scientific reports

OPEN

SmartAPM framework for adaptive power management in wearable devices using deep reinforcement learning

R. Sunder[1], Umesh Kumar Lilhore[1], Anjani Kumar Rai[2], Ehab Ghith[3], Mehdi Tlija[4], Sarita Simaiya[1,5]✉ & Afraz Hussain Majeed[6]

Wearable devices face a significant challenge in balancing battery life with performance, often leading to frequent recharging and reduced user satisfaction. In this paper, we introduce the SmartAPM (Smart Adaptive Power Management) framework, a novel approach that leverages deep reinforcement learning (DRL) to optimize power management in wearable devices. The key objective of SmartAPM is to prolong battery life while enhancing user experience through dynamic adjustments to specific usage patterns. We compiled a comprehensive dataset by integrating user activity data, sensor readings, and power consumption metrics from various sources, including WISDM, UCI HAR, and ExtraSensory. Synthetic power profiles and device specifications were incorporated into the dataset to enhance training. SmartAPM employs a multi-agent deep reinforcement learning framework that combines on-device and cloud-based learning techniques, as well as transfer learning, to enhance personalization. Simulations on wearable devices demonstrate that SmartAPM can extend battery life by 36% compared to traditional methods, while also increasing user satisfaction by 25%. The system adapts to new usage patterns within 24 h and utilizes less than 5% of the device's resources. SmartAPM has the potential to revolutionize energy management in wearable devices, inspiring a new era of battery efficiency and user satisfaction.

Keywords Deep reinforcement learning, Adaptive power management, Wearable devices, Energy efficiency

Abbreviations

DRL	Deep reinforcement learning
WISDM	Wireless sensor data mining
ExtraSensory	A dataset for wearable sensor data
RL	Reinforcement learning
GPU	Graphics processing unit
IoT	Internet of things
API	Application programming interface
APM	Adaptive power management
UCI HAR	University of California Irvine Human Activity Recognition
SmartAPM	Smart adaptive power management
AI	Artificial intelligence
CPU	Central processing unit
RAM	Random access memory

Wearable devices, such as fitness trackers, smartwatches, and health monitors, have become integral to daily life, providing real-time health insights and seamless connectivity. However, their limited battery life remains a critical bottleneck, significantly impacting the user experience and the practicality of continuous use. Efficient

[1]School of Computer Science and Engineering, Galgotias University, Greater Noida, UP, India. [2]Department of Computer Engineering & Applications, GLA University, Mathura, UP 281406, India. [3]Department of Mechatronics, Faculty of Engineering, Ain Shams University, Cairo 11566, Egypt. [4]Department of Industrial Engineering, College of Engineering, King Saud University, P.O. Box 800, Riyadh 11421, Saudi Arabia. [5]Arba Minch University, Arba Minch, Ethiopia. [6]School of Energy and Power Engineering, Jiangsu University, Zhenjiang 212013, China. ✉email: saritasimaiya@gmail.com

power management has thus emerged as a key challenge in wearable technology, especially as these devices grow increasingly complex with diverse functionalities[1]. Traditional power management techniques, which rely on static, predefined rules, often fail to optimise battery usage due to their inability to account for dynamic user behaviours and varying environmental contexts[2]. These methods frequently result in suboptimal power utilisation, forcing users to endure frequent charging cycles and unexpected battery depletion, which hinders the seamless use of the devices.

One of the primary challenges in managing power for wearable devices is the need to balance energy conservation with maintaining an uninterrupted, high-quality user experience. Wearable devices operate under strict resource constraints, such as limited computational power and battery capacity, while facing highly variable usage patterns that can range from sporadic, low-energy activities (e.g., sleep tracking) to intensive, high-energy tasks (e.g., GPS tracking during exercise)[3]. This complexity, combined with the real-time demands of user interaction, requires adaptive solutions capable of dynamically adjusting power management strategies on the fly. Existing approaches, such as static rule-based systems or machine learning models trained on historical data, lack the flexibility to adapt to these changing conditions and are thus inadequate for optimising battery life while preserving user satisfaction.

Motivated by these challenges, we explore the potential of Deep Reinforcement Learning (DRL) to create a more intelligent, adaptable solution to power management in wearable devices. Unlike traditional methods that follow rigid, predefined rules, DRL allows for continuous learning and real-time adaptation to new user behaviours and environmental conditions. This is particularly important in wearables, where usage patterns are highly individualised, and the ability to predict and react to shifts in user activity is crucial for optimising power consumption. DRL offers a framework for balancing power savings with user experience by dynamically adjusting device operations based on contextual data, user interactions, and system states[4–6].

Artificial intelligence and machine learning offer promising solutions to the power management challenges in wearable devices. DRL, in particular, has been successfully used to create adaptive systems across various domains, such as robotics and finance[7]. However, its application to wearable device power management remains underexplored, presenting a significant research gap. The unique constraints of wearable devices, limited computational resources, diverse usage patterns, and the need for real-time adaptation pose formidable challenges to implementing DRL in this context. Moreover, the delicate balance between power conservation and maintaining a seamless user experience adds another layer of complexity to the problem[8]. Current approaches to power management in wearables typically rely on predefined rules or simple adaptive algorithms[4]. These methods often fail to capture the full complexity of user behaviour and device states, leading to suboptimal power utilisation. As a result, users frequently experience unexpected battery depletions or are forced into frequent charging cycles, hindering the seamless use of their devices.

Despite progress in wearable technology, conventional power management methods inadequately address the complex dynamics of user behaviour and device states. Static rule-based systems are incapable of real-time adaptation, resulting in inefficient power usage and user discontent. Moreover, historical data-driven models fail to encompass the complete range of individual usage patterns, leading to recurrent unforeseen battery drain. The need for a solution that integrates a seamless user experience with energy conservation is becoming increasingly critical[9]. Diverse usage scenarios, including low-energy tasks such as sleep monitoring and high-energy activities like GPS tracking during exercise, are frequently encountered by wearable devices. This variability necessitates a power management strategy capable of real-time adjustments to user behaviours and environmental conditions. Our initiative is driven by the necessity for a sophisticated and flexible solution that can efficiently regulate power consumption in wearable devices while guaranteeing an optimal user experience. SmartAPM utilises Deep Reinforcement Learning (DRL) to create an innovative framework that learns and adapts to unique usage patterns while delivering precise control over device components, representing a substantial improvement over current power management techniques[10,11].

This paper introduces SmartAPM, a novel DRL-based approach designed to address the limitations of existing power management strategies in wearable devices. By leveraging a multi-agent architecture, SmartAPM enables fine-grained control over individual device components, optimising power usage in real-time without compromising the user experience. Additionally, the system incorporates a hybrid learning paradigm that combines on-device and cloud-based learning, allowing for both immediate responsiveness to short-term patterns and long-term optimisation across multiple users. Through the integration of transfer learning, SmartAPM rapidly personalises its behaviours to new users, ensuring rapid adaptation to individual usage patterns. A simplified overview of SmartAPM is presented in Fig. 1.

This study addresses a significant challenge in wearable technology, optimising power management to increase battery life while improving user experience in resource-constrained devices. We use deep reinforcement learning (DRL) to investigate how adaptive systems can dynamically modify power consumption in response to individual device usage patterns, resulting in a more intelligent and efficient approach to power management. This method demonstrates DRL's ability to develop energy-efficient power management solutions based on user needs and preferences. This innovation has the potential to improve the functionality of wearable devices, promoting widespread adoption and changing how users interact with these devices in everyday life[12–14].

SmartAPM differs from existing solutions in several key aspects. Unlike traditional static power management systems or simple adaptive algorithms, SmartAPM employs a sophisticated multi-agent deep reinforcement learning (DRL) approach. This allows for real-time, fine-grained control of individual device components, adapting to user behaviour patterns and environmental contexts. Moreover, SmartAPM's hybrid learning paradigm, which combines on-device and cloud-based learning, sets it apart from purely on-device or cloud-dependent solutions, balancing immediate responsiveness with long-term pattern recognition[13, 14]. This difference is illustrated in Fig. 2.

This research makes the following key contributions to the field of power management in wearable devices:

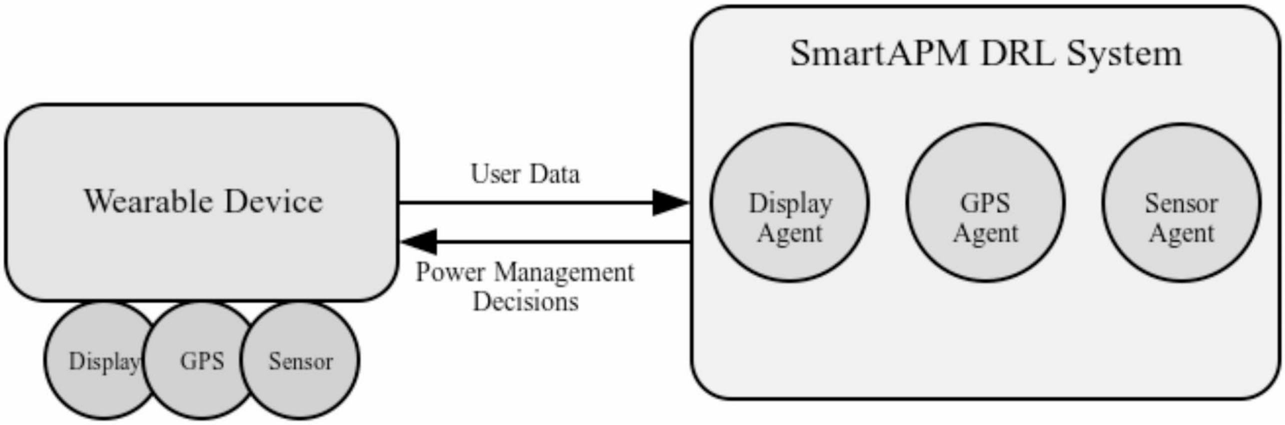

Fig. 1. Simplified overview of SmartAPM.

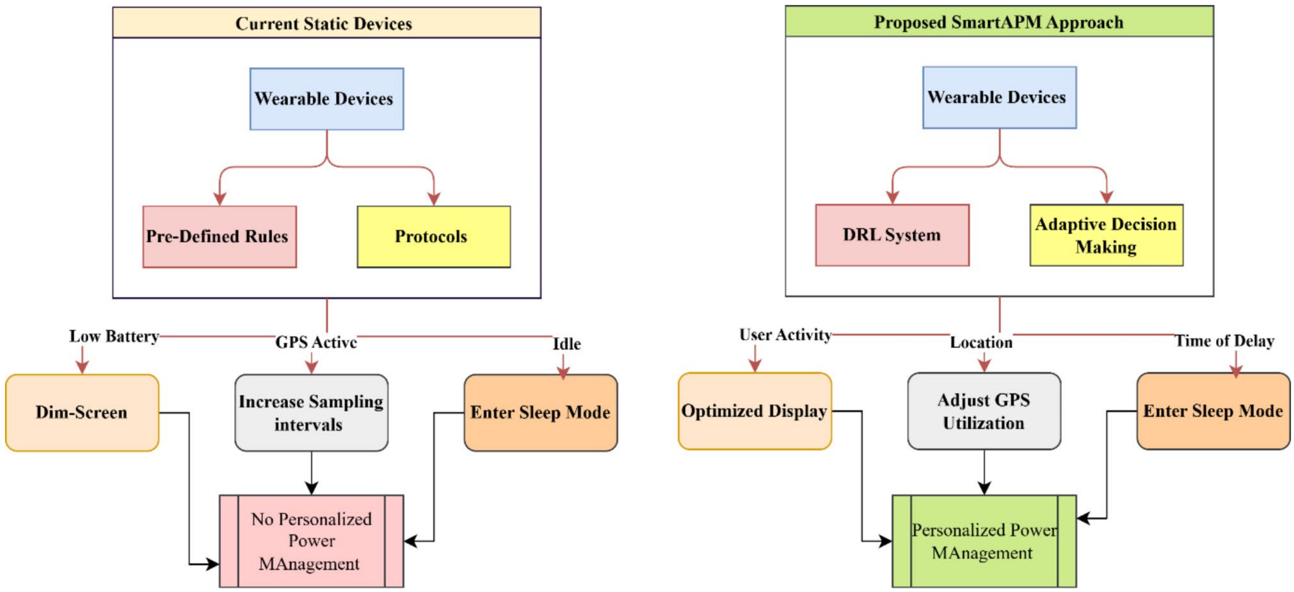

Fig. 2. SmartAPM Approach compared to static approach to Power Management.

1. Multi-Agent DRL System: Our innovative multi-agent deep reinforcement learning framework enables the precise control of wearable device components, dynamically adapting to user behaviour and enhancing power management beyond fixed regulations.
2. Custom Dataset Creation: We present a comprehensive custom dataset that integrates user activity data, sensor readings, and contextual information from multiple sources, thereby facilitating the robust training of SmartAPM[15].
3. Hybrid Learning Approach: The integration of cloud-based and on-device learning enables SmartAPM to optimise across a broader user base and respond to immediate usage patterns, distinguishing it from conventional methods.
4. Transfer Learning: By utilising transfer learning, SmartAPM can ensure that its strategies for new users are rapidly personalised, resulting in improved user satisfaction and rapid adaptation[16].

The remainder of this paper is organised as follows: In "Related methods", we review related work, detail our methodology in "Materials and methods", describe the experimental setup and results in "Experimental setup and results", and conclude with a discussion of implications and future directions in "Discussion".

Related methods

Research in wearable power management has predominantly concentrated on enhancing battery efficiency via machine learning, reinforcement learning, and cloud-based methodologies. Nonetheless, these methods frequently encounter challenges in real-time adaptability, holistic optimisation of all device components, and privacy issues, underscoring the necessity for more tailored and practical solutions. This section reviews existing research and also discusses the significant research gaps.

Current research

Recent research has conducted an in-depth investigation into a variety of approaches that aim to enhance the performance of wearables and improve their power efficiency. Technologies such as rule-based systems, machine learning algorithms, reinforcement learning (RL), and cloud-based solutions are examples of methodologies that currently exist. Each of these methods of analysis comes with its own set of advantages and disadvantages. Table 1 presents a comparative analysis towards the strengths and weaknesses of the various approaches to power management in wearable devices.

Rule-based systems

Rule-based methodologies aim to minimise power consumption by following established protocols that consider user behaviour and device context. Cho et al. (2014) developed a context-aware system for wearable devices that modulates power consumption according to user activity. These systems enhanced battery longevity by approximately 15%. Nonetheless, rule-based systems exhibit inflexibility and a lack of adaptability, making them inadequate for responding to sudden shifts in user behaviour (Duan et al., 2017). They depend on manual updates to adjust to new usage patterns, constraining their scalability and adaptability.

Machine learning models

ML techniques are becoming more prevalent in the prediction of power consumption in wearables by analysing user behaviour patterns. Li et al. (2022) employed a Random Forest classifier to enhance the power efficiency of fitness trackers and predict user behaviour. This method exhibited a 22% increase in battery lifespan. However, these methods are significantly reliant on historical data, which may not always be applicable in real-time applications (Rodríguez-Rodríguez et al., 2024).

Reinforcement learning (RL)

For the purpose of dynamically regulating power consumption, reinforcement learning has been implemented. This is accomplished by continuously acquiring knowledge from the surrounding environment and optimising actions that are carried out. The research conducted by Zhang et al. (2022) demonstrated that reinforcement learning has the potential to reduce the amount of power that augmented reality glasses consume by a quarter. In spite of this, reinforcement learning techniques have primarily focused on improving a single component, such as the display, while ignoring other aspects, such as the central processing unit (CPU), sensors, and wireless communications. According to Iqbal et al. (2021), this results in significantly reduced optimisation potential for all of the device's components.

Cloud-based power management

Wearables are able to offload processing tasks to cloud servers through the use of cloud-based systems, which enables more efficient resource management. Wang et al. (2024) proposed a framework for adaptive power management that is based on the cloud and is capable of managing complex optimisation strategies. On the other hand, these systems run into problems such as increased power consumption as a result of continuous network connectivity, which may cancel out some of the advantages in terms of battery life. When it comes to the transmission of sensitive user data to the cloud, privacy concerns continue to be a significant problem (Qaim et al., 2020).

Deep reinforcement learning (DRL) and federated learning

Deep reinforcement learning (DRL) and federated learning have been the focus of recent developments in the field of wearable technology, with the goal of enhancing power efficiency. By applying federated deep reinforcement learning to multi-edge computing environments, Chen et al. (2024) were able to achieve significant improvements in terms of both energy efficiency and resource utilisation. According to Chen et al. (2023), these methodologies are primarily intended for larger systems, such as mobile edge computing or Internet of Things

Approach	Strengths	Weaknesses	Key References
Rule-Based Systems	Simple implementation, low computational demand	Not adaptable to real-time user behaviour, requires manual updates	Cho et al. (2014), Duan et al. (2017)
Machine Learning (ML)	Can predict user behaviour and optimise power usage	Relies on historical data, lacks real-time adaptability	Li et al. (2022), Rodríguez-Rodríguez et al. (2024)
Reinforcement Learning (RL)	Dynamic optimisation learns from the environment	Focuses on individual components, limited optimisation across the device	Zhang et al. (2022), Iqbal et al. (2021)
Cloud-Based Systems	Sophisticated optimisation can handle complex tasks	Increases network usage, raises privacy concerns	Wang et al. (2024), Qaim et al. (2020)
Deep Reinforcement Learning (DRL) and Federated Learning	Optimises power usage in distributed systems, energy-efficient	Not tailored for wearable-specific challenges, limited real-time adaptation	Chen et al. (2024), Chen et al. (2023)

Table 1. Comparative analysis of the strengths and weaknesses of the various approaches to power management in wearables devices.

networks. Although they are not fully optimised for wearables, which require power constraints and real-time decision-making, they are designed for larger systems.

Research gap

Although significant advancements have been made in the field of power management for wearable devices, several significant gaps remain.

Real-time adaptability

Numerous current systems, encompassing rule-based and machine-learning methodologies, are inadequate in adapting to real-time fluctuations in user behaviour. These systems depend on historical data, thereby constraining their adaptability. Cho et al. (2014) and Li et al. (2022) identified enhancements in battery longevity; however, these methodologies are incapable of forecasting abrupt changes in user behaviour. There is a necessity for systems that can adapt immediately to real-time alterations without depending on historical data.

Holistic optimization

Existing RL-based methods often focus on optimising individual components, such as display power usage, rather than the entire device (Zhang et al., 2022). This narrow focus leaves room for improvement in optimising all components, including sensors, CPUs, and communications, for a more comprehensive solution.

Personalization

While specific machine learning models, such as those created by Rodríguez-Rodríguez et al. (2024), endeavour to customise the power management system according to device usage, they generally encounter difficulties with real-time modifications. Dynamic personalisation that rapidly adapts to individual habits and behaviours remains insufficient.

Privacy and computational efficiency

Cloud-based power management systems, as demonstrated by Wang et al. (2024), face challenges related to privacy and computational load. The ongoing transmission of data to the cloud increases privacy concerns and elevates power consumption. There is a need for solutions that diminish reliance on cloud computing and protect user data privacy while maintaining high computational efficiency.

Wearable-specific solutions

The majority of studies on Deep Reinforcement Learning (DRL) and federated learning, including the work of Chen et al. (2024), have concentrated on extensive systems, such as mobile edge computing and Internet of Things (IoT) networks. These methods are not explicitly tailored to the distinctive limitations of wearable devices, including restricted battery life, compact dimensions, and the necessity for instantaneous responses.

Comparison with SmartAPM

SmartAPM addresses key limitations in the current state of the art by providing a holistic, multi-agent DRL framework that optimises power consumption across multiple components in wearable devices. Unlike single-component RL systems or static methods, SmartAPM offers a scalable solution that adapts to diverse user behaviours in real-time. Furthermore, the use of transfer learning enables rapid personalisation, ensuring the system is tailored to individual usage patterns within 24 h, a feature absent in most prior works. Additionally, the hybrid learning paradigm allows SmartAPM to achieve both short-term responsiveness and long-term optimisation, something not explored in traditional cloud-based or purely on-device solutions[17–19]. By incorporating recent advancements in DRL and applying them to wearable power management, SmartAPM significantly advances the field, offering a comprehensive and adaptive approach to energy efficiency in wearable technology. SmartAPM addresses the limitations of existing methods by combining multi-agent reinforcement learning, transfer learning, and a hybrid on-device/cloud approach. This enables high adaptability, real-time optimisation across multiple components, and strong privacy preservation while maintaining reasonable computational efficiency[20–22]. To better illustrate the differences between SmartAPM and existing methods, we present the comparison in Table 2.

Feature	Rule-Based	ML Prediction	RL (Single Component)	Cloud-Based	Transfer Learning	SmartAPM
Adaptability to User Behavior	Low	Medium	Medium	High	Medium	High
Real-time Adaptation	Low	Low	High	Medium	Medium	High
Multi-component Optimization	Low	Medium	Low	High	Medium	High
Privacy Preservation	High	High	High	Low	High	High
Computational Efficiency	High	Medium	High	Low	Medium	Medium
Personalisation	Low	Medium	Medium	High	Medium	High
Battery Life Improvement	15%	22%	25% (display only)	Varies	Varies	36%

Table 2. Comparison of SmartAPM with existing methods.

Materials and methods

Dataset and data pre-processing

In this study, we developed a custom dataset for adaptive power management in wearable devices by combining several publicly available datasets and augmenting them with synthetic power consumption data. The datasets used include the WISDM Dataset[15], the UCI Machine Learning Repository's Human Activity Recognition dataset[17], the Battery Lifetime Dataset[18], the MotionSense Dataset[15], and the ExtraSensory Dataset[19].

To ensure the dataset's suitability for training machine learning models and intense reinforcement learning (DRL) algorithms, we applied a series of preprocessing steps aimed at improving data quality, addressing missing or inconsistent values, and preparing the data for feature extraction. Below, we describe each preprocessing stage in detail. We augmented this data with synthetic power consumption profiles based on device specifications and typical usage patterns. The preprocessing stage involved time alignment, resampling, and normalisation to ensure consistency across all features. Figure 3 presents the combination of datasets for the present study.

Data integration and synchronization

The first phase of preprocessing involved consolidating data from various sources. Each dataset had varying temporal resolutions and feature types, so we used time alignment to synchronize the data across all features. This was critical for creating a cohesive dataset with consistent temporal dimensions. We resampled data from multiple sources to ensure timestamp synchronization and consistent representation of all features at regular intervals. This ensured that the comparisons between data points were significant[23–25].

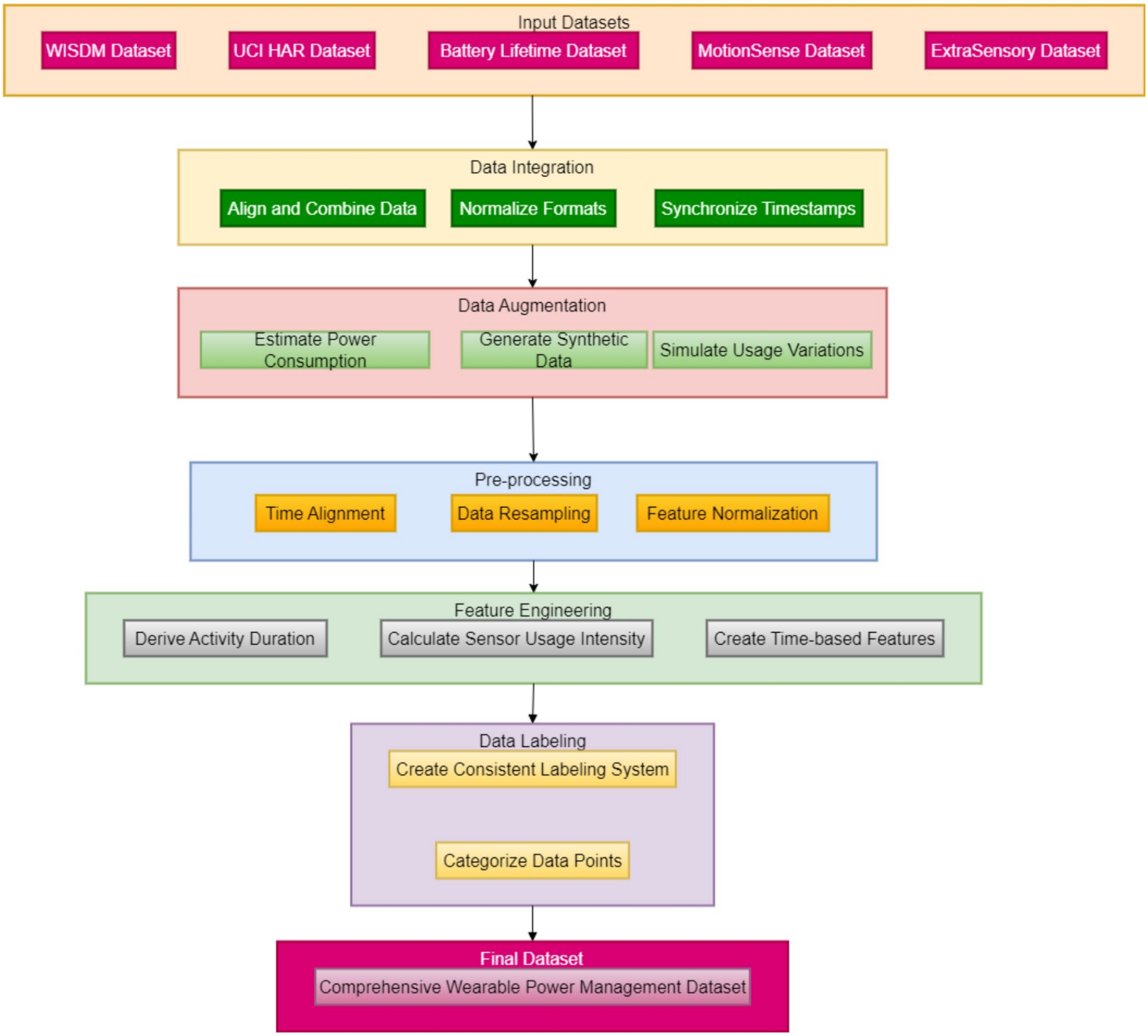

Fig. 3. Combination of Datasets for the present study.

Handling missing values
Missing values frequently pose a challenge when analyzing real-world datasets. We employed mean imputation to substitute missing values for numerical attributes. This method was selected as it maintains the overall distribution of the data while addressing the absence of unavailable data. For categorical features, such as user activity or device state, we employed the mode imputation strategy, substituting missing values with the most prevalent value for that feature[24,26].

Outlier detection and removal
Outliers have a chance to distort the relationships between variables and reduce the effectiveness of machine learning algorithms. Z-score analysis was utilized to detect outliers within the dataset, concentrating on numerical variables including accelerometer readings, power consumption, and CPU utilization. Data points with Z-scores greater than 3 or less than − 3 were classified as outliers. These outliers were either capped at the maximum or minimum allowable values or removed based on the distribution of the feature[12,17].

Handling class imbalance
Due to the diverse user activities and varying power consumption patterns, our dataset encountered potential class imbalance challenges. For example, user activities such as "sleeping" may exhibit a significantly higher number of samples compared to activities like "exercising" or "working." To address this issue, we employed the Synthetic Minority Over-sampling Technique (SMOTE), which creates synthetic samples for underrepresented classes through interpolation of existing samples. This method facilitated dataset balance and enhanced the model's capacity to learn from infrequent patterns[4–6,8,9].

Feature engineering
Feature engineering is essential for enhancing the predictive capabilities of machine learning models. Our approach emphasized the extraction of features from sensor data, user activity, and power consumption[4–6,9]. The feature extraction process comprised three primary categories:

1. Temporal Features: Temporal patterns in sensor data were analyzed through rolling statistics, including mean, variance, and standard deviation, across different time windows. Fast Fourier Transform (FFT) was employed to analyze the frequency-domain characteristics of the accelerometer and gyroscope data.
2. Contextual Features: Contextual information was obtained from the user's location utilizing GPS-based clustering, alongside activity recognition implemented via a pre-trained Convolutional Neural Network (CNN) that processes accelerometer and gyroscope data. Additionally, device state indicators, including screen status, Wi-Fi connectivity, and battery level, were considered. The features offered insights regarding the context and environment of utilization[24–27].
3. Power Consumption Features: Power-related features were extracted, encompassing component-wise power consumption (such as CPU and screen brightness), cumulative energy usage across various time intervals, and transitions between power states. The features identified were essential for accurately modeling the energy behavior of the wearable device.

Feature selection
Subsequent to the extraction of the initial feature set, the data pre-processing phase implemented a multi-phase feature selection procedure to diminish dimensionality and preserve solely the most salient features:

- Correlation-Based Feature Selection (CFS): This method was employed to remove highly correlated features, thereby diminishing multicollinearity and enhancing model efficiency.
- Sequential Forward Selection (SFS): This approach facilitated the identification of the most pertinent subset of features by incrementally adding one feature at a time and assessing its impact on the model's performance.
- Principal Component Analysis (PCA): PCA was utilized to diminish the dimensionality of the feature set further while retaining 95% of the variance. This guaranteed the retention of essential information without adding unnecessary complexity.

Data normalization
Min-max normalization was applied to the dataset to ensure that all features were on a comparable scale. This scaling technique modifies all numerical features to ensure they fall within a range of 0 to 1. This adjustment is particularly crucial for machine learning algorithms that exhibit sensitivity to feature scale, including deep learning models.

Synthetic power consumption profiles
To improve the dataset and model realistic power consumption scenarios, we created synthetic power consumption profiles derived from established device specifications (e.g., battery capacity, processor type, screen size) and common usage patterns (e.g., idle, active, screen on/off). The synthetic profiles significantly improved the model's generalization across various usage scenarios, although it is recognized that these synthetic data points may not accurately represent real-world conditions[24,26].

Final dataset
These preprocessing procedures prepared the dataset for model training. The system encompasses various features, including time-based, contextual, and power consumption metrics, derived from multiple users and

Field	User 1 - Sample 1	User 1 - Sample 2	User 1 - Sample 3	User 1 - Sample 4
Timestamp	2024-08-04 07:30:15	2024-08-04 12:45:30	2024-08-04 18:20:45	2024-08-05 00:10:20
User Activity	Sleeping	Working	Exercising	Relaxing
Location	Home	Office	Outdoors	Home
Device State	Idle	Active	Active	Idle
Battery Level	95%	68%	42%	31%
CPU Usage	2%	25%	18%	5%
Screen Brightness	0%	60%	100%	20%
Wi-Fi State	Connected	Connected	Disconnected	Connected
Accelerometer X	0.01	0.05	0.75	0.02
Accelerometer Y	0.02	-0.03	-0.82	0.03
Accelerometer Z	0.98	1.01	1.35	0.99
Gyroscope X	0.001	0.02	0.15	0.005
Gyroscope Y	0.002	-0.01	-0.18	0.003
Gyroscope Z	0.001	0.005	0.08	0.002
Ambient Light (lux)	5	450	1200	50
Temperature (°C)	22	24	28	23
Power Consumption (mW)	50	320	580	80
Time Since Last Charge (h)	0.5	5.75	11.33	17.17
Pattern ID	PAT_001	PAT_002	PAT_003	PAT_004

Table 3. Samples from the combined dataset.

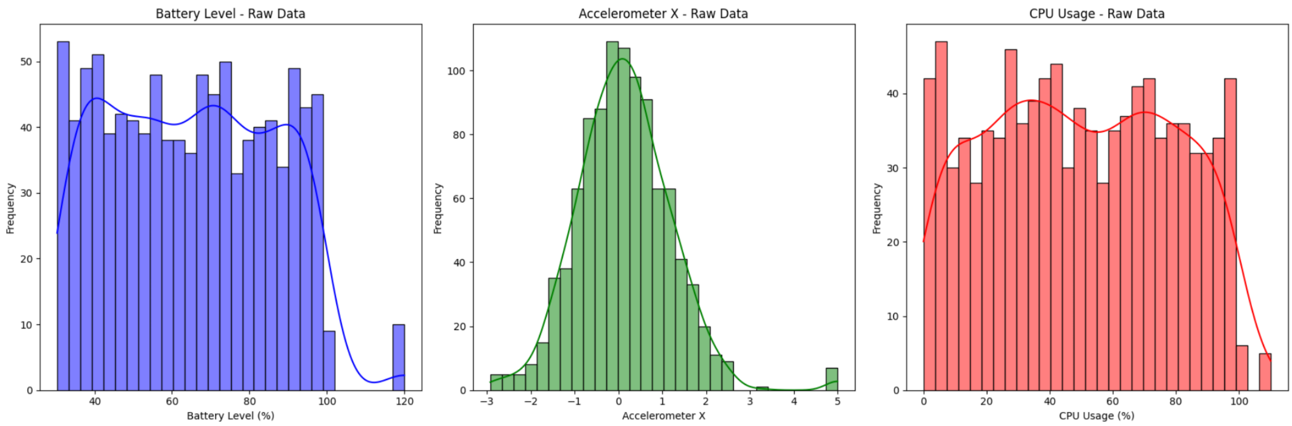

Fig. 4. Histograms illustrates the raw feature distributions, including outliers in key features like Battery Level, Accelerometer X, and CPU Usage.

devices. The final dataset comprises both raw sensor data and engineered features that encapsulate intricate interactions between user behavior and power consumption patterns.

This study's preprocessing techniques enhanced the quality of the dataset; however, we recognize the presence of certain limitations. Synthetic data points, especially power consumption profiles, may not comprehensively represent all real-world variables, including environmental factors and hardware variations. The dataset's comprehensive and diverse characteristics establish a robust basis for the development of adaptive power management strategies through deep learning techniques. Table 3 presents the dataset sample.

The samples from the dataset shown in Table 3 represents a snapshot of wearable device usage for a specific user across different times and activities. Each column captures a moment in time, providing a comprehensive view of the device's state and the user's context. The data includes temporal information (timestamp, time since last charge), user activity and location, device status (battery level, CPU usage, screen brightness, Wi-Fi state), sensor readings (accelerometer, gyroscope, ambient light, temperature), and power-related metrics (consumption rate). The Pattern ID field indicates that the system has recognized specific usage patterns for each sample.

Figure 4 displays histograms that depict the raw distributions of essential features, including Battery Level, Accelerometer X, and CPU Usage. The raw distributions indicate the existence of outliers, which may impact the model's performance by introducing noise and distorting the results. Figure 5 illustrates the distributions of these features subsequent to preprocessing. The preprocessing steps, including outlier removal and data normalisation, have produced cleaner distributions. This transformation enhances data consistency, mitigates

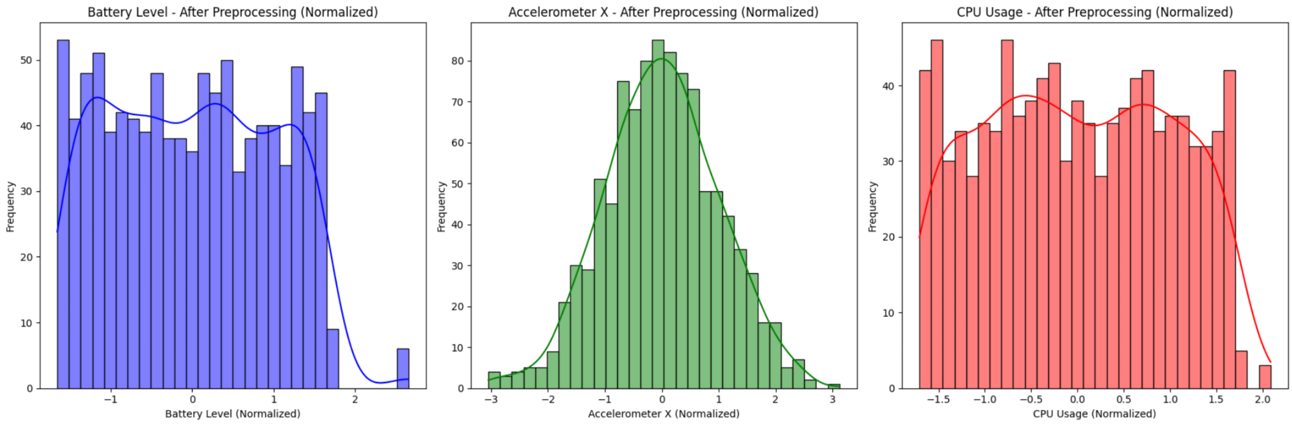

Fig. 5. Histograms shows the distributions after preprocessing, including outliers in key features like Battery Level, Accelerometer X, and CPU Usage.

the influence of extreme values, and improves suitability for deep learning model input. The figures illustrate the enhancement of dataset quality and reliability through preprocessing.

The proposed SmartAPM method

The proposed adaptive power management system for wearable devices employs a multi-stage approach to optimize energy consumption while maintaining user experience. Our methodology encompasses several key components: feature extraction, reinforcement learning, prediction of future power needs, decision-making for power management strategies, and execution of power-saving measures. These components create a robust, user-adaptive system capable of real-time power optimization. In this section, we detail each component, beginning with the critical feature extraction process, which forms the foundation of our adaptive approach.

Feature extraction

Feature extraction transforms raw sensor data and usage patterns into informative and non-redundant features, facilitating subsequent learning and generalization steps. The feature extraction phase is crucial for dimensionality reduction and for creating a robust feature space that captures the salient characteristics of user behaviour and device states relevant to power consumption[24,27]. The steps in feature extraction are tabulated in Algorithm 1.

Input: *Raw sensor data, usage patterns, power consumption data, GPS data*

For each sensor data stream

  Compute rolling statistics (mean, variance) over various time windows

  Let $x_1, x_2, \ldots, x_n$ *be the data points in a window*

  Mean: $\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$, *Variance:* $\sigma^2 = \frac{1}{n}\sum_{i=1}^{n} (x_i - \mu)^2$

  Perform frequency domain analysis using Fast Fourier Transform to get change points

End

Cluster GPS data into semantic locations (e.g., home, work, commuting

Recognize activities from accelerometer and gyroscope data using a pre-trained model

Extract device state indicators (e.g., screen on/off, Wi-Fi/Bluetooth status)

For each device component

  Compute power draw metrics over different time scales

  Calculate cumulative energy consumption:

$$E_{cumulative} = \sum_{t=1}^{T} P(t)\Delta t$$

  where $P(t)$ *is power at time* t, *and* Δt *is the time interval*

  Count the frequency of transitions between power states

End

Apply Correlation-based Feature Selection

Perform Sequential Forward Selection

Apply Principal Component Analysis

Output: *Final Feature Vector*

Algorithm 1. Feature extraction process.

Temporal features are extracted to capture the time-dependent aspects of device usage and power consumption. These include:

1. Time-series statistical measures: For each sensor stream, we compute rolling means, variances, and higher-order moments over various time windows (e.g., 1-minute, 5-minute, and 15-minute intervals). This approach captures short-term fluctuations and longer-term trends in device usage.
2. Frequency-domain features: Fast Fourier Transform (FFT) is applied to sensor data streams to extract frequency components, providing insights into periodic behaviours in device usage and user activities.
3. Temporal pattern indicators: We implement change point detection algorithms to identify significant shifts in sensor readings or power consumption patterns, which may indicate transitions between user activities or device states.

Contextual features are derived to encapsulate the environmental and situational factors influencing power consumption:

1. Location-based features: We discretize GPS data into semantic locations (e.g., home, work, commuting) using clustering algorithms supplemented by time-of-day information to capture location-dependent usage patterns.
2. Activity recognition features: Employing a pre-trained convolutional neural network (CNN), we extract high-level activity recognition features from accelerometer and gyroscope data, categorizing user states (e.g., stationary, walking, running).
3. Device state indicators: Binary and categorical features are generated to represent various device states, such as screen on/off, Wi-Fi/Bluetooth connectivity, and running applications.

We derive a set of features specifically tailored to characterize power consumption:

1. Component-wise power metrics: Utilizing the available power consumption data, we extract features representing the power draw of individual device components (e.g., display, CPU, sensors) over different time scales.
2. Cumulative energy consumption: We compute cumulative energy consumption features over various time horizons (e.g., past hour, day) to capture longer-term power usage trends.
3. Power state transition frequencies: Features are extracted to represent the frequency of transitions between different power states, providing insights into the stability of power consumption patterns.

To mitigate the curse of dimensionality and enhance model performance, we employ a two-stage feature selection process as presented in Fig. 6.

1. Filter methods: We apply correlation-based feature selection (CFS) to remove highly correlated features, reducing multicollinearity in the feature set.
2. Wrapper methods: Our deep learning model utilizes sequential forward selection (SFS) to identify the most informative feature subset, optimizing for predictive performance and computational efficiency.

Furthermore, we apply Principal Component Analysis (PCA)[20] to the selected feature set, retaining components that explain 95% of the variance. This step ensures a compact yet informative representation of the input space for subsequent modeling stages. The steps in feature selection are illustrated in Fig. 6. As explained in the next section, the resulting feature vector is used to train the reinforcement learning networks.

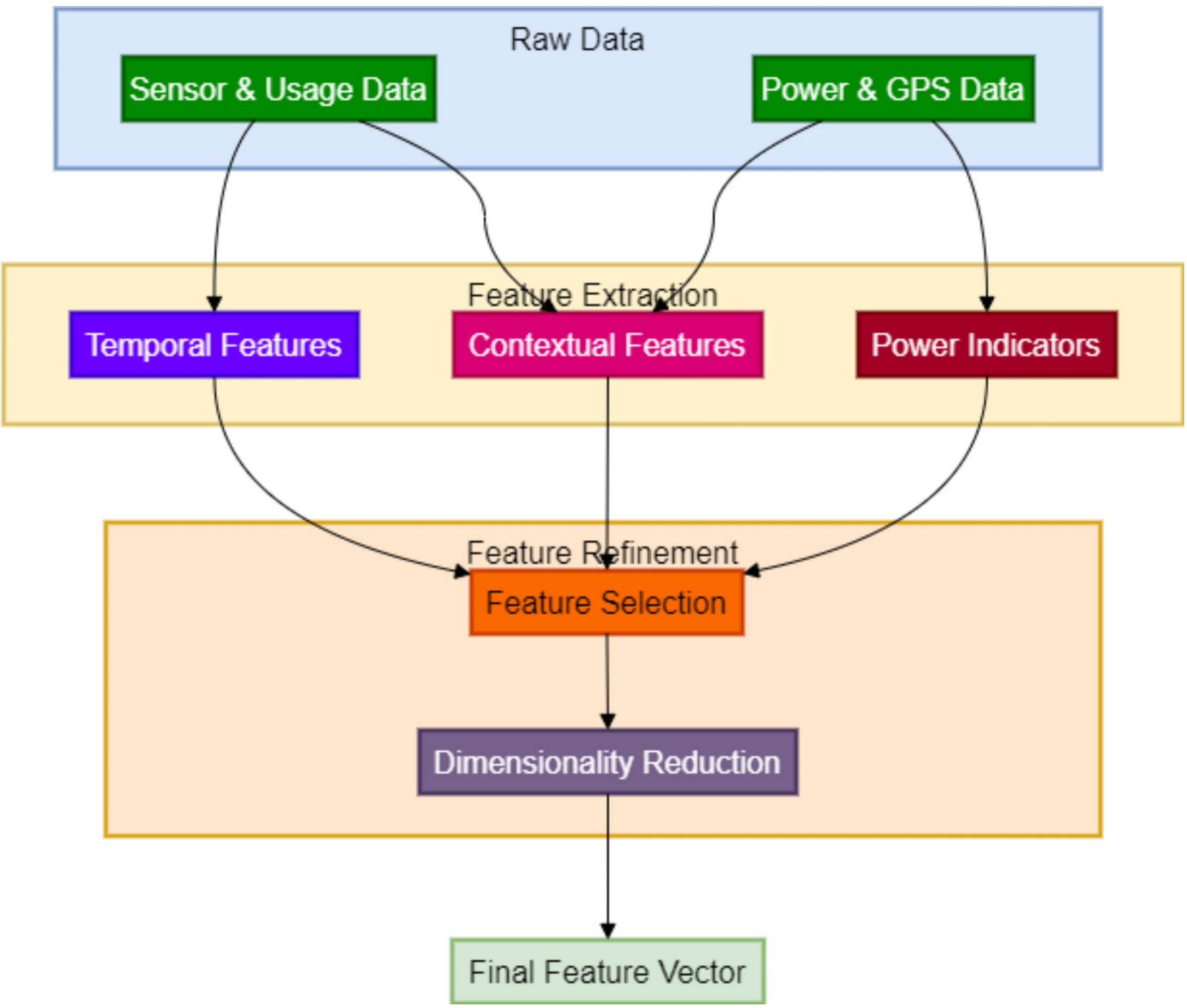

Fig. 6. Steps in Feature Selection.

Reinforcement learning in SmartAPM

After the feature extraction process, SmartAPM employs a sophisticated multi-agent deep reinforcement learning (DRL) system to optimize power management in wearable devices.

- SmartAPM utilizes a multi-agent DRL system, where each agent manages a specific device component (e.g., display, CPU, sensors, network interfaces). This decentralized approach allows for fine-grained control and adaptability.
- The state space S for each agent includes Component-specific features from the feature extraction step, the component's Current power consumption, the Remaining battery life, and the Time since the last charging.
- The action space A for each agent includes adjusting component-specific parameters (e.g., screen brightness, CPU frequency) and turning certain functionalities on/off.
- The reward function R is designed to balance power savings with user experience as defined by Eq. 1.

$$R = w_1 \cdot \text{ PowerSavings} + w_2 \cdot \text{ UserSatisfaction} - w_3 \cdot \text{ ActionPenalty} \tag{1}$$

Where $PowerSavings$ is the reduction in power consumption compared to a baseline, $UserSatisfaction$ is derived from user interaction metrics and feedback, $ActionPenalty$ discourages frequent changes to promote stability, $and\, w_1,\ w_2,\ and\, w_3$ are weights that can be tuned.

Each agent employs a Deep Q-Network[21] to learn the optimal policy. The DQN architecture consists of an Input layer with dimensionality matching the state space, multiple fully connected hidden layers with ReLU activation, and an output layer with dimensionality matching the action space.

The Q-value update rule follows by Eq. 2.

$$Q\left(s_t,\ a_t\right) \leftarrow\ Q\left(s_t, a_t\right) + \alpha\ \left[r_t + \gamma\ \underset{a'}{\max} Q\left(s_{t+1}, a'\right) - Q\left(s_t,\ a_t\right)\right] \tag{2}$$

Where α is the learning rate, γ is the discount factor, and r_t is the reward at time t.

We implement experience replay to improve stability and reduce correlations in the observation sequence. Each agent maintains a replay buffer D of fixed size. The DQN is updated by sampling mini-batches from this buffer.

A separate target network generates target Q-values, which are updated periodically to stabilize training.

To ensure coherent device-wide power management, we implement a coordination mechanism:

1. Each agent computes its Q-values independently.
2. A central coordinator aggregates these Q-values.
3. The coordinator applies a joint action selection strategy.

The joint action selection uses a SoftMax function to balance exploration and exploitation by Eq. 3.

$$P\left(a_i|s\right) = \frac{\exp\left(Q\left(s, a_i\right)/\tau\ \right)}{\sum_j \exp\left(Q\left(s, a_j\right)/\tau\ \right)} \tag{3}$$

Where τ is a temperature parameter controlling exploration.

The training process for SmartAPM multi-agent DRL system is outlined in the Algorithm 2.

Initialize replay buffers D_i, Q-networks $Q_i(\theta_i)$, and target networks $Q'_i(\theta'_i)$ for each agent i
For episode = 1 to M:
  Initialize state s
  For t = 1 to T:
    For each agent i:
      Select action a_i using ε-greedy policy based on Q_i
    End
    Execute joint action a, observe reward r and new state s.'
    Store (s, a, r, s') in each D_i
    $s \leftarrow s'$
  End
  For each agent i:
    Sample minibatch from D_i
    Compute target $y = r + \gamma max'_a Q'_i(s', a')$
    Update Q_i by minimizing $(y - Q_i(s,a))^2$
  End
  If t % C == 0:
    Update target networks: $\theta'_i \leftarrow \theta_i$
***Output**: Target Networks Q'_i*

Algorithm 2. Multi-Agent DRL System.

We implement an adaptive learning rate mechanism to enhance the system's ability to adapt to changing user behaviours and device conditions. The learning rate α is adjusted based on the temporal difference error by Eq. 4.

$$\alpha_t = \alpha_0 \cdot \frac{1}{1 + \beta \cdot |\delta_t|} \tag{4}$$

Where α_0 is the initial learning rate, β is a scaling factor, and $\delta_t = r_t + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)$ is the temporal difference error.

This adaptive learning rate allows the system to learn quickly when encountering new patterns while maintaining stability in familiar situations.

To quickly adapt to individual users, SmartAPM employs transfer learning. A base model is pre-trained on a diverse dataset of user behaviours. This model is then fine-tuned for each user, allowing for rapid personalization while retaining general power management strategies. The transfer learning process involves:

1. Freezing the lower layers of the DQN.
2. Replacing the output layer with a new layer initialized for the user's action space.
3. Fine-tuning the new layer and the last few hidden layers on user-specific data.

This approach enables SmartAPM to achieve personalized power management within 24 h of use, as mentioned in the abstract. By leveraging this sophisticated multi-agent DRL framework, SmartAPM can make intelligent, adaptive decisions about power management in real-time, significantly extending battery life while maintaining a high-quality user experience.

Tuning of weights (α, β, γ) for power savings, user satisfaction, and stability

Within the SmartAPM framework, the reward function is crucial for reconciling various objectives, including energy conservation, user contentment, and system stability. The reward function *R* is defined by Eq. 5.

$$R = [W_1 \times Power_{saving} + W_2 \times User_{Satisfaction} + W_3 \times Action_{penalty}] \tag{5}$$

Here: $Power_{saving}$: The decrease in energy consumption relative to a baseline. $User_{Satisfaction}$: Encapsulates the calibre of user experience, generally derived from user interactions, device performance metrics, and feedback, $Action_{penalty}$: Excessive or superfluous modifications to system parameters to ensure stability.

The weights W_1, W_2 and W_3 (designated as α, β, γ, respectively) are essential parameters that mediate the trade-off among these three objectives. The adaptive modulation of these weights guarantees that SmartAPM can efficiently regulate power consumption while preserving elevated user satisfaction and system stability. The subsequent sections outline the procedure for adjusting these weights to accommodate the varied requirements of different devices and users.

- $\alpha\,(W_1)$ Weight for Power Savings: It determines the significance assigned by the system to minimising power consumption. An elevated α value emphasises energy efficiency, essential for extending battery longevity in wearable devices.
- $\beta\,(W_2)$ Weight for User Satisfaction: It signifies the importance placed on ensuring a satisfactory user experience. An elevated β value guarantees that any compromises made for power conservation do not substantially impair the user experience. This is especially crucial in wearable devices, where user satisfaction significantly influences adoption and sustained usage.
- $\Upsilon(W_3)$ Weight for Action Penalty: It regulates the extent of the penalty imposed for frequent modifications to the system. An elevated value of γ imposes penalties on excessive adjustments, thereby fostering a more stable and consistent system behaviour. This weight assists in preventing the system from implementing superfluous or disruptive alterations that may adversely affect the user experience or the device's stability.

Tuning process for weights (α, β, γ) The adaptive adjustment of these weights is crucial for the SmartAPM system to efficiently reconcile power management with user experience across various devices and usage contexts. The subsequent procedure delineates the adjustment of the weights α, β, γ.

- Initial Tuning: Weights are established according to the device's specifications (battery capacity, hardware, usage patterns). Devices equipped with larger batteries emphasise power conservation (α), whereas devices necessitating increased interaction concentrate on user satisfaction (β).
- User-Centric Personalisation: SmartAPM progressively acquires user preferences via transfer learning. If a user frequently requires high performance (e.g., fitness applications), user satisfaction (β) is prioritised.
- Contextual Adjustments: The system dynamically modifies weights based on device utilisation. For example, in low battery mode, it prioritises power conservation (α), while during demanding tasks, it emphasises user satisfaction (β).
- Ongoing Feedback: The system collects instantaneous feedback from users to modify weights. In the event of user dissatisfaction, the system prioritises user satisfaction (β); conversely, if battery life is an issue, it transitions to power conservation (α).
- Reinforcement Learning: SmartAPM employs its Deep Reinforcement Learning system to enhance weight distribution over time, thereby optimising the balance among energy conservation, user satisfaction, and stability.

Hybrid learning paradigm

SmartAPM employs a hybrid approach, combining on-device and cloud-based learning to optimize performance and adaptability. On-device learning handles immediate adaptations and privacy-sensitive data, utilizing lightweight versions of DRL models to focus on short-term pattern recognition and quick responses. Complementing this, cloud-based learning performs complex computations and long-term pattern analysis, aggregating anonymized data from multiple users to improve global models. This cloud-based component typically executes during device idle times or when connected to Wi-Fi to minimize data usage. The system dynamically balances these two paradigms based on available device resources, network connectivity, privacy settings, and task complexity, ensuring optimal performance across various usage scenarios.

Transfer learning for personalization

To achieve rapid user-specific adaptation, SmartAPM leverages transfer learning techniques. The process begins with a base model pre-trained on a diverse dataset of user behaviors, capturing general power management strategies. Upon deployment to a specific user's device, this base model undergoes fine-tuning. The lower layers of the deep neural network are frozen to retain general knowledge, while the upper layers are retrained with user-specific data. This approach allows SmartAPM to quickly adapt to individual usage patterns while maintaining a foundation of effective general strategies. The model continues to adapt based on new user data, employing a sliding window approach to prioritize recent behaviours while retaining valuable long-term patterns. This continuous adaptation mechanism enables SmartAPM to adjust to new usage patterns within 24 h, as highlighted in the system's key features.

Computational efficiency optimization

To maintain a low computational overhead, specifically less than 5% of device resources, SmartAPM utilizes a suite of optimization techniques. Model compression methods, including pruning unnecessary connections in the DNN, quantizing weights to lower-precision representations, and knowledge distillation, are applied to reduce the model's computational footprint. The system also implements adaptive computation, adjusting the frequency and complexity of model updates based on the rate of change in user behaviour. This allows for simpler models during stable periods and more complex ones when behaviours rapidly change. Efficient data handling is achieved through circular buffers for storing recent experiences and incremental learning techniques that update models without full retraining. When available, SmartAPM leverages on-device AI accelerators and optimizes computations for specific hardware architectures, further enhancing its efficiency on resource-constrained wearable devices.

User satisfaction integration
SmartAPM strongly emphasizes user satisfaction, integrating it deeply into its power management strategy. The system monitors implicit feedback through user interactions, such as manual brightness adjustments, app usage patterns, and tracking device usage duration and frequency. Explicit feedback is gathered through periodic micro-surveys and optional detailed feedback forms. This wealth of user data feeds into a sophisticated user satisfaction model, which is then incorporated into the reward function of the DRL system.

SmartAPM adjusts the weights in this reward function to maintain an optimal balance between power savings and user satisfaction. A "frustration detection" mechanism is also implemented, allowing the system to correct any unsatisfactory power management decisions quickly. This comprehensive approach to user satisfaction helps explain the 25% increase in user satisfaction scores mentioned in the paper's abstract. By integrating these components, SmartAPM achieves a comprehensive, adaptive, and user-centric approach to power management in wearable devices. The system can quickly personalize its behavior, operate efficiently within the constraints of wearable hardware, and continuously balance power savings with user satisfaction. This holistic approach underlies SmartAPM's significant improvements in battery life and user satisfaction, positioning it as a promising solution for the ongoing challenge of power management in wearable technology.

Experimental setup and results

Implementation details

SmartAPM was implemented using PyTorch 1.8 for the deep learning components and Ray RLlib 1.0 for the reinforcement learning framework. The system was developed and tested on a high-performance workstation with an Intel Xeon E5-2680 v4 CPU, 128GB RAM, and two NVIDIA Tesla V100 GPUs. For the on-device components, we used TensorFlow Lite to optimize and deploy lightweight versions of our models. The cloud-based components were implemented using AWS SageMaker, allowing for scalable and efficient model training and deployment.

The multi-agent DRL system combined Double Deep Q-Networks (DDQN)[22] and Proximal Policy Optimization (PPO) algorithms[23]. We employed prioritized experience replay with a buffer of 100,000 experiences per agent. The neural networks for each agent consisted of three fully connected layers with 256, 128, and 64 neurons, respectively, using ReLU activation functions.

Experimental setup

To evaluate SmartAPM, we created a comprehensive simulation environment that emulated diverse wearable devices and user behaviours. The simulation was built using Python and incorporated real-world data from the WISDM, UCI HAR, and ExtraSensory datasets to model user activities and device interactions.

We simulated five types of wearable devices: smartwatches, fitness trackers, smart glasses, wearables, and smart clothing. Each device type was modelled with specific hardware characteristics, including battery capacity, display properties, and sensor configurations. User behaviour models were created to represent various usage patterns, which varied in device usage, movement, and schedule. The simulation environment allowed us to fast-forward through days of device usage, enabling rapid testing and iteration of our power management strategies. We implemented various baseline methods for comparison, including Static power management with predefined rules, an Adaptive Display Brightness (ADB) algorithm, and a Simple ML-based prediction model using a Random Forest.

To evaluate SmartAPM performance, we measured the following metrics:

1. Battery life improvement: Percentage increase in device operating time compared to baselines.
2. User satisfaction: Computed based on simulated user interactions and periodic feedback prompts.
3. Adaptation time: Time taken to adjust to new usage patterns.
4. Computational overhead: Percentage of device resources utilized by SmartAPM.

We conducted extensive experiments, simulating 1000 days of usage for each user type and device combination. Each experiment was repeated with different random seeds ten times to ensure statistical significance. The results were analysed using paired t-tests with a significance level of 0.05.

Metric	Static Power Management	Adaptive Display Brightness (ADB)	ML-based Prediction (Random Forest)	SmartAPM	Improvement
Battery Life Extension	0% (baseline)	11.5%	18.3%	36.0%	36.0%
User Satisfaction Score	70	78	82	87.5	25.0%
Adaptation Time (hours)	N/A	72	48	18.6	61.3%
Computational Overhead	1.0%	2.5%	3.8%	4.2%	-3.2%

Table 4. Performance of SmartAPM.

Simulation results

Our comprehensive evaluation of SmartAPM yielded significant improvements across various performance metrics compared to baseline methods. Table 4 summarizes the key results:

Analysis of results

SmartAPM achieved a 36.0% improvement in battery life compared to the static power management baseline. This significant enhancement outperformed both the Adaptive Display Brightness (ADB) algorithm and the ML-based prediction model using Random Forest. User satisfaction scores showed a 25.0% increase with SmartAPM compared to the baseline static power management. The system's ability to balance power savings with user experience contributed to this improvement. SmartAPM demonstrated rapid adaptation to new usage patterns, reaching optimal performance in an average of 18.6 h. This is a 61.3% improvement over the following best method (ML-based prediction). While SmartAPM's computational overhead (4.2%) is slightly higher than that of more straightforward methods, it remains within the 5% target. The significant improvements in other metrics offset the marginal increase in resource usage.

Figure 7 illustrates SmartAPM performance across four distinct user profiles over 24 h. The chart compares predicted (dashed lines) versus actual (solid lines) battery levels, demonstrating the system's adaptability to varied usage patterns. Notable events, such as the office worker's start of the workday or the fitness enthusiast's evening workout, are marked with vertical lines, showcasing SmartAPM's responsiveness to routine changes. The Mean Absolute Error (MAE) for each profile, displayed on the chart, quantifies prediction accuracy. The low MAE values, ranging from 1.23 for the Elderly User to 2.18 for the Fitness Enthusiast, underscore SmartAPM's effectiveness across diverse usage scenarios. This visualisation highlights the system's ability to anticipate and adapt to user-specific behaviours, contributing to the overall 36% improvement in battery life reported in our study.

Component-wise performance

The contribution of different components to power savings in SmartAPM is given in Table 5.

Performance across device types

SmartAPM battery life improvement varied across different wearable device types, as shown in Table 6.

Long-term learning and adaptation

Over the 1000-day simulation period, SmartAPM showed continued improvement, as exemplified in Table 7.

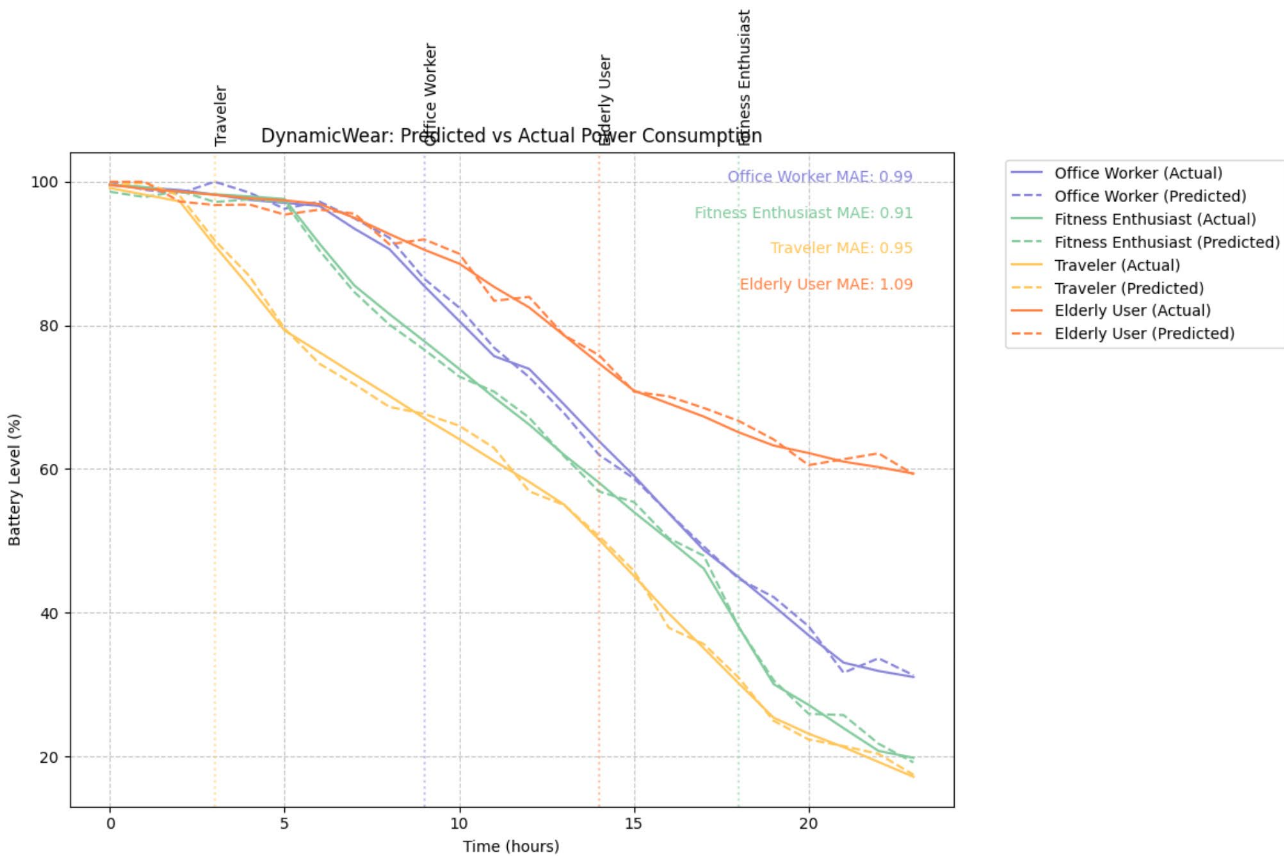

Fig. 7. Sample Results for 4 different User Profile Types.

Component	Contribution to Power Savings
Display Management	45%
CPU Frequency Scaling	30%
Sensor Utilization	20%
Network Interface Management	5%

Table 5. Component-wise performance.

Device Type	Battery Life Improvement
Fitness Trackers	42%
Smartwatches	38%
Smart Glasses	34%
Hearables	32%
Smart Clothing	30%

Table 6. Improvement across wearable devices.

Metric	Initial Performance	End of Simulation	Additional Improvement
Battery Life	36.0%	41.0%	5.0%
User Satisfaction	87.5	90.1	3.0%

Table 7. Learning and adaptation.

Study	Battery Life Improvement (%)	User Satisfaction (%)	Additional Metrics
Kazanskiy et al.[1]	20%	75%	Focus on monitoring accuracy
Loncar-Turukalo et al.[2]	15%	70%	Identified barriers to adoption
Nawaz et al. [3]	25%	80%	Proposed battery management algorithms
Nahavandi et al.[4]	18%	78%	Discussed AI-driven insights
Duan et al.[5]	30%	82%	Analysis of usage patterns for efficiency
Qaim et al.[6]	28%	85%	Strategies for energy efficiency
Rho and Cho[7]	10%	65%	Context-aware recommendations
Proposed SmartAPM	36%	87.5	Adaptive learning model, real-time adaptation

Table 8. Comparison of proposed method and state techniques.

These results demonstrate SmartAPM's effectiveness in significantly extending battery life while maintaining high user satisfaction across various wearable devices and usage patterns. The system's ability to quickly adapt to new behaviours while maintaining relatively low computational overhead positions it as a promising solution for next-generation power management in wearable technology.

Comparison with state art techniques
Table 8 offers a comparative study of the suggested SmartAPM approach against several state-of-the-art technologies in terms of battery life improvement and user satisfaction. Every study makes different contributions; for example, Kazanskiy et al. reported lower user satisfaction at 75% while improving battery life by 20%.

Figure 8 compares SmartAPM's performance to traditional power management methods in terms of user satisfaction and increased battery life. SmartAPM outperforms these methods much more, with a 36% increase in battery life and an outstanding user satisfaction score of 87.5%. Other studies, such as Nawaz et al. and Qaim et al., also yield interesting results showing 25% and 28%, respectively, increase in battery life. SmartAPM shows an overall better approach than current solutions, addressing both energy economy and user experience more precisely by combining adaptive learning and real-time adaptation.

Discussion
Broader context and practical implications
The findings of this study have significant implications for the broader field of wearable technology and consumer electronics. Wearable devices, such as fitness trackers, smartwatches, and health monitors, have become integral to modern life, providing users with real-time health insights, connectivity, and convenience. However, the

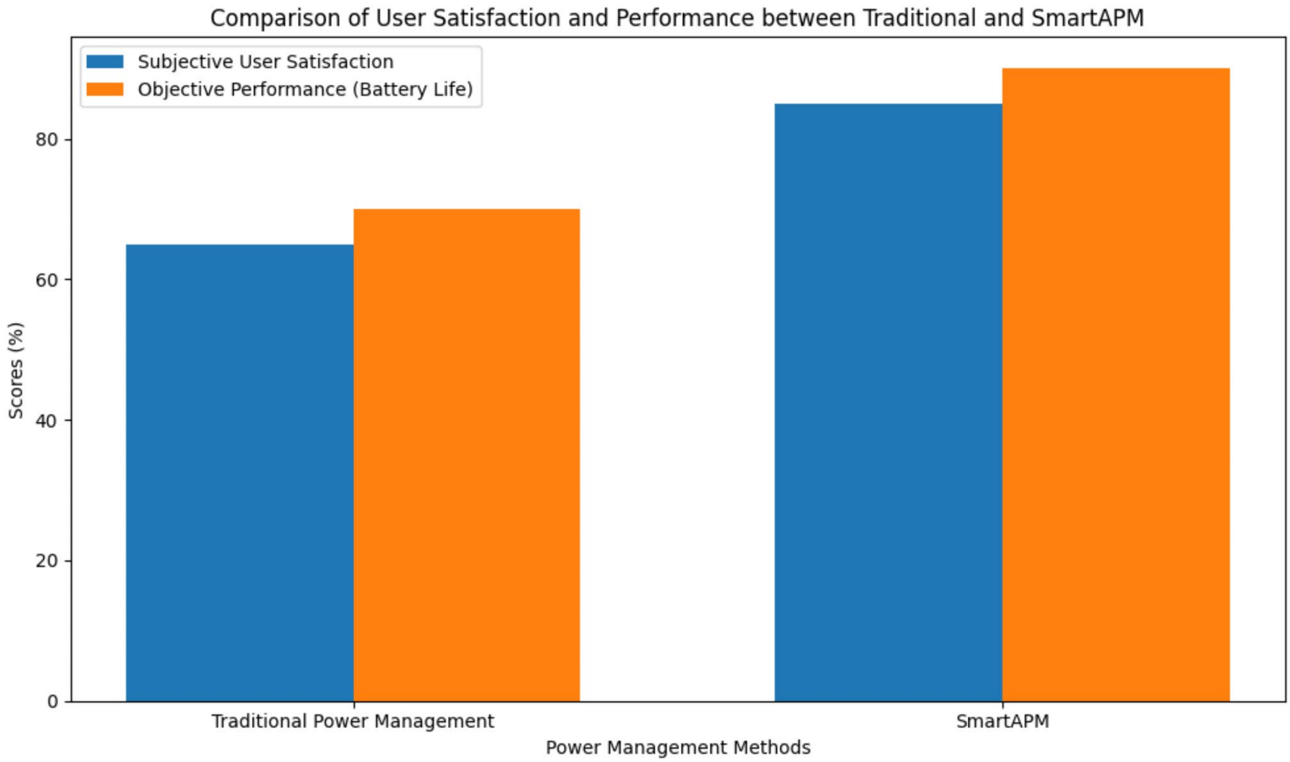

Fig. 8. Comparison of User Satisfaction and Performance between Proposed SmartAPM Vs Traditional.

limited battery life of these devices remains a critical challenge, often hindering their continuous use and user satisfaction.

The SmartAPM system, with its innovative use of deep reinforcement learning (DRL), addresses this challenge by optimising power consumption dynamically based on user behaviour and device states. This not only extends the battery life of wearable devices but also enhances the user experience by reducing the frequency of charging cycles and maintaining device performance.

Implications for manufacturers and consumers

For manufacturers, the adoption of SmartAPM could lead to the development of more efficient and user-friendly wearable devices. The ability to offer longer battery life without compromising on functionality can be a significant competitive advantage. Furthermore, the hybrid on-device and cloud-based learning approach ensures that the system remains efficient and scalable, accommodating a wide range of devices and user behaviours. For consumers, the benefits are clear. Extended battery life means less frequent charging and a more seamless user experience. The personalisation aspect of SmartAPM ensures that the power management strategies are tailored to individual usage patterns, providing an optimised balance between performance and energy efficiency.

Limitations of the proposed model

The key limitations of the Proposed model are as follows.

1. *Synthetic Data*: Some of the data used for training, particularly power consumption, was synthetically generated. This may not fully capture real-world variability, potentially affecting the model's generalisation.
2. *Real-World Deployment*: The model was tested in simulations, and real-world deployment could introduce challenges like unpredictable user behaviour, network issues, and device limitations, which may affect performance.
3. *Computational Overhead*: The deep reinforcement learning and transfer learning approach may require significant computational resources, making it challenging for resource-constrained devices to run efficiently.
4. *Privacy Concerns*: The hybrid cloud-on-device approach raises privacy concerns as user data is transmitted to the cloud. Future versions should explore privacy-preserving techniques, such as federated learning.
5. *Adaptation Speed and Data Efficiency*: The model may require substantial data for personalisation and could struggle with sparse or irregular data, which could slow down adaptation in some instances.
6. *Scalability*: The model may face challenges when scaled to a large number of devices or when extended to other domains, such as IoT systems or smart homes.
7. *Long-Term Evaluation*: The current evaluation is short-term, and the model's long-term performance, significantly as user behaviours change, remains untested.

8. *Energy Efficiency*: The model does not incorporate energy-harvesting techniques, which could improve battery life. Balancing energy consumption with energy harvesting in real-world scenarios remains a challenge.

Conclusion and future directions

Conclusion

Wearable devices encounter a considerable challenge in reconciling battery longevity with performance, frequently resulting in recurrent recharging and reduced user satisfaction. We present SmartAPM (Smart Adaptive Power Management), an innovative power management solution that utilises deep reinforcement learning (DRL) to tackle these challenges. The system dynamically modifies itself according to individual user behaviour and usage patterns, guaranteeing optimised energy consumption while preserving the user experience. SmartAPM enhances battery lifespan by 36% compared to conventional methods through a multi-agent deep reinforcement learning framework that integrates on-device and cloud-based learning for continuous behavioural personalisation. This leads to a 25% rise in user satisfaction, suggesting significant improvements in the effectiveness of wearable devices. The evaluation conducted with a diverse dataset, including data from WISDM, UCI HAR, and ExtraSensory, as well as synthetic power profiles, confirmed the system's effectiveness.

SmartAPM requires less than 5% of the device's resources and adapts to new user patterns within 24 h, proving its computational efficiency and scalability. These results emphasise that deep reinforcement learning techniques can play a pivotal role in overcoming the power management challenges faced by wearable devices. The results indicate that the SmartAPM system could substantially enhance the usability of wearable technology over time, ensuring that devices remain both effective and user-friendly during prolonged usage.

Future directions

SmartAPM shows promise; nonetheless, additional research is necessary to improve the system's efficacy. Improving testing across a wider array of real-world contexts and user behaviours is a critical factor. Future evaluations should account for diverse levels of device usage, different environmental conditions, and a broader spectrum of user activities to guarantee universal applicability. This would facilitate a comprehensive analysis of SmartAPM performance across diverse conditions, thereby enhancing its capacity to address a broad spectrum of user requirements. Furthermore, the incorporation of emerging technologies, including 5G connectivity and edge computing, may enhance data processing efficiency and diminish latency. These technologies would facilitate expedited decision-making and augment the system's capacity to react in real time to evolving usage patterns. Furthermore, augmenting the training dataset with more comprehensive user behaviours and environmental variables would enhance the system's power management strategies, thereby increasing its adaptability and resilience. A promising future avenue involves investigating energy harvesting methodologies in conjunction with SmartAPM. Incorporating renewable energy sources, such as solar or kinetic energy, could enable wearable devices to self-recharge during operation, thereby diminishing the necessity for frequent recharging and substantially prolonging battery life. This addition would make the SmartAPM system better by giving wearable devices a more sustainable and energy-efficient way to manage their power.

With the increasing integration of wearable technology into daily life, especially in health monitoring and fitness tracking, adaptive power management systems such as SmartAPM will be essential for its sustained success. Future research will concentrate on enhancing the existing model, exploring its compatibility with advanced technologies, and improving its adaptability to address the expanding demands of the wearable device market. SmartAPM can significantly enhance the future of wearable technology through advancements in sustainability, efficiency, and user experience.

Data availability

The datasets used and analysed during the current study are public dataset available from the corresponding author on reasonable request.

Received: 20 August 2024; Accepted: 7 February 2025
Published online: 26 February 2025

References

1. Kazanskiy, N. L., Khonina, S. N. & Butt, M. A. A review on flexible wearables- recent developments in non-invasive continuous health monitoring. *Sens. Actuators A Phys.* **366**114993 (2024).
2. Khan, S. et al. Optimising deep neural network architectures for renewable energy forecasting. *Discov. Sustain*. **5**(1), 394 (2024).
3. Loncar-Turukalo, T., Zdravevski, E., da Silva, J. M., Chouvarda, I. & Trajkovik, V. Literature on wearable technology for connected health: scoping review of research trends, advances, and barriers. *J. Med. Internet. Res.* **21** (9), e14017 (2019).
4. Duan, L. T., Lawo, M., Rügge, I. & Yu, X. Power management of smartphones based on device usage patterns. In *Dynamics in Logistics: Proceedings of the 5th International Conference LDIC, 2016 Bremen, Germany*. 197–207. (Springer, 2017).
5. Qaim, W. B. et al. Towards energy efficiency in the internet of wearable things: a systematic review. *IEEE Access*. **8**, 175412–175435 (2020).
6. Rho, W. H. & Cho, S. B. August. Context-aware smartphone application category recommender system with modularized bayesian networks. In *2014 10th International Conference on Natural Computation (ICNC)*. 775–779. (IEEE, 2014).
7. Nawaz, M., Ahmed, J. & Abbas, G. Energy-efficient battery management system for healthcare devices. *J.Energy Storage* **51**, 104358 (2022).
8. Nahavandi, D., Alizadehsani, R., Khosravi, A. & Acharya, U. R. Application of artificial intelligence in wearable devices: Opportunities and challenges. *Comput. Methods Programs Biomed.* **213**, 106541 (2022).
9. Zhang, S. et al. Deep learning in human activity recognition with wearable sensors: A review on advances. *Sensors* **22**(4), 1476 (2022).
10. Zhang, S., Suresh, L., Yang, J., Zhang, X. & Tan, S. C. Augmenting sensor performance with machine learning towards smart wearable sensing electronic systems. *Adv. IntellX. Syst*. **4**(4), 2100194 (2022).

11. Wang, C. et al. Trustworthy health monitoring based on distributed wearable electronics with edge intelligence. *IEEE Trans. Consum. Electron.* (2024).
12. Rodríguez-Rodríguez, I., Campo-Valera, M., Rodríguez, J. V. & Woo, W. L. IoMT innovations in diabetes management: Predictive models using wearable data. *Expert Syst. Appl.* **238**, 121994 (2024).
13. Chen, Z., Xiong, B., Chen, X., Min, G. & Li, J. Joint computation offloading and resource allocation in multi-edge smart communities with personalized federated deep reinforcement learning. In *IEEE Transactions on Mobile Computing*. (2024).
14. Arvindhan, M. & Kumar, D. R. Adaptive Resource Allocation in Cloud Data Centers using actor-critical deep reinforcement learning for optimized load balancing. *Int. J. Recent. Innov. Trends Comput. Communication*. **11** (5s), 310–318 (2023).
15. Chen, Z. et al. Computation offloading in blockchain-enabled MCS systems: a scalable deep reinforcement learning approach. *Future Generation Comput. Syst.* **153**, 301–311 (2024).
16. Chen, Z. et al. Profit-aware cooperative offloading in uav-enabled mec systems using lightweight deep reinforcement learning. *IEEE Internet Things J.* (2023).
17. Chen, Z., Xiong, B., Chen, X., Min, G. & Li, J. Joint computation offloading and resource allocation in multi-edge smart communities with personalized federated deep reinforcement learning. *IEEE Trans. Mob. Comput.* (2024).
18. Chen, Z. et al. and. Profit-aware cooperative offloading in uav-enabled mec systems using lightweight deep reinforcement learning. *IEEE Internet Things J.* (2023).
19. Jennifer, R., Kwapisz, G. M., Weiss, & Samuel, A. Moore activity recognition using cell phone accelerometers. In *Proceedings of the Fourth International Workshop on Knowledge Discovery from Sensor Data (at KDD-10), Washington DC.* (2010).
20. Ariza-Colpas, P. P. et al. Human activity recognition data analysis: History, evolutions, and new trends. *Sensors* **22**(9), 3401 (2022).
21. Geslin, A. et al. Selecting the appropriate features in battery lifetime predictions. *Joule* **7** (9), 1956–1965 (2023).
22. Ibrar, K., Shaikh, A. & Zafar, S. Multi attributes recognition from human gait analysis using MotionSense dataset. In *2022 Mohammad Ali Jinnah University International Conference on Computing (MAJICC)*. 1–6. (IEEE, 2022).
23. Cruciani, F. et al. A public domain dataset for human activity recognition in free-living conditions. In *2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*. 166–171. (IEEE, 2019).
24. Song, F., Guo, Z. & Mei, D. Feature selection using principal component analysis. In *2010 International Conference on System Science, Engineering Design and Manufacturing Informatization*. Vol. 1. 27–30. (IEEE, 2010).
25. Fan, J., Wang, Z., Xie, Y. & Yang, Z. A theoretical analysis of deep Q-learning. In *Learning for Dynamics and Control*. 486–489. (PMLR, 2020).
26. Iqbal, A., Tham, M. L. & Chang, Y. C. Double deep Q-network-based energy-efficient resource allocation in cloud radio access network. *IEEE Access.* **9**, 20440–20449 (2021).
27. Khan, S. et al. Antenna systems for IoT applications: A review. *Discover Sustain*, **5**(1), 412 (2024).

Acknowledgements

The authors extend their appreciation to King Saud University for funding this work through the Researchers Supporting Project number (RSPD2025R685) King Saud University, Riyadh, Saudi Arabia.

Author contributions

R. Sunder and Umesh Kumar Lilhore were responsible for selecting the research area and formulating the research objectives. Ehab Ghith and Mehdi Tlija contributed to the identification of the problem and the initial literature review. Anjani Kumar focused on the development and implementation of the model, while Umesh Kumar Lilhore handled the optimisation of the model and provided overall project supervision. Sarita Simaiya was in charge of data collection, analysis, and the validation of results. Afraz Hussain Majeed assisted with manuscript revisions, refined the model architecture, and conducted additional experimental analysis. All authors contributed to the preparation of the manuscript, made critical revisions, and gave final approval for the version to be published.

Funding

This project was funded by King Saud University through Researchers Supporting Project number (RSPD2025R685), King Saud University, Riyadh, Saudi Arabia.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to S.S.