國立成功大學

資訊工程學系

碩士論文

利用大型語言模型於低精度穿戴式感測行爲辨識

Utilizing Large Language Model for Human Activity Recognition from Coarse-Grained Wearable Sensors

研 究 生：蘇可維

指導教授：莊坤達博士

中華民國一一三年七月

利用大型語言模型於低精度穿戴式感測行爲辨識

學生：蘇可維　　　　　　　　　　　　　　　指導教授：莊坤達博士

國立成功大學資訊工程學系

摘　　要

人類行爲辨識（HAR）是穿戴式感測技術中的重要研究領域，對於健康監測、運動分析和智慧生活應用具有重要意義。傳統的穿戴式感測器往往受限於低精度數據和計算資源限制，難以實現準確的行爲辨識。隨著大型語言模型（LLM）的快速發展，其強大的序列理解和模式識別能力爲解決此問題提供了新的可能。

本論文提出了一種利用大型語言模型進行低精度穿戴式感測行爲辨識的創新方法。我們設計了一個適應性框架，將穿戴式感測器的粗粒度時間序列數據轉換爲語言模型可理解的表示形式，並利用預訓練語言模型的強大表示學習能力來提升行爲辨識的準確性。

我們的方法包含三個核心組件：（1）感測數據預處理和特徵提取模組，將原始感測信號轉換爲結構化序列；（2）基於大型語言模型的行爲模式學習架構，利用注意力機制捕捉時序關係；（3）多模態融合策略，整合不同感測器的信息以提升辨識性能。實驗結果顯示，相比傳統方法，我們的方法在多個公開數據集上都實現了顯著的性能提升，特別是在處理低精度和噪聲數據方面表現優異。

這項研究爲穿戴式設備的智能化發展提供了新的技術路徑，推動了大型語言模型在物聯網和普適計算領域的應用創新。

關鍵字：人類行爲辨識；大型語言模型；穿戴式感測器；低精度數據；時間序列分析；多模態融合

Utilizing Large Language Model for Human Activity Recognition from Coarse-Grained Wearable Sensors

Student: Su-Ko Wei Advisor: Dr. Kun-Ta Chuang

Submitted to Institute of Computer Science and Information Engineering
College of Electrical Engineering and Computer Science
National Cheng Kung University, Tainan, Taiwan, R.O.C.

ABSTRACT

Human Activity Recognition (HAR) is a critical research area in wearable sensing technology, with significant implications for health monitoring, sports analysis, and smart living applications. Traditional wearable sensors are often limited by low-precision data and computational constraints, making accurate activity recognition challenging. The rapid advancement of Large Language Models (LLMs) offers new possibilities with their powerful sequence understanding and pattern recognition capabilities.

This thesis proposes an innovative approach that utilizes Large Language Models for human activity recognition from coarse-grained wearable sensors. We design an adaptive framework that transforms coarse-grained time-series data from wearable sensors into representations comprehensible by language models, leveraging the powerful representation learning capabilities of pre-trained language models to enhance activity recognition accuracy.

Our method comprises three core components: (1) a sensor data preprocessing and feature extraction module that converts raw sensor signals into structured sequences; (2) a Large Language Model-based activity pattern learning architecture that utilizes attention mechanisms to capture temporal relationships; and (3) a multi-modal fusion strategy that integrates information from different sensors to improve recognition performance. Experimental results demonstrate that our approach achieves significant performance improvements over traditional methods across multiple public datasets, particularly excelling in handling low-precision and noisy data.

This research provides a new technical pathway for the intelligent development of wearable devices and advances the application innovation of Large Language Models in the Internet of Things and ubiquitous computing domains.

Keywords: Human Activity Recognition; Large Language Models; Wearable Sensors; Coarse-grained Data; Time Series Analysis; Multi-modal Fusion

Acknowledgment

Contents

List of Tables

List of Figures

Chapter 1 Introduction

Human Activity Recognition (HAR) using wearable sensors has emerged as a fundamental technology for numerous applications including mobile health monitoring, sports analytics, human-computer interaction, and smart environment systems [1, 2]. The ability to automatically identify and classify daily activities from sensor data enables continuous and unobtrusive monitoring of human behavior, providing valuable insights for healthcare, fitness tracking, and lifestyle analysis [3].

A critical and underexplored challenge in HAR is the significant drop in recognition performance when sensor data is downsampled or collected at low temporal resolution, a scenario increasingly common in consumer-grade wearables. Reducing the sampling frequency of sensors such as accelerometers and gyroscopes can dramatically extend battery life and operational time for smart devices, making low-grain data collection highly desirable for real-world, long-term deployment. However, this comes at the cost of information loss, increased noise, and reduced discriminative power, which can severely degrade the accuracy of activity recognition models.

Traditional HAR approaches predominantly rely on motion sensors such as accelerometers, gyroscopes, and magnetometers embedded in smartphones or dedicated wearable devices [4]. While these sensors offer the advantage of being lightweight, power-efficient, and readily available in consumer devices, they often produce coarse-grained data with limited precision compared to specialized research-grade equipment. This limitation poses significant challenges for accurate activity recognition, particularly when distinguishing between subtle variations in movement patterns or activities with similar motion characteristics.

Recent advances in deep learning have revolutionized the field of HAR, with convolutional neural networks (CNNs) and recurrent neural networks (RNNs) showing remarkable performance improvements over traditional machine learning approaches [5, 6]. Deep learning models have demonstrated their ability to automatically extract meaningful features from raw sensor data, reducing the need for manual feature engineering [7, 8]. However, most existing deep learning approaches for HAR are designed for high-quality sensor data and may not perform optimally when dealing with the noisy, low-resolution signals typical of consumer-grade wearable devices [9].

The emergence of Large Language Models (LLMs) and transformer architectures has opened new possibilities for sequence modeling and pattern recognition across diverse domains [10]. Models such as BERT [11] and GPT-3 [12] have demonstrated exceptional capabilities in understanding complex sequential patterns and contextual relationships. While these models were originally designed for natural language processing tasks, their underlying transformer architecture and attention mechanisms show promise for modeling temporal sequences in other domains, including sensor data analysis.

Recent pioneering work has begun to explore the application of LLMs to sensor-based HAR. Most notably, SensorLLM [13] introduced a two-stage framework for aligning large language models with motion sensors for human activity recognition. Their approach involves a Sensor-Language Alignment Stage that introduces special tokens for each sensor channel and automatically generates trend-descriptive text to align sensor data with textual inputs, followed by a Task-Aware Tuning Stage for HAR classification. This work represents an important breakthrough in bridging the gap between natural language processing and sensor-based activity recognition, demonstrating that LLMs can be effectively adapted for HAR tasks through specialized training procedures.

While SensorLLM has established the foundational feasibility of LLM-based HAR, several critical challenges remain largely unaddressed, particularly for real-world deployment scenarios involving downsampled, coarse-grained wearable sensors. First, the computational requirements and memory footprint of large transformer models present significant barriers for deployment on resource-constrained wearable devices, where power efficiency and real-time processing are paramount. Second, existing approaches have primarily focused on high-quality sensor data from controlled laboratory settings, but have not adequately addressed the unique challenges posed by low-frequency, low-precision sensor signals that are characteristic of consumer-grade wearable devices. Third, the temporal resolution and sampling rates of downsampled wearable sensors may not align optimally with the sequence processing capabilities of language models, necessitating specialized data representation and tokenization strategies.

Furthermore, the application of LLMs to sensor-based HAR presents several compelling advantages that remain underexplored in the context of low-grain data. The self-attention mechanism in transformers can effectively capture long-range dependencies in temporal sensor data, potentially identifying subtle patterns that traditional approaches might miss in noisy, low-resolution signals. The pre-training paradigm used in LLMs could enable knowledge transfer from large-scale datasets to smaller, domain-specific HAR datasets, which is particularly valuable when working with limited, downsampled sensor data. Additionally, the contextual understanding capabilities of LLMs may help in disambiguating similar activities by considering broader temporal context, which is crucial when dealing with the reduced discriminative power of low-frequency sensors.

This thesis is specifically motivated by the need to enable accurate and robust HAR on downsampled (low-grain) sensor data, with the ultimate goal of enabling longer battery life and more practical, real-world deployment of smart wearable devices.

The main contributions of this work include:

1. A novel sensor data representation method specifically designed for downsampled, coarse-grained wearable sensor signals that transforms continuous low-frequency sensor data into discrete tokens suitable for processing by Large Language Models, with optimizations for handling noise and reduced temporal resolution.

2. An efficient LLM-based architecture that balances accuracy with computational efficiency for deployment on resource-constrained wearable devices, incorporating novel attention mechanisms and parameter reduction techniques tailored for low-grain sensor data processing.

3. Comprehensive experimental evaluation demonstrating the effectiveness of our approach on multiple public datasets under various levels of sensor downsampling and precision degradation, showing significant improvements over traditional methods and competitive performance with existing LLM-based approaches, particularly in challenging low-grain sensor data scenarios.

4. Analysis of the interpretability and attention patterns learned by the LLM-based HAR model when processing downsampled sensor data, providing insights into how the model adapts to and compensates for reduced sensor precision while maintaining activity recognition accuracy.

The remainder of this thesis is organized as follows: Chapter 2 reviews related work in human activity recognition, deep learning approaches for sensor data analysis, and the application of transformer models to time series data, with particular emphasis on recent developments in LLM-based sensor data processing and the challenges of low-grain data. Chapter 3 presents our proposed methodology, including the downsampled sensor data tokenization approach and the efficient LLM-based HAR architecture. Chapter 4 describes the experimental setup and presents comprehensive evaluation results across multiple datasets and sensor precision levels. Chapter 5 discusses the implications of our findings, limitations of the current approach, and future research directions. Finally, Chapter 6 concludes the thesis and summarizes the key contributions.

Chapter 2 Related Work

2.1 Human Activity Recognition with Wearable Sensors

Human Activity Recognition (HAR) using wearable sensors has emerged as a fundamental research area with applications spanning healthcare monitoring, smart homes, and fitness tracking. Early comprehensive surveys by [1] and [2] established the foundational understanding of sensor-based activity recognition, highlighting the challenges of feature extraction from multi-modal sensor data and the importance of temporal modeling for sequential activities.

The evolution of HAR has been significantly influenced by advances in deep learning methodologies. Traditional approaches relied heavily on hand-crafted features and shallow machine learning models, but the introduction of deep neural networks revolutionized the field. [5] provided a comprehensive survey demonstrating how deep learning techniques could automatically learn hierarchical representations from raw sensor data, eliminating the need for manual feature engineering. This paradigm shift enabled more robust and generalizable activity recognition systems.

2.1.1 Deep Learning Approaches for HAR

The application of deep learning to HAR has yielded remarkable improvements in recognition accuracy and model generalization. [6] pioneered the use of deep convolutional neural networks on multichannel time series data, demonstrating superior performance compared to traditional methods. Their approach effectively captured both spatial relationships between sensor channels and temporal dependencies within activity sequences.

Convolutional Neural Networks (CNNs) have proven particularly effective for sensor data processing. [9] developed real-time HAR systems using CNNs on accelerometer data, achieving impressive accuracy while maintaining computational efficiency suitable for mobile deployment. Similarly, [8] explored smartphone-based HAR using deep learning, showcasing the potential for ubiquitous activity monitoring through everyday devices.

The integration of recurrent architectures has further enhanced temporal modeling capabilities. [7] conducted comprehensive comparisons of deep, convolutional, and recurrent models for wearable-based HAR, establishing best practices for different types of sensor configurations and activity categories. Their work highlighted the importance of model architecture selection based on the specific characteristics of the target application.

2.2 Transformer and Attention Mechanisms in HAR

The transformer architecture, originally introduced by [10], has revolutionized sequence modeling across various domains. The self-attention mechanism's ability to capture long-range dependencies and parallel processing capabilities have made it particularly attractive for sensor data analysis. Recent works have successfully adapted transformer architectures for HAR applications, demonstrating significant improvements over traditional recurrent approaches.

[14] introduced P2LHAP, a novel Patch-to-Label Seq2Seq framework that tackles activity recognition, segmentation, and forecasting simultaneously. Their approach divides sensor data streams into patches served as input tokens, enabling unified processing of multiple HAR tasks within a single model. The patch-based tokenization strategy has proven effective for handling variable-length sensor sequences while maintaining computational efficiency.

Recent advances have also explored specialized transformer variants for sensor data. [15] proposed MoPFormer, a Motion-Primitive Transformer that enhances interpretability by tokenizing sensor signals into semantically meaningful motion primitives. This approach addresses the critical challenge of cross-dataset generalization by learning fundamental movement patterns that remain consistent across different data sources.

[16] adapted the ActionFormer architecture for sensor-based HAR, demonstrating substantial improvements in activity boundary detection and classification accuracy. Their work highlighted the importance of detecting informative channels in multi-sensor configurations, achieving significant performance gains through attention-based channel selection mechanisms.

2.3 Large Language Models for Sensor Data

The recent success of Large Language Models (LLMs) has inspired researchers to explore their application to sensor data understanding. The foundational work in this area includes BERT [11] and GPT-3 [12], which demonstrated the power of large-scale pre-training for natural language understanding. These advances have motivated the development of similar approaches for sensor modalities.

[13] introduced SensorLLM, a pioneering work that aligns large language models with motion sensors for human activity recognition. Their two-stage framework consists of Sensor-Language Alignment for learning shared representations between sensor data and natural language, followed by Task-Aware Tuning for specific downstream applications. SensorLLM demonstrated remarkable capabilities in zero-shot activity recognition and cross-modal reasoning, establishing a new paradigm for sensor data understanding.

Building upon this foundation, [17] developed SensorLM, a comprehensive sensor-language foundation model that processes over 59.7 million hours of data from more than 103,000 participants. Their hierarchical caption generation pipeline captures statistical, structural, and semantic information from sensor data, enabling sophisticated natural language interactions with wearable sensor systems.

[18] explored the application of LLMs for Activities of Daily Living recognition in smart home environments. Their ADL-LLM system transforms raw sensor data into textual representations processed by LLMs for zero-shot recognition, demonstrating the potential for deployment in real-world assisted living scenarios.

[19] presented *HARGPT*, a systematic study investigating whether off-the-shelf LLMs can serve as zero-shot human activity recognizers without any gradient-based fine-tuning. By feeding raw, downsampled (10 Hz) IMU sequences directly into GPT-4 and other large models with carefully crafted role-play and chain-of-thought prompts, HARGPT achieved approximately 80% accuracy on the Capture-24 and HHAR benchmarks. This evidence reveals an emergent ability of LLMs to transform noisy time–series signals into high-level activity semantics purely through prompting, albeit at the expense of large context windows and intensive prompt engineering. In contrast, our two-stage pipeline fine-tunes a compact Chronos encoder–LLaMA

architecture on teacher–student enhanced, coarse-grained data, eliminating elaborate prompt design while enabling efficient on-device inference.

Complementing these empirical findings, [20] provides the first dedicated survey on LLM-driven wearable sensing. The paper synthesizes early trends, publicly available datasets, evaluation protocols, and open challenges—such as privacy, robustness, and energy efficiency—thereby situating the rapidly growing body of LLM-for-HAR research within a broader interdisciplinary landscape.

2.3.1 High-Level Reasoning with LLMs

Beyond basic activity classification, recent research has investigated LLMs' capabilities for high-level reasoning over sensor traces. [21] developed LLMSense, which harnesses LLMs for complex spatiotemporal reasoning tasks such as dementia diagnosis and occupancy tracking. Their framework demonstrates how LLMs can comprehend long-term sensor patterns and make sophisticated inferences that go beyond simple activity labeling.

[22] presented an AI-based system utilizing IoT-enabled ambient sensors and LLMs for complex activity tracking in elderly care. Their approach combines edge device processing with cloud-based LLM reasoning, addressing privacy concerns while maintaining the sophisticated reasoning capabilities required for healthcare applications.

The integration of multiple sensor modalities has been explored by [23], who introduced IoT-LM, a large multisensory language model for the Internet of Things. Their MultiIoT dataset encompasses over 1.15 million samples from 12 modalities, enabling comprehensive multisensory understanding and interactive question-answering capabilities.

2.4 Energy Efficiency and Resource Constraints

A critical challenge in HAR deployment is the balance between recognition accuracy and computational efficiency, particularly for battery-powered wearable devices. This challenge has motivated extensive research into lightweight model architectures and energy-efficient processing strategies.

Traditional approaches to address computational constraints include model compression techniques, quantization, and pruning. However, these methods often require careful tuning and may significantly impact model performance. The development of inherently efficient architectures has emerged as a more promising direction.

[24] introduced the Pretrained Actigraphy Transformer (PAT), designed specifically for time-series wearable movement data with emphasis on computational efficiency. Their lightweight transformer architecture achieves state-of-the-art performance in mental health prediction tasks while maintaining interpretability and deployment feasibility on resource-constrained devices.

2.4.1 Data Sampling and Downsampling Strategies

The relationship between sensor data resolution and recognition performance presents a fundamental trade-off in wearable HAR systems. Higher sampling rates provide richer temporal information but significantly increase computational and energy costs. Conversely, lower sampling rates extend battery life but may compromise recognition accuracy.

This trade-off becomes particularly critical for continuous monitoring applications where devices must operate for extended periods without charging. The challenge of maintaining recognition performance with downsampled sensor data remains largely unexplored in the literature, representing a significant gap that our work addresses.

Current HAR systems typically assume high-frequency sensor data (50-100 Hz or higher), which may not be practical for real-world deployment scenarios where energy efficiency is paramount. The development of effective methods for coarse-grained sensor data processing is essential for enabling practical wearable HAR systems.

2.5 Research Gaps and Motivation

Despite significant advances in HAR research, several critical gaps remain. First, while transformer-based approaches have shown promise, their application to coarse-grained sensor data has received limited attention. Most existing work assumes high-resolution sensor inputs, which may not be feasible in energy-constrained environments.

Second, the integration of LLMs with sensor data has primarily focused on high-frequency, well-sampled data. The effectiveness of LLM-based approaches for downsampled sensor data remains unexplored, despite its practical importance for real-world deployment.

Third, current research lacks comprehensive analysis of the attention patterns and tokenization strategies specifically designed for low-precision sensor data. Understanding how transformer models process coarse-grained sensor information is crucial for developing effective architectures.

Finally, there is insufficient investigation into the fundamental limits of activity recognition with reduced sensor resolution. Establishing these limits and developing methods to approach them is essential for advancing the field toward practical, energy-efficient HAR systems.

Our work addresses these gaps by developing novel tokenization strategies for downsampled sensor data, investigating LLM architectures optimized for resource-constrained environments, and providing comprehensive experimental evaluation across various downsampling levels. Through this research, we aim to bridge the gap between theoretical advances in HAR and their practical deployment in real-world wearable systems.

Chapter 3 Methodology

In this chapter, we present our novel two-stage methodology for enhancing Large Language Model-based Human Activity Recognition (HAR) performance on coarse-grained wearable sensor data. Our approach addresses the fundamental challenge of maintaining recognition accuracy while operating under severe resource constraints typical of wearable devices. The methodology consists of two interconnected stages: (1) **Teacher-Student Data Enhancement** that learns to upsample low-resolution sensor data to high-resolution quality, and (2) **SensorLLM Training** on the enhanced data for improved HAR performance.

3.1 Stage 1: Teacher-Student Data Enhancement

The first stage of our methodology employs a teacher-student learning framework to address the fundamental challenge of limited sensor data quality in resource-constrained environments. This stage learns to enhance coarse-grained, low-resolution sensor data to high-resolution equivalent quality, enabling subsequent stages to operate with richer temporal information despite the original data constraints.

3.1.1 Problem Formulation

Given a dataset of sensor readings collected at different sampling rates, we define high-resolution data $\mathbf{X}^h \in \mathbb{R}^{N \times T_h \times d}$ and low-resolution data $\mathbf{X}^l \in \mathbb{R}^{N \times T_l \times d}$, where N is the number of samples, T_h and T_l are the temporal dimensions with $T_h \gg T_l$, and d is the feature dimension (typically 3 for tri-axial accelerometer data). The downsampling relationship is defined by a factor $r = T_h/T_l$, where r represents the temporal resolution reduction.

The enhancement task aims to learn a mapping function $f : \mathbb{R}^{T_l \times d} \to \mathbb{R}^{T_h \times d}$ that can reconstruct high-resolution sensor data from low-resolution input while preserving the essential

Figure 3.1: Overview of the proposed two-stage methodology for coarse-grained sensor HAR

temporal patterns and activity-relevant features.

3.1.2 Teacher-Student Architecture

Our enhancement framework consists of four key components: a Teacher Encoder, Teacher Decoder, Student Encoder, and Student Decoder, designed to facilitate knowledge transfer from high-resolution to low-resolution domains.

Teacher Network

The teacher network operates on high-resolution sensor data and serves as the source of rich temporal knowledge. The teacher encoder E_T employs a convolutional-LSTM architecture for temporal feature extraction:

$$\mathbf{z}_T = E_T(\mathbf{X}^h; \theta_T^e) \tag{3.1}$$

where θ_T^e represents the teacher encoder parameters. The encoder architecture consists of:

- **Convolutional Layers**: Three 1D convolutional layers with kernel sizes [7, 5, 3], filters [64, 128, 256], and batch normalization with ReLU activation
- **Bidirectional LSTM**: Two-layer bidirectional LSTM with hidden dimension 256, dropout 0.1 for temporal modeling
- **Feature Projection**: Linear layer mapping to final hidden dimension (512 for teacher)

The teacher decoder D_T reconstructs the high-resolution data using progressive upsampling:

$$\hat{\mathbf{X}}^h = D_T(\mathbf{z}_T; \theta_T^d) \tag{3.2}$$

Student Network

The student network operates on low-resolution input and learns to generate high-resolution output through knowledge distillation from the teacher. The student encoder E_S follows the same architectural pattern but with reduced hidden dimensions (256):

$$\mathbf{z}_S = E_S(\mathbf{X}^l; \theta_S^e) \tag{3.3}$$

The student decoder D_S employs progressive upsampling with residual connections to generate enhanced high-resolution data:

$$\hat{\mathbf{X}}^{enhanced} = D_S(\mathbf{z}_S; \theta_S^d) \tag{3.4}$$

The decoder architecture includes:

- **Feature Expansion**: Linear layer expanding encoded features to initial sequence representation
- **Progressive Upsampling**: Multiple ConvTranspose1D layers with 2x upsampling at each stage
- **Residual Connections**: Skip connections between upsampling stages for gradient flow
- **Smoothing Layers**: Additional 1D convolutions for temporal smoothing after each upsampling step

Feature Projector

To enable effective knowledge transfer between teacher and student networks operating in different resolution domains, we introduce a feature projector P that aligns the feature representations:

$$\mathbf{z}_S^{proj} = P(\mathbf{z}_S; \theta_P) \tag{3.5}$$

This ensures that student features can be effectively compared with teacher features for knowledge distillation.

3.1.3 Enhancement Loss Functions

The training of our teacher-student enhancement model employs a multi-component loss function designed to optimize both reconstruction quality and temporal coherence.

Reconstruction Loss

The primary reconstruction loss ensures that the enhanced data closely matches the target high-resolution data:

$$\mathcal{L}_{recon} = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{X}_i^h - \hat{\mathbf{X}}_i^{enhanced}\|_2^2 \tag{3.6}$$

Feature Matching Loss

The feature matching loss facilitates knowledge transfer by aligning student and teacher feature representations:

$$\mathcal{L}_{feature} = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{z}_{T,i} - \mathbf{z}_{S,i}^{proj}\|_2^2 \tag{3.7}$$

Temporal Smoothness Loss

To ensure temporal coherence in the enhanced data, we apply a smoothness constraint:

$$\mathcal{L}_{smooth} = \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T_h - 1} \|\hat{\mathbf{X}}_{i,t+1}^{enhanced} - \hat{\mathbf{X}}_{i,t}^{enhanced}\|_2^2 \tag{3.8}$$

Frequency Domain Loss

To preserve important frequency characteristics, we incorporate a frequency domain loss using the Fourier transform:

$$\mathcal{L}_{freq} = \frac{1}{N}\sum_{i=1}^{N} \|\mathcal{F}(\mathbf{X}_i^h) - \mathcal{F}(\hat{\mathbf{X}}_i^{enhanced})\|_2^2 \tag{3.9}$$

where $\mathcal{F}$ denotes the Fourier transform operation.

Combined Loss Function

The total enhancement loss combines all components with appropriate weights:

$$\mathcal{L}_{enhancement} = \lambda_1 \mathcal{L}_{recon} + \lambda_2 \mathcal{L}_{feature} + \lambda_3 \mathcal{L}_{smooth} + \lambda_4 \mathcal{L}_{freq} \tag{3.10}$$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are hyperparameters controlling the relative importance of each loss component.

3.1.4 Training Procedure

The training of our enhancement model follows a two-phase approach:

Phase 1: Teacher Training - The teacher network is trained exclusively on high-resolution data using reconstruction loss to learn optimal feature representations.

Phase 2: Student Training - The student network is trained using the combined loss function while the teacher network parameters remain frozen, enabling effective knowledge transfer.

3.2 Stage 2: SensorLLM Training on Enhanced Data

The second stage leverages the enhanced sensor data from Stage 1 to train a SensorLLM model for Human Activity Recognition. This stage builds upon the SensorLLM framework [13] while incorporating our novel data enhancement pipeline.

3.2.1 Data Processing Pipeline

The enhanced sensor data undergoes several preprocessing steps to prepare it for SensorLLM training.

Temporal Segmentation

Enhanced sensor data is segmented into fixed-length windows for activity recognition. Given enhanced data $\hat{\mathbf{X}}^{enhanced}$, we extract overlapping windows:

$$\mathbf{W}_i = \hat{\mathbf{X}}^{enhanced}[i \cdot s : i \cdot s + w, :] \tag{3.11}$$

where w is the window size (300 seconds in our implementation), s is the stride, and i indexes the window.

Label Assignment

For each window, we assign activity labels based on a minimum label fraction criterion. A window is labeled with activity class c if:

$$\frac{\text{count}(c \text{ in window})}{\text{window length}} \geq \tau \tag{3.12}$$

where τ is the minimum label fraction threshold (0.5 in our implementation).

Data Tokenization

Following the SensorLLM approach, sensor data is converted into discrete tokens suitable for language model processing. We employ the StanNormalizeUniformBins tokenization method:

$$\text{token}_t = \text{quantize}\left(\frac{\mathbf{x}_t - \mu}{\sigma}, \text{num\_bins}\right) \tag{3.13}$$

where μ and σ are standardization parameters, and quantize() maps continuous values to discrete bins.

3.2.2 Question-Answer Generation

To leverage the natural language capabilities of Large Language Models, we generate question-answer pairs that encode both sensor data and contextual information.

Statistical Summary Generation

For each sensor window, we compute statistical summaries:

$$\text{Summary} = \{\mu_x, \sigma_x, \mu_y, \sigma_y, \mu_z, \sigma_z\} \tag{3.14}$$

where subscripts x, y, z denote the three accelerometer axes.

Correlation Analysis

Cross-correlation patterns between sensor axes are computed:

$$\rho_{ij} = \frac{\text{cov}(\mathbf{x}_i, \mathbf{x}_j)}{\sigma_i \sigma_j} \tag{3.15}$$

These correlations are converted to natural language descriptions (e.g., ”strongly positively correlated”).

Trend Analysis

Temporal trends within windows are analyzed using linear regression slopes and converted to descriptive text.

3.2.3 SensorLLM Architecture Integration

Our enhanced data is integrated into the SensorLLM architecture, which consists of three main components:

Time Series Encoder

We utilize the Chronos time series foundation model as the backbone encoder:

$$\mathbf{h}_{\text{sensor}} = \text{Chronos}(\text{tokens}; \theta_{\text{chronos}}) \tag{3.16}$$

where tokens represent the tokenized sensor data and θ_{chronos} are the pre-trained Chronos parameters.

Language Model Integration

The sensor embeddings are integrated with the LLaMA-3.2-1B language model:

$$\mathbf{h}_{\text{combined}} = \text{LLaMA}([\mathbf{h}_{\text{sensor}}; \mathbf{h}_{\text{text}}]; \theta_{\text{llama}}) \tag{3.17}$$

where $\mathbf{h}_{\text{text}}$ represents text embeddings from the question-answer pairs.

Classification Head

A classification head maps the combined representations to activity classes:

$$\mathbf{p} = \text{softmax}(\mathbf{W}_{\text{cls}}\mathbf{h}_{\text{combined}} + \mathbf{b}_{\text{cls}}) \tag{3.18}$$

3.2.4 Training Configuration

The SensorLLM model is trained using the following configuration optimized for enhanced data:

- **Loss Function**: Weighted cross-entropy loss to handle class imbalance
- **Optimization**: AdamW optimizer with cosine learning rate scheduling
- **Learning Rate**: 2e-3 with 3% warmup ratio
- **Batch Size**: 4 per device with gradient accumulation steps of 8
- **Training Epochs**: 8 epochs with early stopping based on F1-macro score
- **Model Freezing**: LLM parameters frozen, time series encoder frozen, only classification head trainable

3.3 Integration and Evaluation

The integration of our two-stage approach creates a comprehensive pipeline that transforms coarse-grained sensor data into high-quality HAR predictions. The enhanced data from Stage 1 provides the temporal richness necessary for effective Stage 2 training, while the SensorLLM framework in Stage 2 leverages both sensor patterns and contextual understanding for robust activity recognition.

This methodology enables HAR systems to achieve high performance even when operating under severe resource constraints, making it particularly suitable for deployment on consumer wearable devices where battery life and computational efficiency are paramount concerns.

3.4 Experimental Configuration

Our experimental setup encompasses specific configurations for both stages to ensure reproducible and robust results.

3.4.1 Stage 1 Configuration

The teacher-student data enhancement stage operates with the following parameters:

- **Data Specifications**:
  - High-resolution data: 300 timesteps (1Hz sampling, 300 seconds)
  - Low-resolution data: 30 timesteps (0.1Hz sampling, 300 seconds)
  - Upsampling factor: 10x (from 30 to 300 timesteps)
  - Feature dimension: 3 (tri-axial accelerometer data)
- **Model Architecture**:
  - Teacher hidden dimension: 512
  - Student hidden dimension: 256
  - Batch size: 16
  - Loss weights: $\lambda_1 = 1.0$, $\lambda_2 = 0.5$, $\lambda_3 = 1.0$, $\lambda_4 = 1.0$
- **Training Parameters**:
  - Teacher training epochs: 30
  - Student training epochs: 50
  - Learning rate: 1e-3 (both networks)
  - Optimizer: Adam

3.4.2 Stage 2 Configuration

The SensorLLM training stage employs the following configuration:

- **Data Processing**:
  - Window size: 300 seconds
  - Sampling rate: 100Hz (original) downsampled by factors 100DS, 200DS, 500DS, 1000DS
  - Minimum label fraction: 0.5
  - Data balancing ratio: 0.05
  - Stride: 50% overlap between windows
- **Model Configuration**:
  - Base LLM: LLaMA-3.2-1B Instruct
  - Time series encoder: Chronos-T5-Large
  - Tokenization: StanNormalizeUniformBins
  - Model max length: 4096 tokens
  - Number of activity classes: 10 (Capture24 dataset)
- **Training Setup**:
  - Distributed training: 2 GPUs with PyTorch DDP
  - Learning rate: 2e-3 with cosine scheduling
  - Warmup ratio: 3%
  - Batch size: 4 per device, gradient accumulation: 8
  - Training epochs: 8 with early stopping
  - Evaluation metric: F1-macro score
  - Mixed precision: BF16

3.4.3 Dataset Specifications

Our methodology is evaluated on the Capture24 dataset with the following specifications:

- **Participants**: 151 participants selected randomly
- **Train-test split**: 80% training, 20% testing
- **Activity categories**: 10 classes (sleep, sitting, standing, walking, bicycling, vehicle, household-chores, manual-work, sports, mixed-activity)
- **Data balancing**: Aggressive capping with 0.05 balance ratio to handle class imbalance
- **Quality control**: Windows with less than 50% single-label coverage are discarded

Chapter 4 Experiments

4.1 Datasets

4.1.1 Description

We used two public datasets in our experiments. The first dataset, SleepAccel, was collected using an Apple Watch and includes motion data, heart rate, and sleep labels from PSG. Data were collected at the University of Michigan from June 2017 to March 2019, and there are 31 subjects in total. The second dataset, DREAMT, was collected using an Empatica E4 wristband. A total of 100 unique participants were recruited from the Duke University Health System (DUHS) Sleep Disorder Lab to participate in the study between May 2022 and September 2022. The total counts and distribution of the data are shown in the table below.

Table 4.1: Label Counts and Distribution for SleepAccel and DREAMT Datasets

Dataset	**WAKE**	**N1**	**N2**	**N3**	**REM**
Counts					
SleepAccel	2,133	1,624	12,184	3,189	5,427
DREAMT	19,971	8,819	39,687	2,661	8,331
Percentage (%)					
SleepAccel	8.7%	6.6%	49.6%	13%	22.1%
DREAMT	25.1%	11.1%	49.9%	3.3%	10.5%

4.1.2 Preprocessing

Due to the different sampling frequencies of sensors in the datasets, such as the heart rate at 64 Hz and motion at 32 Hz in the DREAMT dataset, and heart rate at 0.2 Hz and motion at 50 Hz in the SleepAccel dataset, it was necessary to align these data for analysis. To synchronize the data and preserve more temporal information, interpolation was used to upsample all data to 64 Hz. However, this process can introduce some artifacts, so future work may consider a compromise approach, such as upsampling some data while downsampling others.

Additionally, because the data includes preparation and measurement start stages, it is possible to accurately determine the start time of measurement for users. Therefore, in addition to three-axis accelerometer data and heart rate data, a relative time difference to the start of measurement (or lights-off) was added as a feature.

To further preprocess the data, Interquartile Range (IQR) normalization was applied to the 3-axis acceleration data and heart rate data. IQR normalization helps to reduce the impact of outliers by scaling the data based on the interquartile range, thus ensuring a more robust and stable input for the model.

4.2 Baselines

We compare our proposed model with one baseline method that aim to reduce computational costs, more baselines would be further implemented:

- **CNN-Transformer**: This model uses four convolutional layers followed by the encoder component of a transformer. It is designed to leverage both convolutional and transformer architectures to achieve efficient feature extraction and sequence modeling.

4.3 Experiment Setting

We set the hyperparameter β in the Class-Balanced Softmax Cross Entropy to 0.99 and the confidence threshold θ to 0.7. The Adam optimizer is used with a learning rate of 0.01, and the batch size is set to 512.

To ensure robust model validation and to prevent overfitting, we employ stratified 5-fold cross-validation. This technique involves splitting the training data into five equal parts, or folds, while maintaining the class distribution within each fold. The model is trained on four folds and validated on the remaining fold. This process is repeated five times, with each fold used exactly once for validation. The cross-validation process helps in assessing the model's performance more reliably, as it ensures that each data point has been used for both training and validation, thereby providing a comprehensive evaluation of the model's ability to generalize to unseen data.

Table 4.2: Experiment Settings

Setting	**Value**
β in Class-Balanced Softmax	0.99
Cross-Validation	Stratified 5-Fold
Learning Rate	0.01
Batch Size	512
Confidence Threshold θ	0.8

4.4 Evaluation Metrics

To comprehensively evaluate the performance of the proposed model, we employ several evaluation metrics that capture different aspects of model effectiveness and efficiency. These metrics include accuracy, macro F1-score, Cohen's kappa, FLOPs (Floating Point Operations), and the number of model parameters.

4.4.1 Accuracy

Accuracy measures the proportion of correctly classified instances among the total instances. It is a straightforward metric but can be misleading in the case of imbalanced datasets. Accuracy is defined as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

where TP stands for True Positives, TN for True Negatives, FP for False Positives, and FN for False Negatives.

4.4.2 Macro F1-Score

The macro F1-score is the harmonic mean of precision and recall, calculated for each class independently and then averaged. This metric gives equal weight to each class, making it particularly useful for evaluating performance on imbalanced datasets. The macro F1-score is defined as:

$$\text{F1-score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{Macro F1-score} = \frac{1}{C} \sum_{i=1}^{C} \text{F1-score}_i$$

where C is the number of classes.

4.4.3 Cohen's Kappa

Cohen's kappa measures the agreement between predicted and true labels, taking into account the possibility of agreement occurring by chance. It is particularly useful for imbalanced datasets. Cohen's kappa is defined as:

$$\kappa = \frac{p_o - p_e}{1 - p_e}$$

where p_o is the observed agreement, and p_e is the expected agreement by chance.

4.4.4 FLOPs (Floating Point Operations)

FLOPs measure the computational complexity of the model by counting the number of floating-point operations required to make a prediction. This metric is crucial for evaluating the computational efficiency and feasibility of deploying the model on resource-constrained devices.

4.4.5 Model Parameters

The number of model parameters is an important metric for understanding the model's complexity and potential overfitting. A model with fewer parameters is generally more efficient and suitable for deployment on devices with limited computational resources.

By using these evaluation metrics, we aim to provide a comprehensive assessment of the model's performance, balancing accuracy, robustness, and computational efficiency to ensure its practical applicability for sleep stage classification on consumer wearable devices.

4.5 Performance Comparison

4.5.1 Performance Result

Method (θ)	Accuracy	F1 Score	Cohen's Kappa	FLOPs (M)	Parameters (K)
CNNTransformer	**0.787**	**0.7133**	**0.6932**	9.8566	384.837
Proposed Method (0.8)	0.7267	0.6476	0.6073	**2.0104**	**32.01**

Table 4.3: Comparison of performance metrics between baseline and proposed methods on the SleepAccel dataset.

Method (θ)	Accuracy	F1 Score	Cohen's Kappa	FLOPs (M)	Parameters (K)
CNNTransformer	**0.697**	**0.6178**	**0.54**	9.8566	384.837
Proposed Method (0.8)	0.6308	0.5377	0.4479	**2.0104**	**32.01**

Table 4.4: Comparison of performance metrics between baseline and proposed methods on the DREAMT dataset.

4.5.2 Performance on Different Confidence Thresholds

4.5.3 Implementation of Confidence Threshold on Baseline

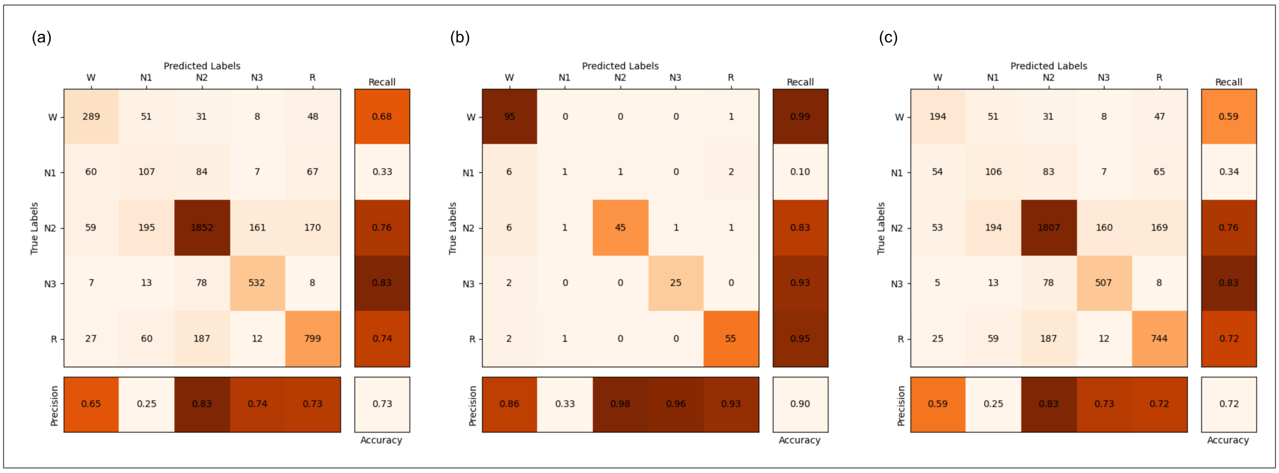

Figure 4.1: Confusion matrix of the proposed method on the SleepAccel dataset: (a) Combined output (b) Simple Classifier output (c) Deep Classifier output

Method (θ)	Accuracy	F1 Score	Cohen's Kappa	FLOPs (M)	Parameters (K)
COPS-CD-1	**0.7423**	**0.6643**	**0.6278**	2.0330	**31.685**
COPS-CD-0.9	0.7354	0.6557	0.6173	2.0256	32.01
COPS-CD-0.8	0.7391	0.6576	0.6227	2.0098	32.01
COPS-CD-0.7	0.7276	0.6496	0.6094	1.9705	32.01
COPS-CD-0.6	0.7130	0.6319	0.5890	**1.8962**	32.01

Table 4.5: Comparison of performance metrics at different confidence thresholds on the SleepAccel dataset.

Method (θ)	Accuracy	F1 Score	Cohen's Kappa	FLOPs (M)	Parameters (K)
COPS-CD-1	**0.6273**	**0.5275**	**0.4353**	2.0330	**31.685**
COPS-CD-0.9	0.6227	0.5187	0.4325	2.0263	32.01
COPS-CD-0.8	0.6197	0.5166	0.4302	2.0157	32.01
COPS-CD-0.7	0.6223	0.5246	**0.4353**	2.0034	32.01
COPS-CD-0.6	0.6201	0.5189	0.4326	**1.9812**	32.01

Table 4.6: Comparison of performance metrics at different confidence thresholds on the DREAMT dataset.

Method (θ)	Accuracy	F1 Score	Cohen's Kappa	FLOPs (M)	Parameters (K)
CDL-CD-1	0.7884	**0.7138**	0.6948	9.8566	**384.837**
CDL-CD-0.9	**0.7907**	0.7123	**0.7019**	9.1792	386.452
CDL-CD-0.8	0.7687	0.6925	0.6727	8.8275	386.452
CDL-CD-0.7	0.7562	0.6734	0.6538	8.3180	386.452
CDL-CD-0.6	0.7077	0.6167	0.5866	**7.6957**	386.452

Table 4.7: Comparison of performance metrics at different confidence thresholds for the CNNTransroemer on the SleepAccel dataset.

Method (θ)	Accuracy	F1 Score	Cohen's Kappa	FLOPs (M)	Parameters (K)
CDL-CD-1	0.6943	0.6202	0.5429	9.8566	384.837
CDL-CD-0.9	0.6812	0.6006	0.5370	9.4794	386.452
CDL-CD-0.8	0.6728	0.5801	0.5246	9.2373	386.452
CDL-CD-0.7	0.6337	0.5443	0.4789	8.9522	386.452
CDL-CD-0.6	0.6248	0.5370	0.4652	8.5095	386.452

Table 4.8: Comparison of performance metrics at different confidence thresholds for the CNNTransroemer on the DREAMT dataset.

Method (θ)	Accuracy	F1 Score	Cohen's Kappa	FLOPs (M)	Parameters (K)
CNNTransformer	0.7870	0.7133	0.6932	9.8566	**384.837**
CNNTransformer (0.8)	**0.7978**	**0.7235**	**0.7071**	**8.9325**	386.452

Table 4.9: Comparison of performance metrics at different confidence thresholds for the CNNTransformer baseline.

Method (θ)	Accuracy	F1 Score	Cohen's Kappa	FLOPs (M)	Parameters (K)
CDL-Deeponly	0.7944	0.7220	0.7030	**9.8566**	**384.837**
CDL-CD-2	0.7963	0.7264	0.7072	**9.8566**	**384.837**
CDL-CD-1	**0.7974**	**0.7282**	**0.7084**	**9.8566**	**384.837**
CDL-CD-0.9	0.7919	0.7211	0.7011	9.8570	385.162
CDL-CD-0.8	0.7849	0.7126	0.6925	9.8570	385.162
CDL-CD-0.7	0.7801	0.7072	0.6853	9.8570	385.162
CDL-CD-0.6	0.7671	0.6921	0.6649	9.8570	385.162

Table 4.10: Comparison of performance metrics at different confidence thresholds for the CNNTransroemer-CD-one on the SleepAccel dataset.

Method (θ)	Accuracy	F1 Score	Cohen's Kappa	FLOPs (M)	Parameters (K)
CDL-Deeponly	0.7944	0.7220	0.7030	**9.8566**	**384.837**
CDL-CD-2	**0.8017**	0.7312	**0.7137**	**9.8566**	**384.837**
CDL-CD-1	0.7951	**0.7263**	0.7049	**9.8566**	**384.837**
CDL-CD-0.9	0.7893	0.7151	0.6991	9.8573	385.482
CDL-CD-0.8	0.7868	0.7105	0.6949	9.8573	385.482
CDL-CD-0.7	0.7799	0.7033	0.6853	9.8573	385.482
CDL-CD-0.6	0.7657	0.6861	0.6638	9.8573	385.482

Table 4.11: Comparison of performance metrics at different confidence thresholds for the CNNTransroemer-CD-two on the SleepAccel dataset.

Method (θ)	Accuracy	F1 Score	Cohen's Kappa	FLOPs (M)	Parameters (K)
CDL-Deeponly	0.7944	0.7220	0.7030	**9.8566**	**384.837**
CDL-CD-2	0.7944	0.7265	0.7086	**9.8566**	**384.837**
CDL-CD-1	**0.7980**	**0.7271**	**0.7091**	**9.8566**	**384.837**
CDL-CD-0.9	0.7978	0.7166	0.6984	9.8573	385.482
CDL-CD-0.8	0.7889	0.7097	0.6955	9.8573	385.482
CDL-CD-0.7	0.7821	0.7083	0.6876	9.8573	385.482
CDL-CD-0.6	0.7589	0.6808	0.6547	9.8573	385.482

Table 4.12: Comparison of performance metrics at different confidence thresholds for the CNNTransroemer-CD-three on the SleepAccel dataset.

Method (θ)	Accuracy	F1 Score	Cohen's Kappa	FLOPs (M)	Parameters (K)
CDL-Deeponly	0.6965	0.6207	0.5450	**9.8566**	**384.837**
CDL-CD-2	0.7003	0.6307	0.5555	**9.8566**	**384.837**
CDL-CD-1	0.7037	0.6343	0.5594	**9.8566**	**384.837**
CDL-CD-0.9	0.6914	0.6125	0.5428	9.8570	385.162
CDL-CD-0.8	0.6952	0.6143	0.5474	9.8570	385.162
CDL-CD-0.7	0.6825	0.5971	0.5316	9.8570	385.162
CDL-CD-0.6	0.6759	0.5872	0.5223	9.8570	385.162

Table 4.13: Comparison of performance metrics at different confidence thresholds for the CNNTransroemer-CD-one on the DREAMT dataset.

Method (θ)	Accuracy	F1 Score	Cohen's Kappa	FLOPs (M)	Parameters (K)
CDL-Deeponly	0.6965	0.6207	0.5450	9.8566	384.837
CDL-CD-2	0.7029	0.6309	0.5587	9.8566	384.837
CDL-CD-1	0.6944	0.6276	0.5504	9.8566	384.837
CDL-CD-0.9	0.6901	0.6034	0.5416	9.8573	385.482
CDL-CD-0.8	0.6721	0.5828	0.5191	9.8573	385.482
CDL-CD-0.7	0.6557	0.5671	0.5009	9.8573	385.482
CDL-CD-0.6	0.6473	0.5538	0.4859	9.8573	385.482

Table 4.14: Comparison of performance metrics at different confidence thresholds for the CNNTransroemer-CD-two on the DREAMT dataset.

Method (θ)	Accuracy	F1 Score	Cohen's Kappa	FLOPs (M)	Parameters (K)
CDL-Deeponly	0.6965	0.6207	0.5450	9.8566	384.837
CDL-CD-2	0.7042	0.6329	0.5605	9.8566	384.837
CDL-CD-1	0.6952	0.6261	0.5519	9.8566	384.837
CDL-CD-0.9	0.6850	0.6080	0.5376	9.8573	385.482
CDL-CD-0.8	0.6617	0.5822	0.5118	9.8573	385.482
CDL-CD-0.7	0.6535	0.5626	0.4975	9.8573	385.482
CDL-CD-0.6	0.6468	0.5517	0.4854	9.8573	385.482

Table 4.15: Comparison of performance metrics at different confidence thresholds for the CNNTransroemer-CD-three on the DREAMT dataset.

Method (θ)	Accuracy	F1 Score	Cohen's Kappa	FLOPs (M)	Parameters (K)
CDL-Deeponly	0.7944	0.7220	0.7030	9.8566	384.837
CDL-CD-2	0.7944	0.7265	0.7086	9.8566	384.837
CDL-CD-1	0.7980	0.7271	0.7091	9.8566	384.837
CDL-CD-0.9	0.7978	0.7166	0.6984	9.8573	385.482
CDL-CD-0.8	0.7889	0.7097	0.6955	9.8573	385.482
CDL-CD-0.7	0.7821	0.7083	0.6876	9.8573	385.482
CDL-CD-0.6	0.7589	0.6808	0.6547	9.8573	385.482

Table 4.16: Comparison of performance metrics at different confidence thresholds for the CNNTransroemer-CD-three on the DREAMT dataset.

Chapter 5 Conclusion

This thesis presents a novel approach to power-efficient sleep stage classification designed for consumer wearable devices. The primary focus has been on developing a deep learning model that balances computational efficiency with classification accuracy, making it suitable for deployment in resource-constrained environments.

The research explored the integration of 3-axis motion data, heart rate, and temporal information relative to light-off as input features. A key contribution of this work is the hierarchical model architecture, which employs a combination of simple and deep classifiers. This architecture allows the model to handle simple classification tasks with minimal computational resources, while reserving more complex feature extraction and classification for a deep classifier. By introducing a confidence-based decision mechanism, the model can dynamically determine whether further processing is necessary, thus optimizing both performance and efficiency.

The implementation of confidence thresholds has proven to be effective in managing the trade-off between computational cost and classification performance. A threshold of 0.8 was identified as providing an optimal balance, improving F1 score and Cohen's kappa while maintaining a low computational footprint. Additionally, the use of multiple simple classifiers at various stages of the CNNTransformer architecture demonstrated the benefits of hierarchical design in enhancing model performance for more challenging tasks.

The performance comparison between the proposed method and the CNNTransformer baseline highlighted significant reductions in computational cost—by approximately 80%—and in the number of parameters—by nearly 90%—on the SleepAccel dataset. While there was a slight trade-off in accuracy, the hierarchical design and confidence-based mechanism led to notable improvements in precision, recall, and overall classification reliability.

In conclusion, the methods developed in this thesis offer a viable solution for power-efficient sleep stage classification on consumer wearable devices. By leveraging hierarchical structures, parameter sharing, and confidence-based decision making, the proposed approach effectively balances the demands of computational efficiency and classification accuracy, paving the way for more practical and scalable sleep monitoring solutions.

Bibliography

[1] Andreas Bulling, Ulf Blanke, and Bernt Schiele. "A tutorial on human activity recognition using body-worn inertial sensors". In: *ACM computing surveys* 46.3 (2014), pp. 1–33.

[2] Oscar D Lara and Miguel A Labrador. "A survey on human activity recognition using wearable sensors". In: *IEEE communications surveys & tutorials* 15.3 (2013), pp. 1192–1209.

[3] Muhammad Shoaib et al. "A survey of online activity recognition using mobile phones". In: *Sensors* 15.1 (2015), pp. 2059–2085.

[4] Kaixuan Chen et al. "A comprehensive survey of deep learning-based human activity recognition". In: *Proceedings of the IEEE* 109.9 (2021), pp. 1601–1622.

[5] Jindong Wang et al. "Deep learning for sensor-based activity recognition: A survey". In: *Pattern Recognition Letters* 119 (2019), pp. 3–11.

[6] Jianbo Yang et al. "Deep convolutional neural networks on multichannel time series for human activity recognition". In: *Twenty-fourth international joint conference on artificial intelligence*. 2015.

[7] Nils Y Hammerla, Shane Halloran, and Thomas Ploetz. "Deep, convolutional, and recurrent models for human activity recognition using wearables". In: *arXiv preprint arXiv:1604.08880* (2016).

[8] Charissa Ann Ronao and Sung-Bae Cho. "Human activity recognition with smartphone sensors using deep learning neural networks". In: *Expert systems with applications* 59 (2016), pp. 235–244.

[9] Andrey Ignatov. "Real-time human activity recognition from accelerometer data using Convolutional Neural Networks". In: *Applied Soft Computing* 62 (2018), pp. 915–922.

[10] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems*. 2017, pp. 5998–6008.

[11] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *arXiv preprint arXiv:1810.04805* (2019).

[12] Tom Brown et al. “Language models are few-shot learners”. In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[13] Zechen Li et al. “Sensorllm: Aligning large language models with motion sensors for human activity recognition”. In: *arXiv preprint arXiv:2410.10624* (2024).

[14] Shuangjian Li et al. “P2LHAP:Wearable sensor-based human activity recognition, segmentation and forecast through Patch-to-Label Seq2Seq Transformer”. In: *arXiv preprint arXiv:2403.08214* (2024).

[15] Hao Zhang et al. “MoPFormer: Motion-Primitive Transformer for Wearable-Sensor Activity Recognition”. In: *arXiv preprint arXiv:2505.20744* (2025).

[16] Kunpeng Zhao, Asahi Miyazaki, and Tsuyoshi Okita. “Detecting Informative Channels: ActionFormer”. In: *arXiv preprint arXiv:2505.20739* (2025).

[17] Yuwei Zhang et al. “SensorLM: Learning the Language of Wearable Sensors”. In: *arXiv preprint arXiv:2506.09108* (2025).

[18] Gabriele Civitarese et al. “Large Language Models are Zero-Shot Recognizers for Activities of Daily Living”. In: *arXiv preprint arXiv:2407.01238* (2024).

[19] Sijie Ji, Xinzhe Zheng, and Chenshu Wu. “HARGPT: Are LLMs Zero-Shot Human Activity Recognizers?” In: *Proceedings of the 2024 IEEE International Workshop on Foundation Models for Cyber-Physical Systems & Internet of Things (FMSys)*. IEEE, 2024, pp. 38–43.

[20] Emilio Ferrara. “Large Language Models for wearable sensor-based human activity recognition, health monitoring, and behavioral modeling: a survey of early trends, datasets, and challenges”. In: *Sensors* 24.15 (2024), p. 5045.

[21] Xiaomin Ouyang and Mani Srivastava. “LLMSense: Harnessing LLMs for High-level Reasoning Over Spatiotemporal Sensor Traces”. In: *arXiv preprint arXiv:2403.19857* (2024).

[22] Yuan Sun and Jorge Ortiz. “An AI-Based System Utilizing IoT-Enabled Ambient Sensors and LLMs for Complex Activity Tracking”. In: *arXiv preprint arXiv:2407.02606* (2024).

[23] Shentong Mo et al. “IoT-LM: Large Multisensory Language Models for the Internet of Things”. In: *arXiv preprint arXiv:2407.09801* (2024).

[24] Franklin Y Ruan et al. "Foundation Models for Wearable Movement Data in Mental Health Research". In: *arXiv preprint arXiv:2411.15240* (2024).