is then repeated with a new state at time $t+1$. Unlike the standard RL setting, there is no transition function in the bandit setting. We defined deterministic policy μ_θ that maps states to specific actions in $\mathcal{A}$, parameterised by θ. A deep contextual bandit agent for continuous actions based upon the DDPG algorithm [13] is used in ADDER to help deal with the one step RL problem. DDPG is a model-free, off-policy reinforcement learning algorithm designed specifically for environments with continuous action spaces. Combining concepts from Deep Q-Learning (DQN) and Actor-Critic methods, DDPG utilizes two neural networks: an actor network that determines the optimal action given a state, and a critic network that evaluates the quality of a given state-action pair. The actor produces a deterministic policy, guiding the agent towards the best perceived actions, while the critic estimates the Q-value of the chosen action, helping to refine the actor's decisions.

Usually, as a solution for MDP problem, Q-value represents the expected cumulative future reward an agent can obtain, starting from state s, taking action a, and thereafter following a specific policy and is described and updated using the Bellman equation. As a bandit problem, the selected action won't affect the future state, thus the optimal one-step Q-value for (x_t, a_t) as follows:

$$Q^*(x_t, a_t) = \mathbb{E}[\mathcal{R}(x_t, a_t)], \tag{2}$$

We refine the critic network using the following mean-squared error (MSE) function to bring our estimated Q-values Q_ϕ closer to the optimal Q^*.

$$L(\phi, \mathbf{E}) = \underset{(x_t, a_t, r_t)\sim\mathbf{E}}{\mathbb{E}}\left[\left(Q_\phi(x_t, a_t) - r_t\right)^2\right], \tag{3}$$

where ϕ represents the critic network parameters. Given that our focus is solely on the immediate rewards of each step, rather than the cumulative reward, we removed the target networks.

Since the actor network produces a deterministic action for each state, a behaviour policy β is used for exploration, by adding noise sampled from a one-step Ornstein-Uhlenbeck process $\mathcal{N}$ to the deterministic actor policy:

$$\beta = \mu(x_t|\theta_t^\mu) + \mathcal{N}, \tag{4}$$

We update the deterministic policy gradient for the contextual bandit setting by:

$$\begin{aligned}\nabla_\theta \mathbf{J}_\beta(\mu_\theta) &= \int_{\mathcal{X}} \rho^\beta \nabla_\theta \mu_\theta(a|x) Q^\mu(x, a) ds, \\ &= \mathbb{E}_{x_t\sim\mathbf{E}}\left[\nabla_\theta \mu_\theta(x_t) \nabla_a Q^\mu(x_t, a_t)|_{a_t=\mu_\theta(x_t)}\right],\end{aligned} \tag{5}$$

where, $\rho^\beta(x)$ defines the state distribution visited under policy β, which in the bandit setting is equivalent to sampling states from our environment $\mathbf{E}$.

TABLE II
CURRENT CONSUMPTION PARAMETERS

Symbol	Description	Value
I_{PSM}	PSM floor current	2.7μA
I_{PO}	Average current during PO	5913μA
I_{eDRX}	Average current during idle edrx	23.82μA
I_{inact}	Average current during RRC inactivity	15345μA

IV. EVALUATION

In this section, we will first describe the configuration of ADDER, followed by an evaluation of ADDER from three perspectives: comparing ADDER to other baseline methods in scenarios involving nonstationary and infrequent traffic, demonstrating the utility of ADDER across various application services with distinct objectives regarding energy conservation and latency, and finally, we examine the impact of traffic prediction errors on the performance of ADDER with a real-life use case.

A. Simulator

We use a simulator to generate the data samples needed to train our model. The simulator creates packets based on a random packet arrival probability within a given time window. It precisely mimics the behavior of UE as it transitions between different states according to an energy-saving mechanism when handling downlink packets. We record the energy consumption and resulting latency for training. The energy consumption in the simulator is calculated based on the specifications from the nRF9160 by Nordic Semiconductor, a System-in-Package (SiP) that facilitates low-power cellular IoT designs with its modem support. This SiP is compatible with both PSM and eDRX power conservation techniques. Table IV-A provides the power characteristics of the nRF9160 rev2 chip operating in NB-IoT network mode with a voltage of 3.7v. To simplify the simulation, we assume that during each TAU from PSM, we switch to the RRC connection state for 2 seconds. It is important to note that no data transmission occurs within the duration, equivalent to the RRC inactivity state. Our simulation environment is built on Gymnasium's APIs [14]. While there is an ns-3-based model for LTE energy conservation, ns-3's extensive components tend to slow down data acquisition. Our simulator has significantly improved data collection speed, approximately eight times faster than the ns-3-based model.

B. DDPG configuration

Since target networks are removed from the DDPG algorithm, we only need to configure the actor and critic networks. The actor-network comprises three hidden layers, each employing a ReLU activation function. It culminates in a Tanh activation function in the output layer. The output is then mapped to the appropriate parameter value ranges, and the resultant action values are input into the simulator. These hidden layers consist of 64 units, 32 units, and the dimension of the action space, respectively. Meanwhile, the critic network utilizes two ReLU hidden layers comprising 32 units to process the observations. It combines these processed observations with the actions and passes the result through three additional ReLU hidden layers. These hidden layers in the critic network also consist of 64 units, 32 units, and finally, 1 unit. During the training process, we employ a batch size of 256, and both the actor and critic networks use a learning rate of 0.0005.

C. Performance

Baseline Methods To evaluate our approach, we compare it with two baselines. The first baseline, EDRX, uses a static DRX cycle matching the delay tolerance parameter. Our second comparison is AC-DRX, an Actor-critical-based approach that combines tabular V value appraisal as a critique with a stochastic actor policy for improved energy efficiency.

To compare AC-DRX with ADDER, we need to modify AC-DRX. First, the original AC-DRX framework defines decision windows (DWs) based on a predetermined number of incoming requests. Each incoming request is associated with a corresponding RRC state, and these states are subsequently used to determine the action to be taken for the next DW. We redefine the DW to be measured by the number of steps (corresponding to each TAU update, as detailed in Section 1) instead of the number of incoming requests. This modification introduces a new state into AC-DRX to account for cases when no packets are received within a step. Furthermore, the action in the original AC-DRX framework is confined to determining the length of the DRX cycle. It does not encompass the timing configurations for PSM, which is referred to as the RRC idle mode in their setup. To align with our method's action space for comparative analysis, we adapted the original AC framework to include the parameter setting for PSM. This required a transition to a DDPG strategy better suited to our broader action space. For the critic component in AC-DRX, we replaced the tabular approach for estimating the V value for a state with a deep Q-learning algorithm for a state-action pair. We retained their rule for updating the V value, which utilizes accumulated rewards to inform the Q-learning process. We continue using their defined reward function to ensure the model benefits from their established conceptual groundwork.

Setup In our analysis, we set a fixed periodic TAU period for each step and configured the AC-DRX model with DWs spanning five steps each. While the packet arrival probability remains constant during a single DW (five steps for ADDER), it varies across different DWs to replicate the fluctuating peak and off-hour traffic patterns frequently observed in IoT environments. ADDER is provided with the arrival probability and a threshold. This threshold determines the extent of savings achieved when the actual usage probability falls below it. ADDER 10 and ADDER 40 come with thresholds of 10 and 40, respectively, symbolizing the service's varying needs for real-time responses. The higher the threshold, the greater the demand for immediate responses. Our primary objective in this section is to compare ADDER to AC-DRX and EDRX in nonstationary and infrequent traffic scenarios and showcase how ADDER adapts to service latency requirements according to the given threshold.

Results Fig. 3 and 4 show the performance of the methods, evaluated over time windows of one and two hours per step, respectively. The time window lengths are pertinent to applications with infrequent traffic, such as actuators receiving commands every few hours. The figures employ a Cumulative Distribution Function (CDF) plot for energy consumption analysis, measured in 100-step increments. For the one-hour time window, approximately 7.5% of the samples for the ADDER 40 had latencies over 10 secondsm which results from the threshold that disregards packets that are unlikely to arrive, yielding substantial energy savings. ADDER 40 predominantly shows energy usage under 30 Joules per 100 steps, which stands out against the other methods exceeding 30 Joules. Notably, ADDER 40 achieves a 26% reduction in total energy consumption compared to EDRX. Meanwhile, ADDER 10 consistently shows latencies under 10 seconds, with an average of around 5 seconds and a maximum of 10 seconds, but it only offers a 5.7% energy saving over EDRX. Clearly, adjusting the threshold allows ADDER to modify its performance in response to the demand for low latency. Both EDRX and AC-DRX maintain median latencies below one second. However, AC-DRX, designed to focus on energy efficiency, shows a more varied latency range and realizes an energy saving of 2.9% .

For the two-hour time window, ADDER 40 is more energy-efficient, saving approximately 32.7% more energy than EDRX, while ADDER 10 saves around 5.9%. ADDER 40 exhibits a 9% occurrence of samples with latency exceeding 10 seconds, whereas all the samples for ADDER 10 stay within the latency boundary. We observed that ADDER 40 consumed less energy than in a one-hour time window, resulting in ADDER's tendency to adopt riskier strategies as traffic becomes less frequent. ADDER switches to PSM when the probability is low. When the probability is above the threshold but remains below a certain level, ADDER uses a combination of PSM and TAU to save energy. On the other hand, AC-DRX consumes more energy than EDRX. This is because AC-DRX lacks information about the probability of packet arrival, often assuming that a higher packet arrival rate during the on-duration of a DRX cycle in a DW implies a lower probability of packet arrival in the next DW. Consequently, AC-DRX is inclined to enter PSM for extended off durations to maximize rewards while increasing the frequency of packet checks during the RRC idle mode to avoid potential penalties on latency. However, their reward function does not consider the variations in energy costs associated with extended sleep periods in PSM, the RRC idle mode, and the energy consumption for TAU after PSM. This oversight is one of the reasons why AC-DRX performs less effectively.

D. Case Study: Shared Micro-mobility Vehicle in Austin, Texas

In this section, we explore ADDER's performance under nonstationary traffic conditions without providing it with packet arrival probabilities. Recognizing that a packet arrival

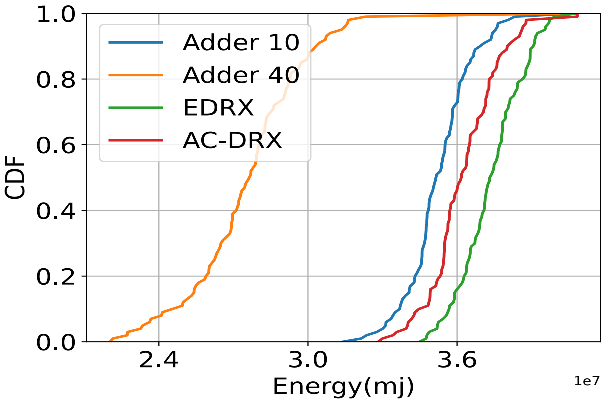

Fig. 3. CDF for energy consumption when a single packet randomly arrives within a one-hour time window

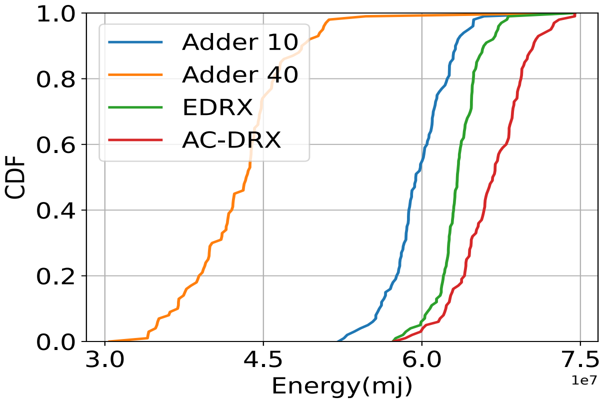

Fig. 4. CDF for energy consumption when a single packet randomly arrives within a two-hour time window

estimator can only offer approximations, which might affect ADDER's effectiveness, we delve into a case study of scooter sharing in Austin, Texas, to assess potential performance degradation due to traffic prediction errors. Despite scooters having larger batteries and less concern for communication energy than bike sharing, the user behavior and environmental impacts on usage patterns between the two are similar enough to make the scooter-sharing scenario relevant to our study. Given our emphasis on energy-sensitive applications, we will refer to"scooters" as "bikes" in our analysis to maintain clarity.

1) Data Overview: Our dataset comprises two components: the usage statistics and the contextual information. The usage statistics source from the Austin Transportation Department's "Shared Micro-mobility Vehicle Trips" report [15] related to a scooter sharing service. We selected a subset of data from the West Campus neighborhood in central Austin, where the records from March 2019 to June 2019 are notably comprehensive and consistent. This dataset features device ID, census tract numbers for starting and concluding locations, trip duration, and a 15-minute interval marking the trip's initiation and conclusion (for safeguarding user privacy). The accompanying contextual data encapsulate local events and weather information. Publicly available data sources provided information on campus happenings, such as school vacations and national holidays. We extracted hourly historical records from Wunderground [16] for weather data, narrowing our focus to the Austin-Bergstrom International Airport Station. This station is located just 9 miles from West Campus and offers the most detailed weather records nearby.

The demand-supply ratio indicates the probability that a bike will be unlocked, highlighting service dynamics during both high-demand (peak) and low-demand (off-peak) periods. This ratio is derived by dividing the forecasted bike demand by the bike service companies' supply. However, the specific data on the number of bikes in operation is not provided, necessitating an estimation of the total bike supply. This estimation is based on the anticipated bike demand and the average turnover rate, which measures how frequently a bike is used within a given fleet size and demand context. Our analysis adopts a turnover rate assumption of twice per vehicle, in line with the estimation approach outlined in [17]. By dividing the total number of bike trips by this turnover rate, we can estimate the available number of bikes, thereby enabling us to calculate the demand-supply ratio.

2) Performance of Context-aware Predictor: The usage data are divided into training and testing sets. Any missing values are filled in with their respective mean values, and the dataset is subjected to min-max normalization. To assess the stationarity of the usage data, we utilize the Augmented Dickey-Fuller (ADF) test. This test, known as a unit root test, measures the impact of a trend on a time series. The results, which include the p-value and ADF statistics from the ADF test, are presented in Table III. Since the p-value is below 0.05, it confirms that the series is indeed stationary. Therefore, time series-based algorithms can be employed for usage prediction.

The ARIMA Model We first apply an ARIMA model to the bike usage dataset and then analyze the residual errors. The PACF plot in Fig. 5 highlights a distinct peak at lag 1, suggesting nonseasonal behavior. The ACF follows a diminishing trend, pointing to an $AR(1)$ component. This suggests a nonseasonal model component of $(1, 0, 0)$. Regarding seasonal patterns, the seasonal period is defined as $S = 24$. The ACF and PACF exhibit significant autocorrelation at lags of 24 and 48. A notable grouping is seen around lag 24 in the ACF. Additionally, the PACF presents peaks at two intervals of S, leading us to choose $AR(2)$. Consequently, the seasonal portion of the model is designated as $(2, 1, 0, 24)$.

The LSTM Model We then employ LSTM for traffic forecasting. In bike-sharing, the time series data are converted into instances with an input-output structure with a lag of 24. The chosen model comprises two hidden layers, each containing 50 neurons. The Adam algorithm is utilized as the optimization method, and the mean squared error (MSE) is the loss metric. The set parameters include a learning rate of 0.0008, a dropout rate of 0.2, a batch size of 50, and 800 epochs.

Context-aware Predictor The ADDER constructs a Context-aware Predictor utilizing a neural network (NN).

TABLE III
ADF TEST RESULTS ON THE TIME SERIES.

	ADF Statistic	***p-value***	***Critical Value 1%***	***Critical Value 5%***	***Critical Value 10%***
Value	-4.71	$8.12e^{-5}$	-3.43	-2.86	-2.57

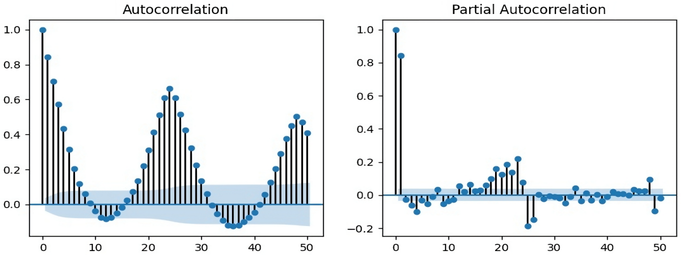

Fig. 5. ACF and PACF

TABLE IV
RESULTS OF THE MLPREGRESSOR PERFORMANCES FOR THE BIKE-SHARING CASE WITH 3 DIFFERENT LOSS FUNCTIONS.

Method	***Test***			***Rainy***		
	MAE	***MSE***	***RMSE***	***MAE***	***MSE***	***RMSE***
ARIMA	14.42	359.65	18.96	27.82	1588.60	39.86
LSTM	19.7	591.29	24.31	32.592	2010.48	44.83
NN	14.27	346.92	18.63	21.09	899.22	29.99
Method	***Holiday***			***Windy***		
	MAE	***MSE***	***RMSE***	***MAE***	***MSE***	***RMSE***
ARIMA	24.59	789.10	28.09	85.59	10069.63	100.34
LSTM	20.58	500.52	22.37	21.54	731.06	27.038
NN	11.07	176.43	13.28	22.05	752.02	27.42

Features influencing commuting behaviors, such as hourly temperature, rainfall, wind intensity, holidays, day of the week, and hour of the day, are employed to train this NN. This network, specifically a multilayer perceptron (MLP) regression, consists of three hidden layers with five hidden units in each layer. The Adam algorithm is chosen for optimization, and the ReLU function is used for activation.

To assess the performance of the models, we used three metrics: mean absolute percentage error (MAPE), mean squared error (MSE), and root mean squared error (RMSE). In addition to the standard test set, we evaluated the models on data from rainy days, holidays, and windy days. We selected data for these specific test sets using control variables, filtering out entries with normalized hourly precipitation or wind speeds greater than 0.5. The accuracy results are presented in Table IV.

The Neural Network (NN) model outperforms the other two methods regarding predictive accuracy on rainy days and holidays, while it performs similarly to LSTM on windy days. Since ARIMA and LSTM are primarily time series models, it is expected that the NN would outperform them in less time-dependent situations. This finding is consistent with previous studies on bike-sharing system demand prediction [18]–[20], which suggest that both regular (e.g., time and weather) and opportunistic contextual factors (e.g., social and traffic events) play essential roles in forecasting bike usage patterns.

3) Decision Making based on the Context: The discrepancy between the anticipated and actual traffic patterns could result in the selection of an alternative action, impacting energy efficiency and latency. In this paragraph, we demonstrate how this variance in performance plays out using the bike-sharing scenario as an example. Fig. 6 depicts the performance for ADDER act (based on actual packet arrival probability), ADDER pred (based on estimated packet arrival probability), and EDRX approaches, respectively. We have set the threshold at the 30th percentile of the daily packet arrival probability for both ADDER act and ADDER pred. The latency plot for the three methods reveals that most latency measurements range from 0 to 10 seconds, averaging around 5 seconds. The box plot does not include outliers, which shows that around 9% of ADDER act samples have latencies over 10 seconds, while about 12.5% of ADDER pred samples exceed this latency threshold, suggesting that prediction errors can increase latency. Regarding energy consumption, both ADDER act and ADDER pred are more efficient than EDRX, with the ADDER act curve higher than ADDER pred, indicating greater energy efficiency when using actual data. The experimental results demonstrate that devices equipped with the nRF9160 NB-IoT module save 14.5% of energy using actual probability and 12.9% with predicted probability over a one-day period.

V. LIMITATION AND FUTURE WORK

The current simulator oversimplifies energy-saving mechanisms in state transitions by assuming a constant power profile and ignoring link quality factors such as signal strength and noise levels. Creating a more practical simulator that accounts for these real-world complexities is a significant undertaking. This is particularly true when considering link quality, which demands domain knowledge or extensive samples for ML-based approaches which necessitate a testing phase to determine the cost of different actions and retraining after deployment if environmental conditions change.

Aside from creating a more realistic simulator, it's essential to recognize that in actual systems, the action space is not continuous but rather high-dimensional. The current parameter configuration relies on binary numbers to represent different parameter values. To effectively use the Adder's action spaces, a more efficient approach for expressing parameter values with minimal signaling is required.

While we sidestep the energy-intensive process of online training, the implications of this approach on network providers' energy usage, particularly in terms of scheduling parameters, remain uncertain. Moreover, fluctuating dynamic DRX cycles for various applications within the same cell

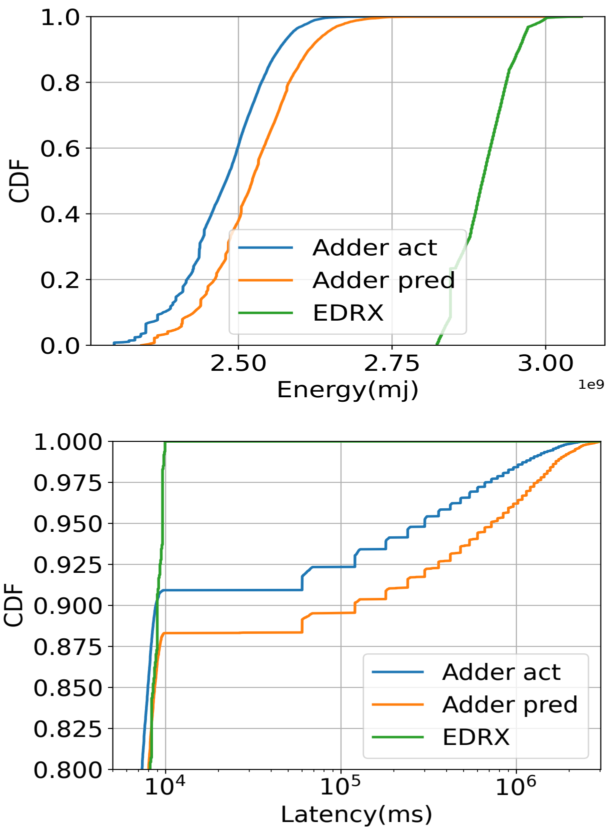

Fig. 6. Adder's performance, in terms of energy consumption and latency, based on packets generated according to the records from the Shared Micromobility Vehicle in Austin, Texas.

complicate the scheduling process, which may lead to less efficient spectrum use. Exploring the potential of reinforcement learning to address the challenges of setting energy parameters and optimizing spectrum and time scheduling could be a promising direction for future research.

VI. Conclusion

We introduced our research on Adder, an adaptive data-driven method for adjusting energy-saving mechanism parameters. Adder utilizes contextual data for IoT applications and employs a machine learning-based approach on the network side to dynamically adjust RRC parameter settings. Our research demonstrate that Adder offers a customizable balance between energy efficiency and latency, allowing IoT service providers to tailor energy-saving network services to specific service requirements. We assessed the practicality of our approach through simulations using real-world data from a mobility-sharing service.

VII. Related work

Earlier efforts [21] adapt the DRX mechanism for different services such as web browsing, VoIP service, video and multimedia services, ultra-reliable low latency communication and others. These services demand rapid responses, so DRX in the RRC connection is commonly employed to conserve energy for such services. On the other hand, for IoT applications with limited energy resources and infrequent traffic, DRX in RRC idle mode and PSM are more frequently used [22]–[25]. The tradeoff between latency and energy motivates many studies to develop models for setting DRX parameters [26]–[28]. Adaptive DRX parameter methods have been proposed to strike this balance by considering traffic characteristics. For example, a Counter-Driven Adaptive DRX scheme adjusts DRX cycles based on traffic characteristics through cooperation between base stations and the end devices, and reduces the signaling required for RRC reconfiguration [29]. Other earlier work has compared the performance of the DRX mechanism with adjustable and non-adjustable DRX cycles for bursty packet data traffic [30]. Quality of service (QoS) or Channel Quality Indicator (CQI) is essential for service latency requirements; DRX cycles can be extended or reduced to satisfy applications' real-time needs [31]–[33]. Researchers have also started using AI methods to automatically adjust DRX configurations. For instance, there have been proposals to use a trained recurrent neural network (RNN) model to implement dynamic short or long sleep cycles [34]. Long short-term memory (LSTM) approaches for forecasting traffic have also been suggested [35], [36] and these have been extended to 5G scenarios in [37]. Finally, there have been RL-based approaches for adjusting DRX cycles by learning traffic statistics and testing them with different packet arrival distributions [7].

References

[1] 3rd Generation Partnership Project (3GPP), "LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control (MAC) protocol specification," 3rd Generation Partnership Project (3GPP), Technical Specification TS 136 321 V15.2.0, 2018.

[2] ——, "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2," 3rd Generation Partnership Project (3GPP), Technical Specification TS 136 300 V14.3.0, 2017.

[3] ——, "Architecture enhancements to facilitate communications with packet data networks and applications," 3rd Generation Partnership Project (3GPP), Technical Specification TS 123 682 V15.5.0, 2018.

[4] ——, "Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS); Stage 3," 3rd Generation Partnership Project (3GPP), Technical Specification TS 124 301 V15.4.0, 2018.

[5] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[6] J. Zhou, G. Feng, T.-S. P. Yum, and S. Qin, "Actor-critic algorithm based discontinuous reception (drx) for machine-type communications," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 1–7.

[7] J. Zhou, G. Feng, T.-S. P. Yum, M. Yan, and S. Qin, "Online learning-based discontinuous reception (drx) for machine-type communications," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5550–5561, 2019.

[8] A. Hoglund, D. P. Van, T. Tirronen, O. Liberg, Y. Sui, and E. A. Yavuz, "3gpp release 15 early data transmission," *IEEE Communications Standards Magazine*, vol. 2, no. 2, pp. 90–96, 2018.

[9] F. Michelinakis, A. S. Al-Selwi, M. Capuzzo, A. Zanella, K. Mahmood, and A. Elmokashfi, "Dissecting Energy Consumption of NB-IoT Devices Empirically," *IEEE Internet of Things Journal*, 2020.

[10] J. Mocnej, A. Pekar, W. K. Seah, and I. Zolotova, *Network traffic characteristics of the IoT application use cases*. School of Engineering and Computer Science, Victoria University of Wellington, 2018.

[11] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[12] J. Patterson and A. Gibson, *Deep learning: A practitioner's approach*. " O'Reilly Media, Inc.", 2017.

[13] P. Duckworth, B. Lacerda, K. Vallis, and N. Hawes, "Reinforcement learning for bandits with continuous actions and large context spaces," 2022, unpublished.

[14] M. Towers, J. K. Terry, A. Kwiatkowski, J. U. Balis, G. d. Cola, T. Deleu, M. Goulão, A. Kallinteris, A. KG, M. Krimmel, R. Perez-Vicente, A. Pierré, S. Schulhoff, J. J. Tai, A. T. J. Shen, and O. G. Younis, "Gymnasium," 2023. [Online]. Available: https://zenodo.org/record/8127025

[15] C. of Austin Transportation Department, "Shared micromobility vehicle trips," Last updated: February 20, 2021, https://data.austintexas.gov/Transportation-and-Mobility/Shared-Micromobility-Vehicle-Trips/7d8e-dm7r.

[16] "Wunderground austin,tx weather history," https://www.wunderground.com/history/daily/us/tx/austin/KAUS, accessed: 2021-05-31.

[17] M. Hua, X. Chen, S. Zheng, L. Cheng, and J. Chen, "Estimating the parking demand of free-floating bike sharing: A journey-data-based study of nanjing, china," *Journal of Cleaner Production*, vol. 244, p. 118764, 2020.

[18] L. Chen, D. Zhang, L. Wang, D. Yang, X. Ma, S. Li, Z. Wu, G. Pan, T.-M.-T. Nguyen, and J. Jakubowicz, "Dynamic cluster-based over-demand prediction in bike sharing systems," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2016, pp. 841–852.

[19] K. Gebhart and R. B. Noland, "The impact of weather conditions on bikeshare trips in washington, dc," *Transportation*, vol. 41, no. 6, pp. 1205–1225, 2014.

[20] L. Consulting, "2013 Capital Bikeshare Member Survey Report,," 2020.

[21] K.-H. Lin, H.-H. Liu, K.-H. Hu, A. Huang, and H.-Y. Wei, "A survey on drx mechanism: Device power saving from lte and 5g new radio to 6g communication systems," *IEEE Communications Surveys & Tutorials*, 2022.

[22] T. Tirronen, A. Larmo, J. Sachs, B. Lindoff, and N. Wiberg, "Reducing energy consumption of lte devices for machine-to-machine communication," in *2012 IEEE Globecom Workshops*. IEEE, 2012, pp. 1650–1656.

[23] H.-C. Wang, C.-C. Tseng, G.-Y. Chen, F.-C. Kuo, and K.-C. Ting, "Accurate analysis of delay and power consumption of lte drx mechanism with a combination of short and long cycles," in *The 15th International Symposium on Wireless Personal Multimedia Communications*. IEEE, 2012, pp. 384–388.

[24] S. Jin and D. Qiao, "Numerical analysis of the power saving in 3gpp lte advanced wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 4, pp. 1779–1785, 2012.

[25] Y. Y. Mihov, K. M. Kassev, and B. P. Tsankov, "Analysis and performance evaluation of the drx mechanism for power saving in lte," in *2010 IEEE 26-th Convention of Electrical and Electronics Engineers in Israel*, 2010, pp. 000 520–000 524.

[26] K. Zhou, N. Nikaein, and T. Spyropoulos, "Lte/lte-a discontinuous reception modeling for machine type communications," *IEEE Wireless Communications Letters*, vol. 2, no. 1, pp. 102–105, 2013.

[27] A. T. Koc, S. C. Jha, R. Vannithamby, and M. Torlak, "Device power saving and latency optimization in lte-a networks through drx configuration," *IEEE Transactions on wireless communications*, vol. 13, no. 5, pp. 2614–2625, 2014.

[28] H. Ramazanali and A. Vinel, "Performance evaluation of lte/lte-a drx: A markovian approach," *IEEE Internet of Things Journal*, vol. 3, no. 3, pp. 386–397, 2015.

[29] E. Liu, J. Zhang, and W. Ren, "Adaptive drx scheme for beyond 3g mobile handsets," in *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*, 2011, pp. 1–5.

[30] S. Fowler, R. S. Bhamber, and A. Mellouk, "Analysis of adjustable and fixed drx mechanism for power saving in lte/lte-advanced," in *2012 IEEE International Conference on Communications (ICC)*, 2012, pp. 1964–1969.

[31] S. Gao, H. Tian, J. Zhu, and L. Chen, "A more power-efficient adaptive discontinuous reception mechanism in lte," in *2011 IEEE Vehicular Technology Conference (VTC Fall)*. IEEE, 2011, pp. 1–5.

[32] J. Liang, J. Chen, H. Cheng, and Y. Tseng, "An energy-efficient sleep scheduling with qos consideration in 3gpp lte-advanced networks for internet of things," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 3, no. 1, pp. 13–22, 2013.

[33] T. Kolding, J. Wigard, and L. Dalsgaard, "Balancing power saving and single user experience with discontinuous reception in lte," in *2008 IEEE International Symposium on Wireless Communication Systems*, 2008, pp. 713–717.

[34] M. L. Memon, M. K. Maheshwari, N. Saxena, A. Roy, and D. R. Shin, "Artificial intelligence-based discontinuous reception for energy saving in 5g networks," *Electronics*, vol. 8, no. 7, p. 778, 2019.

[35] M. L. Memon, M. K. Maheshwari, D. R. Shin, A. Roy, and N. Saxena, "Deep-drx: A framework for deep learning–based discontinuous reception in 5g wireless networks," *Transactions on Emerging Telecommunications Technologies*, vol. 30, no. 3, p. e3579, 2019.

[36] Y. Tian and L. Pan, "Predicting short-term traffic flow by long short-term memory recurrent neural network," in *2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*, 2015.

[37] M. L. Memon, M. K. Maheshwari, N. Saxena, A. Roy, and D. R. Shin, "Artificial intelligence-based discontinuous reception for energy saving in 5g networks," *Electronics*, vol. 8, no. 7, 2019. [Online]. Available: https://www.mdpi.com/2079-9292/8/7/778