Multimedia Tools and Applications
https://doi.org/10.1007/s11042-025-21086-5

Human activity recognition using transformer-based positional encoding in a federated learning framework

Mohammad Ariaeimehr[1] · **Reza Ravanmehr**[1]

Received: 16 September 2024 / Revised: 3 July 2025 / Accepted: 2 August 2025

Abstract
Human Activity Recognition (HAR) aims to identify specific movements or actions based on sensor data. Existing techniques utilizing deep neural networks have several limitations, including insufficient accuracy in activity classification and a high computational cost. Moreover, centralized approaches that employ server-side analysis raise significant concerns regarding privacy. This research uses the transformer technique to address the first issue, leveraging the attention mechanism. This mechanism enables modeling dependencies regardless of the distance in the input or output sequences. The Transformer is not sensitive to the order of input tokens and lacks an inherent understanding of sequential positioning; thus, an additional encoding is added to the input tokens. However, this encoding can limit the model's ability to capture fine-grained temporal relationships and may hinder optimization efficiency, leading to slower convergence. For this purpose, the proposed method adds positional encoding directly to the attention matrix at each head of the attention module. To address the second issue, Federated Learning (FL) is employed to maintain the privacy of user-sensitive data collected from wearable sensors. In our Federated Learning model, users only need to upload their local model weights to the server to generate training results, allowing sensitive data to remain on the user's device. Experimental results demonstrate that MHAT-FL achieves state-of-the-art performance across PAMAP2, MotionSense, and Opportunity datasets, attaining F1-scores of 99.52%, 99.11%, and 99.79%, respectively. Moreover, MHAT-FL demonstrates strong privacy preservation by maintaining low Membership Inference Attack (MIA) accuracy, utilizing federated methods such as FedAvg and FedPer, with MIA accuracies of 16.67% on PAMAP2, 16.64% on MotionSense, and 16.66% on Opportunity. These results highlight MHAT-FL's effectiveness in advancing Human Activity Recognition while ensuring strong privacy preservation, making it a reliable and secure solution for decentralized data scenarios.

Keywords Human activity recognition · Deep neural networks · Federated learning · Transformer · Positional encoding · Self-Attention

✉ Reza Ravanmehr
r.ravanmehr@iau.ac.ir

Mohammad Ariaeimehr
m.ariaeimehr@iau.ir

[1] Department of Computer Engineering, CT.C, Islamic Azad University, Tehran, Iran

Published online: 18 September 2025

1 Introduction

Human activity recognition (HAR) has emerged as one of the most significant research areas in the last decade, identifying human activities based on sensor data [1]. This area has gained considerable attention due to its vital role in applications such as healthcare [2], fitness [3], and skill assessment [4]. Despite the challenges in this field, technologies such as smartphones and personal tracking devices have made data collection inexpensive and common [5].

Numerous studies have been conducted to improve activity recognition systems by providing suitable models. Despite the progress made in this research area, challenges such as the accuracy of activity recognition, data privacy, communication efficiency, data heterogeneity, and the extensive need for computing power still require attention.

In recent years, various methods of deep neural networks and hybrid models have been developed to address these challenges. However, a unified framework to deliver effective HAR and balance all these requirements is missing.

One of the critical issues in HAR is data privacy, as sensitive user data is often required to be collected and processed. Federated Learning (FL) has been introduced to address the data privacy challenge that arises when machine learning models require sensitive user data for training purposes. In FL, users only need to upload their local model weights to the server to generate training results, allowing sensitive data to remain on the user's device [6–8]. To this end, the user first uploads the model parameters to the server, which then uses the weighted average method to collect the user's model parameters, build the updated model, and send it to all participating users.

In this study, to ensure data privacy, facilitate efficient communication, and address data heterogeneity arising from diverse sensor sources, two federated learning techniques, FedAvg and FedPer, were employed to resolve the following issues:

1. Data privacy: In traditional approaches, users' personal activity data must be transferred to relevant servers, and the model must be built centrally, which increases privacy concerns. Using FedAvg and FedPer models enables models to learn from distributed data while enhancing privacy.
2. Communication efficiency: Transferring large amounts of data from user devices to centralized servers can consume a significant amount of bandwidth. FedAvg and FedPer address this problem to a large extent by sharing only model updates, not the raw data.
3. Data heterogeneity: Real-world data in HAR applications are often heterogeneous due to the variety of sensors used, which may affect model performance. The FedPer method was specifically designed for this purpose, to introduce personalized updates and mitigate the impact of heterogeneity.

The primary difference between FedAvg and FedPer lies in their approach to model updates. FedAvg updates by averaging the weights, assuming a uniform data distribution. However, FedPer introduces a personalization factor concerning the data distribution of individual devices. This makes FedPer more effective in handling non-IID (Independent and Identically Distributed) data. The FedAvg model may face challenges related to high variance in model performance due to non-IID data distribution. Using FedPer, which assigns more

weight to clients that provide more valuable updates, alleviates this problem and results in more consistent performance. Additionally, traditional FL methods may not adequately meet clients' specific needs and characteristics, resulting in suboptimal outcomes. Using FedPer, it is possible to provide a more personalized model for each client, thus improving overall performance.

Apart from the challenges mentioned above, the accuracy of recognizing human activities is another fundamental issue in this field. HAR often includes time series data that requires models to capture temporal dependencies. Recently, Transformer models have garnered significant attention for this purpose. These models can capture long-range dependencies with the self-attention mechanism and model temporal relationships in the data. Many studies have been conducted on this matter, combining the Transformer model with other methods or attempting to address existing challenges by modifying the standard Transformer model [9–11]. Traditional Transformer models utilize positional encoding at the input to track the sequence's order. Applying positional encoding to the input and capturing the required information for the model can be challenging, especially for longer sequences.

The model presented in this article can better manage this issue by applying positional encoding to the attention matrix at each Transformer attention head. The attention mechanism in the Transformer enables the model to focus on the most important parts of the input data. By applying positional encoding to the attention matrix, the model can comprehend the order and position of the data within the sequence, thereby enhancing its ability to understand the context and relationships within the sequences.

Overall, the primary research questions this study aims to address are as follows:

- How does a Transformer model, by applying positional encoding on the attention matrix, improve the performance of HAR?
- How can Federated Learning be effectively leveraged to enhance data privacy in HAR?
- How can a unified framework be developed to effectively balance data privacy concerns, communication efficiency, and data heterogeneity of a HAR system?
- Which parameters most significantly influence prediction accuracy in a HAR application?

To comprehensively address the aforementioned research questions, a Federated Learning framework called Multi-Head Attention Transformer using Federated Learning (MHAT-FL) has been introduced. This framework uses a Transformer with positional encoding on the attention matrix in a Federated Learning environment. MHAT-FL enhances the model's performance on both the client and server sides by improving the accuracy of activity classification, ensuring data privacy, collecting heterogeneous data, and aggregating model parameters. In particular, this method increases the model's efficiency in identifying human activities by applying positional encoding on the attention matrix in each attention head. By applying positional encoding to each attention head, the model captures sequential dependencies more effectively, enhancing pattern recognition and accuracy in activity classification. This approach enriches input sequence representation, optimizing the model's performance within the federated learning framework.

In developing the MHAT-FL framework, this research explores how Federated Learning techniques can be effectively integrated into HAR to enhance data privacy and communication efficiency. To achieve this, two Federated Learning strategies, FedAvg and FedPer,

were employed to enable collaborative model training without transferring sensitive user data to a central server, thereby ensuring that data remains localized on users' devices. These techniques share model updates rather than raw data, significantly reducing communication overhead and associated costs. Recognizing the challenge of data heterogeneity across users, the FedPer method was utilized to support a personalized model that adapts to local data distributions, thereby making the model more generalizable across diverse users' activity patterns.

The specific objectives of this research are threefold. First, to investigate how Transformer-based architectures, particularly those incorporating attention mechanisms and multi-head modules, can be leveraged to enhance precision and reduce detection latency in HAR. Second, to explore how Federated Learning can be effectively leveraged to improve data privacy. Third, to conduct a comprehensive and systematic evaluation of the proposed approach across multiple HAR datasets, benchmarking its performance against existing state-of-the-art methods.

Briefly, research contributions are as follows:

- • Developing a novel Federated Learning using FedAvg and FedPer to address data privacy and heterogeneity in Human Activity Recognition. While FL has been widely used for privacy-preserving training, its application in HAR with both FedAvg and FedPer remains underexplored. Our study provides insights into how these aggregation techniques affect performance in HAR tasks under real-world data distributions.
- • Proposing a Transformer model based on a novel positional encoding strategy that assigns different positional encodings to each attention head, enabling the model to capture fine-grained temporal features more effectively, and contributing to the theory and design of Transformer architectures for time-series data.
- • Introducing an integrated client-server Transformer training framework under federated settings, which constitutes a new learning paradigm for HAR by bridging the gap between centralized high-performance models and distributed privacy-aware training.
- • Conducting comprehensive empirical validation across diverse HAR datasets (PAMAP2, MotionSense, and Opportunity) to evaluate the performance of MHAT-FL. Compared to state-of-the-art baselines, MHAT-FL consistently shows superior performance.

The remainder of the paper is organized as follows: Sect. 2 presents an overview of related work. Section 3 discusses the details of the MHAT-FL. Section 4 presents the evaluation results of MHAT-FL and compares them with several baselines. Finally, Sect. 5 presents conclusions and future research directions.

2 Related works

This section reviews previous works in the field of HAR, focusing on two primary categories: Transformer-based deep learning methods and Federated Learning approaches. To ensure this review reflects the latest advancements and current state-of-the-art techniques, we primarily considered articles published after 2020. These categories were selected because the core focus of our approach lies in the Transformer-based deep learning methods

within a Federated Learning framework, which enables a thorough investigation of the various dimensions and perspectives of the research questions presented in the previous chapter. Moreover, we aimed to include studies that represent state-of-the-art techniques addressing key challenges in HAR, particularly concerning data privacy, accuracy, and the need for personalized performance across diverse user scenarios.

2.1 Transformer-Based deep learning

The study selection criteria for this section emphasize novel feature extraction techniques, demonstrated model performance, and potential applicability to current challenges in HAR, including data heterogeneity, computational efficiency, and real-world deployment. Since Transformers are known for their capability to model long-range dependencies and capture spatial and temporal patterns in sequential data, we focused on studies that integrated Transformer architectures with other deep learning techniques to enhance performance.

Various hybrid models have been proposed to address the inherent complexities of HAR. Pramanik et al. focused on using a Transformer-based reverse attention mechanism in human activity recognition [9]. Moreover, a 3-block CNN architecture has been proposed, where each block consists of 2 levels of convolutions. This approach aims to enhance feature extraction and representation by employing a top-down feature fusion method. Zhang et al. presented ConvTransformer, a deep learning model for human activity recognition using wearable sensors. This model combines Convolution, Transformer, and attention mechanisms to extract and highlight important features before classification [10]. Xiao et al. presented the CapMatch model, which utilizes a Transformer network. This model identifies local and global patterns, augments data, and uses knowledge distillation to transfer knowledge between models [11]. Foumani et al. introduced the ConvTran model, which utilizes convolution to reduce sequence length and employs tAPE and eRPE encodings prior to the Transformer. This model introduces two new Positional Encoding methods: tAPE for absolute position and eRPE for relative position, enabling the Transformer model to better handle long-time series [12]. Pareek et al. presented a new model for recognizing human activities that combines a 3D CNN with an attention layer. This model enhances activity recognition by utilizing spatial and temporal features and employs a 3D CNN Transformer to capture input features [13]. Gu et al. presented the RMPCT-Net model for radar-based human activity recognition, which uses a multi-channel parallel CNN and Transformer model. With two inputs and four channels, this model extracts features and transfers them to the classification layer to classify activities [14]. Lee et al. presented a system that uses a filtering network and LSTM to process unfiltered data in real-time. This system utilizes a filtering network to learn representations of sensor data related to activity transitions [15]. Yang et al. presented an unsupervised continuous authentication system called CALL based on mobile device sensor data. This system employs a one-dimensional autoencoder and a shuffle low-rank Transformer (SLRT) to extract spatial and temporal characteristics from sensor time series data, utilizing reconstruction error and similarity with the original data for authentication [16]. Kim et al. presented a new conformer-based model for recognizing human activities using inertial sensors. This model extracts temporal dependencies from time series sensor data using two convolutional layers and self-attention. It provides good performance and increases computational efficiency using only two blocks. Its hybrid neural

network structure enables robust sequential modeling with improved computational performance compared to other architectures [17].

2.2 FL-Based deep learning

In the context of Federated Learning (FL), we selected studies that demonstrate the integration of FL with various deep learning architectures, focusing on their ability to address data privacy concerns, maintain robust model performance, and meet the practical challenges of real-world deployment in HAR. FL is particularly valuable in HAR scenarios, where sensitive user data is often collected from personal devices such as smartphones and wearables, and must remain decentralized for privacy preservation. The selected studies were evaluated based on their contributions to addressing current issues in HAR, including data scarcity, limited labeled data, non-IID data, and device heterogeneity.

Recent research highlights how FL can be synergistically combined with deep learning methods to ensure privacy while tackling these issues. Yu et al. introduced the FedHAR framework, which is based on semi-supervised online learning. The authors presented an algorithm for unsupervised gradient computation and an unsupervised gradient aggregation strategy that solves the concept instability and convergence problems [6]. Presotto et al. proposed FedAR, a method for activity recognition in mobile devices. By presenting a Federated Learning framework, they solved the data scarcity problem and overcame the limitations of labeled data [7]. Gao et al. integrated Federated Learning with semi-supervised learning for activity recognition. They tackled the challenge of unlabeled clients by training autoencoders in an unsupervised manner on client data and employing supervised learning on the server [18]. Pham et al. developed a method to recognize human activities using wearable sensors based on 3D CNNs. By utilizing Federated Learning, they preserved privacy in sensor data while 3D CNNs captured local spatial and temporal correlations [19]. Wang et al. introduced a hybrid unified learning framework called Hydra for human activity recognition. This framework employs a hybrid model using BranchyNet and Federated training methods to reduce data heterogeneity, delivering superior performance compared to other baselines [20]. Bu et al. presented an attention-based approach to federated human activity recognition considering non-IID data distribution. This method identifies pair-by-pair cooperation between similar devices using a learned similarity matrix, resulting in each client receiving an updated personal model that is a weighted combination of models from similar devices [21]. Li et al. introduced REWAFL, which aims to enhance the learning process by considering device battery energy levels and wireless transmission rates. This algorithm optimizes local iterations based on transmission rates by selecting participants using a new utility function and providing an adaptive local calculation policy [22]. Chai et al. developed a joint personalization method based on profile similarity to recognize human activities using wearable sensors. By calculating similarity based on gender, age, height, and weight profiles, this method offers higher accuracy in recognizing activities from sensor data [23]. Park et al. introduced a new method called AttFL for Federated Learning, which is suitable for processing mobile sensor data. AttFL reduces the computational burden by using attention modules to extract data features [24].

The studies on human activity recognition employing Transformer/Federated Learning models are summarized in Table 1, which includes the advantages and disadvantages of each work, as well as the datasets used for evaluation purposes.

Table 1 Summary of previous works

Article	Advantages and Disadvantages	DataSets
Yu et al. [6], 2021	+FedHAR's personalized federated framework enables personalized learning and collaboration between multiple users. It overcomes conceptual instability and convergence issues. -Limited scope of application due to specific design, reliance on specific data/task characteristics, and evaluation on small datasets.	RealWorld, HAR-UCI
Presotto et al. [7], 2022	+Considering privacy concerns by using federated learning. -Limited positions of mobile devices and does not provide a detailed comparison of recognition performance, considering different types of activities or user profiles.	WISDM, MobiAct
Pramaniko et al. [9], 2023	+Assigning higher weights to distinct features using a deep inverse Transformer. -Amplification of certain areas of the input characteristics by deep inverting Transformers.	MHEALTH, USC-HAD, WHARF, UTD-MHAD1, UTD-MHAD2
Zhang et al. [10], 2023	+Utilizing the strengths of CNN and Transformer to extract local and global features. The attention mechanism helps highlight important features for classification. -Increased training complexity due to the combination of several models.	OPPORTUNITY, PAMAP2, SKODA, USC-HAD
Xiao et al. [11], 2023	+Combining different learning techniques such as supervised, unsupervised, contrastive learning, and knowledge distillation to improve performance. Requires minimal labeled data for training. -Data reinforcement and complex training methods are needed. Limited insight is available into learned representations and how different techniques contribute to overall performance.	HAPT, WISDM, UCI HAR
Foumani et al. [12], 2024	+Proposing new positional encoding methods (tAPE and eRPE) specifically designed to classify time series data with Transformers. Integrates positional Encoding into the new ConvTran model, achieving advanced results on benchmark datasets. - Limited comparison to non-deep learning methods. Lack of theoretical analysis.	UEA, Ford Challenge, Actitracker
Pareek et al. [13], 2024	+The attention layer allows for better modeling of temporal dependencies. This approach addresses CNN limitations by considering only local areas and traditional temporal modeling techniques. -Training 3D CNNs on data requires significant computing resources. The attention mechanism imposes additional parameters, increasing the model's complexity.	Weizmann, UCF101
Gu et al. [14], 2023	+Combining CNN and Transformer networks effectively to extract spatial and temporal features from the input. It uses both time-range and micro-Doppler maps to obtain richer activity information. -Poor classification performance for similar activities, such as lifting and drinking. The model was evaluated on a limited dataset of six activities in controlled indoor/outdoor environments.	University of Grasse Human Activity Dataset
Lee et al. [15], 2023	+Presenting an efficient method for handling unfiltered sensor data for real-time tasks. The attention mechanism in the filtering network helps distinguish refined and unrefined sensor states. -The fixed size of the input window may not be optimal, potentially delaying predictions. The performance comparison is not fully comprehensive.	Milan, Kyoto8, Kyoto11

Table 1 (continued)

Article	Advantages and Disadvantages	DataSets
Yang et al. [16], 2024	+Providing an unsupervised method for continuous authentication. A low-rank Transformer model is used that is suitable for mobile device deployment. -Evaluates only Accuracy and EER without examining other metrics like FRR/FAR.	UCI HAR, WISDM HARB
Kim et al. [17], 2022`	+Includes data from various sensor modalities such as Accelerometer, Gyroscope, etc. A large amount of data was collected from multiple participants. -Imbalance of the WISDM dataset. Some hyperparameters, such as the number of blocks, headers, etc., were tuned, but there was no systematic hyperparameter analysis.	WISDM, UCI-HAR, PAMAP2
Gao et al. [18], 2023	+Personalization and efficient use of the network. -Imbalance of data distribution.	UCI-HAR, PAMAP2
Pham et al. [19], 2024	+Effective 3D CNN modeling and privacy-preserving through unified learning and encryption. -Slight accuracy loss due to encryption and increased complexity compared to non-private techniques.	Daily and Sport Activities (Sport), Daily Life Activities (DaLiAc)
Wang et al. [20], 2024	+User clustering reduces the impact of heterogeneous data distributions. It accommodates devices' heterogeneous computing capabilities through its hybrid model design. -A more detailed division of device types increases complexity.	HHAR, MobiAct, HARBox
Bu et al. [21], 2024	+Proposing a new pairwise federated framework that can handle heterogeneous and non-IID data distributions commonly seen in mobile HAR scenarios. Reduces communication overhead between clients and servers. -Constant online presence of devices for model updates may not be feasible in real-world scenarios.	UCI-HAR, WISDM, UniMiB-SHAR, HARBOX
Li et al. [22], 2024	+Considering the remaining battery energy level and correcting the heterogeneous wireless transmission rate. Introduces a utility function that jointly optimizes statistical, global latency, and local energy utilities to improve training efficiency. -Performance depends on accurate latency and power consumption estimation, which may be challenging on diverse real devices. Hypersensitivity to parameters like α and β, which are scaling coefficients to balance different metrics regarding the model accuracy, latency, and energy efficiency.	MNIST, Shakespeare, HAR
Chai et al. [23], 2024	+Proposing an effective solution to the challenge of data heterogeneity in federated learning. Personalizes global models for each client based on similarity, improving generalization. Robust to changes in client selection ratio and dataset size. -It fails to analyze sensitivity to changes in computing resources, and real-world impact needs to be demonstrated through applications and user participation.	RealWorld, SisFall
Park et al. [24], 2023	+Various applications (ECG, activity, sound) to evaluate the generalizability of the proposed method. Uses real-world data from mobile phone sensors for practical applications. -Data distribution across devices/subjects may not fully represent real-world conditions.	MIT-BIH electrocardiogram (ECG), HAR-UCI, RAVDESS

As evident from the above table, previous methods in HAR, particularly those incorporating Transformer models and Federated Learning frameworks, face numerous recurring challenges. These include handling non-IID data distributions among clients, ensuring high model accuracy when there is limited or unlabeled data, mitigating resource constraints on edge devices, and protecting user privacy without compromising performance. In addition, scalability, personalization, and adapting to changes in real-world sensor data remain significant challenges in the literature. Our proposed approach addresses several of these critical challenges consistently highlighted in the reviewed studies. The selection of these challenges is motivated by their prevalence in existing works and their strategic importance in driving the next generation of HAR systems. The primary objective of our work is to enhance HAR models by developing models that deliver high accuracy and robustness while being easily scalable and adaptable to various deployment contexts and user scenarios, while also protecting privacy.

In summary, the significant challenges inherent to any HAR system, also evidenced by the studies reviewed, are outlined as follows:

- Scalability: Machine learning models struggle with large-scale data significantly as the number of clients and data volume increase. Federated learning addresses this by training the HAR model on the client side and aggregating the weights on the server side, distributing the computing load, and providing scalability. The FL decentralized approach of MHAT-FL in both FedAvg and FedPer enables parallel processing on both the client and server sides, reducing training time.
- Data Privacy: Data privacy is a critical concern in HAR. Client data often contains sensitive and private information that must be protected during training. Using Federated Learning in MHAT-FL helps maintain data privacy by transferring only model updates, not raw data.
- Customization of the Model: A centralized model of HAR may not meet the specific needs of each client. Common methods in HAR lack personalization capabilities and follow a one-size-fits-all approach. Federated Learning of MHAT-FL enables the customization of models to meet client-specific needs. Each client can have a personalized and optimized model by adapting the global model based on their data.
- Recognition Accuracy: Standard positional encoding of Transformer models does not account for the relative positions of data elements in a sequence, which can be crucial in specific contexts. Using positional encoding on the attention matrix, rather than the input, enables the model to understand the context better and capture the relative positions of the data. To this end, the Transformer model of MHAT-FL develops a novel positional encoding approach on each attention head to accurately recognize human activities and improve the training time.

3 Proposed approach: MHAT-FL

This research introduces the Multi-Head Attention Transformer with Federated Learning (MHAT-FL) framework, a novel approach that synergistically integrates a Transformer model with advanced Federated Learning techniques. This innovative framework is specifically designed to enhance the accuracy of human HAR while simultaneously ensuring

robust data privacy for users. The development of MHAT-FL is grounded in established scientific principles and draws extensively upon the latest advancements in both Transformer architecture and Federated Learning methodologies. The central advance of MHAT-FL lies in its novel approach of directly adding positional encoding to the attention matrix, rather than focusing solely on input token positions. The model benefits from this combination, achieving enhanced accuracy in detecting temporal patterns within sensor datasets. The precision of Human HAR systems depends on how well they recognize when activities occur during sequences because this timing affects their performance. By integrating positional data into the attention score calculation, MHAT-FL enables the model to dynamically adjust the weights assigned to different locations within the input sequence. The model can recognize minor changes in activity patterns through its enhanced flexibility in understanding data contexts more effectively. This framework substantially improves classification accuracy, enabling it to differentiate complex actions that share similar attributes yet show differences in time frames and event sequences.

Moreover, the integration of Federated Learning techniques within the MHAT-FL framework serves a dual purpose: it preserves user privacy by ensuring that sensitive data remains localized on client devices while leveraging the diverse datasets available across multiple clients for model training. This collaborative yet privacy-conscious approach allows the framework to learn from a broader range of user contexts and preferences, further enhancing its generalizability and effectiveness.

The architecture of MHAT-FL is illustrated in Fig. 1. It is worth mentioning that MHAT-FL refers to a specific architectural design proposed in this study, rather than a general framework with multiple possible instantiations. It represents a unified and concrete architecture that integrates Transformer-based multi-head attention mechanisms within a Federated Learning setting, explicitly tailored for HAR.

It begins with the data preprocessing phase, which is critical for ensuring that the raw sensor data is transformed into a suitable format for model training. Then, a Transformer model is developed on both the client and server sides. This duplicate setup enables decentralized learning, where local models are trained on client devices using datasets from individual users. The server collects these local model updates, thus enabling a collaborative learning process without compromising data privacy. Moreover, the MHAT-FL framework establishes a groundbreaking approach by integrating positional encoding within the attention matrix to enhance the model's capability to understand temporal patterns and contextual connections in sensor data, leading to precise activity recognition. Employing Federated Learning strategies, such as FedAvg and FedPer, enables the model to gain knowledge from decentralized data sources without compromising users' privacy. MHAT-FL achieves this by combining model updates rather than data, thereby not only complying with privacy regulations but also creating a trustworthy environment for users.

3.1 Preprocessing

The section describes the strategies used for data preprocessing and feature extraction, which are crucial stages in preparing data for the implementation of different learning algorithms. Data preprocessing plays a key role in improving the quality of the dataset by reducing noise and making the data more suitable for analysis. This step encompasses several key operations, including denoising and segmentation, feature extraction, and feature selection.

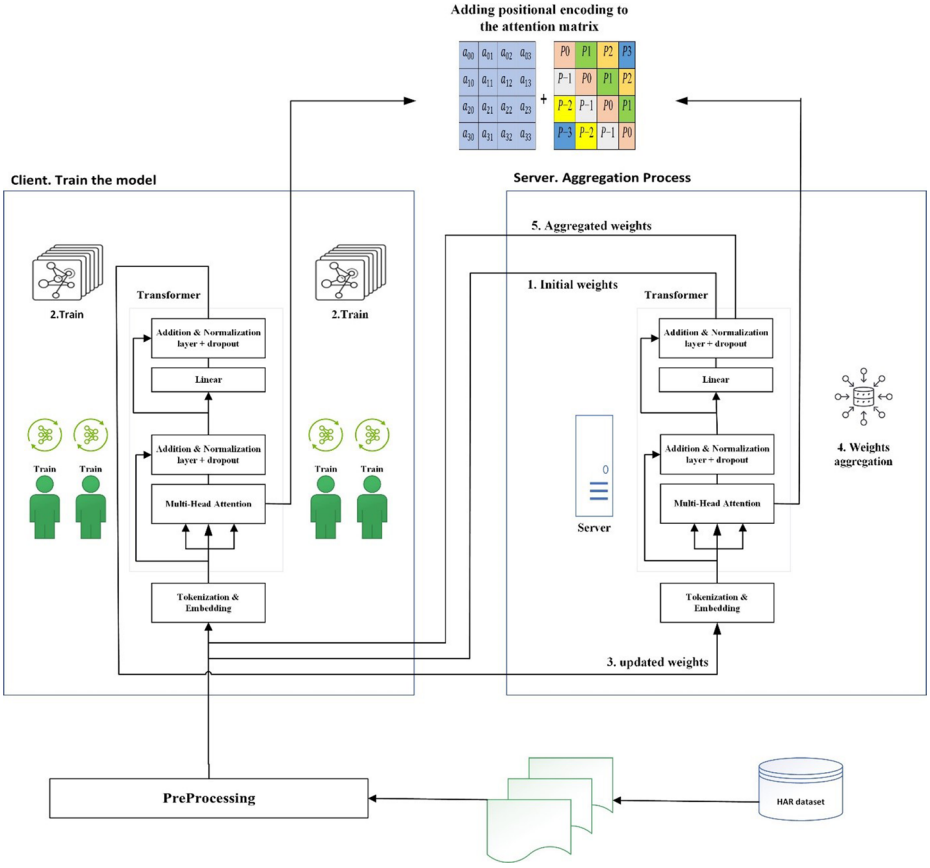

Fig. 1 MHAT-FL architecture

This research utilizes the Scikit-learn library and the LabelBinarizer to one-hot encode labels. Following established protocols in the literature, the dataset is split, allocating 80% for training and the remaining 20% for testing. The data is segmented into varying window sizes tailored to each specific dataset. For the PAMAP2, MotionSense, and Opportunity datasets, the window sizes are (32, 36), (72, 1), and (300, 108), respectively. These selections ensure an optimal balance between capturing temporal information, focusing on motion patterns, and adequately representing long-term dependencies and complex activities.

Algorithm 1 presents the pseudocode for the preprocessing steps and outlines the systematic approach taken in this research.

Algorithm 1: Data preprocessing

```
Input  Human_Activity_Data (dataset);
Output Preprocessed Data;

data = Load dataset;
labels = Load labels;
label_binarizer = LabelBinarizer();
labels_one_hot = label_binarizer_fit_transform(labels);
train_data, test_data, train_labels, test_labels = split_train_test (data, labels_one_hot, test_size=0.2);
input_window_size (PAMAP2)-(32,36); (MotionSense)-(72,1); (Opportunity)-(300,108)
```

3.2 MHAT-FL transformer architecture

The architecture of the MHAT-FL framework is underpinned by a Transformer model that incorporates several key components: multi-head attention mechanisms, feedforward networks, normalization layers, and dropout layers. Central to this architecture are the multi-headed attention layers and feedforward networks, which are interconnected through residual connections. Residual connections allow gradients to propagate through the deep neural network, improving training efficiency.

As depicted in Fig. 2, the initial layer of the architecture is a dense (fully connected) layer. This layer is responsible for learning complex features by applying a linear transformation to the input data and introducing a nonlinear activation function. Specifically, the Rectified Linear Unit (ReLU) activation function introduces nonlinearity to the model, enabling it to uncover complex patterns in the data. Following the dense layer, the model employs a multi-head attention layer. This part of the architecture is crucial as it enables the model to interactively select various sections of the input sequence

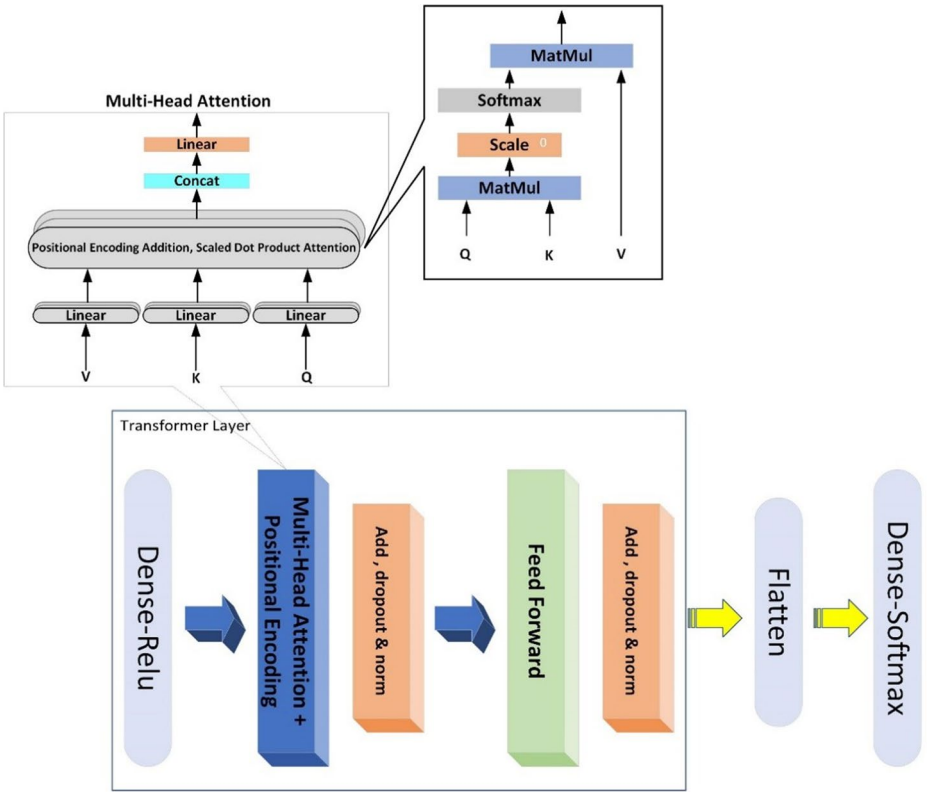

Fig. 2 Transformer architecture in MHAT-FL

while making predictions. Such a focus is necessary for recognizing the relationships in sequential data. The multi-headed attention principle operates by conducting several parallel attention operations, each generating distinct subsets of the representation space, which enables the model to maintain its comprehension of the input. In this architecture, positional encodings are integrated into the attention matrix at each attention head. This integration preserves the positional information of tokens within the input sequence, thereby enhancing the model's contextual awareness.

After the multi-head attention layer, dropout, and normalization layers are applied, the output of the attention mechanism is combined with its input via the residual connection, which enables the training of deeper networks by eliminating the vanishing gradient problem. Normalization is subsequently applied across features to stabilize and accelerate the training process. A Feed Forward Neural Network (FFN) is utilized following the attention and normalization stages. The FFN comprises two linear transformations with a ReLU activation in between. This component processes each position independently, allowing the model to treat tokens differently. The output from the feedforward layer is further subjected to dropout and normalization, just like the earlier stages. The architecture concludes with a flattening layer that converts the multidimensional output from the preceding layer into a one-dimensional array, preparing it for the final output layer. This final layer is another dense layer, which employs a softmax activation function. The softmax function transforms the raw output scores into probabilities that sum to one, facilitating the interpretation of model outputs as class probabilities. This structured approach ensures that the MHAT-FL framework effectively captures the complexities of human activity recognition while leveraging the strengths of the Transformer architecture.

3.2.1 Client-Side transformer

The client-side Transformer model operates by first receiving a raw input tensor from the "Encoder" class. This tensor undergoes a flattening process, transforming its multidimensional structure into a one-dimensional format suitable for subsequent processing. Following this, a dense layer is applied, utilizing a softmax activation function to generate the output tensor. The softmax function is essential in this context, as it converts the raw output scores into probabilities, which can be interpreted as the likelihood of each class. In this context, we consider clients to be data shards, which are used to split the training dataset into ten separate data shards. Each shard is passed to each client for processing in parallel, and the model can be trained locally on each client. This enables the models to learn from multiple datasets while also ensuring the privacy of user data. Additionally, all processing of client data is managed through the TensorFlow framework, which determines where and how the data is allocated and manipulated throughout the training process. The structured approach employed in the client-side Transformer model ensures that each client's unique dataset contributes to the framework's overall learning objectives.

Algorithm 2 presents the pseudocode outlining the operations of the client-side Transformer.

Algorithm 2: Client-side transformer

```
Input  Human Activity Data
Output Updated Weights
client_side_Transformer_federated  _learning :
  foreach client in clients:
      local_model = Transformer
      local_model.load_weights_from_the_global_model
      for epoch in range(num_epochs):
        for batch in data:
          loss = local_model.train_on_batch(batch)
        endfor
      endfor
      updated_weights = local_model.get_the_updated_local_model_weights
      send_to_server(updated_weights)
  endfor
```

3.2.2 Server-Side transformer

The Transformer model on the server side employs an "Encoder" instance with specific parameters. The input tensor passes through this instance, and the resulting tensor is then flattened and converted from a three-dimensional tensor to a two-dimensional tensor. Finally, an output tensor is generated using a dense layer and a softmax activation function.

The server-side Transformer model plays several vital roles in this process:

1. Aggregation of weights: The server receives and aggregates the updated weights from the clients. This step typically involves calculating a weighted average of client-side weights, which can be based on factors such as the number of training samples or device capabilities. The server-side Transformer model is responsible for performing this aggregation operation.
2. Updating the global model: After the aggregation stage, the server updates its global model by integrating the aggregated weights. The server-side Transformer ensures that the global model is updated correctly.
3. Redistribution: The new weights are sent to the client devices once the global model is updated. The server model facilitates the distribution of the updated global model to the client devices, enabling them to commence the next local training session.

The pseudocode demonstrating the server-side Transformer in Algorithm 3 shows the organized and structured method used to aggregate weights in the model.

Algorithm 3: Server-side transformer

```
Input  Trained weights from clients
Output Aggregated weights
server_side_model_aggregation(server_model, client_weights):
  aggregated_weights = []
  foreach client_weight in client_weights:
    aggregated_weights.append (client_weight)
  endfor
  updated_weights = server_model_aggregation (aggregated_weights)
  server_model.update_weights (updated_weights)
  send_to_clients (server_model)
```

3.2.3 Attention mechanism and positional encoding

To enable the model to attend to specific context relationships within the given input sequence, the attention mechanism of the Transformer is utilized. This mechanism computes attention scores, which quantify the contribution of some input segments to the output. In the Transformer framework, these scores are derived from the interaction of queries, keys, and values. Specifically, the attention mechanism computes a dot product between the query vector and all key vectors, applying a softmax function to yield normalized attention weights. The final output is formulated as a weighted sum of the corresponding value vectors. When analyzing sequential data, the order of elements is crucial. The standard attention module in the Transformer does not account for this and treats all positions equally, which can lead to misunderstandings. We utilize positional encoding as a key component of the Transformer setup to address this issue. Positional encoding provides essential information about the location of each token in the input sequence, enabling the model to understand the data's order and structure.

In MHAT-FL, a specific form of fixed positional encoding is employed that utilizes sine and cosine functions across various frequencies. By integrating this positional encoding into the attention matrix, we embed positional information directly within the computation of attention scores. This integration is efficient in situations like HAR, where the order of sensor data is crucial in determining what people are doing. For example, the sequences "Walk-Then-Run" and "Run-Then-Walk" demonstrate different orders of action, resulting in distinct understandings of what is happening. Positional encoding is consistently applied across all attention heads to ensure that the model appropriately accounts for this sequential order in the computation of attention scores.

The attention mechanism is defined using Eq. (1), where Attention(Q, K, V) computes the softmax operation to derive the attention weights. Here, Q, K, and V represent the inputs' Query, Key, and Value projections:

$$\text{Attention}\left(Q, K, V\right) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

Then, using the output of the attention mechanism from the above equation, a new latent state $head$ is created, as shown in Eq. (2):

$$\text{head}_i = \text{Attention}\left(QW_i^Q,\ KW_i^K,\ VW_i^V\right) \tag{2}$$

In the above equation, the terms QW, KW, and VW project the queries, keys, and values. The multi-headed attention combines the concatenated hidden states $head$ from Eq. (2) across all attention heads to produce the final output, calculated using Eq. (3):

$$\text{MultiHead}\left(Q, K, V\right) = \text{Concat}\left(\text{head}_1, \ldots, \text{head}_n\right)W^O \tag{3}$$

The Feedforward Network (FFN) is described using two linear transformations with ReLU activation in between, as shown in Eq. (4):

$$\mathrm{FFN}(x) = \max\left(0,\ xW_1 + b_1\right) W_2 + b_2 \tag{4}$$

In the above equation, W_1, b_1, and W_2, b_2 denote the weights and biases for the two linear layers. Figure 3 shows an overview of the attention mechanism process.

As previously mentioned, in MHAT-FL, positional encoding was added to the attention matrix in each attention head. The positional encoding matrix has dimensions of (sequence_length, d_model), where sequence_length represents the length of the input sequence, and d_model denotes the dimensions of the model's hidden states. The positional encoding matrix can be computed as follows:

$$PE(pos, 2i) = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{\mathrm{model}}}}}\right) \tag{5}$$

$$PE(pos, 2i+1) = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{\mathrm{model}}}}}\right) \tag{6}$$

Where pos indicates the position of the token in the sequence, i indicates the dimension index in the hidden states, and d_model is the dimensions of the hidden states of the model. Two element-wise matrices are added to the attention matrix token to integrate the positional encoding into the attention mechanism. This integration occurs before the self-attention operation in each attention head.

If we consider the attention matrix token as X, and the positional encoding matrix as PE, the equation for adding the positional encoding at each attention head can be shown as:

$$X' = X + PE \tag{7}$$

Where $X\prime$ is the new attention matrix token with position information added. Equation (7) allows the Transformer model to implicitly encode the positional information of the tokens in the self-attention mechanism, helping the model understand the order of the input sequence.

Algorithm 4 shows the pseudocode of the Transformer model of MHAT-FL with positional encoding.

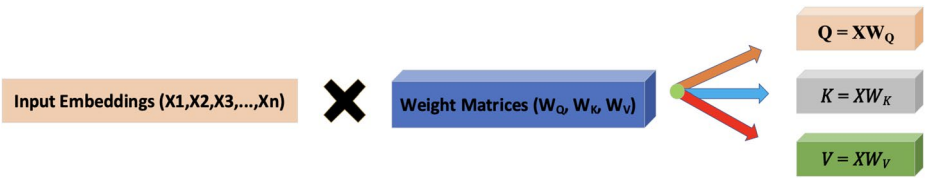

Fig. 3 Overview of attention mechanism

Algorithm 4: Transformer model with positional encoding applied to the attention matrix

```
Input Human Activity Data
Output Trained Model Weights
EncoderLayer :
    num_of_layers = 1
    server_side_model_dimension = 64
    client_side_model_dimension = 128
    num_heads = 4
    feed_forward_dimention=1024
    Dropout_rate=0.1
    foreach input_data do
        apply data embedding
        Multi-head attention :
            foreach head do
                Scaled dot product attention :
                    foreach query, key, and value, do
                                calculate dot product between query and key.transpose()
                        apply softmax to obtain attention scores
                                calculate the weighted sum of value based on attention scores
                        Positional Encoding:
                            foreach position and dimension do
                                    calculate position * angle rates, then
                                    calculate encoded scores
                                    apply positional Encoding to attention scores
                                return output
                            endfor
                    endfor
            endfor
        apply layer normalization
        apply Dropout
        Feedforward:
            activation = relu
            foreach inputs do
                apply a dense layer with  hidden dimension
                apply a dense layer with embedding dimension
            endfor
        return output
        apply layer Normalization
        apply Dropout
    endfor
    return weights
```

3.3 MHAT-FL federated learning

Federated learning represents a decentralized machine learning paradigm that enables collaborative model training while preserving the confidentiality of raw data. This approach is particularly advantageous in Human Activity Recognition (HAR), as it offers significant privacy protections and enhances scalability within the federated learning framework; several methodologies exist, including Federated Averaging (FedAvg) and Federated Personalized Learning (FedPer). The FedPer methodology is characterized by its personalization capabilities, allowing each client to tailor the model to its specific local dataset. In this framework, clients independently compute updates based on their local data and contribute personalized gradients to the global model. This individualized approach fosters improved model performance across diverse client environments. In contrast, the FedAvg methodology aggregates model updates from multiple clients to construct a unified global model. In this process, the central server computes the average of the weights derived from local models, generating new aggregated weights and an updated global model.

The integration of federated learning with the Transformer architecture in the MHAT-FL capitalizes on the strengths of both methodologies. This hybrid approach enables practical federated training on sensitive or distributed datasets while achieving high-performance outcomes.

Algorithm 5 demonstrates the pseudocode of the Transformer model and Federated Learning in MHAT-FL.

Algorithm 5: Transformer model with Federated Learning in MHAT-FL

```
Input Human Activity Data
Output Model and test accuracy
Initialize client models and server model
Import Transformer_model
train_federated_learning(datasets, client_learning_rate, server_learning_rate):
        function federated_average (models) :
            average_weights = empty_weights ()
          for model in models, do
              weights = get_weights (model)
              average_weights + = weights
          endfor
            average_weights /= length(models)
            return average_weights
        function client_update(data, client_model):
            Load data to the client
            Run Transformer_model
            Update the client_model's parameters
        function server_aggregation (client_models):
             aggregated_weights = federated  _average(client_models)
             update_weights(server_model, aggregated_weights)
            for dataset in datasets, do
                client_model = initialize_Transformer_model()
                server_model = initialize_Transformer_model()
              for epoch in range(num_epochs) do
                    for data in the dataset, do
                        client_update(data, client_model)
                    endfor
                     server_aggregation(client_models)
                     update_server_weights(server_model, server_learning_rate)
              endfor
                test_model_on_dataset(server_model, dataset)
            endfor
datasets = [Pamap2, Opportunity, MotionSense]
client_learning_rate = [0.01, 0.001]
server_learning_rate = [0.01, 0.001]
```

3.3.1 MHAT-FL federated learning: FedAvg approach

Federated Averaging (FedAvg) is a well-known algorithm in federated learning. It enables multiple clients or devices to collaborate on training a global model while maintaining the privacy and security of their local data. The primary objective of FedAvg is to minimize the discrepancies between local and global model parameters, thereby enhancing the overall accuracy of the model. This technique is particularly effective in domains such as HAR, where the temporal order of sensor data points is critical for accurately identifying and clas-

sifying activities. For example, Walk-then-Run and Run-then-Walk show different action orders, which can lead to varied interpretations.

The FedAvg algorithm includes the following steps:

1. Initialization: The global model parameters are denoted by W_0 are initialized. Then, the number of communication rounds is set by T, and the learning rate is denoted by η. The fraction of selected clients in each round is indicated by C.
2. Communication round: First, A fraction of clients C is randomly selected from the existing clients. Then, for each selected client, the parameters of the global model W_t are sent to the client. Client *i* trains its local model using its local dataset, denoted by D_i. Then, the parameters of the local model after training are calculated, denoted by W_{it}. The client's updated weights are computed as $\delta i = (|Di|)/(\sum_j |D_j|)$, where $|Di|$ is the dataset size of client *i*, and the denominator represents the sum of dataset sizes across all selected clients. Finally, the updated parameters $(\delta i * Wi_t)$ are sent to the server.
3. Aggregation of weights: The server receives updated parameters from all selected clients. Then, the weighted average of the received parameters is calculated to obtain the parameters of the new global model for the next round, indicated by W_{t+1} according to Eq. (8):

$$W_{t+1} = \frac{\sum_i (\delta_i \cdot W_{it})}{\sum_i (c_i)} \tag{8}$$

4. Steps 2–3 are repeated for T communication rounds.

FedAvg pseudocode is presented in Algorithm 6. In the below algorithm, the local mini-batch size is Z, the number of local epochs is F, and the learning rate is η. These factors determine the index of the ***D*** clients.

Algorithm 6: FedAvg in MHAT-FLw

Input Human Activity Data
Output Averaged Weights
Server Starts:
    initialize w_0
    foreach round l= 1, 2, 3,... **do**
        k ← maximum (C · D, 1)
        S_l ← (random set of *k* clients)
        foreach client D ∈ S_l in parallel **do**
            $w_{l+1}^D \leftarrow ClientUpdate(D, w_l)$
        endfor
        $w_{l+1} \leftarrow \sum_{D=1}^{D} \frac{n_D}{n} w_{l+1}^D$
    endfor
Client Update (D, w):
    Z ← (split ρ_Z into batches of size Z)
    foreach local epoch i from 1 to F, **do**
        for batch z∈ Z, **do**
            $w \leftarrow w - \eta \nabla L(w; z)$
        endfor
    endfor
return w to the server

3.3.2 MHAT-FL federated learning: FedPer approach

The FedPer approach distinguishes itself from traditional FedAvg by facilitating the collection of personalized models from multiple edge devices or clients while ensuring the confidentiality of individual user data. This method prioritizes user privacy and data security by allowing clients to retain their local datasets while still contributing to the global model.

The FedPer mathematical relation can be represented as Eq. (9):

$$\theta_{i(t+1)} = \theta_i(t) - \eta \nabla L_i(\theta_i(t), D_i) \tag{9}$$

In the above equation, $\theta_i(t)$ denotes the parameters of the client i in round t, η represents the learning rate, which determines the step size or update speed of the model parameters, $\eta \nabla L_i\left(\theta_i(t), D_i\right)$ represents the gradient or derivative of the loss function with respect to the model parameters $\theta_i(t)$, where $L_i\left(\theta_i(t), D_i\right)$ is the loss function for client i using local data D_i.

FedPer pseudocode is presented in Algorithm 7.

Algorithm 7: FedPer in MHAT-FL

Input Human Activity Data
Output Client-Specific Weights
Server starts:
    initialize $w0$
    foreach round l= 1, 2, 3,... **do**
        k ←maximum (C · D, 1)
        S⍰←(random set of k clients)
        foreach client D∈Sl in parallel **do**
            $w_{l+1} \leftarrow \left(\frac{1}{D}\right) * \sum_{D=1}^{D} \frac{n_D}{n} w_{l+1}^{D}$
        endfor
        $w_{l+1} \leftarrow \sum_{D=1}^{D} \frac{n_D}{D} w_{l+1}^{D}$
    endfor
Client Update (D, w):
        Z ← (split ρ_Z into batches of size Z)
        foreach local epoch i from 1 to F, **do**
            for batch z∈ Z, **do**
                $w \leftarrow w - \eta \nabla L(w; Z)$
            endfor
        endfor
return w to the server

4 Evaluation

This section presents a comprehensive evaluation of MHAT-FL. Section 4.1 introduces the datasets, and Sect. 4.2 describes the hardware and software configurations used to implement MHAT-FL. The evaluation criteria are introduced in Sect. 4.3, along with their mathematical relationships. Section 4.4 reports the optimal hyperparameter and Transformer

configuration settings across datasets. Section 4.5 extensively evaluates MHAT-FL for HAR using more comprehensive scenarios and metrics. Section 4.6 focuses on privacy assessment, evaluating the resilience of membership inference attacks (MIA) and weight update perturbations. Section 4.7 and 4.8 provide insights through ablation studies and computational complexity analysis. Finally, Sect. 4.9 evaluates the performance of MHAT-FL in comparison to state-of-the-art HAR models.

4.1 Datasets

Human activity recognition involves identifying and classifying human activities using sensors embedded in electronic devices. Various sensors, including accelerometers, gyroscopes, and magnetometers, are used in data collection. This study evaluates the proposed model using three publicly available datasets: MotionSense, Opportunity, and PAMAP2. These datasets are notable for their size, diversity, and completeness, which distinguishes them from other available datasets, as shown in Table 2.

The PAMAP2 dataset [25], developed as part of a European project aimed at enhancing the quality of life, particularly for the elderly, comprises 12 core activities and six optional activities. This dataset comprises various measurements from inertial sensors placed on different parts of the volunteers' bodies, encompassing 52 dimensions.

The MotionSense dataset [26] contains time-series data from Accelerometer, Gyroscope, and magnetometer sensors collected using an iPhone 6 at a rate of 50 Hz. This data was gathered from 24 volunteers with diverse characteristics, including six daily activities performed in 15 different efforts.

The Opportunity dataset [27] is an extensive multimodal sensor dataset that captures data from various sensors, including Accelerometers, Gyroscopes, magnetometers, and inertial measurement units. This data, collected from 12 volunteers while they performed 17 different activities, was sampled at 30 Hz, providing a precise temporal aspect to the dataset.

4.2 Hardware/Software configurations

The Google Colab infrastructure has been utilized to implement MHAT-FL. Considering the model has been developed using Federated Learning techniques and the Transformer model simultaneously, significant processing power was required, necessitating the Colab Pro+service, which provides an NVIDIA A100 GPU with 40 GB of HBM2 memory and 6912 CUDA cores, supported by Intel Xeon Scalable processors and 83.5 GB of RAM.

Table 2 Specifications of the datasets

Datasets	Classes	Number of participants	Sensors
PAMAP2	12	9	Accelerometer, Gyroscope, and Magnetometer
MotionSense	6	24	Accelerometer and Gyroscope
Opportunity	5	4	Accelerometer, Gyroscope, and Magnetometer

The execution duration for each model was approximately 1.5 h, based on 100 rounds. MHAT-FL has been implemented using Python and machine learning libraries, including Tensorflow, NumPy, PyPlot, Pandas, and Matplotlib.

4.3 Evaluation criteria

To accurately evaluate the performance of the MHAT-FL, we have selected a set of evaluation metrics that are widely recognized in the literature for their effectiveness in measuring model performance in classification tasks [1, 5]. In this study, we employed Accuracy, F1-Score, Precision, and Recall as our primary evaluation metrics. These are the primary assessment metrics for the effectiveness of the Transformer-based architecture and the federated learning framework in accurately recognizing human activities through decentralized datasets. As the framework's primary objective is to enhance classification performance while maintaining privacy, the selection of these metrics is thus appropriate to convey the model's practical utility. On the other hand, these criteria were chosen because they are well-known and used to evaluate classification performance in the HAR and machine learning communities. We want to ensure that our evaluation is comparable to previous studies, so we use a well-established assessment using the standard metrics described above:

- Accuracy (Acc): This metric is the most prevalent measure of classification performance, defined as the ratio of correctly predicted observations to the total number of observations. It is a fundamental measure of the model's accuracy in identifying the correct cases, as shown in Eq. (10):

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \tag{10}$$

- F1-score (F1): The F1-score is useful when dealing with unbalanced classes because it considers both Precision and Recall. This metric, represented in Eq. (11), is critical for understanding the trade-off between false positives and false negatives, with higher values indicating better model performance. The F1-score helps us understand how accurate our model is, which is important in areas like HAR, where figuring out specific actions matters:

$$\text{F1}-\text{score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{11}$$

- Precision (P): Precision measures how many of the positive predictions are correct. This metric is important because it helps us see how well the model avoids false positives:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{12}$$

- Recall (R): shows the proportion of actual positive predictions among all possible positive predictions. This measure is crucial for assessing the model's sensitivity, particularly in HAR applications, where accurately identifying actions is essential for achieving good results.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{13}$$

(TP: true positive, TN: true negative, FP: false positive, and FN: false negative)

To evaluate the data privacy aspects of the MHAT-FL federated learning environment, the following criteria are considered:

- The Membership Inference Attack (MIA) is a type of privacy threat in federated learning, where an adversary attempts to determine whether a particular data point was included in the training set. The MIA accuracy is defined as the percentage of correctly predicted membership of a data point in the training set:

$$\text{MIA Accuracy} = \frac{\text{Number of correctly predicted membership}}{\text{Total number of data points}} \times 100\% \tag{14}$$

- Weight Distance (Update Perturbation Metric): In addition to the MIA metric, we assess privacy using the weight distance metric, which measures the Euclidean distance between the model weights before and after local training with Gaussian noise:

$$D = \sqrt{\sum_{i=1}^{n} \left\| W_i^{\text{before}} - W_i^{\text{after}} \right\|_F^2} \tag{15}$$

Finally, the Confusion Matrix, which is a powerful tool for comprehensively summarizing the model's classification performance, has been utilized. It offers a clear breakdown of correct and incorrect predictions across all categories, enhancing our understanding of the model's strengths and weaknesses in predicting specific human actions.

4.4 MHAT-FL configuration settings

To ensure reproducibility and provide clarity on the implementation details, we present the configuration settings used for training the proposed MHAT-FL model. These settings are organized into two main categories: (1) general deep learning parameters, such as learning rate and batch size, and (2) Transformer-specific parameters, including the number of attention heads, hidden dimensions, and positional encoding strategy.

Table 3 provides these hyperparameters, and the details of both configuration settings are discussed in the following subsections.

Table 3 Configuration settings for the MHAT-FL Model

General Hyperparameter	Value	Transformer Hyperparameter	Value
Optimizer	Adam	Number of Attention Heads	4
Loss Function	Categorical Cross-Entropy	Hidden Dimension (Server)	64
Number of Epochs	100	Hidden Dimension (Client)	128
Number of Clients	10	Feedforward Network Dimension	1024
Batch Sizes	16, 32, 64	Dropout Rate	0.1
Learning Rates	0.01, 0.001	Activation Function (All Layers)	ReLU (Dense/FFN Layer), Softmax (Dense Output)

4.4.1 Hyperparameter settings

This section evaluates various hyperparameters of MHAT-FL, including batch size and learning rate, in both FedAvg and FedPer models across the PAMAP2, MotionSense, and Opportunity datasets. The Adam optimizer, known for its robustness to noisy gradients, was also employed. It is common in Federated Learning due to the variation in data quality and quantity across clients. We set the number of epochs to 100 and the number of clients to 10. This number of clients ensures that the data is sufficiently distributed, capturing a wide range of patterns and reducing the risk of overfitting to subsets of data. Categorical Cross Entropy was used as the loss function.

For the PAMAP2 dataset, a sequence length of 32 time steps was chosen to balance the capture of temporal information and maintain computational efficiency. Each time step has 36 features, representing multiple sensor readings that provide a rich representation of activities. For the MotionSense dataset, a sequence length of 72 time steps was chosen to capture sufficient temporal patterns in the motion data. Each time step has only one feature, simplifying the model and allowing for efficient capture of temporal dynamics. For the Opportunity dataset, a sequence length of 300 time steps ensures that long-term dependencies and complex activities are captured. Each time step has 108 features, containing detailed sensor data that comprehensively represent the environment and activities.

The following discussion presents the optimal values of batch sizes and learning rates for the FedAvg and FedPer models using the PAMAP2, MotionSense, and Opportunity datasets.

Table 4 shows the results of different batch sizes and learning rates on the PAMAP2, motionsense, and opportunity datasets. For PAMAP2, the best result was achieved with a batch size of 32 and a learning rate of 0.001, resulting in an accuracy of 0.9952. For motionsense, the optimal result was achieved with a batch size of 32 and a learning rate of 0.001, attaining an accuracy of 0.9910. For opportunity, the best result was achieved with a batch size of 64 and a learning rate of 0.01, resulting in an accuracy of 0.9978.

Evaluation of accuracy using FedAvg for PAMAP2, motionsense, and opportunity datasets the evaluation results for different batch sizes and learning rates using FedPer on the PAMAP2, motionsense, and opportunity datasets are presented in Table 5. For PAMAP2, the best result was obtained with a batch size of 16 and a learning rate of 0.001, achieving

Table 4 Evaluation of accuracy using FedAvg for PAMAP2, MotionSense, and opportunity datasets

PAMAP2			MotionSense			Opportunity		
Batch Size	Learning Rate	Accuracy	Batch Size	Learning Rate	Accuracy	Batch Size	Learning Rate	Accuracy
16	0.01	0.9771	16	0.01	0.9904	16	0.01	0.9957
32	0.01	0.9794	32	0.01	0.9809	32	0.01	0.9914
64	0.01	0.9782	64	0.01	0.9833	64	0.01	**0.9978**
16	0.001	0.9944	16	0.001	0.9881	16	0.001	0.9936
32	0.001	**0.9952**	32	0.001	**0.9910**	32	0.001	0.9936
64	0.001	0.9939	64	0.001	0.9869	64	0.001	0.9808

Table 5 Evaluation of accuracy using FedPer for PAMAP2, MotionSense, and opportunity datasets

PAMAP2			MotionSense			Opportunity		
Batch Size	Learning Rate	Accuracy	Batch Size	Learning Rate	Accuracy	Batch Size	Learning Rate	Accuracy
16	0.01	0.9717	16	0.01	0.9845	16	0.01	**0.9979**
32	0.01	0.9782	32	0.01	0.8964	32	0.01	0.9701
64	0.01	0.9790	64	0.01	0.9821	64	0.01	0.9680
16	0.001	**0.9944**	16	0.001	0.9905	16	0.001	0.9638
32	0.001	0.9942	32	0.001	**0.9911**	32	0.001	0.9744
64	0.001	0.9935	64	0.001	0.9810	64	0.001	0.9744

an accuracy of 0.9944. For motionsense, the optimal result was achieved with a batch size of 32 and a learning rate of 0.001, attaining an accuracy of 0.9911. For opportunity, the best result was achieved with a batch size of 16 and a learning rate of 0.01, resulting in an accuracy of 0.9979.

We now provide a summary of the evaluation of different hyperparameter settings for MHAT-FL using FedAvg and FedPer, as presented in Table 6.

4.4.2 Transformer settings

For the Transformer model used in MHAT-FL, the output dimensions of 64 and 128 have been selected for the server-side and client-side Transformers, respectively. The higher dimension on the client side is chosen to train the model, prioritizing performance and

Table 6 Optimal hyperparameter setting of MHAT-FL

Dataset	FL Model	Optimum Batch Size	Optimum Learning Rate	Accuracy
PAMAP2	FedAvg	32	0.001	0.9952
PAMAP2	FedPer	16	0.001	0.9944
MotionSense	FedAvg	32	0.001	0.9910
MotionSense	FedPer	32	0.001	0.9911
Opportunity	FedAvg	64	0.01	0.9978
Opportunity	FedPer	16	0.01	0.9979

accuracy by leveraging local computing resources. Conversely, the lower dimension on the server side emphasizes efficiency in communication, aggregation, and reducing computational load. The number of attention heads has been selected as 4, allowing the model to focus on different parts of the input sequence simultaneously. This multi-head attention mechanism enhances the model's ability to understand complex patterns and relationships within sequences, thereby improving overall performance. For the feedforward network dimension and dropout rate, 1024 and 0.1 were selected. The high dimension of the Feedforward network provides substantial capacity for learning complex representations, thus increasing the model's ability to process and transform input data efficiently. This leads to improved performance in capturing intricate patterns and dependencies. The chosen Dropout rate introduces regularization to the model, preventing overfitting. In addition to the above settings, Table 7 specifies the Transformer model's configurations for different datasets, including the layers of the Transformer model and the output shape of each layer.

4.5 Comprehensive evaluation of MHAT-FL

Based on the optimal hyperparameters obtained in Table 7, a more comprehensive evaluation of MHAT-FL using FedAvg and FedPer models is conducted on the PAMAP2, MotionSense, and Opportunity datasets, employing broader factors in this subsection.

Table 7 Specifications of the transformer model of MHAT-FL

PAMAP2 dataset		
Layer (type)	Output shape	Param
input_1 (InputLayer)	[(None, 32, 36)]	0
encoder (Encoder)	(None, 32, 64)	151,424
flatten (Flatten)	(None, 2048)	0
dense_7 (Dense)	(None, 12)	24,588
Total params: 176,012 (687.55 KB)		
Trainable params: 176,012 (687.55 KB)		
MotionSense dataset		
Layer (type)	**Output shape**	**Param**
input_1 (InputLayer)	[(None, 72, 1)]	0
encoder (Encoder)	(None, 72, 64)	149,184
flatten (Flatten)	(None, 4608)	0
dense_7 (Dense)	(None, 6)	27,654
Total params: 176,838 (690.77 KB)		
Trainable params: 176,838 (690.77 KB*)*		
Opportunity dataset		
Layer (type)	**Output shape**	**Param**
input_1 (InputLayer)	[(None, 300, 108)]	0
encoder (Encoder)	(None, 300, 64)	156,032
flatten (Flatten)	(None, 19200)	0
dense_7 (Dense)	(None, 5)	96,005
Total params: 252,037 (984.52 KB)		
Trainable params: 252,037 (984.52 KB)		

4.5.1 Evaluation of MHAT-FL using FedAvg

For the evaluation purpose of MHAT-FL using the FedAvg and FedPer models on the PAMAP2, MotionSense, and Opportunity datasets, various evaluation metrics such as Precision, Recall, F1-Score, and the Confusion Matrix are considered to assess the model's performance.

Table 8 Shows the results of the mentioned metrics for the PAMAP2 dataset. The model achieved high values in precision, recall, and F1-Score, demonstrating its ability to handle the complexity of real-world data. The confusion matrix in Fig. 4 provides a detailed analysis of the model's performance across 12 different human activity classes. Classes 0, 1, 3, 4, 5, 6, 9, and 11 exhibit nearly perfect prediction accuracy. Minor misclassifications are observed in classes 2, 7, 8, and 10, where a small number of samples are incorrectly classified.

In Table 8, M.avg, or Macro Average represents the average values of Precision, Recall, and F1-Score for all classes. W.avg or Weighted Average indicates the weighted average of these evaluation metrics, where the weight of each class is based on the number of its samples, giving larger classes a greater impact on the average calculation.

Table 9 Shows the results of the mentioned metrics for the motionsense dataset. The model achieved high values in precision, recall, and F1-Score, demonstrating its ability to handle the complexity of real-world data. The confusion matrix in Fig. 5 provides a detailed analysis of the model's performance across six different human activity classes. Classes 3, 4, and 5 exhibit nearly perfect prediction accuracy. Minor misclassifications are observed in classes 0, 1, and 2, where a small number of samples are incorrectly classified.

Table 10 shows the results of the mentioned metrics for the Opportunity dataset. The model achieved high values in Precision, Recall, and F1-Score, demonstrating its ability to handle the complexity of real-world data. The Confusion Matrix in Fig. 6 provides a detailed analysis of the model's performance across five different human activity classes: classes 0, 2, 3, and 4 exhibit nearly perfect prediction accuracy. Minor misclassifications are observed in class 1, where a small number of samples are incorrectly classified.

Table 8 Evaluation of MHAT-FL using FedAvg for recognizing 12 human activities in PAMAP2 dataset

Actions	Class	Precision	Recall	F1-Score	Support
Lyi	0	0.9990	0.9990	0.9990	3862
Sit	1	0.9978	0.9929	0.9954	3667
stand	2	0.9771	0.9989	0.9879	3758
walk	3	0.9956	0.9971	0.9964	4534
running	4	1.0000	0.9979	0.9989	1895
cycling	5	0.9994	1.0000	0.9997	3287
nord walk	6	0.9962	0.9978	0.9970	3685
ascend stair	7	0.9908	0.9865	0.9886	2293
descend stair	8	0.9919	0.9873	0.9896	2119
Vacuum clean	9	0.9986	0.9992	0.9989	3542
ironing	10	0.9994	0.9864	0.9928	4699
rope jump	11	0.9980	1.0000	0.9990	1021
M.avg		0.9953	0.9952	0.9953	38,362
W.avg		0.9952	0.9952	0.9952	38,362

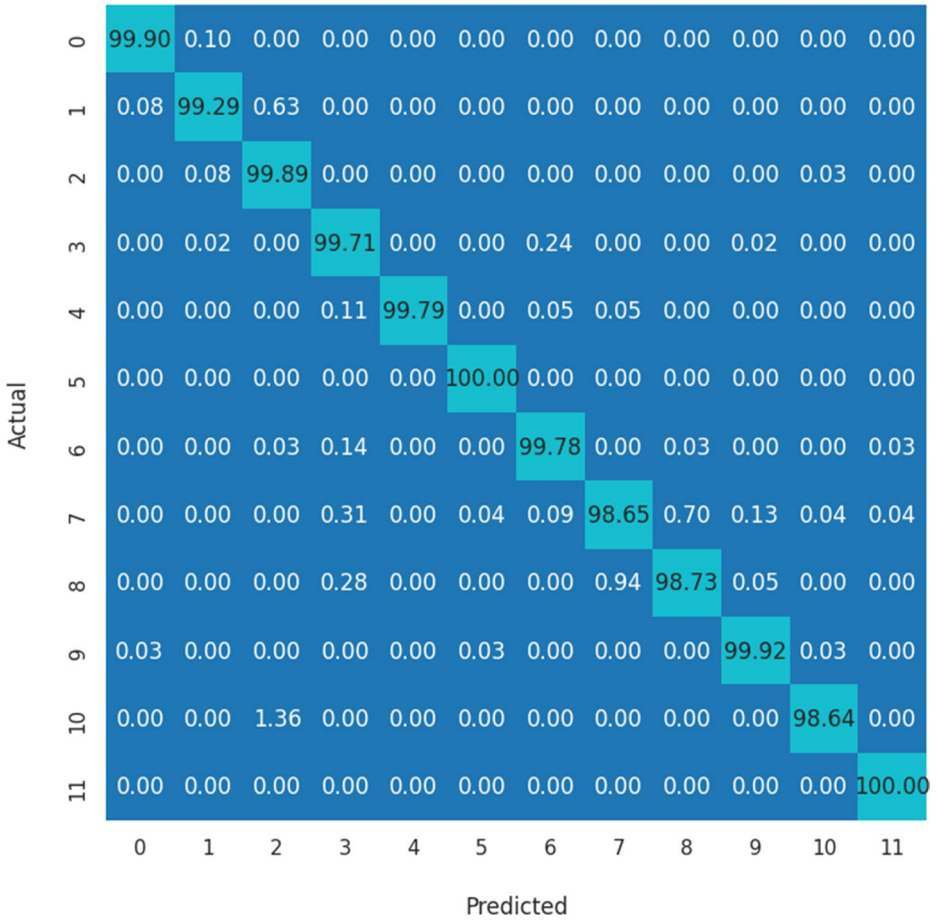

Fig. 4 MHAT-FL confusion matrix of 12 human activities using FedAvg in PAMAP2 dataset

Table 9 Evaluation of MHAT-FL using FedAvg for recognizing 6 human activities in MotionSense dataset

Action	Num	Precision	Recall	F1-Score	Support
wlk	0	0.9600	0.9863	0.9730	73
sit	1	0.9836	1.0000	0.9917	60
std	2	0.9730	0.9600	0.9664	75
ups	3	1.0000	0.9943	0.9971	174
jog	4	1.0000	1.0000	1.0000	135
dws	5	1.0000	0.9935	0.9968	155
M.avg		0.9861	0.9890	0.9875	672
W.avg		0.9912	0.9911	0.9911	672

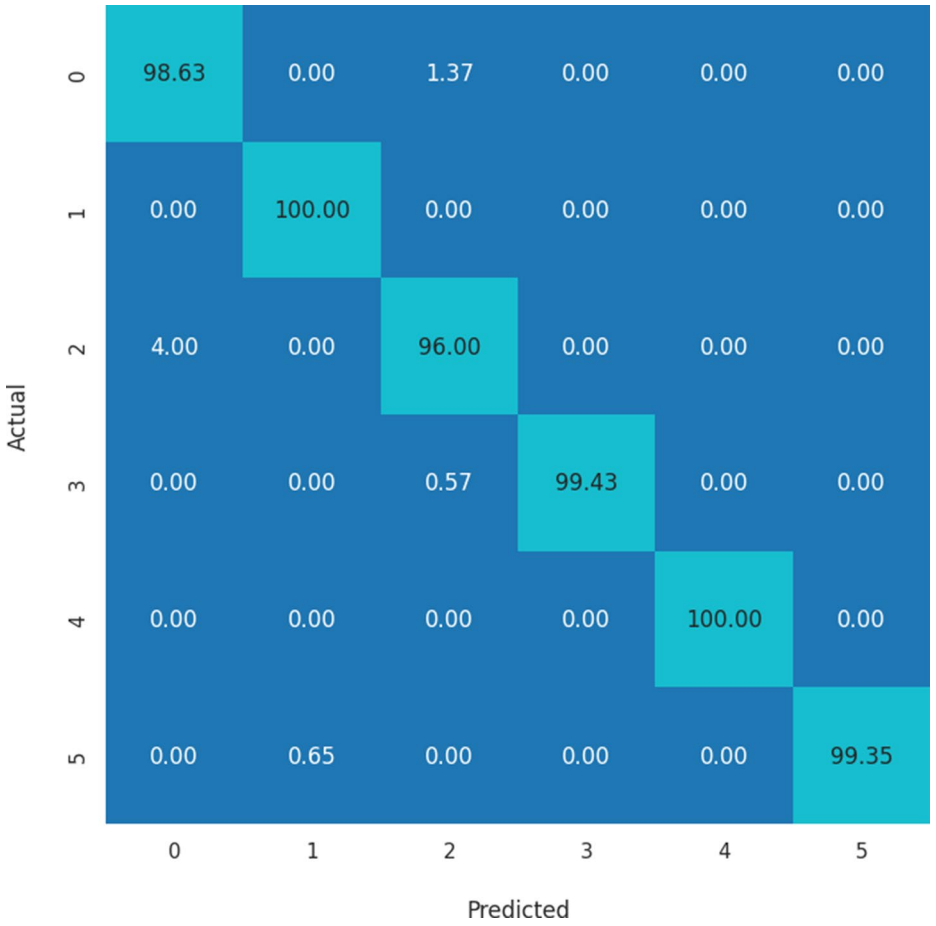

Fig. 5 MHAT-FL confusion matrix of 6 human activities using FedAvg in MotionSense dataset

Table 10 Evaluation of MHAT-FL using FedAvg for recognizing 5 human activities in opportunity dataset

Action	Num	Precision	Recall	F1-Score	Support
Relaxing	0	1.0000	1.0000	1.0000	28
Coffee time	1	0.9828	1.0000	0.9913	57
Early morning	2	1.0000	1.0000	1.0000	110
Cleanup	3	1.0000	1.0000	1.0000	117
Sandwich time	4	1.0000	0.9936	0.9968	157
M.avg		0.9966	0.9987	0.9976	469
W.avg		0.9979	0.9979	0.9979	469

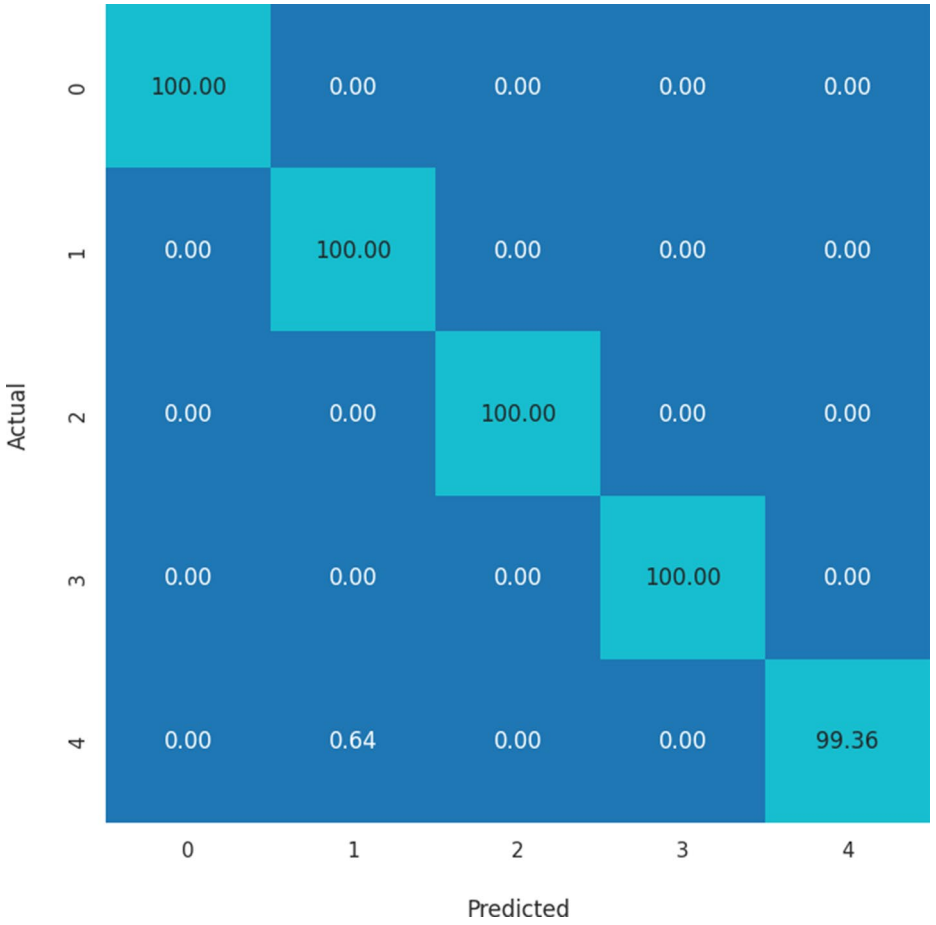

Fig. 6 MHAT-FL confusion matrix of 5 human activities using FedAvg in opportunity dataset

4.5.2 Evaluation of MHAT-FL using FedPer

Table 11 shows the results of the mentioned metrics for the PAMAP2 dataset. The model achieved high values in precision, recall, and F1-Score, demonstrating its ability to handle the complexity of real-world data. The confusion matrix in Fig. 7 provides a detailed analysis of the model's performance across 12 different human activity classes. Classes 0, 1, 3, 4, 5, 6, 9, and 11 exhibit nearly perfect prediction accuracy. Minor misclassifications are observed in classes 2, 7, 8, and 10, where a small number of samples are incorrectly classified.

Table 12 shows the results of the mentioned metrics for the motionsense dataset. The model achieved high values in precision, recall, and F1-Score, demonstrating its ability to handle the complexity of real-world data. The confusion matrix in Fig. 8 provides a detailed analysis of the model's performance across six different human activity classes. Classes 0, 2, and 4 exhibit nearly perfect prediction accuracy. Minor misclassifications are observed in classes 1, 3, and 5, where a small number of samples are incorrectly classified.

Table 11 Evaluation of MHAT-FL using FedPer for recognizing 12 human activities in PAMAP2 dataset

Actions	Num	Precision	Recall	F1-Score	Support
Lyi	0	0.9992	0.9987	0.9990	3828
sit	1	0.9971	0.9910	0.9940	3760
stand	2	0.9758	0.9995	0.9875	3750
walk	3	0.9959	0.9970	0.9964	4637
running	4	1.0000	0.9970	0.9985	1988
cycling	5	0.9976	0.9994	0.9985	3265
nord walk	6	0.9956	0.9984	0.9970	3653
ascend stair	7	0.9894	0.9842	0.9868	2281
descend stair	8	0.9911	0.9848	0.9880	2044
Vacuum clean	9	0.9977	0.9957	0.9967	3497
ironing	10	0.9981	0.9867	0.9923	4728
rope jump	11	0.9947	1.0000	0.9973	931
Accuracy				0.9944	38,362
M.avg		0.9943	0.9944	0.9943	38,362
W.avg		0.9945	0.9944	0.9944	38,362

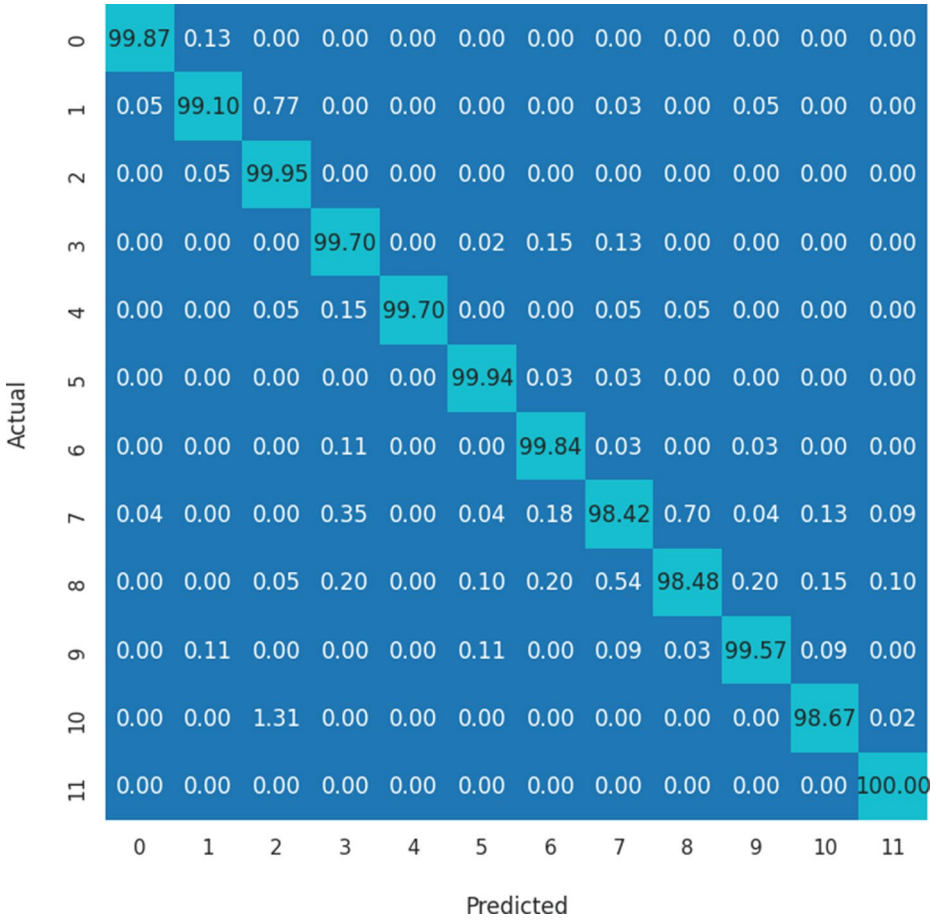

Fig. 7 MHAT-FL Confusion Matrix of 12 human activities using FedPer in the PAMAP2 dataset

Table 12 Evaluation of MHAT-FL using FedPer for recognizing 6 human activities in MotionSense dataset

Action	Num	Precision	Recall	F1-Score	Support
wlk	0	0.9945	0.9945	0.9945	182
sit	1	0.9714	0.9714	0.9714	70
std	2	1.0000	1.0000	1.0000	150
ups	3	1.0000	0.9815	0.9907	54
jog	4	1.0000	1.0000	1.0000	162
dws	5	0.9455	0.9630	0.9541	54
Accuracy				0.9911	672
M.avg		0.9852	0.9851	0.9851	672
W.avg		0.9912	0.9911	0.9911	672

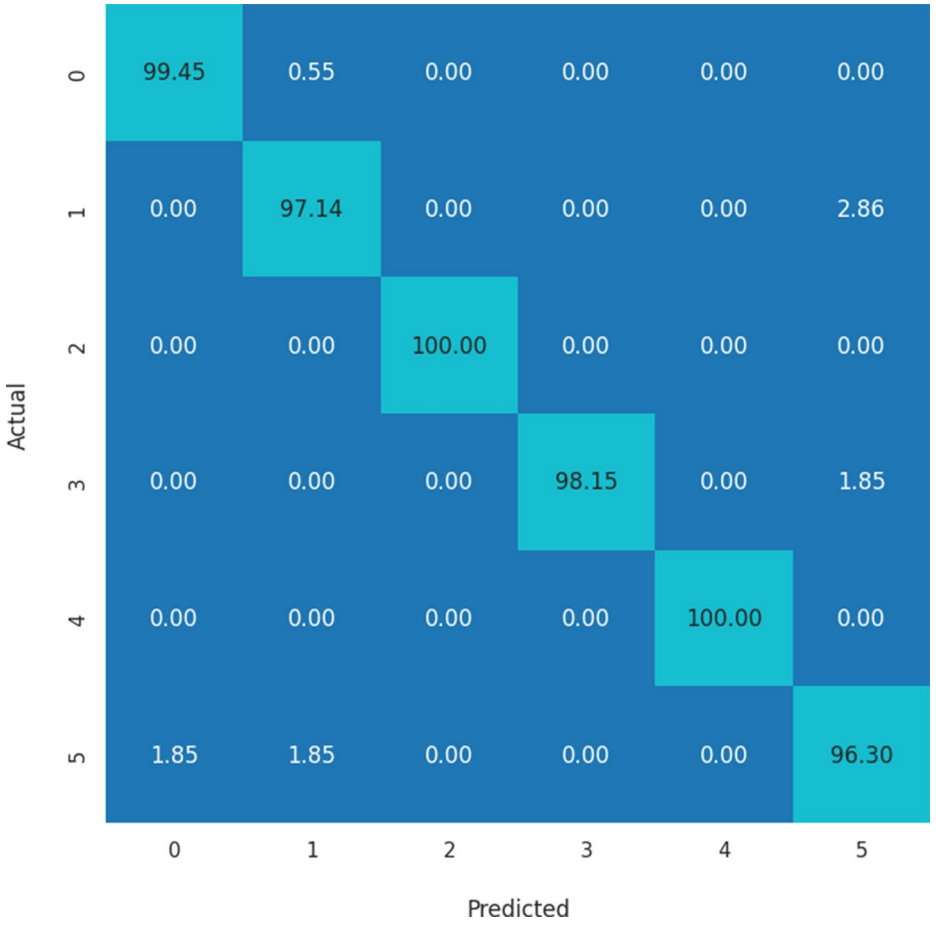

Fig. 8 MHAT-FL confusion matrix of 6 human activities using FedPer in the MotionSense dataset

Table 13 shows the results of the mentioned metrics for the Opportunity dataset. The model achieved high values in Precision, Recall, and F1-Score, demonstrating its ability to handle the complexity of real-world data. The Confusion Matrix in Fig. 9 provides a

Table 13 Evaluation of MHAT-FL using FedPer for recognizing 5 human activities in opportunity dataset

Action	Num	Precision	Recall	F1-Score	Support
Relaxing	0	1.0000	1.0000	1.0000	28
Coffee time	1	1.0000	1.0000	1.0000	57
Early morning	2	0.9910	1.0000	0.9955	110
Cleanup	3	1.0000	0.9915	0.9957	117
Sandwich time	4	1.0000	1.0000	1.0000	157
Accuracy				0.9979	469
M.avg		0.9982	0.9983	0.9982	469
W.avg avg		0.9979	0.9979	0.9979	469

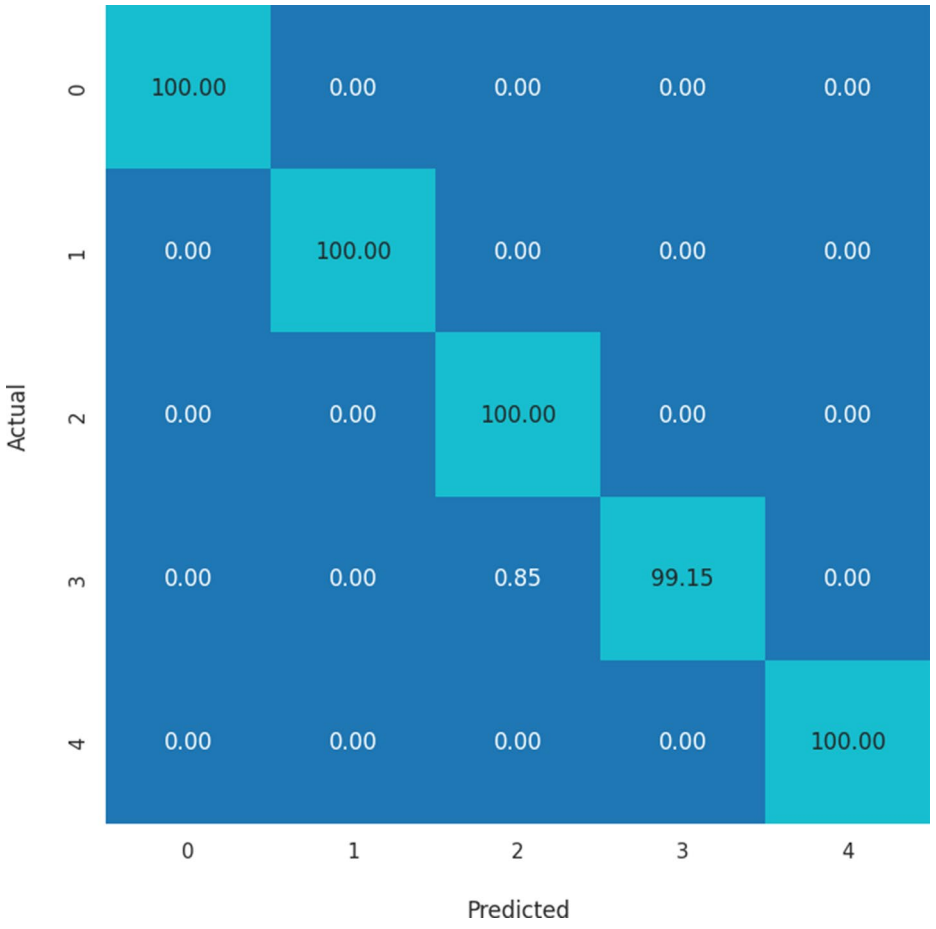

Fig. 9 MHAT-FL Confusion Matrix of 5 human activities using FedPer in the opportunity dataset

detailed analysis of the model's performance across five different human activity classes. Classes 0, 1, and 4 exhibit perfect prediction accuracy. Minor misclassifications are observed in classes 2 and 3, where a small number of samples are incorrectly classified.

4.6 MHAT-FL privacy assessment

In this section, we systematically assess the privacy risks associated with federated learning by evaluating two complementary metrics: (1) the magnitude of client weight update distances and (2) the effectiveness of membership inference attacks (MIA). The weight update distance quantifies the extent to which each client's local model diverges from the global model during training, providing insight into the potential exposure of client-specific information through model parameters [53]. We also empirically measure MIA accuracy, a widely adopted standard for quantifying privacy leakage in federated settings [51, 52]. MIA aims to determine whether a specific data sample was included in the training set, with accuracy significantly above a random guess, suggesting potential vulnerability. The random guess rate, in this context, is simply the reciprocal of the number of classes in the dataset (i.e., 1/number of classes), reflecting the expected baseline performance of an uninformed attacker (for example, 8.33% for 12 classes in PAMAP2). Recent studies have highlighted the importance of MIA accuracy as a core privacy metric, demonstrating that, even when raw data are never shared, localized updates may still leak membership status unless specialized defenses are employed [51, 52]. By jointly analyzing these metrics across datasets and personalization methods, we aim to provide a comprehensive privacy risk profile for standard (FedAvg) and personalized (FedPer) federated learning approaches.

The following subsections present a detailed empirical privacy evaluation by reporting, for each dataset and method, the distribution of MIA confidence scores for both member and non-member samples and the dynamics of client weight update distances throughout communication rounds. Together, these metrics visualize and substantiate the privacy-preserving properties of MHAT-FL.

4.6.1 PAMAP2 dataset

For the PAMAP2 dataset employing the FedAvg algorithm, the analysis of client weight update distances in communication round 100 indicates that client weight update distances remain substantial, with average magnitudes of 9.00 across clients (Fig. 10). Utilizing the FedPer method on the PAMAP2 dataset, the client weight update distances in communication round 100 are averaging 9.90 across clients (Fig. 11).

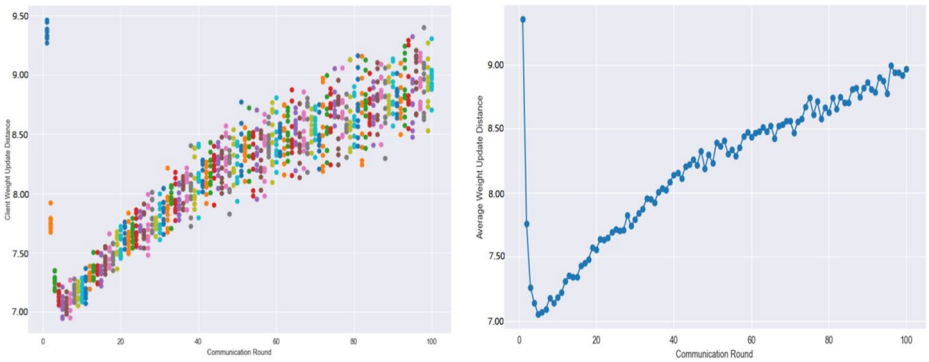

Fig. 10 Client average model weight update distances in FedAvg across communication rounds on PAMAP2

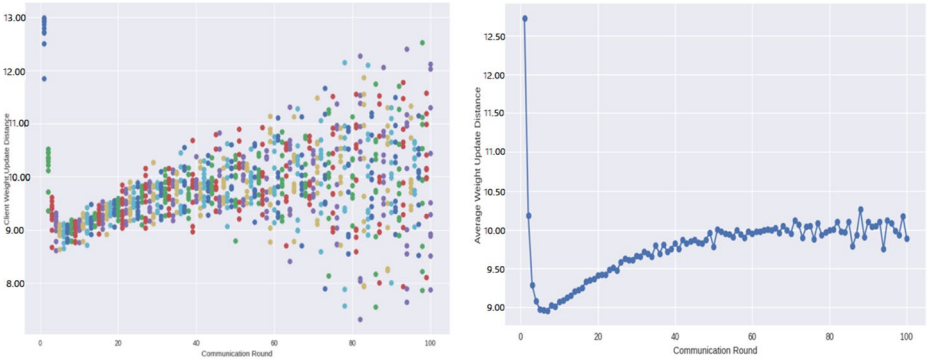

Fig. 11 Client average model weight update distances in FedPer across communication rounds on PAMAP2

Tables 14 and 15 summarize the distribution of member and non-member predictions above and below the MIA attack threshold for FedAvg and FedPer methods. Both groups exceed the threshold with approximately similar confidence levels, confirming that the attack cannot distinguish between members and non-members.

The associated MIA confidence threshold crossings for FedPer on PAMAP2 the associated MIA accuracy for fedavg.is only 16.67%, which is modestly above the random guess rate of 8.33% for the 12-class setting of the PAMAP2 dataset, suggesting that neither standard nor personalized aggregation results in meaningful privacy leakage (Table 16). Thus, while clients exhibit meaningful local learning, the risk of exposing individual participation in the federated training remains limited in this setting. Utilizing the FedPer method, MIA accuracy remains at 16.67%, comparable to fedavg. This consistency suggests that the increased diversity in client updates introduced by FedPer does not significantly elevate the threat of membership inference on this dataset, and privacy risk remains suitable even with personalized model components (Table 16)

Table 14 MIA confidence threshold crossings for FedAvg on PAMAP2

Group	Above Threshold	Below Threshold	Total
Members	139,300	700	140,000
Non-Members	39,600	400	40,000

Table 15 MIA confidence threshold crossings for FedPer on PAMAP2

Group	Above Threshold	Below Threshold	Total
Members	148,000	2,000	150,000
Non-Members	38,500	1,500	40,000

Table 16 MIA accuracy and client weight update distance for PAMAP2 dataset

Method	Average Client Weight Update Distance Communication Round 100	Membership Inference Attack Accuracy
FedAvg	9.00	16.67%
FedPer	9.90	16.67%

4.6.2 Opportunity dataset

In FedAvg, Analysis of the per-client weight update distances reveals a strikingly narrow and stable range, with nearly all clients' updates falling in 5.00 at communication round 100 (Fig. 12). This tight clustering of update magnitudes demonstrates high learning homogeneity, suggesting that the Opportunity dataset induces similar optimization behavior across all participating clients. In practical terms, this means clients share similar data distributions and convergent local training dynamics, minimizing the potential for any client to inadvertently "stand out" or leak unique information through its model updates.

Examining the per-client weight update distances under FedPer, we observe that the mean and spread of update magnitudes remain virtually indistinguishable from those seen with FedAvg. As implemented by FedPer, personalization does not induce any notable increase in the dispersion of client update trajectories over time; the range of values for client weight updates continues to cluster at 5.00 at communication round 100 (Fig. 13). This high degree of statistical similarity suggests that enabling model personalization has a negligible effect on this dataset's underlying collaborative optimization process, reinforcing the impression of dataset-intrinsic homogeneity.

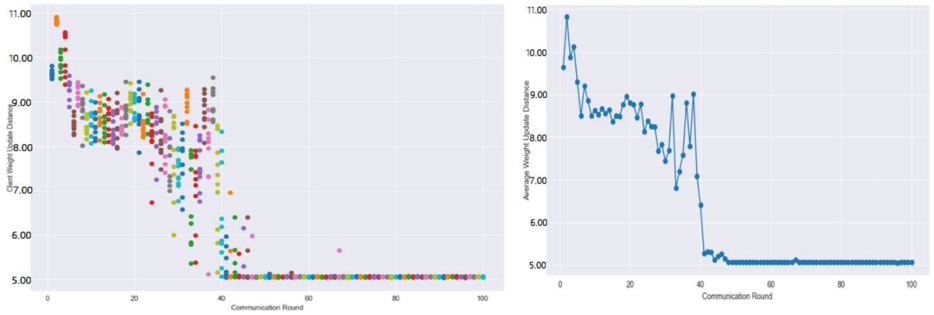

Fig. 12 Client average model weight update distances in FedAvg across communication rounds on Opportunity

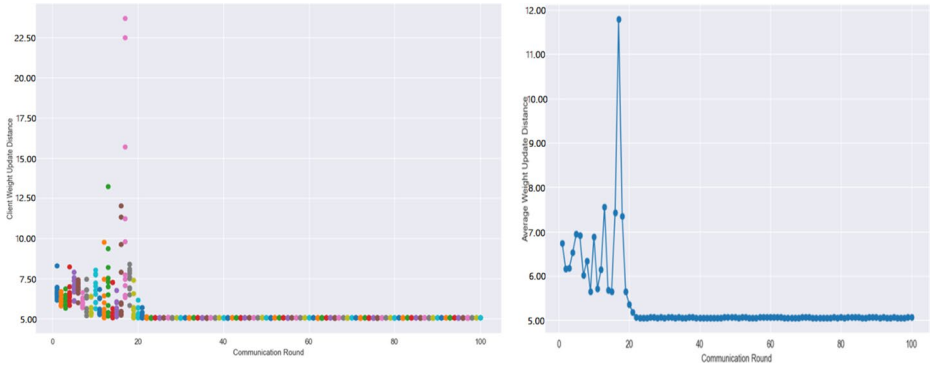

Fig. 13 Client average model weight update distances in FedPer across communication rounds on Opportunity

Tables 17 and 18 summarize the distribution of member and non-member predictions above and below the MIA attack threshold for FedAvg and FedPer on the Opportunity dataset. All members (1,900) and non-member (440) samples lie above the threshold, with both groups showing almost identical confidence distributions. This pronounced overlap confirms that the attack is unable to distinguish between members and non-members, resulting in performance no better than random guessing. Thus, it demonstrates no practical privacy risk for either method.

As discussed above, with the Opportunity dataset and FedAvg algorithm, client weight update distances converge rapidly and stabilize at 5.00 across all clients during 100 communication rounds. This homogeneity in the update distance is reflected in a low and stable MIA accuracy of 16.66% (Table 19). Given that the random guess rate for membership inference is 20% (calculated as one divided by the five activity classes in the Opportunity dataset), the observed MIA accuracy does not indicate a significant privacy risk. The consistently low and tightly grouped distances suggest that client models remain highly similar, reducing the likelihood of information leakage via model updates.

Applying FedPer to the Opportunity dataset leads to nearly identical weight update distance patterns as FedAvg, with all clients' distances clustered at five around 100 communication rounds. The minimal variance and sustained low MIA accuracy (16.66%) indicate adequate privacy protection under membership inference (Table 19). The results indicate that, on Opportunity, personalization via FedPer does not introduce additional privacy risk compared to FedAvg, as client updates are already highly homogeneous.

Table 17 MIA confidence threshold crossings for FedAvg on opportunity

Group	Above Threshold	Below Threshold	Total
Members	1,900	0	1,900
Non-Members	440	0	440

Table 18 MIA confidence threshold crossings for FedPer on opportunity

Group	Above Threshold	Below Threshold	Total
Members	1,900	0	1,900
Non-Members	440	0	440

Table 19 MIA accuracy and client weight update distance for opportunity dataset

Method	Average Client Weight Update Distance Communication Round 100	Membership Inference Attack Accuracy
FedAvg	5.00	16.66%
FedPer	5.00	16.66%

4.6.3 MotionSense dataset

In FedAvg, the plot of per-client weight update distances reveals an extremely narrow range, with all client curves confined tightly between 4.34 and 4.41 across all training rounds (Fig. 14). This remarkable consistency suggests that each client's data distribution is highly similar, indicating that the learning task is sufficiently uniform across clients. The absence of significant divergence or outlier update dynamics speaks to both the stability of the federated optimization process and the homogeneous nature of MotionSense activity recognition.

For FedPer, a detailed analysis of the per-client weight update distances reveals an even tighter clustering of update magnitudes compared to the FedAvg baseline (Fig. 15). Across all communication rounds, client updates remain consistently low, and the associated scatter, indicative of participant variability, is further reduced. This heightened synchronization suggests that personalization, as implemented in the FedPer paradigm,

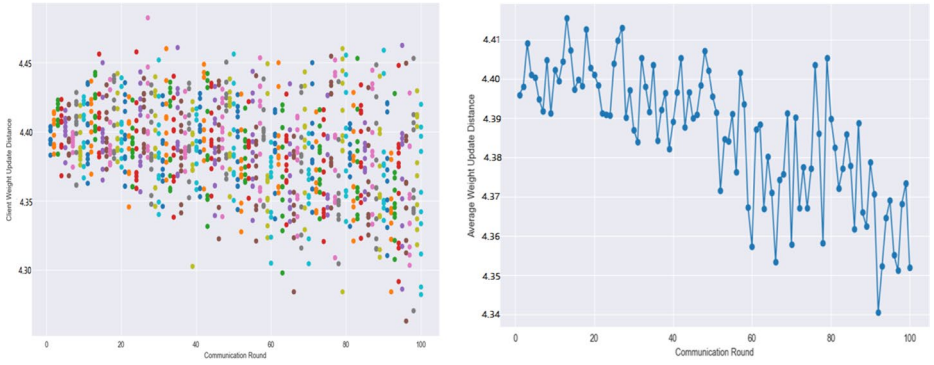

Fig. 14 Client average model weight update distances in FedAvg across communication rounds on MotionSense

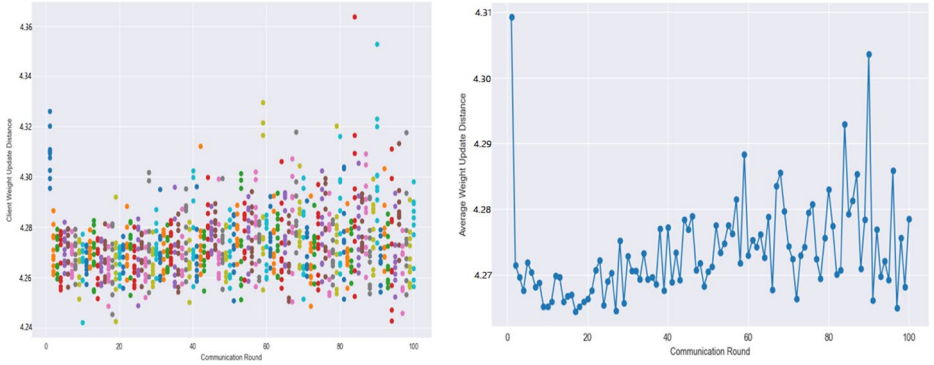

Fig. 15 Client average model weight update distances in FedPer across communication rounds on MotionSense

encourages clients to focus their updates more narrowly, particularly on the segments of the model that most closely reflect their data. As a result, overall model drift is minimized, and the update trajectories of all clients exhibit exceptional convergence.

Tables 20 and 21 present the distribution of member and non-member predictions above and below the MIA attack threshold for MotionSense using both FedAvg and FedPer. In both cases, nearly all members (2,480/2,500 for FedAvg; 2,450/2,500 for FedPer) and non-members (780/800 for FedAvg; 770/800 for FedPer) exceed the threshold, with only minor dispersion below. This high overlap indicates that the model's confidence scores provide no practical distinction between members and non-members. As a result, the MIA operates at random guessing levels, confirming that neither approach introduces meaningful privacy leakage for MotionSense.

Across all datasets, MIA accuracy remains in the range of 16.64–16.67%, which is slightly above the random guess rate and suggests a suitable level of resilience against membership inference attacks. This risk appears most pronounced in PAMAP2, where larger weight update distances and higher variability hint at greater susceptibility than the more stable MotionSense and Opportunity results. While FedPer's personalized updates can increase weight distances, particularly for datasets like PAMAP2, potentially raising privacy risks, they do not significantly deteriorate privacy for MotionSense.

The distribution of client weight update distances highlights the degree of personalization and data diversity in federated learning. Higher and more variable distances (e.g., in PAMAP2) reflect greater differences in client data and stronger local adaptation, while tightly clustered, lower distances (as seen in MotionSense and Opportunity) indicate more similarity in client distributions and faster model convergence. These results suggest that the choice of method and the underlying dataset play key roles in shaping how much clients diverge from the global model during training.

Table 20 MIA confidence threshold crossings for FedAvg on motionsense

Group	Above Threshold	Below Threshold	Total
Members	2,480	20	2,500
Non-Members	780	20	800

Table 21 MIA confidence threshold crossings for FedPer on MotionSense MIA confidence threshold crossings for fedper.on motionsense as discussed above, for the motionsense dataset, both FedAvg and fedper. approaches yield modest client weight update distances over communication rounds 1 to 100, with values at communication round 100 measured at 4.35 for FedAvg and 4.28 for fedper. The magnitude of these updates generally decreases or remains stable as training progresses, indicating effective model convergence. This stability in model updates is accompanied by consistently low membership inference attack (MIA) accuracy (16.64% for both methods), which closely aligns with the random guess rate of 16.67% for this 6-class problem reflects strong resistance to privacy leakage and demonstrating that both global (FedAvg) and personalized (FedPer) federated learning frameworks effectively preserve participant privacy on motionsense (Table 22)

Group	Above Threshold	Below Threshold	Total
Members	2,450	50	2,500
Non-Members	770	30	800

Table 22 MIA accuracy and client weight update distance for motionsense dataset

Method	Average Client Weight Update Distance Communication Round 100	Membership Inference Attack Accuracy
FedAvg	4.35	16.64%
FedPer	4.28	16.64%

4.7 Ablation study

To empirically validate the contribution of different components in MHAT-FL, such as the positional encoding method, we conducted an ablation study comparing the following configurations:

- Standard positional encoding: This version employs the widely used input-level sinusoidal positional encoding introduced in the original Transformer architecture of MHAT-FL (Table 23).
- Positional encoding at the input level: All positional encodings are applied at the input level (Table 24).
- No positional encoding: All positional encodings are removed to evaluate the importance of explicit positional information (Table 25).
- Without a feedforward layer: The feedforward layer is removed to analyze its contribution to learning complex feature transformations (Table 26).
- Without a normalization layer: The normalization layer is removed to assess its effect on model stability and convergence (Table 27).

Table 23 Accuracy of MHAT-FL with standard positional encoding

Dataset	FL Model	Optimum Batch Size	Optimum Learning Rate	Accuracy
PAMAP2	FedAvg	32	0.001	0.9952
PAMAP2	FedPer	16	0.001	0.9944
MotionSense	FedAvg	32	0.001	0.9910
MotionSense	FedPer	32	0.001	0.9911
Opportunity	FedAvg	64	0.01	0.9978
Opportunity	FedPer	16	0.01	0.9979

Table 24 Accuracy of MHAT-FL with positional encoding in the input

Dataset	FL Model	Batch size	Learning Rate	Accuracy
PAMAP2	FedAvg	32	0.001	0.9930
PAMAP2	FedPer	16	0.001	0.9934
MotionSense	FedAvg	32	0.001	0.9869
MotionSense	FedPer	32	0.001	0.9845
Opportunity	FedAvg	64	0.01	0.9914
Opportunity	FedPer	16	0.01	0.9936

Table 25 Accuracy of MHAT-FL without positional encoding

Dataset	FL Model	Batch size	Learning Rate	Accuracy
PAMAP2	FedAvg	32	0.001	0.9935
PAMAP2	FedPer	16	0.001	0.9943
MotionSense	FedAvg	32	0.001	0.9881
MotionSense	FedPer	32	0.001	0.9881
Opportunity	FedAvg	64	0.01	0.9936
Opportunity	FedPer	16	0.01	0.9829

Table 26 Accuracy of MHAT-FL without feedforward layer

Dataset	FL Model	Batch size	Learning Rate	Accuracy
PAMAP2	FedAvg	32	0.001	0.9573
PAMAP2	FedPer	16	0.001	0.9598
MotionSense	FedAvg	32	0.001	0.9809
MotionSense	FedPer	32	0.001	0.9774
Opportunity	FedAvg	64	0.01	0.9275
Opportunity	FedPer	16	0.01	0.9616

Table 27 Accuracy of MHAT-FL without normalization layer

Dataset	FL Model	Batch size	Learning Rate	Accuracy
PAMAP2	FedAvg	32	0.001	0.9878
PAMAP2	FedPer	16	0.001	0.9776
MotionSense	FedAvg	32	0.001	0.9892
MotionSense	FedPer	32	0.001	0.9905
Opportunity	FedAvg	64	0.01	0.9850
Opportunity	FedPer	16	0.01	0.9638

The models were evaluated using the same training and evaluation settings on the PAMAP2, MotionSense, and Opportunity datasets. The accuracies for each configuration are summarized in Tables 23, 24, 25, 26 and 27.

The proposed MHAT-FL model, which incorporates positional encoding directly into each attention head (Table 17), consistently achieves the highest accuracy across all datasets and federated learning models, highlighting its superior capability in capturing temporal dependencies. When positional encoding is applied at the input level (Table 24), the accuracy drops slightly, indicating that embedding positional information within attention heads yields more expressive representations. Removing positional encoding altogether (Table 25) leads to a further reduction in performance, confirming the importance of explicit position information. Additionally, ablating the feedforward layer (Table 26) results in a significant decline in accuracy, particularly on the Opportunity dataset, underscoring its crucial role in learning complex feature transformations. The absence of the normalization layer (Table 27) also degrades accuracy, though less severely, suggesting that normalization contributes to training stability and model regularization. Together, these results validate the effectiveness of our proposed positional encoding method and highlight the importance of key Transformer components for achieving optimal HAR performance.

4.8 Computational complexity analysis

Given the importance of understanding both the theoretical resource requirements and the practical efficiency of the MHAT-FL, we present our computational complexity analysis in two parts. First, we offer a theoretical complexity analysis, examining the computational cost of each key component of MHAT-FL in terms of input dimensions and model parameters. We then complement this with an empirical complexity analysis that focuses on the number of model parameters, memory footprint, and inference time for all datasets evaluated in this study.

4.8.1 Theoretical complexity analysis

The overall theoretical computational complexity of the MHAT-FL model is determined by the cost of processing each input sequence with a multi-layer Transformer encoder and the cost of global model aggregation during each round of federated learning, as calculated in the following:

Each client processes input sequences of length n and embedding dimension d through L Transformer layers. The computational complexity per sequence is dominated by the multi-head self-attention and feedforward sub-layers, yielding $O(n^2d + nd^2)$ per layer, or $O(L \cdot [n^2d + nd^2])$ in total.

For the federated learning aggregation part, the server aggregates the model updates from *N* clients, with each model parameter vector of size *P*, resulting in a communication and aggregation complexity of *O(NP)* per round.

Considering both client-side computations and server-side aggregation, the total cost per communication round is:

$$O_{\text{round}}\left(\text{MHAT}-\text{FL}\right) = N \cdot L \cdot \left(n^2d + nd^2\right) + N \cdot P \tag{16}$$

Finally, total complexity over C rounds can be expressed as the following equation:

$$O_{\text{total}}\left(\text{MHAT}-\text{FL}\right) = C \cdot \left[N \cdot L \cdot \left(n^2d + nd^2\right) + N \cdot P\right] \tag{17}$$

4.8.2 Empirical complexity analysis

To assess the empirical complexity of the proposed MHAT-FL model, we evaluate and report three key metrics: parameter count, memory footprint, and inference time. The results for these metrics were obtained across different datasets and model configurations.

First, we calculated the trainable parameter count and memory footprint for each dataset:

- PAMAP2 Dataset: The model consists of 176,012 parameters, corresponding to approximately 687.55 KB of memory.
- MotionSense Dataset: The model consists of 176,838 parameters, corresponding to approximately 690.77 KB of memory.
- Opportunity Dataset: The model comprises 252,037 parameters, corresponding to approximately 984.52 KB of memory.

Then, we calculated the practical inference time for processing one input sequence on a typical edge device. The inference times were approximated based on the number of floating-point operations (FLOPs) and scaled according to the processing capabilities of mid-range edge hardware. Considering the relatively low model size and efficient Transformer configuration, the inference times are calculated as follows:

- PAMAP2: 30 ms per sequence.
- MotionSense: 28 ms per sequence.
- Opportunity: 40 ms per sequence.

These inference times are feasible for near-real-time applications on edge devices with moderate computing power.

Therefore, the proposed MHAT-FL model introduces a manageable computational cost that aligns with the resource constraints of edge devices in a Federated Learning setup. The Transformer's output dimension was intentionally kept low on the server side to reduce communication overhead and computational burden. Furthermore, the model's moderate memory footprint (all under 1 MB) ensures compatibility with edge platforms. A summary of the empirical complexity results is provided in Table 28.

4.9 Comparison of MHAT-FL

This section compares MHAT-FL with a series of related studies in human activity recognition to better evaluate its position better. Baseline articles have been selected to represent the most advanced technology in human activity recognition using wearable sensors and deep learning techniques. Since MHAT-FL utilizes the Transformer model within a Federated Learning framework, several Transformer and FL models have been selected as baselines. Moreover, some hybrid deep neural network models have also been utilized to assess MHAT-FL more comprehensively.

The Transformer-based baselines include ConvTransformer, TTN, SelfPAB, DIN, and CNN-MHA, all of which are designed to capture advanced temporal and spatial relationships in human activity data using state-of-the-art attention and transformer mechanisms:

- ConvTransformer [10]: CNN, Transformer, attention mechanism for spatial features extraction, temporal correlations, and relevant weighting, respectively.
- TTN [42]: self-attention two-stream (temporal and spatial stream) Transformer network.
- SelfPAB [43]: pre-trained transformer encoder network.
- DIN [44]: dual-branch interactive network using CNNs with Transformer.
- CNN-MHA [45]: CNN architecture incorporating a multi-head attention mechanism.

Table 28 Summary of empirical complexity analysis across different datasets

Dataset	Sequence Length	Feature Dimensions	Total Parameters	Memory Footprint	Inference Time
PAMAP2	32	36	176,012	687.55 KB	30 ms
MotionSense	72	1	176,838	690.77 KB	28 ms
Opportunity	300	108	252,037	984.52 KB	40 ms

For the Federated Learning category, models such as UniHAR, FLAME, and ProtoHAR represent the latest approaches for privacy-preserving, decentralized HAR with a focus on handling data heterogeneity and communication efficiency:

- UniHAR [39]: self-supervised and federated learning techniques to train a generalized feature extraction model.
- FLAME [46]: federated learning solution for multi-device environments.
- ProtoHAR [47]: a novel FL framework that leverages activity prototypes to address the data heterogeneity challenge.

The hybrid baselines are included to ensure a comprehensive evaluation and encompass the most recent designs integrating CNNs, RNNs, Graph Neural Networks, and self-supervised learning techniques commonly cited as state-of-the-art for HAR tasks with wearable sensors:

- All-MLP [32]: lightweight network architecture that is entirely built on MLP layers.
- CAPSLSTM [41]: single capsule layer to extract spatial features and an LSTM layer to capture temporal dependencies.
- HAR-DeepConLG [28]: different blocks of CNN, LSTM, and GRU to extract spatial features.
- ConvAE-LSTM [29]: different blocks of CNN, AE, and LSTM layers for feature extraction, dimensionality reduction, and temporal modeling.
- Contrastive Supervision [30]: Generalization of contrastive learning in an intensely supervised setting, which aims to learn time series augmentation invariances.
- Att-ResBiGRU [31]: different blocks of CNN, BiGRU, and attention for extracting spatial and time-series features and optimizing the final recognition features' efficiency.
- 1D-CNN-BiLSTM [35]: 1D-CNN and BiLSTM for extracting features and capturing sequential dependencies.
- FFT/WVT-CNN [36]: 2-D FFT/WVT(Wigner-Ville Transform) and VGG-16 CNN.
- SS-HAR [37]: deep CNN in a self-supervised manner.
- HAR-CT [38]: deep neural network with convolutional layers (CNN), followed by network compression using Ternary Weights Network (TWN).
- GE-EnsemCNN-HAR [40]: Knowledge Graphs and Graph Convolutional Networks for feature engineering and ensemble CNN to capture hierarchical features.
- AReNet [48]: parallel blocks of 1D CNN and a fusion layer that aggregates the extracted features using a maximum operation, along with a cascade learning approach.
- 1D-CNN HAR [49]: a 1D-CNN-based deep learning model, with optimized hyperparameters and architecture.
- ICGNet [50]: a hybrid Inception-inspired CNN-GRU network which can automatically extract local features and capture long-term dependencies.

The comparison results of MHAT-FL with other baselines using the PAMAP2 dataset are presented in Table 29. MHAT-FL achieves the highest accuracy of 0.9952 among all baselines using FedAvg.

Table 29 Comparison of MHAT-FL with other baselines using PAMAP2 dataset

Method	F1-Score
ConvTransformer [10]	0.9900
HAR-DeepConLG [28]	0.9789
Convae-lstm [29]	0.9446
Contrastive Supervision [30]	0.9297
Att-ResBiGRU [31]	0.9644
all-MLP [32]	0.9199
TTN [42]	0.9800
SelfPAB [43]	0.8559
DIN [44]	0.9205
CNN-MHA [45]	0.9507
FLAME [46]	0.5300
ProtoHAR [47]	0.8733
AReNet [48]	0.9920
1D-CNN HAR [49]	0.9200
ICGNet [50]	0.9762
MHAT-FL: FedAvg	**0.9952**
MHAT-FL: FedPer	**0.9944**

The comparison results of MHAT-FL with other baselines using the MotionSense dataset are presented in Table 30. MHAT-FL achieves the highest accuracy of 0.9911 using FedPer among all baselines.

The comparison results of MHAT-FL with other baselines using the Opportunity dataset are presented in Table 31. MHAT-FL achieves the highest accuracy of 0.9979 among all baselines using FedAvg.

As mentioned in the above tables, MHAT-FL outperformed other baselines based on F1-Score metrics. Positional encoding in the attention matrix enhances the model's ability to understand sequential information in the data, aiding in activity recognition and motion measurement. Indeed, the positional encoding in the Transformer architecture of MHAT-FL helps the model better understand the temporal dynamics of the data, thereby improving its performance in activity detection. The use of Federated Learning methods, particularly FedAvg and FedPer, improves the model's performance by enabling training on a more extensive and diverse dataset, leading to better generalization and performance.

Table 30 Comparison of MHAT-FL with other baselines using motionsense dataset

Method	F1-Score
1D-CNN-BiLSTM [35]	0.9189
FFT/WVT-CNN [36]	0.9206
SS-HAR [37]	0.9304
HAR-CT [38]	0.9618
UniHAR-A [39]	0.7301
GE-EnsemCNN-HAR [40]	0.9608
CAPSLSTM [41]	0.8301
MHAT-FL: FedAvg	**0.9900**
MHAT-FL: FedPer	**0.9911**

Table 31 Comparison of MHAT-FL with other baselines using opportunity dataset

Method	F1-Score
ConvTransformer [10]	0.7300
Convae-lstm [29]	0.9554
all-MLP [32]	0.9175
Ullah, S [33]	0.9561
Xu, H [34]	0.9604
TTN [42]	0.6900
SelfPAB [43]	0.8616
DIN [44]	0.9155
FLAME [46]	0.5705
AReNet [48]	0.9920
MHAT-FL: FedAvg	**0.9979**
MHAT-FL: FedPer	**0.9979**

5 Conclusion

The research area of Human Activity Recognition (HAR) focuses on developing automatic techniques to identify daily life activities from sensor signals. HAR enables detailed monitoring and analysis of daily activities using modern smartphones equipped with sensors such as Accelerometers and Gyroscopes. This technology holds great potential in various fields, including healthcare, fitness, security, and smart environments.

The proposed approach in this research, MHAT-FL, developed a novel Transformer architecture with positional encoding on the attention matrix in each attention head in a Federated Learning framework. Federated Learning in MHAT-FL preserves data privacy, enhances communication efficiency, and addresses the issue of data heterogeneity by utilizing FedAvg and FedPer techniques. Utilizing the Transformer architecture on the client and server sides, which is equipped with positional encoding on each attention head, achieves more accurate human activity recognition and improves training time. To assess the performance of MHAT-FL, a comprehensive evaluation using various metrics was conducted on three public datasets of different sizes, including PAMAP2, MotionSense, and Opportunity. The evaluation results indicate that MHAT-FL outperforms recent state-of-the-art methods in the HAR scope.

While the proposed approach demonstrates promising improvements in HAR, it is important to acknowledge several potential limitations. Transformer-based models, although powerful, typically introduce higher computational overhead compared to lightweight architectures. In addition, Federated Learning in heterogeneous environments may encounter convergence issues due to device capabilities, data distributions, and intermittent connectivity, which can lead to complications. Finally, this work assumes a baseline level of data quality and constant sensor availability. In practice, noisy sensor signals or missing data can be handled before processing. In this regard, several promising directions can be pursued in future research:

- Investigate and analyze various positional encoding methods to assess their effectiveness in enhancing the performance of the Transformer architecture in a Federated Learning environment.

- Explore the potential of using adaptive positional encoding methods to dynamically adjust the encoding based on data distribution or learn an adaptive encoding function during training.
- Develop alignment strategies or interpolation techniques to address positional encoding misalignment across different clients in Federated Learning settings due to data heterogeneity. This could involve alignment-based positional encoding or learning shared positional embeddings.
- Integrate and evaluate advanced privacy-preserving mechanisms, such as Differential Privacy, secure aggregation, or sophisticated noise injection techniques, within the federated learning pipeline, aiming to strengthen resistance against membership inference and other privacy attacks while preserving model utility.

Author contributions Ariaeimehr: Concept, Methodology, Implementation, and Evaluation. Ravanmehr: Concept, Supervision, Validation, and Editing.

Funding information Not Applicable.

Data availability Data used in this research are public datasets.

Declarations

Competing interests The authors have no conflict of interest to declare.

Informed consent Not Applicable.

References

1. Dentamaro V, Gattulli V, Impedovo D, Manca F (2024) Human activity recognition with Smartphone-Integrated sensors: a survey. Expert Syst Appl 246:123143. https://doi.org/10.1016/j.eswa.2024.123143
2. Islam MM, Nooruddin S, Karray F, Muhammad G (2023) Multi-level feature fusion for multimodal human activity recognition in internet of healthcare things. Inf Fusion 94:17–31. https://doi.org/10.1016/j.inffus.2023.01.015
3. Bodhe R, Sivakumar S, Sakarkar G et al (2024) Outdoor activity classification using smartphone-based inertial sensor measurements. Multimedia Tools Appl. https://doi.org/10.1007/s11042-024-18599-w
4. Bassani G, Filippeschi A, Avizzano CA (2021) A dataset of human motion and muscular activities in manual material handling tasks for biomechanical and ergonomic analyses. IEEE Sens J 21(21):24731–24739. https://doi.org/10.1109/JSEN.2021.3113123
5. Hussain Z, Sheng QZ, Zhang WE (2020) A review and categorization of techniques on device-free human activity recognition. J Netw Comput Appl 167:102738. https://doi.org/10.1016/j.jnca.2020.102738
6. Yu H, Chen Z, Zhang X, Chen X, Zhuang F, Xiong H, Cheng X (2021) Fedhar: Semi-supervised online learning for personalized federated human activity recognition. IEEE Trans Mob Comput 22(6):3318–3332. https://doi.org/10.1109/TMC.2021.3136853
7. Presotto R, Civitarese G, Bettini C (2022) Semi-supervised and personalized federated activity recognition based on active learning and label propagation. Pers Ubiquit Comput 26(5):1281–1298. https://doi.org/10.1007/s00779-022-01688-8
8. Xiao Z, Xu X, Xing H, Song F, Wang X, Zhao B (2021) A federated learning system with enhanced feature extraction for human activity recognition. Knowl-Based Syst 229:107338. https://doi.org/10.1016/j.knosys.2021.107338
9. Pramanik R, Sikdar R, Sarkar R (2023) Transformer-based deep reverse attention network for multi-sensory human activity recognition. Eng Appl Artif Intell 122:106150. https://doi.org/10.1016/j.engappai.2023.106150

10. Zhang Z, Wang W, An A, Qin Y, Yang F (2023) A human activity recognition method using wearable sensors based on convtransformer model. Evol Syst 14:939–955. https://doi.org/10.1007/s12530-022-09480-y
11. Xiao Z, Tong H, Qu R, Xing H, Luo S, Zhu Z, Song F, Feng L (2023) CapMatch: semi-supervised contrastive transformer capsule with feature-based knowledge distillation for human activity recognition. IEEE Trans Neural Networks Learn Syst 1–15. https://doi.org/10.1109/TNNLS.2023.3344294
12. Foumani NM, Tan CW, Webb GI, Salehi M (2024) Improving position encoding of transformers for multivariate time series classification. Data Min Knowl Discov 38:22–48. https://doi.org/10.1007/s10618-023-00948-2
13. Pareek G, Nigam S, Singh R (2024) Modeling transformer architecture with attention layer for human activity recognition. Neural Comput Appl 36:5515–5528. https://doi.org/10.1007/s00521-023-09362-7
14. Gu M, Chen Z, Chen K et al (2024) RMPCT-Net: a multi-channel parallel CNN and transformer network model applied to HAR using FMCW radar. SIViP 18:2219–2229. https://doi.org/10.1007/s11760-023-02894-4
15. Lee TH, Kim H, Lee D (2023) Transformer-based early classification for real-time human activity recognition in smart homes. Proceedings of the 38th ACM/SIGAPP symposium on applied computing:410–417. https://doi.org/10.1145/3555776.3577693
16. Yang Z, Li Y, Zhou G (2024) Unsupervised sensor-based continuous authentication with low-rank transformer using learning-to-rank algorithms. IEEE Trans Mob Comput 1–17. https://doi.org/10.1109/TMC.2024.3353209
17. Kim YW, Cho WH, Kim KS, Lee S (2022) Inertial-measurement-unit-based novel human activity recognition algorithm using conformer. Sensors (Basel) 22(10):3932. https://doi.org/10.3390/s22103932
18. Gao L, Konomi SI (2023) Personalized federated human activity recognition through semi-supervised learning and enhanced representation. Adjunct proceedings of the 2023 ACM international joint conference on pervasive and ubiquitous computing & the 2023 ACM international symposium on wearable computing 463–468. https://doi.org/10.1145/3594739.3610739
19. Pham CH, Huynh-The T, Sedgh-Gooya E, El-Bouz M, Alfalou A (2024) Extension of physical activity recognition with 3D CNN using encrypted multiple sensory data to federated learning based on multi-key homomorphic encryption. Comput Methods Programs Biomed 243:107854. https://doi.org/10.1016/j.cmpb.2023.107854
20. Wang P, Ouyang T, Wu Q, Huang Q, Gong J, Chen X (2024) Hydra: hybrid-model federated learning for human activity recognition in heterogeneous devices. J Syst Architect 147:103052. https://doi.org/10.1016/j.sysarc.2023.103052
21. Bu C, Zhang L, Cui H, Cheng D, Wu H, Song A (2024) Learn from others and be yourself in federated human activity recognition via attention-based pairwise collaborations. IEEE Trans Instrum Meas. https://doi.org/10.1109/TIM.2024.3351260
22. Li Y, Qin X, Geng J, Chen R, Hou Y, Gong Y, Pan M, Zhang P (2024) REWAFL: residual energy and wireless aware participant selection for efficient federated learning over mobile devices. IEEE Trans Mob Comput 1–15. https://doi.org/10.1109/TMC.2024.3365477
23. Chai Y, Liu H, Zhu H, Pan Y, Zhou A, Liu H, Liu J, Qian Y (2024) A profile similarity-based personalized federated learning method for wearable sensor-based human activity recognition. Inf Manage 103922. https://doi.org/10.1016/j.im.2024.103922
24. Park J, Lee K, Lee S, Zhang M, Ko J (2023) Attfl: a personalized federated learning framework for time-series mobile and embedded sensor data processing. Proc ACM Interact Mob Wearable Ubiquitous Technol 7(3):1–31. https://doi.org/10.1145/3610917
25. Reiss A, Stricker D (2012) Introducing a new benchmarked dataset for activity monitoring. 16th international symposium on wearable computers - IEEE 108–109. https://doi.org/10.1109/ISWC.2012.13
26. Malekzadeh M, Clegg RG, Cavallaro A, Haddadi H (2019) Mobile sensor data anonymization. Proceedings of the international conference on internet of things design and implementation 49–58. https://doi.org/10.1145/3302505.3310068
27. Chavarriaga R, Sagha H, Calatroni A, Digumarti ST, Tröster G, del Millán R, Roggen JD (2013) The opportunity challenge: a benchmark database for on-body sensor-based activity recognition. Pattern Recognit Lett 34(15):2033–2042. https://doi.org/10.1016/j.patrec.2012.12.014
28. Ding W, Abdel-Basset M, Mohamed R (2023) HAR-DeepConvLG: hybrid deep learning-based model for human activity recognition in IoT applications. Inf Sci 646:119394. https://doi.org/10.1016/j.ins.2023.119394
29. Thakur D, Biswas S, Ho ES, Chattopadhyay S (2022) Convae-lstm: convolutional autoencoder long short-term memory network for smartphone-based human activity recognition. IEEE Access 10:4137–4156. https://doi.org/10.1109/ACCESS.2022.3140373

30. Cheng D, Zhang L, Bu C, Wu H, Song A (2023) Learning hierarchical time series data augmentation invariances via contrastive supervision for human activity recognition. Knowl-Based Syst 276:110789. https://doi.org/10.1016/j.knosys.2023.110789
31. Mekruksavanich S, Jitpattanakul A (2024) Device position-independent human activity recognition with wearable sensors using deep neural networks. Appl Sci 14(5):2107. https://doi.org/10.3390/app14052107
32. Wang S, Zhang L, Wang X, Huang W, Wu H, Song A (2024) PatchHAR: a MLP-like architecture for efficient activity recognition using wearables. IEEE Trans Biometrics Behav Identity Sci 6(2):169–181. https://doi.org/10.1109/TBIOM.2024.3354261
33. Ullah S, Pirahandeh M, Kim DH (2024) Self-attention deep ConvLSTM with sparse-learned channel dependencies for wearable sensor-based human activity recognition. Neurocomputing 571:127157. https://doi.org/10.1016/j.neucom.2023.127157
34. Xu H, Li J, Yuan H, Liu Q, Fan S, Li T, Sun X (2020) Human activity recognition based on gramian angular field and deep convolutional neural network. IEEE Access 8:199393–199405. https://doi.org/10.1109/ACCESS.2020.3032699
35. Lowe YJ, Lee CP, Lim KM (2022) Wearable sensor-based human activity recognition with hybrid deep learning model. Inf MDPI 9(3):56. https://doi.org/10.3390/informatics9030056
36. Zebhi S (2022) Human activity recognition using wearable sensors based on image classification. IEEE Sens J 22(12):12117–12126. https://doi.org/10.1109/JSEN.2022.3174280
37. Rahimi Taghanaki S, Rainbow MJ, Etemad A (2021) Self-supervised human activity recognition by learning to predict cross-dimensional motion. Proceedings of the 2021 ACM international symposium on wearable computers 23–27. https://doi.org/10.1145/3460421.3480417
38. Jaberi M, Ravanmehr R (2022) Human activity recognition via wearable devices using enhanced ternary weight convolutional neural network. Pervasive Mob Comput 83:101620. https://doi.org/10.1016/j.pmcj.2022.101620
39. Xu H, Zhou P, Tan R, Li M (2023) Practically adopting human activity recognition. Proceedings of the 29th annual international conference on mobile computing and networking 1–15. https://doi.org/10.1145/3570361.3613299
40. Ghalan M, Aggarwal RK (2024) Novel human activity recognition by graph engineered ensemble deep learning model. IFAC J Syst Control 27:100253. https://doi.org/10.1016/j.ifacsc.2024.100253
41. Khan P, Kumar Y, Kumar S (2022) CapsLSTM-based human activity recognition for smart healthcare with scarce labeled data. IEEE Trans Comput Social Syst 11(1). https://doi.org/10.1109/TCSS.2022.3223343
42. Xiao S, Wang S, Huang Z, Wang Y, Jiang H (2022) Two-stream transformer network for sensor-based human activity recognition. Neurocomputing 512:253–268. https://doi.org/10.1016/j.neucom.2022.09.099
43. Logacjov A, Herland S, Ustad A, Bach K (2024) SelfPAB: large-scale pre-training on accelerometer data for human activity recognition. Appl Intell 54(6):4545–4563. https://doi.org/10.1007/s10489-024-05322-3
44. Tang Y, Zhang L, Wu H, He J, Song A (2022) Dual-branch interactive networks on multichannel time series for human activity recognition. IEEE J Biomed Health Inf 26(10):5223–5234. https://doi.org/10.1109/JBHI.2022.3193148
45. Tan TH, Chang YL, Wu JR, Chen YF, Alkhaleefah M (2023) Convolutional neural network with multi-head attention for human activity recognition. IEEE Internet Things J 11(2):3032–3043. https://doi.org/10.1109/JIOT.2023.3294421
46. Cho H, Mathur A, Kawsar F (2022) Flame: federated learning across multi-device environments. Proc ACM Interact Mob Wearable Ubiquitous Technol 6(3):1–29. https://doi.org/10.1145/3550289
47. Cheng D, Zhang L, Bu C, Wang X, Wu H, Song A (2023) Protohar: prototype guided personalized federated learning for human activity recognition. IEEE J Biomed Health Inf 27(8):3900–3911. https://doi.org/10.1109/JBHI.2023.3275438
48. Boudjema A, Titouna F, Titouna C (2024) AReNet: cascade learning of multibranch convolutional neural networks for human activity recognition. Multimedia Tools Appl 83(17):51099–51128. https://doi.org/10.1007/s11042-023-17496-y
49. Kaya Y, Topuz EK (2024) Human activity recognition from multiple sensors data using deep CNNs. Multimedia Tools Appl 83(4):10815–10838. https://doi.org/10.1007/s11042-023-15830-y
50. Dua N, Singh SN, Semwal VB, Challa SK (2023) Inception inspired CNN-GRU hybrid network for human activity recognition. Multimedia Tools Appl 82(4):5369–5403. https://doi.org/10.1007/s11042-021-11885-x
51. Abbasi Tadi A, Dayal S, Alhadidi D, Mohammed N (2023) Comparative analysis of membership inference attacks in federated and centralized learning. Information 14(11):620. https://doi.org/10.3390/info14110620

52. Bai L, Hu H, Ye Q, Li H, Wang L, Xu J (2024) Membership inference attacks and defenses in federated learning: a survey. ACM-CSUR 57(4):Article89. https://doi.org/10.1145/3704633
57. Li W, Fan K, Zhang J, Li H, Lim WYB, Yang Q (2025) Enhancing security and privacy in federated learning using low-dimensional update representation and proximity-based defense. IEEE Trans Knowledge Data Eng 37(6): 3372–3385 https://doi.org/10.1109/TKDE.2025.3539717

Mohammad Ariaeimehr received his B.Sc. degree in Electronic Technology Engineering from the Marand Branch of Islamic Azad University, Iran, in 2014, and his M.Sc. degree in Software Engineering from the Central Tehran Branch of Islamic Azad University, Iran, in 2025. His research interests include Human activity recognition, transformer-based and deep learning architectures, positional encoding techniques, and federated learning frameworks, with applications in healthcare monitoring, biomedical signal analysis, aerospace mission support, and intelligent autonomous systems.

Reza Ravanmehr graduated in computer engineering from Shahid Beheshti University, Tehran, in 1996. After that, he gained his M.Sc. and Ph.D. degrees, both in computer engineering, from IAU, Science and Research Branch, Tehran, in 1999 and 2004, respectively. His main research interests are social network analysis, distributed/parallel systems, and large-scale data management systems. He is currently an Associate Professor in the Department of Computer Engineering at the Central Tehran Branch, IAU.