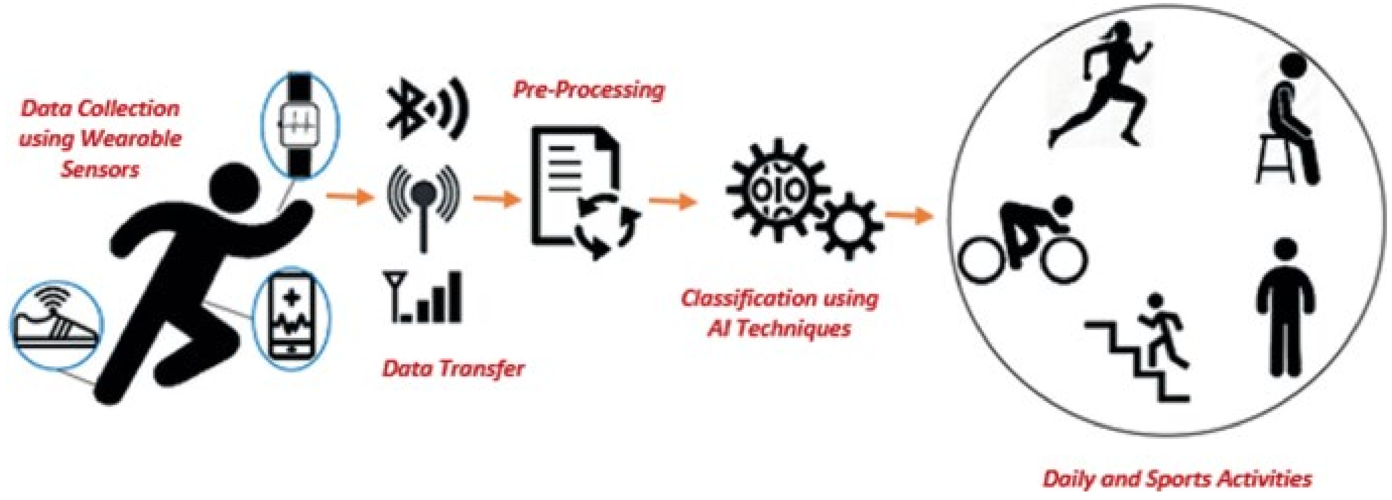
Lightweight Human Activity Recognition with Model Distillation



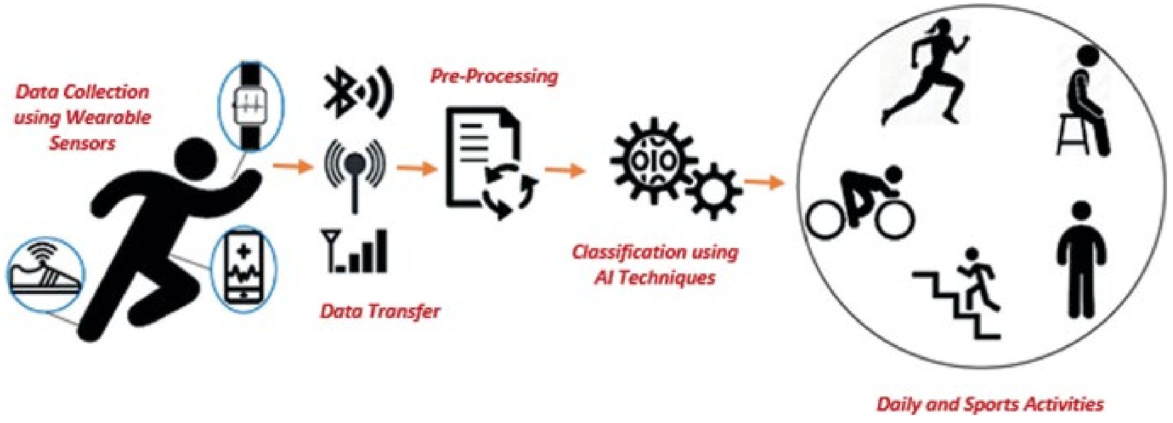
Student name: Longzhao Gong

Student ID number: 20034766

Applications	Reasearch Tasks	Dataset	Data Processing	DL models	Performance Measures and Results
PAMAP2 processing	Data Processing	PAMAP2	• Complete heart rate data • Align sample rate • Data cleansing	N/a	N/a
mHealth processing	Data Processing	mHealth	• Bandpass filtering of ECG signals • Calculate heart rate • Data cleansing	N/a	N/a
Data Fusion&Enhancement	Data Processing	mHealth&PAMAP2	Align mappings	N/a	N/a
Model Training	Train Teacher model	Processed Dataset	GaussianNoise	LSTM+SimpleRNN	Accuracy 99%
Offline Distillation	Train offline Student model	Processed Dataset	N/a	LSTM	Accuracy 97%
Online Distillation	Train online Student model	Processed Dataset	N/a	LSTM	Accuracy 99%

Outline

- Introduction
- Applications
- Datasets
- Data Processing
- Deep Learning Models
- Performance Results
- Discussions
- Conclusions
- Challenges
- Future Directions
- References



- **Introduction**

This project focuses on human activity recognition using time-series data from wearable sensors. By leveraging teacher-student knowledge distillation, the aim is to improve model efficiency while maintaining accuracy for deployment on resource-constrained edge devices.

- **Applications**

Healthcare Monitoring: Recognizing daily activities for elderly care and rehabilitation.

Fitness Tracking: Providing insights into physical activity for personalized fitness plans.

IoT Devices: Enhancing smart wearables to monitor activity in real time.

Industrial Safety: Monitoring workers' postures to prevent injuries.

Datasets

1. MHEALTH Dataset

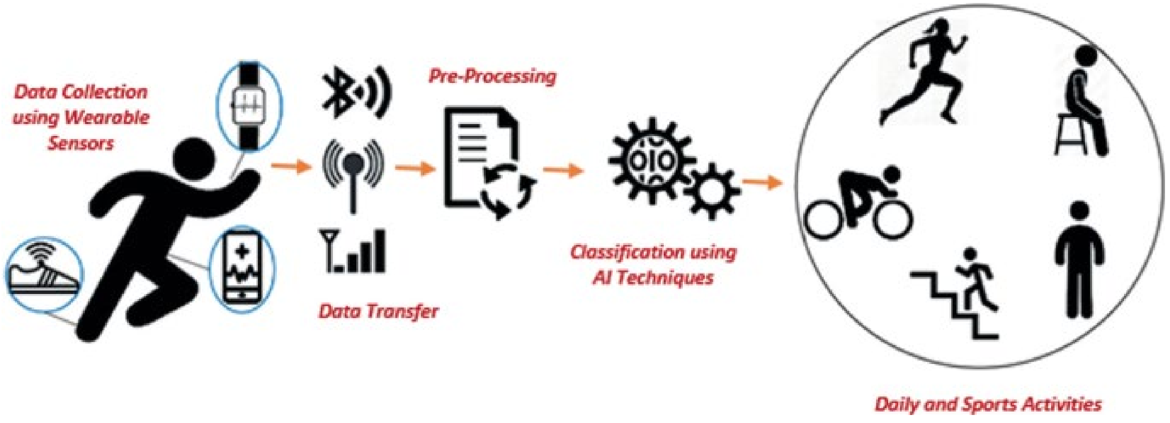
- **Data Content**: The MHEALTH dataset contains motion and physiological signals captured from different body parts, including chest accelerometer, ankle accelerometer, wrist accelerometer, gyroscope, magnetometer, and electrocardiogram (ECG) signals. The data was collected from 10 volunteers during different physical and daily activities.
- **Sampling Rate**: All sensor data was sampled at 50 Hz.
- **Sensor Placement**: Sensors were positioned on the chest, left ankle, and right wrist, allowing them to capture movement and physiological state from different parts of the body.
- **Activities**: The dataset includes 12 types of daily activities, such as standing still, sitting, lying down, walking, running, cycling, climbing stairs, and more.

2. PAMAP2 Physical Activity Monitoring Dataset

- **Data Content**: The PAMAP2 dataset consists of a collection of motion and physiological data captured from various body parts. It aims to recognize multiple daily activities, including sitting, standing, walking, running, cycling, climbing stairs, etc.
- **Sampling Rate**: The sensor data was sampled at 100 Hz, while the heart rate data was approximately sampled at 9 Hz.
- **Sensor Placement**: The data was collected from three wearable devices placed on the chest, left ankle, and wrist, capturing accelerometer, gyroscope, magnetometer, and temperature data.
- **Activities**: The dataset includes 18 types of daily activities, covering both static (e.g., sitting, lying down) and dynamic activities (e.g., walking, running, cycling, jumping).

Data Processing

- **PAMPA data processing**
  - ➢ Resampling rate from 100 Hz to 50 Hz
- **mHealth data processing**
  - ➢ Use ECG data to get Heart Rate
    1. Bandpass Filter
    2. R Peak Detection
    3. Heart Rate Calculation
- **Align sensor positions**
- **Align activity labels**
- **Dataset Fusion**
  - ➢ Class Imbalance
    1. SMOTE
    2. Under sampler
  - ➢ Sliding Window with overlap
  - ➢ Feature and Label Segmentation



Model Training

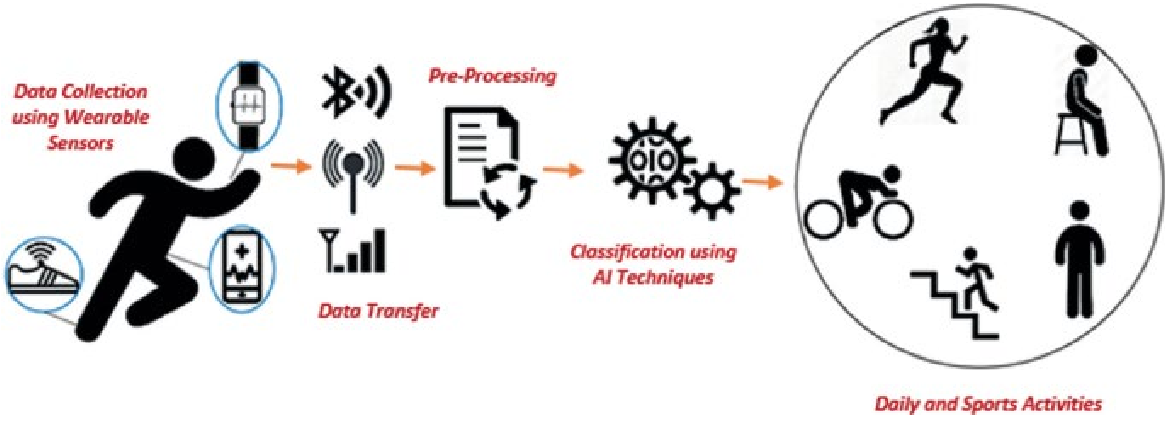
Teacher Model

- **GaussianNoise Layer**: Adds Gaussian noise with a standard deviation of 0.1 to the input data.
- **LSTM Layer**:
  - Units: 128
  - Return sequences
  - Regularization: L2 with a penalty of 0.01 to prevent overfitting.
- **BatchNormalization Layer**
- **Dropout Layer**: Drops 40% of the neurons randomly to prevent overfitting.
- **SimpleRNN Layer**:
  - Units: 64
  - Regularization: L2 with a penalty of 0.01
- **Dropout Layer**: Drops 40% of the neurons randomly.
- **Dense Layer**:
  - Units: 64
  - Activation: ReLU
  - Regularization: L2 with a penalty of 0.01
- **Dropout Layer**: Drops 40% of the neurons randomly.
- **Dense Layer** (Output Layer):
  - Units: Number of classes (same as y_train.shape[1])
  - Activation: Softmax for multi-class classification

Model Training

Teacher Model

- **SGD Optimizer**
  - Learning Rate (learning_rate=1e-3)
  - Momentum (momentum=0.9)
  - Nesterov (nesterov=True)
- **Early Stopping Callback**
  - Monitor (monitor='val_loss')
  - Patience (patience=5):
  - Restore Best Weights (restore_best_weights=True)
- **Class weights**
  - (class_weight='balanced') make the model pay more attention to underrepresented classes.

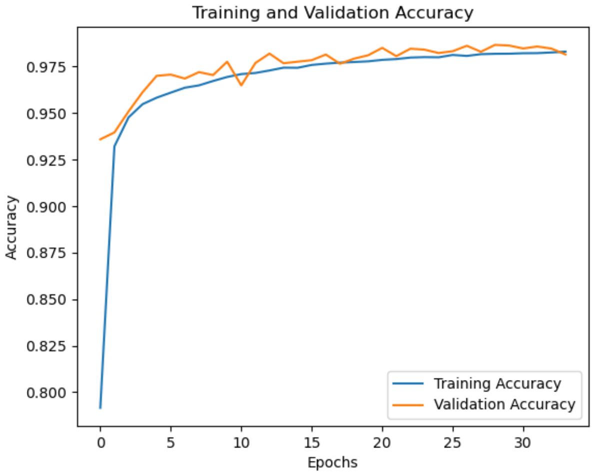


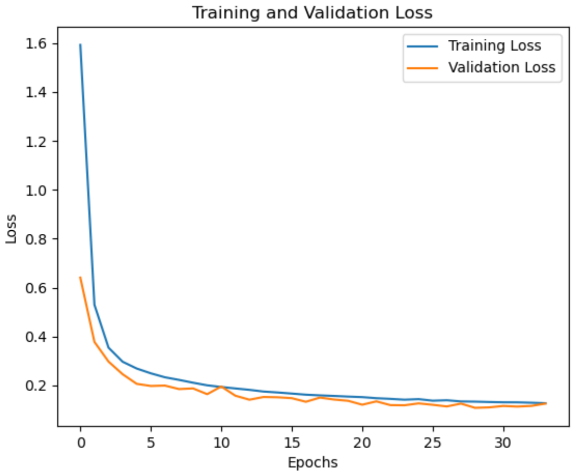
Performance Results

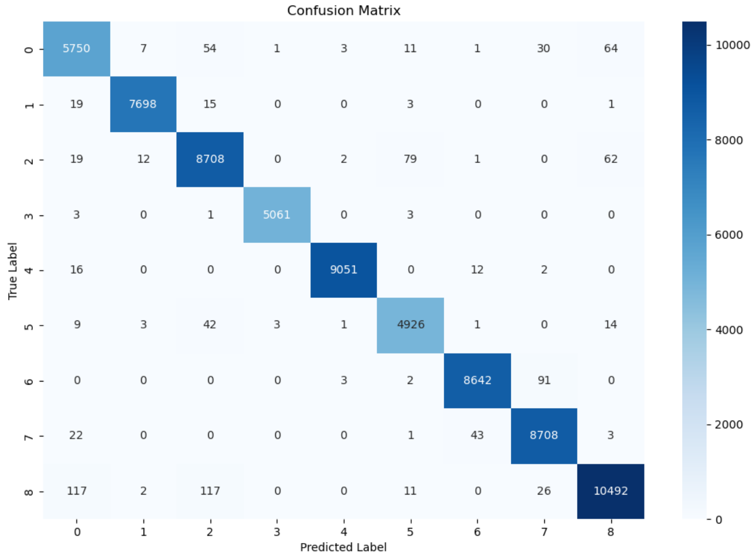
Teacher Model

Classification Report:

	precision	recall	f1-score	support
0	0.97	0.97	0.97	5921
1	1.00	1.00	1.00	7736
2	0.97	0.98	0.98	8883
3	1.00	1.00	1.00	5068
4	1.00	1.00	1.00	9081
5	0.98	0.99	0.98	4999
6	0.99	0.99	0.99	8738
7	0.98	0.99	0.99	8777
8	0.99	0.97	0.98	10765
accuracy			0.99	69968
macro avg	0.99	0.99	0.99	69968
weighted avg	0.99	0.99	0.99	69968







Model Training

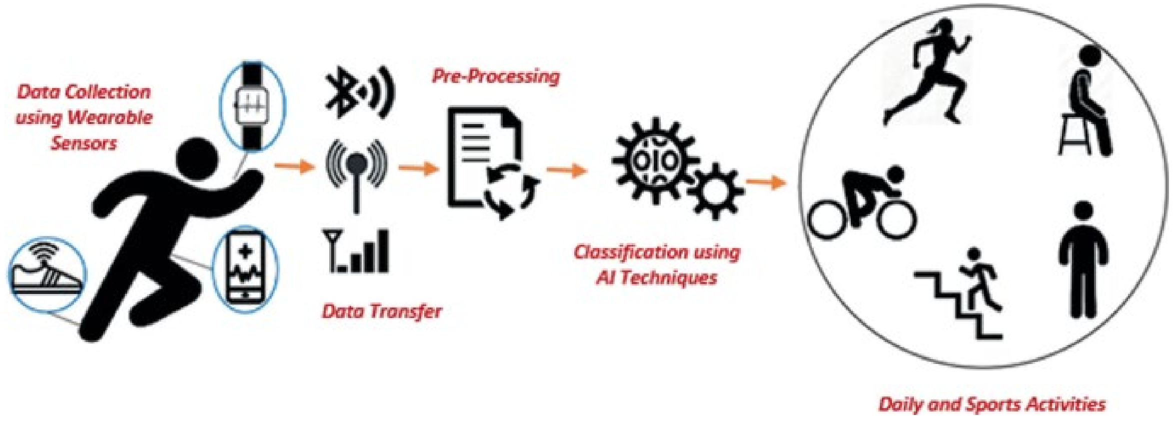
Offline Knowledge Distillation

- **Load data**
- **Distillation loss function**
  - Hard Label Loss(Categorical Crossentropy)
  - Soft Label Loss(KL Divergence)
  - Overall Loss(alpha * hard_loss + (1 - alpha) * soft_loss)
- **Build Student Model**

Model Training

Student Model -Offline Knowledge Distillation

- LSTM Layer,
  - 64 units
  - L2 regularization
  - Outputs sequences (return_sequences=True)
- Batch Normalization (batch normalization)
- Dropout (rate=60%)
- LSTM Layer
  - 32 units
- Batch Normalization
- Dropout (60%)
- Dense Layer
  - 32 units
  - ReLU activation function
  - L2 regularization
- Dropout (60%)
- Output Dense Layer
  - Softmax for multi-class classification
  - L2 regularization

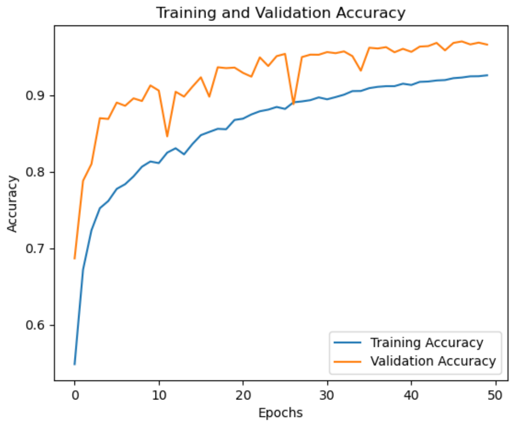


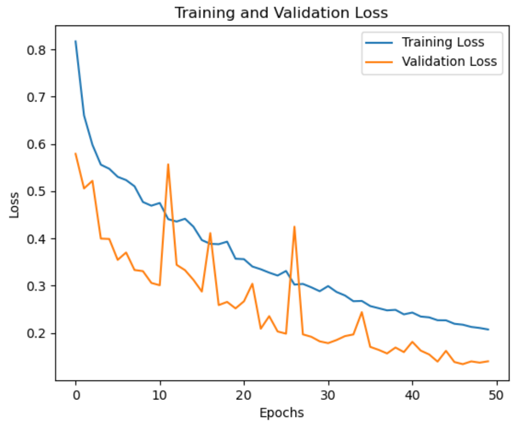
Performance Results

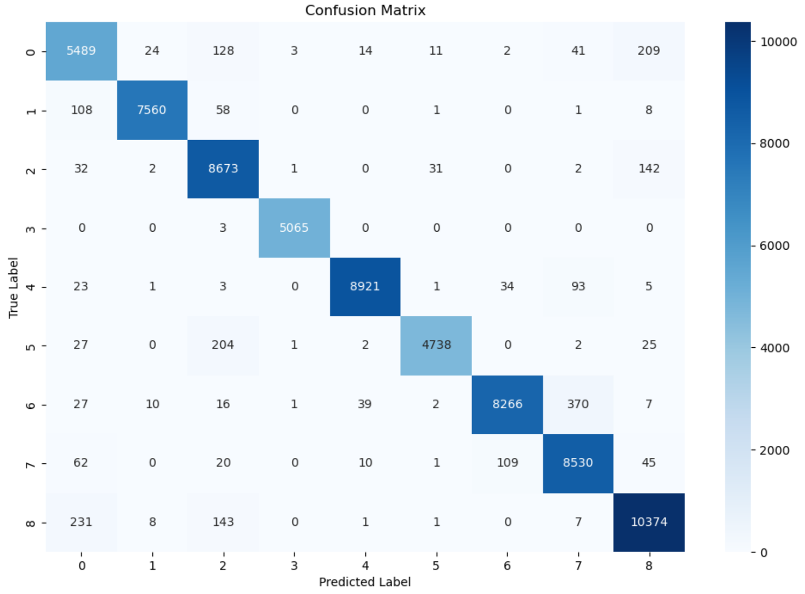
Offline Knowledge Distillation

Classification Report:

	precision	recall	f1-score	support
0	0.91	0.93	0.92	5921
1	0.99	0.98	0.99	7736
2	0.94	0.98	0.96	8883
3	1.00	1.00	1.00	5068
4	0.99	0.98	0.99	9081
5	0.99	0.95	0.97	4999
6	0.98	0.95	0.96	8738
7	0.94	0.97	0.96	8777
8	0.96	0.96	0.96	10765
accuracy			0.97	69968
macro avg	0.97	0.97	0.97	69968
weighted avg	0.97	0.97	0.97	69968







Model Training

Online Knowledge Distillation

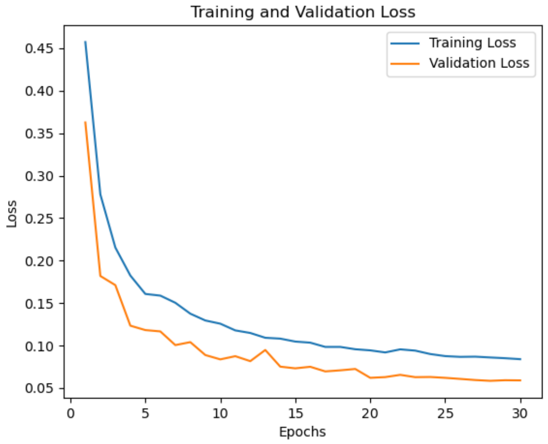
- **Load data**
- **Distillation loss function(same as Offline)**
  - Hard Label Loss
    - ➢ Computes the cross-entropy loss between the true labels and the student's predictions
  - Soft Label Loss
    - ➢ Computes the KL divergence between the teacher model's soft labels and the student's predictions
  - Overall Loss(alpha * hard_loss + (1 - alpha) * soft_loss)
- **Distillation loss(**tf.GradientTape()**)**
  - Compute distillation loss dynamically
  - Compute the gradients of the loss with respect to the model's parameters
  - Update the model weights based on the computed gradients
- **Build Student Model(same as Offline)**

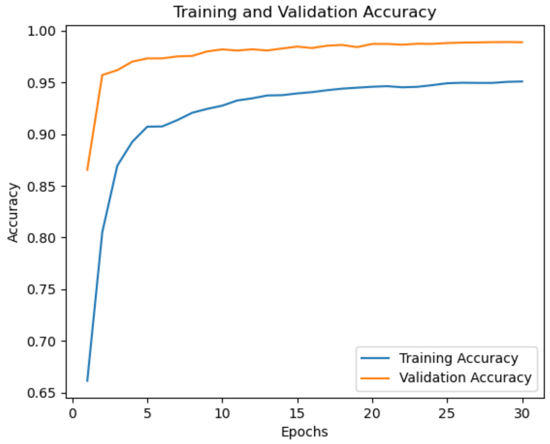
Performance Results

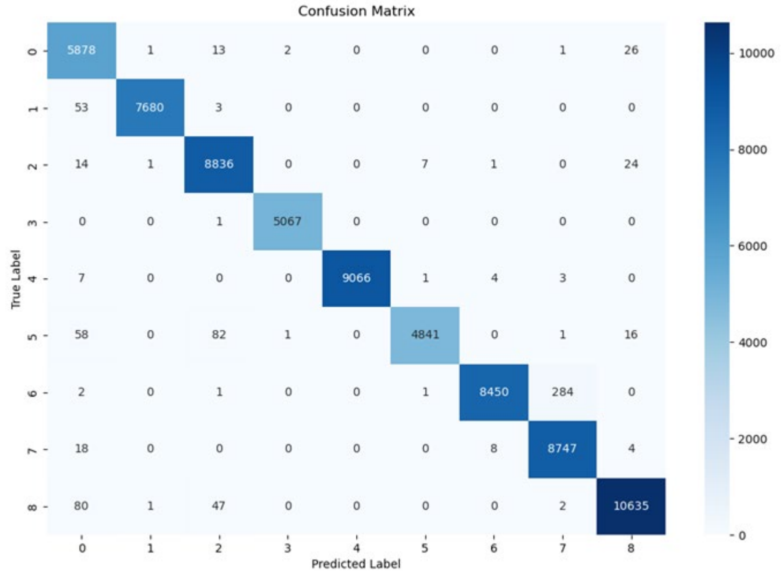
Online Knowledge Distillation

Classification Report:

	precision	recall	f1-score	support
0	0.96	0.99	0.98	5921
1	1.00	0.99	1.00	7736
2	0.98	0.99	0.99	8883
3	1.00	1.00	1.00	5068
4	1.00	1.00	1.00	9081
5	1.00	0.97	0.98	4999
6	1.00	0.97	0.98	8738
7	0.97	1.00	0.98	8777
8	0.99	0.99	0.99	10765
accuracy			0.99	69968
macro avg	0.99	0.99	0.99	69968
weighted avg	0.99	0.99	0.99	69968







Discussions

0. Validation accuracy is higher than training accuracy

- **Dropout**
  - Dropout randomly ignores some neurons, reducing the model's dependency on specific neurons, which enhances the model's generalization. This leads to a drop in performance during training, but during validation, Dropout is disabled, allowing the model to utilize all neurons, which can result in better performance on the validation set.
- **L2 Regularization**
  - L2 regularization applies constraints on the weights to prevent the model from overfitting. This suppresses the model's capacity during training, but during validation, regularization does not affect the model's inference, which may cause better performance on the validation set compared to the training set.
- **Batch Normalization**:
  - Batch Normalization computes the mean and variance based on the current batch of data, which may cause fluctuations in the performance of each mini-batch. However, during validation, Batch Normalization uses the running average and variance from the entire training data, resulting in more stable performance. This could also contribute to the model's better performance on the validation set compared to the training set.
- **Early Stopping callback**
  - Early Stopping stops training process as soon as the performance on the validation set starts to deteriorate. This ensures that the model achieves optimal generalization performance on the validation set but also means that the model may not be fully converged on the training set, leading to situations where the performance on the validation set might be better than on the training set.

Discussions

1. Data Processing

- **Cross-Dataset Integration**:
  - Integrated the **PAMAP2** and **mHealth** datasets to enhance sample diversity and address potential generalization issues that arise when using only one dataset.
- **Data Preprocessing**:
  - **Missing Value Imputation**: Managed incomplete data using imputation techniques to mitigate their impact on model training.
  - **Feature Normalization**: Applied **Min-Max scaling** to sensor data (such as accelerometer, gyroscope, and temperature) to ensure that features are on a comparable scale.
- **Feature Selection**:
  - Selected key sensor features like **accelerometer**, **heart rate**, and **gyroscope** based on task requirements.
- **Label Alignment**:
  - Addressed label discrepancies between datasets to ensure consistent activity labels, allowing for unified model training.

Discussions

2. Model Design

- **Teacher Model**:
  - Designed a deep **LSTM + SimpleRNN** model to capture both **long-term and short-term dependencies** in sequential data.
  - Integrated techniques like **Gaussian noise addition**, **L2 regularization**, and **Dropout** to improve model generalization.
- **Student Model**:
  - Developed a lightweight **LSTM-based student model** to reduce computational complexity while maintaining sequential data modeling capabilities.
  - Focused on creating an efficient architecture suitable for **resource-constrained edge devices**.

Discussions

3. Knowledge Distillation

- **Hard and Soft Label Balancing**:
  - Experimented with the weighting parameter (α) in the distillation loss to balance **hard labels** (classification targets) and **soft labels** (teacher model predictions).
- **Online vs. Offline Distillation**:
  - Compared **online distillation** (real-time predictions from the teacher model) and **offline distillation** (pre-saved teacher model predictions) to evaluate differences in efficiency and model performance.
- **Distillation Loss Construction**:
  - Constructed a custom **distillation loss** combining **hard label cross-entropy** with **soft label KL divergence**, analyzing the trade-off between focusing on ground truth labels versus teacher predictions.

Discussions

4. Model Training

- **Learning Rate Scheduling**:
  - Used an **exponential decay learning rate** scheduler to maintain a balance between training convergence speed and final model performance.
- **Batch Size Selection**:
  - Investigated the impact of different **batch sizes** on training efficiency and model performance.
- **Class Imbalance Handling**:
  - Addressed **class imbalance** by incorporating **class weights** into the loss function to prevent overfitting to more frequent classes.
- **Early Stopping**:
  - Implemented **early stopping** to prevent overfitting, accelerate training, and find the optimal point to stop training.

Discussions

5. Performance Evaluation

- **Evaluation Metrics**:
  - Evaluated model performance using standard metrics like **precision**, **recall**, **F1-score**, and **confusion matrices**.
- **Training and Validation Curve Analysis**:
  - Analyzed **training and validation loss/accuracy curves** to understand model convergence behavior and detect overfitting risks.
- **Comparison of Online Distillation and Offline Training**:
  - Compared the student model's performance against the teacher model, validating the effectiveness of the **knowledge distillation** process in both online and offline settings.

Discussions

6. Applications and Deployment

- **Adaptation for Edge Devices**:
  - Optimized the **student model** for deployment on **resource-constrained edge devices**, focusing on reducing computational requirements while maintaining accuracy.
- **Real-World Scenarios**:
  - Envisioned possible applications in **activity recognition** for **IoT** devices and **health monitoring** systems.
- **Model Compression and Optimization**:
  - Explored further **compression** and **optimization techniques** to improve inference speed without sacrificing accuracy, including knowledge distillation.

Discussions

7. Project Challenges

- **Cross-Dataset Integration**:
  - Overcame challenges in aligning **activity labels** and **feature structures** across different datasets to enable coherent training.
- **Hardware Constraints**:
  - Managed **GPU memory limitations** by adjusting virtual memory allocation and reducing **batch size**.
- **Distillation Effectiveness**:
  - Investigated techniques to **maximize the performance gain** of the student model through effective **knowledge distillation**.

Conclusions

- **The Teacher Model is 1500Kb, but the Online-Student Model is 180Kb, and they have the same performance.**
- **Effective Cross-Dataset Integration**
  - Integrating the PAMAP2 and mHealth datasets provided significant diversity in the data, enhancing model generalizability.
- **Model Design Balances Complexity and Efficiency**
  - **Teacher Model**: The use of a more complex LSTM + SimpleRNN architecture for the teacher model was effective in capturing both long-term and short-term temporal dependencies. The addition of techniques like Dropout and Gaussian noise proved instrumental in enhancing the model’s generalization.
  - **Student Model**: Designing a lightweight student model allowed for computational efficiency while maintaining sufficient accuracy for HAR tasks. This was particularly useful for deployment in resource-constrained environments such as edge devices.

Conclusions

- **Knowledge Distillation Effectiveness**
  - **Offline Distillation**: Reduced computational demands during training by leveraging pre-saved teacher predictions, making it suitable for environments with limited computational resources.
  - **Online Distillation**: Allowed the teacher to adapt in real-time, which improved the interaction between teacher and student models, but at the cost of increased computational load
- **Training Strategies and Optimizations Lead to Robust Generalization**
  - **Early Stopping and Regularization**: Provided good generalization performance on the validation set. The model learned well without overly depending on any specific features of the training data.
  - **Learning Rate Scheduling**: The use of exponential decay helped balance convergence speed and achieving a strong final model. This resulted in consistent improvements in both training and validation metrics.
  - **Class Imbalance Handling**: Using a combination of SMOTE, under-sampling, and class-weighted loss provided effective mitigation of class imbalance, ensuring the model was not biased towards the majority classes and performed well across all activities.

Challenges

- **Training Complexity:**
  - **High Computational Cost:** Training deep models like LSTM with large input windows required significant computational resources.
  - **Resource Limitation:** Limited GPU memory necessitated reducing batch sizes and carefully managing memory usage, slowing down model convergence.
- **Knowledge Distillation Challenges:**
  - **Optimization of Distillation Process:** Balancing between hard label and soft label contributions in the distillation loss was challenging. Finding the optimal value for the temperature parameter also required substantial experimentation.
- **Generalization to Real-World Scenarios:**
  - **Noise and Variability:** Sensor data captured in real-world environments is subject to variability and noise, affecting model reliability when tested outside the dataset environment.
  - **Model Deployment on Edge Devices:** The constraints of edge devices, including memory and computational limitations, made deployment challenging despite the knowledge distillation process.

Future Directions

- **Advanced Training Techniques:**
  - **Transfer Learning:** Apply transfer learning from pre-trained models to speed up the training process and improve model accuracy for activity recognition tasks.
  - **Curriculum Learning:** Introduce curriculum learning by first training the model on simpler activities before gradually moving on to more complex ones, helping to improve overall performance.
- **Model Compression for Edge Deployment:**
  - **Quantization and Pruning:** Further compress the student model using quantization and pruning techniques to make it even more efficient for deployment on edge devices.
  - **Hardware-Specific Optimization:** Tailor the model specifically for edge hardware, such as ARM processors, to improve efficiency and reduce latency during inference.
- **Improving Model Robustness and Generalization:**
  - **Augmentation Techniques:** Use advanced data augmentation techniques to simulate noise and variability encountered in real-world settings, thereby enhancing the robustness of the model.
  - **Federated Learning:** Explore federated learning approaches to train models using distributed sensor data, maintaining privacy and improving robustness without aggregating data centrally.
- **Exploring Other Distillation Techniques:**
  - **Multi-Teacher Distillation:** Utilize multiple teacher models, each specialized in different activities, to improve the knowledge passed on to the student model.
  - **Ensemble Student Models:** Create an ensemble of student models trained through distillation to improve final performance, especially for more complex activities.

References

- Pienaar S W, Malekian R. Human activity recognition using LSTM-RNN deep neural network architecture[C]//2019 IEEE 2nd wireless africa conference (WAC). IEEE, 2019: 1-5.
- Murad A, Pyun J Y. Deep recurrent neural networks for human activity recognition[J]. Sensors, 2017, 17(11): 2556.
- Chen K, Zhang D, Yao L, et al. Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities[J]. ACM Computing Surveys (CSUR), 2021, 54(4): 1-40.
- Hinton G. Distilling the Knowledge in a Neural Network[J]. arXiv preprint arXiv:1503.02531, 2015.
- Gonçalves P H N, Bragança H, Souto E. Efficient Human Activity Recognition on Wearable Devices Using Knowledge Distillation Techniques[J]. Electronics, 2024, 13(18): 3612.
- Xu Q, Wu M, Li X, et al. Reinforced Cross-Domain Knowledge Distillation on Time Series Data[C]//The Thirty-eighth Annual Conference on Neural Information Processing Systems.
- Qiu S, Zhao H, Jiang N, et al. Multi-sensor information fusion based on machine learning for real applications in human activity recognition: State-of-the-art and research challenges[J]. Information Fusion, 2022, 80: 241-265.
- Hamad R A, Yang L, Woo W L, et al. Joint learning of temporal models to handle imbalanced data for human activity recognition[J]. Applied Sciences, 2020, 10(15): 5293.

References

- Gou J, Yu B, Maybank S J, et al. Knowledge distillation: A survey[J]. International Journal of Computer Vision, 2021, 129(6): 1789-1819.
- Polino A, Pascanu R, Alistarh D. Model compression via distillation and quantization[J]. arXiv preprint arXiv:1802.05668, 2018.
- Yuan Z, Yang Z, Ning H, et al. Multiscale knowledge distillation with attention based fusion for robust human activity recognition[J]. Scientific Reports, 2024, 14(1): 12411.
- Chen Z, Zhang L, Cao Z, et al. Distilling the knowledge from handcrafted features for human activity recognition[J]. IEEE Transactions on Industrial Informatics, 2018, 14(10): 4334-4342.
- Romero A, Ballas N, Kahou S E, et al. Fitnets: Hints for thin deep nets[J]. arXiv preprint arXiv:1412.6550, 2014.
- Li K, Yu L, Wang S, et al. Towards cross-modality medical image segmentation with online mutual knowledge distillation[C]//Proceedings of the AAAI conference on artificial intelligence. 2020, 34(01): 775-783.
- Ni J, Ngu A H H, Yan Y. Progressive cross-modal knowledge distillation for human action recognition[C]//Proceedings of the 30th ACM International Conference on Multimedia. 2022: 5903-5912.
- Shuvo M M H, Islam S K, Cheng J, et al. Efficient acceleration of deep learning inference on resource-constrained edge devices: A review[J]. Proceedings of the IEEE, 2022, 111(1): 42-91.