WiTS：A Wi-Fi-Based Human Action Recognition via Spatio-Temporal Hybrid Neural Network

Pengcheng Gao*
School of Cyber Security, Gansu University of Political Science and Law, Lanzhou 730070, Gansu, China

Abstract—**Human action recognition has many applications in different scenarios. With the advancement of wireless sensing and the widespread deployment of Wi-Fi devices, the perception technology of Wi-Fi channel state information (CSI) has shown great potential. Related studies identified actions by capturing specific attenuation and distortion features caused by human posture on CSI. These methods are less susceptible to the effects of lighting and object occlusion. However, they have yet to adequately extract information within CSI. The challenge of enhancing model performance through the comprehensive utilization of information features within different dimensions remains an imperative area. To address this, a spatio-temporal hybrid neural network model named WiTS is proposed. It integrates the advantages of different neural networks, using CNN to extract spatial features, combining TCN and Bi-LSTM for dual temporal dimension modeling, and incorporating Transformer's global attention mechanism to achieve comprehensive extraction and multi-level fusion of spatio-temporal features. Additionally, this study further optimizes the original WiTS model from three aspects. The Experiment on WiAR and CSIAR datasets show that the model achieves average accuracy rates of 95.75% and 96.71%, respectively, with F1-scores exceeding 96%. The model has only 2.19 million parameters and less than 560 million FLOPs, offering significant advantages in terms of lightweight design, making it suitable for deployment on limited-computing edge terminals while meeting real-time requirements.**

Keywords—Wi-Fi CSI; human action recognition; deep learning

I. INTRODUCTION

The advent of wireless sensing technology has led to significant advancements in perception technology based on Wi-Fi CSI, which has demonstrated considerable potential. It can be applied to many fields, with the most notable being human action recognition [1-2]. For instance, in the field of smart homes, wireless sensing could recognize users' hand gestures to control electronic devices. In the gaming field, action capture can be used to manipulate characters in real time to enhance the gaming experience. In the field of security, CSI changes in monitored areas can be captured to detect anomalies. In the medical and elderly care fields, wireless sensing technology can be applied for respiratory monitoring, fall detection, etc. [3]. A Wi-Fi-based human action recognition method has been developed that can detect human movements in a signal coverage environment. It utilizes sensors to collect data such as CSI and received signal strength (RSS). This method is distinct from both visual perception methods and those that employ wearable devices for perception, and it possesses three primary advantages. Firstly, the Wi-Fi-based perception is unaffected by lighting conditions and can be used in any lighting environment. Secondly, Wi-Fi signals are less susceptible to obstruction by objects and can penetrate obstacles such as walls, furniture, and appliances, enabling accurate recognition of human movements behind obstructions. Thirdly, unlike visual data collection methods, this approach abstracts human movements by identifying the specific disturbances they cause to signals, offering inherent advantages in terms of privacy protection and making it suitable for deployment in private spaces such as bedrooms and bathrooms [4]. At present, methods for human action recognition using Wi-Fi sensing could be broadly classified into two categories: machine learning methods and deep learning methods. The content below will introduce pertinent research on action recognition based on these two methods.

In the field of machine learning-based methods, numerous researchers have explored this domain. As posited by He et al. [5], the WiG system was the inaugural system to utilize CSI for this task. The Birge-Massart filter was utilized to denoise the signal, thereby preserving significant information. The Local Outlier Factor and Support Vector Machine (SVM) were employed to extract feature data and perform action recognition. In a related study, Zhang et al. [6] extracted domain-independent gesture features at lower signal levels. They proposed a general method applicable to different environments that effectively achieves cross-domain recognition. Xian et al. [7] modified the K-means algorithm to facilitate the recognition of fine-grained actions. Dang et al. [8] combined the Dynamic Time Warping (DTW) algorithm and SVM to match and recognize different actions, effectively improving recognition accuracy. The amplitude and phase difference of subcarrier levels in wireless signals was found to be correlated with human actions by Hao et al. [9]. They combined K-means and Bagging algorithms to optimize SVM, achieving the recognition task at a low computational cost. Huang et al. [10] discovered that Nonlinear Phase Error Variation (NLPEV) data in CSI exhibits good stability and sensitivity to actions, and utilized it to achieve effective HAR in Co-channel Interference (CCI) scenarios. Chelli et al. [11] developed a machine learning framework for action recognition based on average Doppler shift, using KNN for action classification, achieving good action recognition performance. Cheng et al. [12] utilized CSI phase differences to construct an extended matrix for action feature extraction and employed a Gaussian mixture-hidden Markov model to identify CSI feature data. This approach has been shown to reduce system computation time and improve the fault tolerance rate of segmentation.

In contrast to machine learning, which necessitates the laborious process of manual feature extraction, methods based on deep learning have undergone rapid development in recent years. Zou et al. [13] employed CNNs to extract the most significant features from signals for the purpose of action recognition. Muaaz et al. [14] proposed a system named Wi-Sense, which employs CSI ratio methods to mitigate noise and phase shift effects. This is followed by the generation of spectral maps from preprocessed data and the training of a CNN model using spectral images. Huan et al. [15] proposed a novel activity segmentation method utilizing signal variance differences between action and non-action segments to achieve a balance between robustness and property. Duan et al. [16] proposed a sorting algorithm based on subcarrier correlation and inversion to extract different user signals. The authors combined a bidirectional GRU with an attention mechanism and a convolutional neural network to achieve multi-user action recognition. Meng et al. [17] employed a sparse recovery approach for identifying the primary paths affected by human activities, and constructed a matrix based on the phase differences between adjacent antennas. Thereafter, they proposed a bidirectional GRU structure with an attention mechanism for automatic learning and feature extraction from the matrix. Gao et al. [18] developed a geometric representation of different human hand gestures and used a backtracking search algorithm to recognize them. Cui et al. [19] employed CNN to extract features between different subcarriers and proposed an integrated architecture comprising multiple layers of perception, random forests, and SVM to improve recognition accuracy.

In summary, there are several challenges associated with using machine learning methods for action recognition. Manual feature extraction requires significant labor costs and is subject to the loss of implicit yet crucial information, which complicates the effective differentiation of similar actions. Furthermore, the efficacy of this method is contingent upon the availability of high-quality data; in the absence of such data, the recognition accuracy cannot attain a high level. While deep learning methods have demonstrated efficacy in the extraction of CSI, existing approaches generally utilize spatial features in signals, failing to effectively extract features from the temporal domain. Therefore, developing a model which could parse the rich radio signal strength and phase variation information contained in CSI signals to extract features from both the temporal and spatial dimensions remains an urgent problem to be solved. To address this issue, this study proposes a Wi-Fi-based human action recognition model using a spatio-temporal hybrid neural network. The primary contributions of this paper are as follows:

1) A hybrid architecture model, designated as WiTS, was developed, incorporating TCN, CNN, Bi-LSTM, and Transformer networks, which can effectively extract and fuse the temporal and spatial feature information in Wi-Fi CSI to achieve high-precision human action recognition.

2) The structure of the CNN module was optimized by using a spatial attention block to enhance important spatial locations and replacing traditional convolutions with partial convolutions [20] to reduce parameters.

3) An attention mechanism was employed into the Bi-LSTM to enhance the sensitivity and expressive capability to key information segments.

4) The Sparrow Search Algorithm (SSA) [21] was employed for hyperparameter optimization, further enhancing the performance of model.

The rest part is divided into five sections: Section II introduces related research. Section III provides a comprehensive illustration of the improved methodology. Section IV chiefly presents the experimental configuration and dataset. Section V shows the results of ablation experiments and comparative experiments. Section VI summarizes the entire work.

II. Related Works

To address the issues previously mentioned regarding Wi-Fi CSI in human action recognition and achieve more comprehensive extraction of features, this paper utilizes four deep learning models, CNN, TCN, Bi-LSTM, and Transformer, to construct a hybrid architecture. The subsequent content will offer a concise overview of these seminal networks.

Convolutional Neural Network (CNN) is a type of deep learning model that has been specifically developed for processing grid-like information, which includes convolution, pooling, and fully connected layers. Convolution layers obtain features through local receptive fields and weight sharing. Pooling layers like max pooling and average pooling reduce data dimensions and enhance translation invariance. Fully connected layers are employed for final classification or regression tasks [22]. The advantage of CNNs lies in their hierarchical feature learning capability, where shallow convolutions capture low-level features, while deep networks extract high-level semantic features [23]. In recent years, CNNs have achieved significant progress in image recognition, object detection and other domains, thus becoming an integral component of deep learning research.

Temporal Convolutional Network (TCN) is a network to process time series data. The main structure of TCN principally comprises causal convolution, dilated convolution, and residual connection structure. It ensures temporal dependency through causal convolution; combines dilated convolution to expand the receptive field, thereby effectively capturing long-term dependencies; and introduces residual connections to moderate the training difficulties of deep networks [24]. Compared to traditional recurrent neural networks (RNNs), TCN has faster training speeds and more stable gradients, and they perform exceptionally well in various temporal tasks [25].

Long Short-Term Memory (LSTM) is a type of RNN to address the long-term dependency issue inherent in general RNNs [26]. LSTM introduces three gating mechanisms and the concept of a cell state, in which information can be added, deleted, or modified. The forget gate decides which information will be removed from the cell state. The input gate decides which new information will be added to it. The function of the output gate is to decide which information will be output from it. During the update process, the cell state is primarily updated through element-wise multiplication and addition, and gradients are linearly propagated along the time axis during backpropagation. This enables LSTM to reliably learn long-term dependencies in sequences and effectively mitigate the gradient vanishing or exploding issues that plague traditional RNNs. Bidirectional Long Short-Term Memory (Bi-LSTM) was proposed by Mike Schuster and Kuldip K. Paliwal [27]. This network consists of a forward LSTM and a backward LSTM connected in parallel. The forward layer reads the sequence in time order to obtain the preceding information, while the backward layer reads the sequence in reverse time order to obtain the subsequent information. This allows the network to use both past and future context, furthering the accuracy of tasks such as classification and prediction.

The Transformer model was first proposed by Vaswani et al. [28] in 2017 and completely varied the paradigm of natural language processing. In contrast to conventional RNN and CNN, the Transformer utilizes attention mechanisms as its primary approach to capture global dependencies in input sequences. This enables parallel computing and facilitates more efficient modeling of long-range dependencies. The core structure includes multi-head attention and position-wise feed-forward networks, and it also incorporates residual connections and layer normalization to optimize the training process. In contemporary applications, the Transformer has been employed not only in natural language processing but also in a variety of downstream tasks, including computer vision, speech processing, and multimodal learning through hybrid architectures that integrate the Transformer with other models [29].

III. Improved Methodology

This paper develops a hybrid architecture network named WiTS based on CNN, TCN, Bi-LSTM, and Transformer to fuse temporal and spatial feature information. The network structure is shown in Fig. 1.

For spatial feature extraction, a CNN branch is utilized to construct a feature encoder that captures spatial local correlations of the input data through local connections and weight sharing. For temporal feature extraction, the temporal encoder is constructed using dilated causal convolutions from TCN to leverage its exponentially increasing receptive field and capture temporal sequence features. Simultaneously, a Bi-LSTM branch utilizes its gated memory mechanism to explicitly model short-term to long-term dependencies in the temporal dimension. This complements TCN's feature extraction and forms a dual-temporal dimension modeling approach that combines parallel capture with recursive memory. To fully examine the deep correlations between different features, the spatio-temporal features extracted by the three branches are integrated and import to the Transformer that uses its self-attention mechanism to establish global dependencies. This architecture complements the spatial feature extraction capabilities of CNN, the temporal feature extraction capabilities of TCN and Bi-LSTM, and the global context modeling capabilities of Transformer, thereby achieving a more comprehensive and deeper feature representation of complex spatio-temporal patterns.

This paper also introduces three improvements based on this model, namely improvements to the LSTM module, improvements to the CNN module, and hyperparameter optimization. The subsequent subsections will provide a detailed introduction to the optimization content.

A. LSTM Module Improvement

In comparison with the conventional Bi-LSTM module, this model incorporates an attention mechanism, assigning distinct weights to the output of each time step of the LSTM, enabling the model to concentrate on the most critical segments in the sequence rather than simply relying on the last hidden state. Fig. 2 shows the optimized LSTM module. The calculation process of the attention mechanism is as follows.

Assume that the output of Bi-LSTM is:

$$X = [h_1, h_2, \cdots, h_T] \tag{1}$$

Where $h_t \in R^{256}$, $X \in R^{B \times T \times 256}$, and 256 is the feature dimension of each time step output of Bi-LSTM. Subsequently, the output h_t of each time step undergoes linear transformation and Tanh activation:

$$e_t = \tanh(w^T h_t + b) \tag{2}$$

Where $w \in R^{256}$ and $b \in R$.

After the transformation and activation of the output of each time step, the attention scores of all time steps are to be concatenated:

$$e = [e_1, e_2, \cdots, e_T] \in R^{B \times T} \tag{3}$$

For each sample, all time step scores are normalized using Softmax to obtain attention weights:

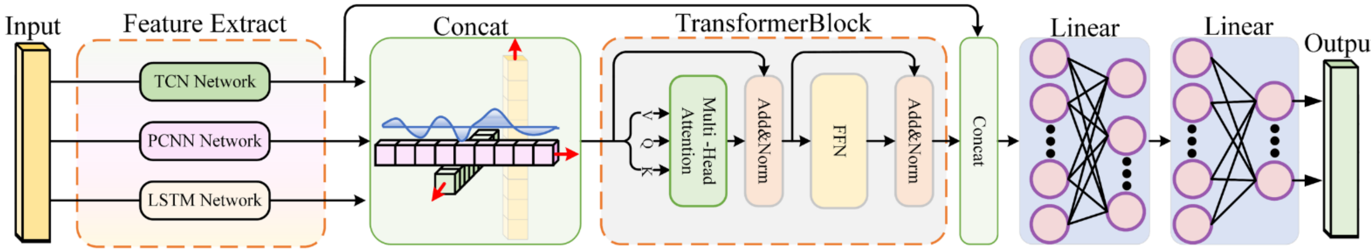

Fig. 1. The structure of WiTS.

$$\alpha_t = \frac{\exp(e_t)}{\sum_{k=1}^{T}\exp(e_k)} \quad (4)$$

where, $\alpha = [\alpha_1, \alpha_2, \cdots, \alpha_T] \in R^{B\times T}$.

Weight the total outputs of all time steps of the Bi-LSTM using attention weights to get the global feature vector of the whole sequence after attention weighting:

$$s = \sum_{t=1}^{T}\alpha_t h_t \quad (5)$$

where, $s \in R^{B\times 256}$, h_t is the output of Bi-LSTM at time step t, and α_t is the attention weight at that time step.

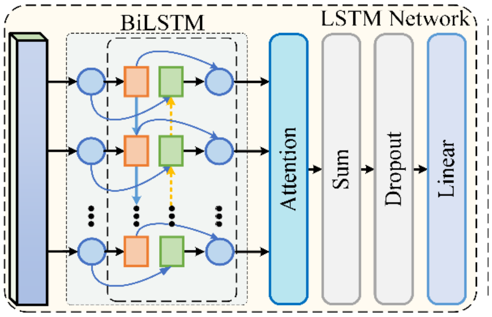

Fig. 2. The structure of optimized Bi-LSTM module.

The calculation process reveals that, in contrast to the conventional Bi-LSTM module, which utilizes the final moment as the output, the enhanced Bi-LSTM prioritizes time steps that are designated as crucial by the attention mechanism within the sequence. It also integrates information from all-time steps within the sequence, thereby offering a more comprehensive approach. This design effectively improves the ability to model long sequences and complex dependencies, and it also increases the richness and discriminative power of feature expression.

B. CNN Module Improvement

To reduce the inference time, partial convolution is utilized in CNN instead of the traditional convolution structure in CNN networks. Partial convolution was proposed by Jierun Chen et al [20]. They believe that although the information in the feature maps to be detected is gradually extracted and aggregated by the model as the depth of the convolutional neural network increases, the feature maps of different layers often contain a lot of redundant information. Consequently, they enhanced the convolution block to reduce the processing of such repetitive information. The structure of Partial Convolution is shown in Fig. 3. The primary implementation method involves the combination of a $k \times k$ convolution with c_p channels and a 1×1 convolution with $c - c_p$ channels to form a hammer-shaped convolution structure, which replaces the traditional $k \times k$ convolution structure with c channels. In this convolution form, only a portion of the channels utilize $k \times k$ convolution, while the remaining portions are processed using 1×1 convolution kernels. This approach has been demonstrated to markedly reduce parameters necessary for convolution operations, thereby enabling more efficient feature extraction. Additionally, it allows for the adjustment of the proportion of two types of convolutions to minimize potential loss of useful feature information.

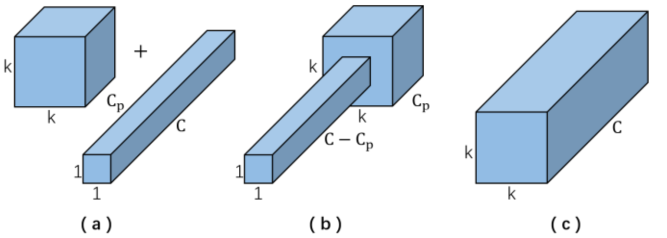

Fig. 3. (a) Convolutional variants; (b) A convolutional framework consisted of one PConv, and one 1*1 Conv; (c) One regular convolutional structure.

To enhance the weights of important spatial locations in the input feature map while suppressing unimportant regions, a spatial attention mechanism module was added to the CNN module, whose structure is shown on the left part of Fig. 4. The primary working principle is as follows: First, the average and maximum values are calculated for the input feature map along the channel dimension, yielding two single-channel feature maps. Second, these two maps are concatenated to form a two-channel feature map. Third, a convolutional layer is utilized to reduce the two-channel feature map to a one-channel feature map, thereby obtaining a spatial attention weight map. The weights are then normalized to the range of 0 to 1 using a Sigmoid activation function. This weight map is then multiplied by the original feature map to enhance important spatial locations.

This improved CNN module is called the PCNN module, and its structure is shown on the right part of Fig. 4.

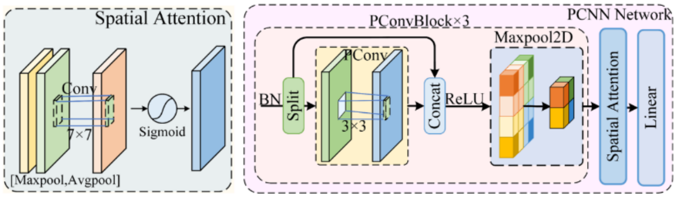

Fig. 4. The structure of PCNN module.

C. Hyperparameter Optimization

The SSA was utilized for the purpose of hyperparameter optimization, with the objective of further enhancing the model's detection capabilities. SSA is a swarm intelligence optimization method inspired by sparrows' foraging and anti-predation behaviors. The algorithm simulates the collaboration and division of labor among sparrows during the process of searching for food and evading predators to solve complex optimization problems. The algorithm first randomly generates a set of sparrow individuals, with each individual representing a solution. The classification of sparrows is typically divided into three distinct categories: The roles of producer, scrounger, and scout are delineated. Producers are responsible for global search and guiding the population's movement. Scroungers follow Producers and perform local search. Scouts are tasked with escaping local optima. After each iteration, the fitness of each sparrow (i.e. the quality of the solution) is evaluated, and roles and positions are adjusted accordingly. This process is repeated iteratively until the maximum number of iterations is reached or convergence conditions are met. The derivation process of the sparrow search algorithm is illustrated below.

Initialization of the population is the first step. Assuming a population size of N, the position of each sparrow is represented as $X_i = (X_{i1}, X_{i2}, \cdots X_{id})$, where d is the dimension of the problem. The initialization formula is as follows:

$$X_i^0 = LB + rand() \times (UB - LB) \quad (6)$$

In this formula, LB and UB represent the lower and upper bounds of the variable, respectively, and $rand()$ denotes a random number within the interval [0,1].

The number of producers is calculated as $PD \times N$, where PD is typically set within the range of 0.2 to 0.3. The position update formula is as follows:

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t \cdot \exp\left(\frac{-i}{\alpha \cdot T}\right), R_2 < ST \\ X_{i,j}^t + Q \cdot L, \quad R_2 \geq ST \end{cases} \quad (7)$$

In this formula, $X_{i,j}^t$ denotes the position of the ith sparrow in the jth dimension at generation t. The constant α is typically equal to 1, T signifies the maximum iterations, R_2 is a number in the range [0, 1], ST is the safety threshold, which is typically equal to 0.8, Q is a number following a standard distribution, and L is a vector of all ones. In scenarios where $R_2 < ST$, the sparrow is in a safe state and engages in a global search employing exponential decay. In scenarios where, $R_2 \geq ST$, the sparrow perceives a threat and employs Gaussian perturbation to augment its capacity to evade local optima.

The number of scroungers is $(1 - PD) \times N$, and the position update is based on Formula 8.

$$X_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{X_{worst,j}^t - X_{i,j}^t}{i^2}\right), & f_i > f_g \\ X_{i,j}^t + \left|X_{i,j}^t - X_{best,j}^t\right| \cdot \frac{A^+}{A^T \cdot A}, & f_i = f_g \end{cases} \quad (8)$$

$X_{worst,j}^t$ represents the position of the worst sparrow in the jth dimension in the tth generation, $X_{best,j}^t$ represents the position of the best sparrow in the jth dimension in the tth generation, f_i is the fitness of the ith sparrow, and f_g is the global optimal fitness. Q is a random number following a normal distribution, A is a random vector following a normal distribution, and A^+ is the Moore-Penrose generalized inverse of A. When the follower's fitness is worse than the global optimum, it moves away from the worst individual to avoid getting stuck in a local optimum. When the fitness equals the global optimum, it moves closer to the best individual to enhance development capabilities.

The scouts randomly select a portion of the scroungers, with the number being $SD \times N$. The value of SD is typically between 0.1 and 0.2. The following formula is for position updating:

$$X_{i,j}^{t+1} = X_{best,j}^t + \beta \cdot \left|X_{i,j}^t - X_{best,j}^t\right| \quad (9)$$

Among these, $X_{i,j}^{t+1}$ represents the position of the ith sentinel in the jth dimension in the $t + 1$ generation, $X_{best,j}^t$ represents the position of the globally optimal individual in the jth dimension in the tth generation, $X_{i,j}^t$ represents the position of the ith sentinel in the jth dimension in the tth generation, and β is a random number following a normal distribution. When the population becomes stuck in a local optimum, the scouts randomly jump to new positions near the optimal individual, thereby increasing the chance of discovering new optimal solutions and enhancing the diversity of the global search. After each iteration, the fitness of each sparrow is calculated, and the roles of producers, scroungers, and scouts are dynamically adjusted based on fitness rankings to enhance the algorithm's global optimization capability and convergence speed. The application of SSA to the task of optimizing deep learning model hyperparameters involves the mapping of these parameters to the SSA search space and the definition of a fitness function, thereby enabling an automatic search for the optimal combination.

Subsequent experiment results demonstrate that these three optimizations based on the original model have all had positive effects, and that this hybrid architecture significantly improves performance in human action recognition tasks compared to a single network structure.

IV. Sensing Principle and Environment Configuration

A. Wi-Fi Sensing Principle

In an indoor environment, a Wi-Fi sensing system is composed of two primary components: a Wi-Fi signal transmitter (Tx) and a Wi-Fi signal receiver (Rx). When Wi-Fi signals propagate through stationary objects within a room, such as floors, ceilings, and furniture, the angles and paths of reflection remain relatively stable. Consequently, in the absence of external interference, the CSI signals received at each time stamp remain largely unchanged. However, in environments characterized by dynamic targets, such as individuals walking or performing specific actions indoors, the position and posture of the human body undergo fluctuations at distinct temporal intervals. This causes channel interference such as amplitude attenuation and phase distortion when the Wi-Fi signal propagates to the human body, resulting in changes to the CSI received at the receiver. Channel interference caused by the same action is generally similar, while channel interference caused by different actions varies significantly [30].

Wi-Fi technology adheres to the IEEE 802.11 standard, employing OFDM to modulate signals. OFDM divides the spectrum into multiple mutually orthogonal subcarriers, which do not interfere with each other. This frequency division method enables the modulated signal to resist multipath propagation and interference, thereby improving signal robustness. Therefore, this method is widely used in the field of wireless communication. The used dataset is comprised of data collected from wireless network cards that are following this standard. Through Wi-Fi CSI signals, detailed amplitude and phase data can be obtained. This data can be used for channel state analysis to understand the multipath effects and interference conditions during signal propagation.

The process of Wi-Fi signal transmission is shown in Formula 10:

$$\vec{Y} = H\vec{X} + n \tag{10}$$

$\vec{Y}$ means the received signal vector, $\vec{X}$ means the transmitted signal vector, H denotes CSI, which describes the channel frequency response of the subcarriers between the transmitter and receiver, and n denotes the noise vector.

The CSI can be expressed by the following formula:

$$H = \begin{bmatrix} H_{1,1} & \cdots & H_{1,M} \\ \vdots & \ddots & \vdots \\ H_{N,1} & \cdots & H_{N,M} \end{bmatrix} \tag{11}$$

M represents the number of subcarriers. Typically, a larger bandwidth corresponds to a greater number of subcarriers. $H_{N,M}$ is the CSI signal transmitted by the Mth subcarrier at time N.

The CSI measurement value for the Mth subcarrier at time t can be expressed by Formula 12:

$$H_t = |H_t| e^{j\angle H_t} \tag{12}$$

$|H_t|$ represents the amplitude of the Mth subcarrier at that moment, and $\angle H_t$ represents its phase data.

CSI can be further decomposed into static and dynamic components. The former is caused by stationary objects in the environment, while the latter is caused by human movements. Therefore, CSI can also be expressed in the formula below:

$$H = \sum_{s=1}^{S} \alpha_s(f_k) e^{-j2\pi f_k \tau_s} + \sum_{d=1}^{D} \alpha_d(f_k, t) e^{-j2\pi f_k \tau_d(t)} \tag{13}$$

The first part represents the static component, where S denotes the number of static paths, $\alpha_s(f_k)$ denotes the static attenuation coefficient, and τ_s denotes the time delay of the static path. The second part represents the dynamic component, where D denotes the number of dynamic paths, $\alpha_d(f_k, t)$ denotes the dynamic attenuation coefficient, and $\tau_d(t)$ denotes the time delay of the dynamic path. Unlike the static component, the dynamic component varies with time, meaning that changes in the dynamic component occur during the period from the start of an action to its completion. Therefore, Wi-Fi-based action recognition primarily achieves action classification by identifying the characteristic changes in the dynamic components of different actions.

B. Experimental Environment and Dataset

All experiments in this research were conducted under identical conditions to eliminate the effect of environmental factors on the results. The operating environment and main parameter configurations are shown in Table I and Table II.

In this study, two public datasets, CSIAR [30] and WiAR [31], were selected to train the model and verify its recognition performance. The categories of human actions included in the datasets are shown in Table III.

The WiAR dataset was collected in a meeting room measuring 6m × 8m, with 4m between the transmitter and receiver, and a distance of 1m between the volunteer performing the actions and the transmitter. The data collection was carried out by three volunteers, each performing 16 types of coarse-grained actions, with each action repeated 30 times, resulting in a total of 1,440 samples. This study selected 10 of these actions for experimentation, including 7 upper-limb actions, 2 lower-limb actions, and 1 full-body action. The CSIAR dataset was collected in an office environment, with 3m between the transmitter and receiver. Six volunteers participated in the experiment, with each volunteer performing one activity within 20 seconds, repeated 20 times. The 6 actions in this dataset are all coarse-grained actions and are all full-body actions. Overall, the WiAR dataset primarily focuses on upper-body actions and lacks full-body actions, while the 6 actions covered by CSIAR are all full-body actions, effectively supplementing the missing types in the former and better reflecting the model's ability to recognize various human movements.

Since each action in the dataset is executed multiple times, the CSI signals obtained via the Wi-Fi transceiver contain multiple actions. To separate each action so that each sample contains only one action process, this study employs a sliding window algorithm to partition the data.

In this study, the size of the sliding window and the step size L_0 for each slide are both set to 120. Thus, each time the window moves forward by one step, a sample of length L_0 is obtained. By performing multiple consecutive slides, a series of samples of length L_0 can be obtained. The dataset used was collected using a transmitter with one antenna and a receiver with three antennas, along with a modified Intel 5300 network card. The number of subcarriers was 30. After dividing each subcarrier using the sliding window algorithm and then integrating the results, the input data for the model was obtained. The input data has a dimension of 120×30, where 120 represents the step size and 30 represents the number of subcarriers.

TABLE I. OPERATION ENVIRONMENT CONFIGURATION

Equipment category	Equipment name
CPU	16 cores Intel(R) Xeon(R) Gold 6430
GPU	RTX 4090 (24GB)
Memory	120G
Python version	3.10.8
Pytorch version	2.1.2
CUDA version	12.1

TABLE II. MAIN PARAMETER SETTINGS

Parameter name	Parameter value
Epochs	50
Batchsize	128
Learning rate	1e-4
Optimizer	Adam

TABLE III. THE INFORMATION OF DATASET

Dataset	No. of actions	Type of actions
WiAR	10	High arm wave, Horizontal arm wave, Two hands wave, Draw x, Draw tick, High throw, Toss paper, Bend, Forward kick, Side kick
CSIAR	6	Stand up, Sit down, Lie down, Walk, Run, Fall,

V. Results and Analysis

The experiments in this section consist of two parts. The first part introduces PCNN, attention mechanism, and SSA parameter optimization based on the original WiTS model to conduct ablation experiments and analyze the results. The second part selects three commonly used models and compares them with the improved WiTS model to conduct comparative experiments and analyze the results.

A. Ablation Experiment

This research sequentially adds the corresponding improvement measures and conducts ablation experiments under the same experimental conditions to verify that each improvement introduced in this study enhances the performance of the original model in human action recognition. The results of these experiments are shown in Table IV.

Model A in the table represents the original WiTS model. Model B represents a model in which the convolutional structure of the CNN module is replaced with a partial convolutional structure and a spatial attention block is incorporated. Model C represents a model in which an attention mechanism is added to the LSTM module based on Model B. The last row represents the final model with SSA parameter optimization based on Model C, i.e., the model with all three improvements. The results show that all three improvement strategies have a positive effect on model performance, with varying degrees of improvement in accuracy across the WiAR and CSIAR datasets. Compared to the original model without any improvements, the accuracy is 93.84% and 94.57%, with parameters and Flops of 2.25M and 561.33M, respectively. For Model B, which replaced some convolutions, the accuracy improved slightly in both datasets despite a decrease of 0.06M in the number of parameters and 1.91M in Flops compared to the original model. Comparing Models B and C, the attention mechanism added in Model C does not significantly increase computational complexity, yet its accuracy improved by 0.62% and 1.26%, respectively, demonstrating the effectiveness of this improvement method. The final model achieved the best performance in ablation experiments, with accuracy reaching 95.75% and 96.71% on the two datasets, respectively, representing improvements of 1.91% and 2.14% over the unmodified model, and also showing a significant improvement over Model C. This further demonstrates that SSA can iteratively identify optimal hyperparameter configurations to enhance model performance.

TABLE IV. Results of Ablation Experiments

Model	PCNN	Attention mechanism	SSA	WiAR Acc./%	CSIAR Acc./%	Param. /M	Flops/M
A	×	×	×	93.84	94.57	2.25	561.33
B	√	×	×	93.86	94.60	2.19	559.42
C	√	√	×	94.48	95.86	2.19	559.42
Final	√	√	√	95.75	96.71	2.19	559.42

TABLE V. Results of Comparative Experiments

Dataset	Model	Precision/%	Recall/%	F1-score/%	Accuracy/%
WiAR	CNN	87.15	87.04	86.89	86.84
	TCN	86.82	85.60	85.89	85.14
	Bi-LSTM	80.88	78.37	78.49	78.13
	WiTS	**96.34**	**96.13**	**96.06**	**95.75**
CSIAR	CNN	87.26	86.11	86.10	86.57
	TCN	81.65	80.20	80.32	80.54
	Bi-LSTM	79.42	79.36	79.20	79.71
	WiTS	**97.13**	**96.52**	**96.39**	**96.71**

B. Comparative Experiment

To further validate the advancement of the proposed WiTS in human action recognition methods, this study selected three different deep learning models for comparison experiments. The experimental results are shown in Table V.

From the results, WiTS outperformed other models in all performance metrics in both experiments, while CNN ranked second in all results. Compared to CNN, in the WiAR dataset, the WiTS model improved Precision, Recall, F1-score, and Accuracy by 9.19%, 9.09%, 9.17%, and 8.91%, respectively; in the CSIAR dataset, these four metrics improved by 9.87%, 10.41%, 10.29%, and 10.14%, respectively, all showing significant improvements. In the experiments using the WiAR dataset, TCN ranked third in all metrics, closely following CNN and significantly outperforming Bi-LSTM. In another set of experiments, while TCN still ranked third, its performance metrics lagged significantly behind CNN, only slightly outperforming Bi-LSTM. The results indicate that a single network structure performs inconsistently across different datasets, making it challenging to accurately recognize human actions in real-world scenarios. In contrast, the WiTS model, which achieves deep integration of spatial and temporal features, demonstrates significantly superior performance compared to other models, fully validating the model's effectiveness.

VI. Conclusion and Future Work

A. Summary of Contributions

In the field of human action recognition using Wi-Fi CSI, the extraction of features from Wi-Fi signals using deep learning methods for the classification of actions has emerged as a prominent research direction in recent years. Nevertheless, these methods have yet to adequately extract information from the signal. The challenge of fully utilizing the information capabilities from both spatial and time dimensions to enhance model performance remains an urgent issue to be solved. To address this, this paper proposes a spatio-temporal hybrid neural network model named WiTS. This model integrates the advantages of CNN, TCN, Bi-LSTM, and Transformer. It uses CNN to extract spatial features, combines TCN and Bi-LSTM to achieve dual temporal dimension modeling, and incorporates the global attention mechanism of Transformer to comprehensively extract and multi-levelly fuse spatio-temporal features. Furthermore, this study optimizes and innovates the original WiTS model. The incorporation of partial convolution and spatial attention structures within the CNN module has been demonstrated to effectuate a reduction in model parameters and FLOPs, while preserving the model's capacity for feature extraction. Additionally, the attention mechanism is incorporated into the Bi-LSTM module to dynamically assign weight to temporal features, thereby enabling the model to prioritize key action segments. Finally, the SSA is employed to optimize hyperparameters, further enhancing model performance. Experiments on two complementary datasets, WiAR (primarily half-body actions) and CSIAR (primarily full-body actions), demonstrate that the model exhibits strong generalization capabilities across different action granularities, achieving average recognition accuracies of 95.75% and 96.71%, significantly outperforming single-network architectures. The F1-score exceeds 96% in both cases, addressing the issue of performance fluctuations in single-network models across different scenarios. Moreover, the enhanced model possesses a mere 2.19 million parameters and fewer than 560 million FLOPs offering advantages in terms of lightweight design and making it suitable for deployment on limited-computing edge devices while meeting real-time requirements.

B. Limitations and Future Work

Despite the WiTS model's demonstrated efficacy in action recognition accuracy and lightweight design, certain limitations persist. On the one hand, the experiments were validated exclusively on the WiAR and CSIAR public datasets, which exhibit limited diversity in action categories and scenarios. The exclusion of more complex scenarios, such as signal interference and varying room layouts, may impact the model's generalization capabilities in real-world settings. Secondly, the present research focuses primarily on the optimization of algorithms without addressing the practical deployment challenges that have been identified, such as dynamic environmental changes and long-term stability on edge devices.

To address these issues, the plan of future work encompasses three key areas. First, constructing larger, multi-scenario datasets incorporating more diverse human actions to enhance the trained model's generalization and robustness. Second, further optimizing the model architecture by exploring more adaptive learning mechanisms, enabling the model to dynamically adjust to environmental changes and user variations. Finally, deploy the WiTS model on real-world edge device platforms, conduct practical scenario testing, optimize inference efficiency and energy consumption, and advance its practical application in fields such as smart homes and health monitoring.

References

[1] Z. Shi, Q. Cheng, J. A. Zhang, and R. Y. Da Xu, “Environment-robust WiFi-based human activity recognition using enhanced CSI and deep learning,” IEEE Internet of Things Journal, vol. 9, no. 24, pp. 24643–24654, 2022.

[2] Y. Zhang, F. Zhang, Y. Jin, Y. Cen, V. Voronin, and S. Wan, “Local correlation ensemble with GCN based on attention features for cross-domain person Re-ID,” ACM Transactions on Multimedia Computing, Communications and Applications, vol. 19, no. 2, pp. 1–22, 2023.

[3] I. Ahmad, A. Ullah, and W. Choi, “WiFi-based human sensing with deep learning: Recent advances, challenges, and opportunities,” IEEE Open Journal of the Communications Society, vol. 5, pp. 3595–3623, 2024.

[4] Z. Wei, W. Chen, S. Ning, W. Lin, N. Li, B. Lian, X. Sun, and J. Zhao, “A survey on WiFi-based human identification: Scenarios, challenges, and current solutions,” ACM Transactions on Sensor Networks, vol. 21, no. 1, pp. 1–32, 2025.

[5] W. He, K. Wu, Y. Zou, and Z. Ming, “WiG: WiFi-based gesture recognition system,” in 2015 24th International Conference on Computer Communication and Networks (ICCCN), Las Vegas, NV, USA, 2015, pp. 1–7.

[6] Y. Zhang, Y. Zheng, K. Qian, G. Zhang, Y. Liu, C. Wu, and Z. Yang, “Widar3.0: Zero-effort cross-domain gesture recognition with Wi-Fi,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 11, pp. 8671–8688, 2021.

[7] X. Meng, L. Feng, H. Chen, T. Chen, J. Ma, A. Wang, D. Liu, and Y. Zhao, “Just-in-time human gesture recognition using WiFi signals,” Chinese Journal of Electronics, vol. 30, no. 6, pp. 1111–1119, 2021.

[8] X. Dang, Y. Liu, Z. Hao, X. Tang, and C. Shao, “Air gesture recognition using WLAN physical layer information,” Wireless Communications and Mobile Computing, vol. 2020, no. 1, p. 8546237, 2020.

[9] Z. Hao, Y. Duan, X. Dang, Y. Liu, and D. Zhang, “Wi-SL: Contactless fine-grained gesture recognition uses channel state information,” Sensors, vol. 20, no. 14, p. 4025, 2020.

[10] J. Huang, B. Liu, C. Miao, Y. Lu, Q. Zheng, Y. Wu, J. Liu, L. Su, and C. W. Chen, “PhaseAnti: An anti-interference WiFi-based activity recognition system using interference-independent phase component,” IEEE Transactions on Mobile Computing, vol. 22, no. 5, pp. 2938–2954, 2021.

[11] A. Chelli, M. Muaaz, and M. Pätzold, “Actrec: a wi-fi-based human activity recognition system,” in 2020 IEEE International Conference on Communications Workshops (ICC Workshops), Dublin, Ireland, 2020, pp. 1–6.

[12] X. Cheng and B. Huang, “CSI-based human continuous activity recognition using gmm-hmm,” IEEE Sensors Journal, vol. 22, no. 19, pp. 18709-18717, 2022.

[13] H. Zou, Y. Zhou, J. Yang, H. Jiang, L. Xie, and C. J. Spanos, “WiFi-enabled device-free gesture recognition for smart home automation,” in 2018 IEEE 14th International Conference on Control and Automation (ICCA), Anchorage, AK, USA, 2018, pp. 476–481.

[14] M. Muaaz, A. Chelli, M. W. Gerdes, and M. Pätzold, “Wi-Sense: A passive human activity recognition system using Wi-Fi and convolutional neural network and its integration in health information systems,” Annals of Telecommunications, vol. 77, no. 3, pp. 163–175, 2022.

[15] H. Yan, Y. Zhang, Y. Wang, and K. Xu, "WiAct: A passive WiFi-based human activity recognition system," IEEE Sensors Journal, vol. 20, no. 1, pp. 296–305, 2019.

[16] P. Duan, C. Li, J. Li, X. Chen, C. Wang, and E. Wang, "WISDOM: Wi-Fi-based contactless multiuser activity recognition," IEEE Internet of Things Journal, vol. 10, no. 2, pp. 1876–1886, 2022.

[17] W. Meng, X. Chen, W. Cui, and J. Guo, "WiHGR: A robust WiFi-based human gesture recognition system via sparse recovery and modified attention-based BGRU," IEEE Internet of Things Journal, vol. 9, no. 12, pp. 10272–10282, 2021.

[18] R. Gao, W. Li, J. Liu, S. Dai, M. Zhang, L. Wang, and D. Zhang, "WiCGesture: Meta-motion-based continuous gesture recognition with Wi-Fi," IEEE Internet of Things Journal, vol. 11, no. 9, pp. 15087–15099, 2023.

[19] W. Cui, B. Li, L. Zhang, and Z. Chen, "Device-free single-user activity recognition using diversified deep ensemble learning," Applied Soft Computing, vol. 102, p. 107066, 2021.

[20] J. Chen, S. H. Kao, H. He, W. Zhuo, S. Wen, C. H. Lee, and S. H. G. Chan, "Run, don't walk: chasing higher FLOPS for faster neural networks," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, Canada, 2023, pp. 12021–12031.

[21] J. Xue and B. Shen, "A novel swarm intelligence optimization approach: sparrow search algorithm," Systems Science & Control Engineering, vol. 8, no. 1, pp. 22–34, 2020.

[22] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," Journal of Big Data, vol. 8, no. 1, p. 53, 2021.

[23] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in European conference on computer vision, D. Fleet, T. Pajdla, B. Schiele and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 818–833.

[24] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," arXiv preprint arXiv:1803.01271, 2018.

[25] J. Fan, K. Zhang, Y. Huang, Y. Zhu, and B. Chen, "Parallel spatio-temporal attention-based TCN for multivariate time series prediction," Neural Computing and Applications, vol. 35, no. 18, pp. 13109–13118, 2023.

[26] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 2016.

[27] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," IEEE Transactions on Signal Processing, vol. 45, no. 11, pp. 2673–2681, 1997.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez and Ł. Kaiser, "Attention is all you need," in Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach California USA, 2017, pp. 6000–6010.

[29] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," AI Open, vol. 3, pp. 111–132, 2022.

[30] S. Yousefi, H. Narui, S. Dayal, S. Ermon, and S. Valaee, "A survey on behavior recognition using WiFi channel state information," IEEE Communications Magazine, vol. 55, no. 10, pp. 98–104, 2017.

[31] L. Guo, L. Wang, C. Lin, J. Liu, B. Lu, J. Fang, Z. Liu, Z. Shan, J. Yang, and S. Guo, "Wiar: A public dataset for WiFi-based activity recognition," IEEE Access, vol. 7, pp. 154935–154945, 2019.