A CSI-based Position Independent Gesture Recognition System Using Deep Learning

***Abstract*—Gesture recognition is one of the prominent technologies that allows mobile devices to recognize and respond to human gestures as input. Recently, Wi-Fi-based gesture recognition systems that use CSI data (channel state information) are becoming increasingly popular due to their privacy-preserving and non-intrusive nature. Previous Wi-Fi sensor-based gesture recognition systems typically operated from fixed positions, limiting their usefulness in real-world scenarios where users interact from multiple locations within a room. They did not work on all gesture types or position-independent systems. Their system's accuracy was insufficient. To solve these issues, this paper delivers a CSI-based position-independent gesture recognition system using machine and deep learning techniques. Our work divided the experimental room into a 3x3 grid and collected training data from each of these nine positions. Our work collected testing data from diverse positions in the room to test different machine (ML) and deep learning (DL) models trained with the training data set. This work used a variety of ML and DL models to classify four gestures. This paper discovered that the LSTM model outperforms all other methods tested, with an 81.11% accuracy value. The results highlighted that the proposed system improved the accuracy of existing schemes by at least 1.6%.**

***Index Terms*—WiFi, CSI, Position Independent, Gesture Recognition, IoT, Deep learning, HCI.**

I. Introduction

The significant advancement of IoT (internet of things) has given the study of HCI (human-computer interfacing) enormous popularity, with a focus on optimizing the interaction between users and computers to meet users' needs. Human gesture recognition systems (HGRS) are gaining attraction as a key strategy for providing HCI alternatives in real-world scenarios such as operating smart home appliances. Computer vision, wearable sensors, and Wi-Fi sensing-based gesture recognition techniques are among the most commonly used techniques [1], [2], [3]. Some sensor-based techniques [4] require users to attach gadgets regularly, which can be inconvenient for real-time use. Computer vision-based methods require users to be within the camera range and have adequate illumination, which may violate user privacy. Wi-Fi signal-based systems, on the other hand, have sparked widespread interest due to their privacy-preserving and non-intrusive recognition capabilities [5], [6], [7], [8], [9], and [10]. The widespread deployment of Wi-Fi infrastructure in indoor spaces, such as offices, laboratories, and other indoor areas, has created ideal conditions for indoor wireless sensing [11], [12]. Other benefits of WiFi-based solutions include the ability to use existing Wi-Fi infrastructure and the availability of hands-free input, which is especially useful when users' hands are filthy, damp, busy, or covered in gloves, making touch input challenging. Among the various Wi-Fi sensing methods, the two most commonly used for gesture recognition are RSSI, or Received Signal Strength Information, and CSI, or Channel State Information [13], [14], [15]. H. Abdelnasser et al. developed a technique for recognizing gestures using RSSI signals [5]. Because the RSSI signal only contains coarse-grained information, it is simple to extract this value from any device, but it performs less accurately in gesture detection. Gesture recognition using CSI data, on the other hand, takes advantage of WiFi signal propagation properties to recognize and comprehend hand gestures. At the sub-carrier level, CSI records the signal amplitude, phase, and reflections caused by human movement. This enables fine-grained tracking, resulting in exceptional performance [16], [17], [18]. CSI data is impervious to both environmental and anthropogenic changes. Furthermore, it achieves excellent accuracy with fewer user profiles, as demonstrated by H. Ahmed et al. in their survey [6]. However, existing CSI-based gesture recognition systems are limited, particularly in terms of user position variability. They did not use ESP-32 microcontroller-based systems for data collection [19]. Previous works (e.g., [17], [18], [20], [21], [22], [23]) struggle to maintain high accuracy when users change positions or require the user to carry extra devices, limiting their practical usability in real-world scenarios such as operating smart home appliances. They did not work on various gesture types or real-time data collection. Most of their works have extremely low accuracy results. They did not investigate machine learning, shaplet learning, or deep learning methods in their research. Most of their works did not include performance comparisons to previous works. To outperform these issues, this paper proposes a position-independent and CSI-based gesture recognition system using ML, IoT, and DL technologies. First, we divided the room into a 3x3 grid and collected training data from each of the nine positions. This paper collects CSI data using an ESP-32 microcontroller. Then we collected testing data from various positions in the room to test different ML and DL models that had been trained using the train data set. We utilized a variety of ML and DL models. We used ensemble learning methods such as XGB, LGBM, and Random Forest, as well as a voting classifier, to combine these tree models. We also used a learning shapelet model to take advantage of our data's time series properties. For deep learning, we used CNN and LSTM models. This paper compares the accuracy value and other performance values of various ML and DL models. Finally, this paper

classified four gestures using the LSTM model, which has a high accuracy value in comparison with existing methods. The second section discusses existing work. Section three presents a detailed overview of the developed system. Section 4 contains the detailed results. Section five hints at the paper's concluding remarks with key future research points.

II. Literature Review

This section delivers discussion regarding existing gesture recognition methods with a brief explanation. In [24], the authors developed a hand activity detection system that utilizes Wi-Fi cards and the RSSI signal to track hand movements. It enables hands-free drawing without the use of wearable's or specialized hardware. In [5], the authors used Wi-Fi signal and mobile device for gesture identification without using any ML based prediction or training method. This system uses a Discrete Wavelet Transformation (DWT) method to reduce noise in RSSI values. The used a single access point around the mobile device. In [25], the authors introduced a CSI-based system for device-free human activity recognition that achieves atleast 75% accuracy. In [26], the authors employed deep learning techniques for gesture detection. However, in untrained situations, this performance drops to 67%. In [27], the author recognizes 276 sign gestures by using CSI data and CNN model. However, their classification work is limited by the specific distance between the access point device. In [28], the authors used CSI to recognize hand motions in Wi-Fi-enabled system environments. For successful classification, the antenna spacing must fall within a specific range. Hong Li et al. developed WiFinger [29], which allows users to accurately input text on Wi-Fi devices. They also stated that, for stable and clear finger gesture patterns in the CSI stream, transceivers should be placed close together, as patterns weaken with increasing distance in this system. They did not use shaplet learning [30] to predict the best target variable or feature and instead examined the time series properties of CSI data. In [31], the authors used 802.11-based CSI data to classify human activity. Using meta learning techniques, they were able to achieve a 79.5 percent accuracy. In [32], the authors employed random forest based gesture classification mechanism by using both microcontroller devices and Wi-Fi data. In [1], the authors used HMM model for gesture classification and achieved an accuracy of 67.8. The aforementioned existing works did not create a CSI-based device-free independent system for gesture recognition with high accuracy that incorporates ML and DL techniques. To address these limitations, this paper collects a real-time dataset and creates a CSI-based device-free position-independent recognition system for gestures that employs ML, shaplet learning, and DL methods.

III. Proposed Scheme

The system architecture of our CSI-based, position-independent gesture recognition system is shown in Figure 1, which consists some phases: establishing the experimental setup, collecting raw CSI training and testing data, CSI data preprocessing, parsing, and labeling, train the ML and DL models, gesture classification, and result comparison.

A. Hardware configuration

This work used two ESP-32 microcontrollers, one as a receiver and the other as a transformer (transmit and receive CSI data over a Wi-Fi network). To do so, we first installed the ESP-IDF framework [19]. Then we set up the ESP-IDF to enable ESP32 flashing. One of the ESP32s is configured with Active-AP to serve as an access point or transmitter. This configuration allows the ESP32 to transmit CSI data while operating as an access point. The other ESP32 is flashed with Active-STA so that it can receive the transmitted CSI data. This step connects the two ESP32s via Wi-Fi, allowing for the retrieval of CSI data from the receiver.

B. CSI data collection

To collect CSI data, we first divided the room into a 3x3 grid and marked the center of each grid cell. We then repeated each of the four gestures (clap, wave, push, and circle) ten times at each of these nine locations, yielding 360 sets of training data in total. Each gesture was carried out independently for 20 seconds. Following that, we repeated all four gestures for 20 seconds in five randomly selected positions throughout the room. This process resulted in 20 sets of test data. The data collection process is illustrated in Fig. 2. After collecting all of the data, it was time to begin pre-processing. We generated a CSV file (raw dataset) from our experimental setup (Fig. 3).

C. Data preporcessing

After creating the CSV file, we individually parsed each set of data. During the pre-processing step, we filtered the data and only kept the high throughput values. Here, sig_mode=1 denotes a high throughput signal. After filtering, we extracted CSI data from 26 columns of the dataset. Each row in this column represents a data packet containing 128 subcarriers, with each pair representing a complex number. Then, after analyzing the data, we discovered that 24 of the 128 sub-carriers had a constant null value. We removed the null subcarriers from our data sets, leaving only 104 complex subcarriers. The formula for extracting the amplitude and phase of any complex number z = x + iy is given by: amplitude z=$\sqrt{x^2 + y^2}$ and phase θ=$tan^{-1}(\frac{y}{x})$. After extracting 52 amplitude and 52 phase subcarrier values, we concatenated and labeled the data set, which contained 52 amplitude and 52 phase subcarriers. Similarly, we filtered, parsed, and labeled the test dataset.

D. Gesture Classification Using ML and DL Method

After labeling and creating the final train and test datasets, we used ensemble learning, shapelet learning (two machine learning methods), CNN, and LSTM (two deep learning methods) to classify gestures in our data set. Initially, we imported both the training and testing data in CSV format for each machine learning or deep learning technique. In our ensemble learning approach, we used three different models:

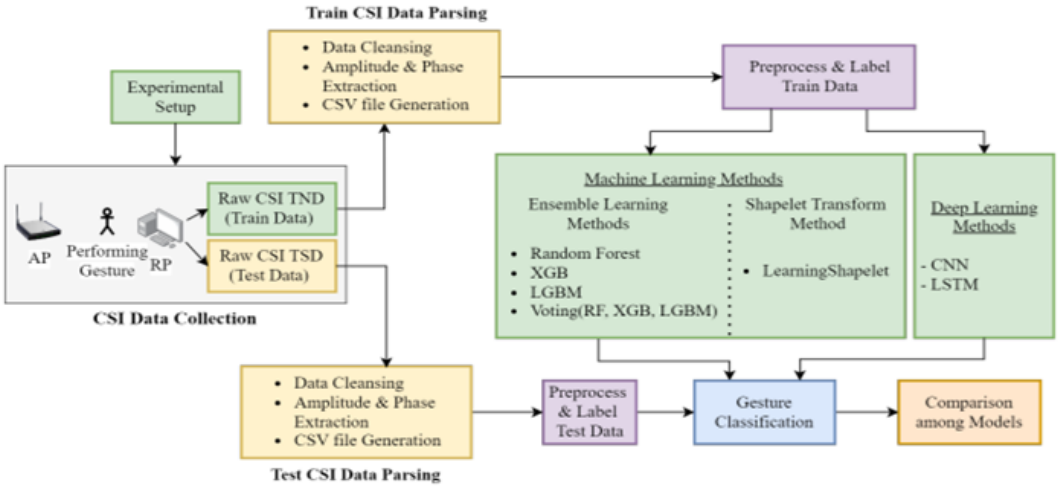

Fig. 1. Methodology diagram of our proposed system

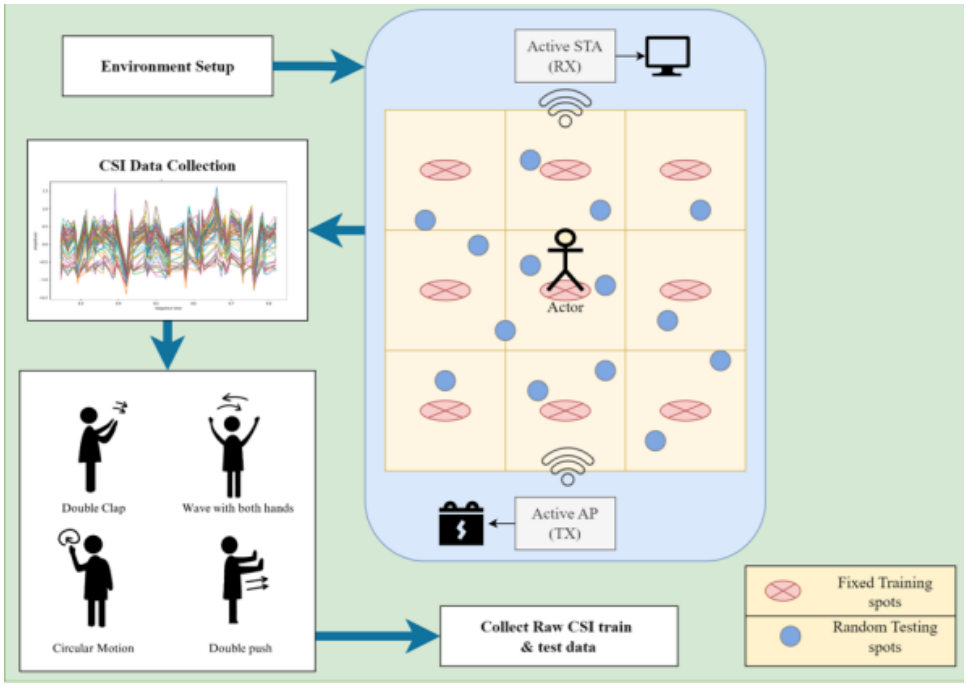

Fig. 2. Overview of data collection

TABLE I
COMPARISON AMONG ML AND DL MODEL FOR GESTURE RECOGNITION

Model	Accuracy	Precision	Recall	F1 score
Learning shaplet	74.5%	77.7%	74.5%	74.0%
CNN (amplitude)	73.31%	73.71%	73.31%	73.51%
CNN (phase)	59.77%	62.84%	59.77%	61.27%
CNN (amplitude and phase)	76.55%	77.48%	76.55%	77.01%
CNN (weighted avg.)	78.13%	78.69%	78.13%	78.41%
LSTM	81.11%	81.10%	81.11%	81.11%

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	type	role	mac	rssi	rate	sig_mode	mcs	bandwidth	smoothing	not_sounding	aggregation	stbc	fec_coding
2	CSI_DATA	STA	30:AE:A4:9C:A5	-67	11	0	0	0	0	0	0	0	0
3	CSI_DATA	STA	30:AE:A4:9C:A5	-67	11	0	0	0	0	0	0	0	0
4	CSI_DATA	STA	30:AE:A4:9C:A5	-74	11	1	3	1	1	1	0	0	0

(a) First few columns CSI data

S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE
local_timestamp	ant	sig_len	rx_state	real_time_set	real_timestamp	len	CSI_DATA					
23753	0	34	0	0	0.211462	128	[34 32 2 0 0 0 0 0 0 0 0 0 0 0 22 -21 23 -19 26 -18 24 -18 24 -17 25 -15 26 -14 23 -14 22 -12 22 -12 24 -12 21 -13					
37148	0	131	0	0	0.224475	128	[-125 48 8 0 0 0 0 0 0 0 0 0 0 0 6 28 5 27 2 30 2 30 0 30 -2 29 -2 29 -4 26 -3 26 -4 26 -4 26 -3 24 -2 24 -2 23 -2 23					
41305	0	135	0	0	0.228841	384	[-121 112 8 0 0 0 0 0 0 0 0 0 0 0 7 -25 7 -24 10 -22 12 -25 14 -21 12 -23 11 -23 11 -20 16 -17 15 -21 17 -17 15 -18					

(b) Last few columns of CSI data

Fig. 3. CSI data in CSV format

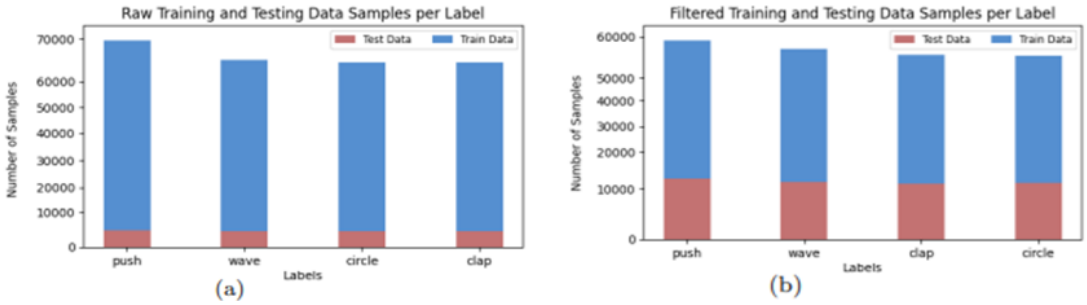

Fig. 4. Train-Test Distribution, Before and After Filtering

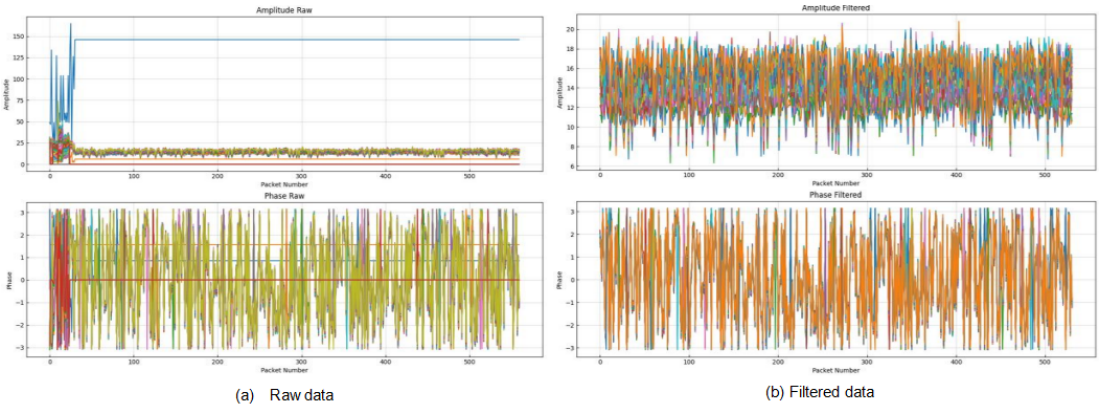

Fig. 5. Amplitude and Phase Variation (Double Clap Gesture)

LGBM, XGBoost (XGB), and Random Forest. To reduce individual biases while improving overall model performance, we used the hard voting ensemble method. This involved combining the predictions from the Random Forest, XGBoost, and LightGBM classification models. The final output prediction is the class that receives the most votes from the models in each sample. We also investigated the use of the shapelet transform method. A shapelet can be termed as short segment of a time series that accurately predicts the target variable. To take advantage of our CSI data's time-series property, we used the Shapelet learning model [30]. This technique designed for time series classification tasks, entails identifying discriminative subsequences, or shapelets, within temporal data. To implement this model, we first converted our dataset from 2D to 3D, with each instance of time series data representing a 200-length window containing 52 amplitude subcarriers. Then, rather than trying out every possible shapelet, we discovered near-optimal top-K shapelets by capturing their interactions. The CNN model was built with several layers, including Convolutional (Conv1D) layers, normalization, pooling, dense, and a dropout layers. We used Conv1D, or one-dimensional convolution, to extract features from our sequential data. It involves applying filter or kernel windows to the input sequence. In our system, we used 64 filters in the 1st layer and 128 filters in the 2nd layer with kernel size 3. Batch normalization helps to stabilize learning

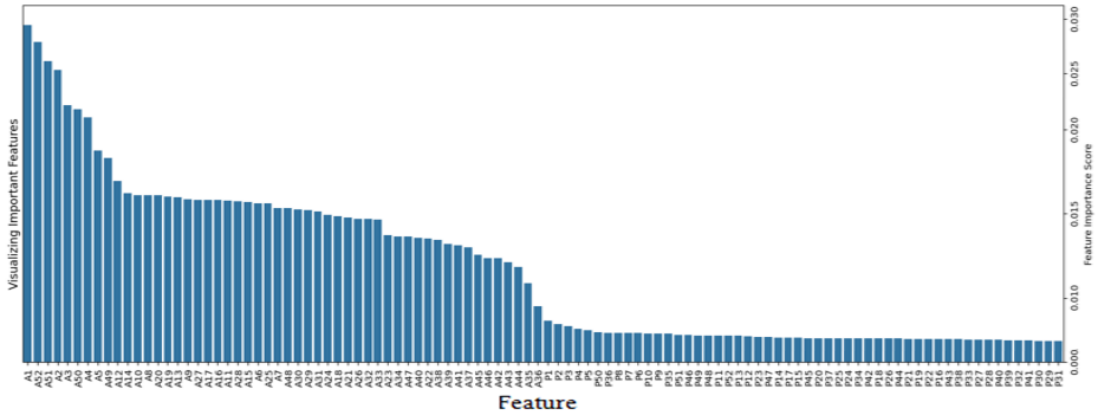

Fig. 6. Feature Importance

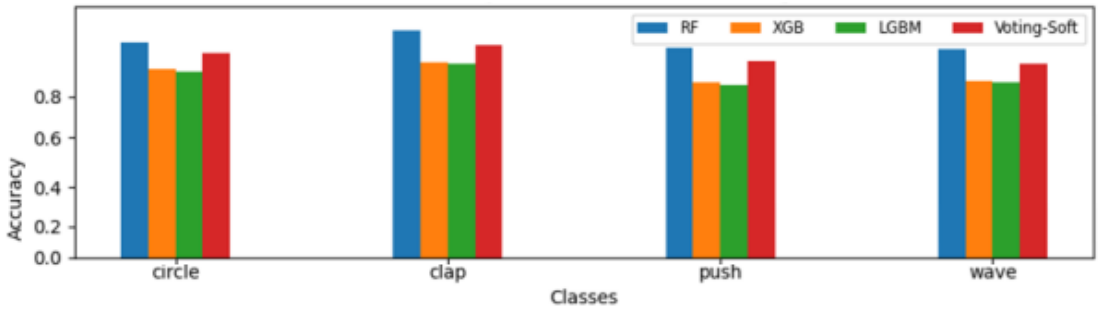

Fig. 7. Performance comparison (ensemble model)

by normalizing the previous layer's output. In our case, a pooling size of 2 indicates that the layer reduces the feature dimension by half. The Flatten layer reduces the multidimensional output of previous layers to a single long feature vector. The dense layer uses the extracted features (i.e., by Conv1D and pooling layers) to perform classification or regression task. The dropout layer prevents overfitting by randomly removing units (and their connections) during training. For the activation function, we used ReLU for the internal layers and Softmax for the output layer. While the LSTM model consists LSTM layer, a dense layer, and softmax activation function. We start the model with an LSTM layer of 64 units. We specify the input shape parameter as number of time steps and number of features per time step. Following the LSTM, there is a dense (fully connected) layer containing four units for each gesture class. We used the softmax activation function because it is common for multi-class classification tasks and produces a probability distribution over the classes. This LSTM model is typically used for our experiment with sequential time series data. Each of these models was then tested on the testing dataset to determine their performance.

IV. Results and Analysis

For this work, we collected 360 sets of training data from nine fixed positions and 20 sets of testing data from randomly selected positions. After filtering and removing these 380 sets of CSI data, we had a total of 2,29,490 rows (see Figure 4). We visualized the variations in the parsed CSI data (amplitude and phase) for each gesture class before and after filtering. Filtering involved selecting data based on high-throughput signals and removing all null subcarriers. We also looked into the significance of amplitude and phase values as features in this system (see Figure 5 for double cap gesture data). Figure 6 depicts the feature importance of all 52 amplitude (prefix 'A') and 52 phase (prefix 'P') subcarriers. In our CSI data, we discovered that amplitude subcarriers are more significant than phase subcarriers. We evaluated our proposed position-independent gesture classification model using a variety of metrics. Figure 7 shows that the random forest (RF) model outperforms the other ML classifiers in terms of accuracy, recall, and F1 scores. To take advantage of our data's time-series properties, we trained a Shapelet Learning model. For our research, we obtained 5 shapelets of size 20, which best predict time-series data. Table I shows shaplet learning model

TABLE II
Comparison with existing works

Existing works	Tools, method	Accuracy
K. Ohara et al., [1]	Intel 5300, HMM	67.8%
Y. Zhang et al., [31]	802.11 based CSI and meta learning	79.5%
M. Bastwesy et al., [32]	ESP 32, random forest	72%
Proposed work	ESP 32 and LSTM	81.11%

has an accuracy of 74.5%. We investigated the performance of our CNN model using only amplitude subcarriers, only phase subcarriers, or both amplitude and phase subcarriers as features. Table I delivers that the CNN model's accuracy for various features ranges from 59% to 78%. Table I shows that the LSTM model has an accuracy value of 81.11%. After comparing both ML and DL approaches, we can conclude that LSTM offers better performance and is thus chosen for the gesture recognition task. Table II compares our findings to three existing methods. (e.g., [1], [31], [32]) We can see that Y. Zhang et al. [31] achieved the second highest accuracy of 79.5%. Unlike previous works, our proposed gesture classification scheme uses an LSTM model and offers a better accuracy value of 81.11%, which is at least 1.6% better than the existing works. The LSTM model performs well in this context because it captures dependencies and patterns in sequential data. It excels at understanding the progression of events over time, which is critical for interpreting gestures in our application.

V. Conclusion

This paper describes a WiFi, IoT, and CSI-based position-independent gesture recognition system for smart home appliances. This paper collects CSI data using two ESP32 microcontrollers. In the pre-processing phase, this paper removes null subcarriers and refines the data using high-throughput filtering. This paper then extracted important information about CSI amplitude and phase from the filtered data, which we used to train our models. To choose the best prediction model, this paper thoroughly compares various ML and DL approaches, such as different ensemble learning techniques, the shaplet learning method, and deep learning models such as CNNs and LSTMs. Our results highlight that the LSTM method achieved the highest accuracy of 81.11%, slightly surpassing the Random Forest's accuracy of 80.50%, demonstrating the efficacy of long-term memory for this type of sequential data. The performance comparison results also showed that the proposed LSTM-based gesture recognition system achieves 1.6% more accuracy gain than previous works. In the future, this work can be expanded to include more challenging scenarios such as CSI data collection with more people, more noise, more gesture considerations, explainable AI-based advanced feature extraction, and security considerations using block chain technology.

References

[1] K. Ohara et al., "Preliminary Investigation of Position Independent Gesture Recognition Using Wi-Fi CSI," PerCom Workshops, Greece, 2018, pp. 480-483.

[2] D. De et al., "Multimodal Wearable Sensing for Fine-Grained Activity Recognition in Healthcare," IEEE Internet Computing, vol. 19, no. 5, pp. 26-35, 2015.

[3] H. Mliki et al., "Human activity recognition from uav-captured video sequences," Pattern Recognition, vol. 100, pp. 1-10, 2020.

[4] J. Shotton et al., "Real-time human pose recognition in parts from single depth images," CVPR, USA, 2011, pp. 1297-1304.

[5] H. Abdelnasser et al., "WiGest: A ubiquitous WiFi-based gesture recognition system," INFOCOM, Hong Kong, 2015, pp. 1472-1480.

[6] H. F. T. Ahmed et al., "Device free human gesture recognition using wi-fi csi: A survey," Engineering Applications of Artificial Intelligence, vol. 87, pp. 1-10, 2020.

[7] S. Tan et al., "Commodity WiFi Sensing in Ten Years: Status, Challenges, and Opportunities," in IEEE IoT Journal, vol. 9, no. 18, pp. 17832-17843, 2022.

[8] Z. Wang et al., "Wi-Fi CSI-Based Behavior Recognition: From Signals and Actions to Activities," in IEEE Communications Magazine, vol. 56, no. 5, pp. 109-115, May 2018.

[9] N. Damodaran et al., "Device free human activity and fall recognition using wifi channel state information (csi)," in CCF Transactions on Pervasive Computing and Interaction, vol. 2, pp. 1–17, 2020.

[10] Y. Zeng et al., "Fullbreathe: Full human respiration detection exploiting complementarity of csi phase and amplitude of wifi signals," ACM IMWUT, vol. 2, no. 3, pp. 1–19, 2018.

[11] B. Yu et al., "WiFi-Sleep: Sleep Stage Monitoring Using Commodity Wi-Fi Devices," in IEEE Internet of Things Journal, vol. 8, no. 18, pp. 13900-13913, 15 Sept.15, 2021.

[12] Z. Wang et al., "A survey of user authentication based on channel state information," Wireless Communications and Mobile Computing, vol. 2021, pp. 1–16, 2021.

[13] Y. Ma et al., "Wifi sensing with channel state information: A survey," ACM Computing Surveys (CSUR), vol. 52, no. 3, pp. 1–36, 2019.

[14] Q. Pu et al., "Whole-home gesture recognition using wireless signals, " MobiCom, 2013, pp. 27–38.

[15] J. Liu et al., "Tracking vital signs during sleep leveraging off-the-shelf wifi," MobiHoc conference, 2015, pp. 267–276.

[16] Z. Wang et al., "A Survey on Human Behavior Recognition Using Channel State Information," IEEE Access, vol. 7, pp. 155986-156024, 2019.

[17] M. De Sanctis et al., "Wibecam: Device free human activity recognition through wifi beacon-enabled camera," in Proceedings of the 2nd workshop on Workshop on Physical Analytics, 2015, pp. 7–12.

[18] Z. Yang et al., "From rssi to csi: Indoor localization via channel response, " ACM Comput. Surveys, vol. 46, no. 2, pp. 1– 32, 2013.

[19] L. Espressif Systems, "Getting started with esp32," https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/index.html, last accessed on June 2024.

[20] S. M. Hernandez et al., "Lightweight and Standalone IoT Based WiFi Sensing for Active Repositioning and Mobility," IEEE 21st WoWMoM, Ireland, 2020, pp. 277-286.

[21] M. Oudah et al., "Hand gesture recognition based on computer vision: A review of techniques," journal of Imaging, vol. 6, no. 8, pp. 1-10, 2020.

[22] C. Zhu et al., "Wearable Sensor-Based Hand Gesture and Daily Activity Recognition for Robot-Assisted Living," in IEEE Transactions on SMC, vol. 41, no. 3, pp. 569-573, May 2011.

[23] F. Adib et al., "See through walls with wifi!," ACM SIGCOMM, 2013, pp. 75–86.

[24] L. Sun et al., "Widraw: Enabling handsfree drawing in the air on commodity wifi devices," 21st MobiCom, 2015, pp. 77–89.

[25] W. Xi et al., "Device-free human activity recognition using csi," CSAR workshop, 2015, pp. 31–36.

[26] Q. Zhou et al., "A Device-Free Number Gesture Recognition Approach Based on Deep Learning," 12th CIS conference, China, 2016, pp. 57-63.

[27] Y. Ma et al., "Signfi: Sign language recognition using wifi," Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, vol. 2, no. 1, pp. 1–21, 2018.

[28] Z. Tian et al., "WiCatch: A Wi-Fi Based Hand Gesture Recognition System," in IEEE Access, vol. 6, pp. 16911-16923, 2018.

[29] H. Li et al., 'Wifinger: Talk to your smart devices with finger-grained gesture," UBICOMP conference, 2016, pp. 250–261.

[30] J. Grabocka et al., "Learning time-series shapelets," 20th ACM SIGKDD, 2014, pp. 392– 401.

[31] Y. Zhang et al., "Human Activity Recognition Across Scenes and Categories Based on CSI," in IEEE Transactions on Mobile Computing, vol. 21, no. 7, pp. 2411-2420, July 2022.

[32] M. Bastwesy et al., "Tracking On-Desk Gestures Based on Wi-Fi CSI on Low-Cost Microcontroller," ICMU conference, Japan, 2023, pp. 1-6.