WiFi Sensing-Based Human Activity Recognition For Smart Home Applications Using Commodity Access Points

Gad Gad*, Iqra Batool*, Mostafa Fouda†, Shikhar Verma‡, Zubair Md Fadlullah*
*Department of Computer Science, Western University, London, ON, Canada
†Department of Electrical and Computer Engineering, Idaho State University, Pocatello, ID, USA
‡School of Informatics, Kochi University of Technology, Kochi, Japan
Emails: ggad@uwo.ca, ibatool2@uwo.ca, mfouda@ieee.org, shikhar.verma@ieee.org, zfadlullah@ieee.org

***Abstract*—The ubiquitous presence of WiFi-enabled devices in residential households creates a foundation for deploying WiFi sensing in smart home applications. This technology offers advantages such as preserving users' privacy, efficient resource consumption, and sidestepping conventional sensing limitations, such as a restricted field of view and occlusion. However, WiFi sensing comes with its own technical challenges, including hardware-level noise and sensitivity to temporal and environmental changes, leading to shifts in data distribution across time, environments, and subjects, ultimately degrading the performance of machine learning (ML) models.**

To mitigate distribution drift and train robust WiFi sensing models, previous work employed data-cleaning and processing methods, such as phase sanitization. These techniques are resource-intensive, slowing the entire WiFi sensing pipeline. This work presents a lightweight time-invariant feature representation method for WiFi sensing data. To evaluate our proposed method, we train ML models on self-collected, labeled Channel State Information (CSI) data from commodity WiFi-enabled devices across multiple smart home applications, including room occupancy detection, human activity recognition, and indoor localization. Results show an improvement up to 1.6x accuracy compared with previous methods with significantly less preprocessing time (250x).

***Index Terms*—WiFi sensing, channel state information, human activity recognition, rolling variance, deep learning.**

I. Introduction

WiFi Channel State Information (CSI) has emerged as a promising modality for contactless sensing in smart home applications. Unlike camera- or wearable-based systems, CSI sensing is privacy-preserving, requires no additional hardware beyond commodity WiFi devices, and is robust to occlusion and lighting conditions

However, CSI signals are susceptible to hardware-induced phase noise, sampling rate drift, and environmental changes that cause distribution shift between training and deployment, degrading classifier performance over time.

Prior work has addressed the phase-noise problem through sanitisation techniques such as SHARP [**?**], which recovers a clean phase signal via sparse delay-domain reconstruction. In this paper, we show that phase information is *not necessary*: a simple rolling-variance transform applied to CSI amplitude captures the temporal fluctuation energy caused by human activity, suppresses slow environmental drift, and consistently outperforms phase-based representations.

The main contributions of this work are:

1) **CSI time-invariant feature representation and resource-efficient pipeline.** We propose a rolling-variance transform that extracts time-invariant activity signatures from CSI amplitude alone, eliminating the need for phase information.
2) **Multi-task labeled datasets.** Open-source classification CSI datasets from commodity WiFi microcontrollers (ESP32-C6) across two environments (home, office) and three classification tasks: human presence detection, human activity recognition, and indoor localization.

The remainder of the paper is organised as follows. Section II presents the CSI signal model and hardware setup. Section III describes the datasets. Section IV details the proposed two-phase method. Section V covers experimental setup and evaluation metrics. Section VI presents results, ablation studies, and discussion.

II. System Model

Channel State Information (CSI) represents the complex channel response between transmitter and receiver in OFDM systems [**?**], [**?**]. For subcarrier k at time n, the CSI matrix is $\mathbf{H}_k[n] \in \mathbb{C}^{N_r \times N_t}$. The measured channel frequency response (CFR) includes hardware-induced phase offsets:

$$\bar{H}_m(n) = H_m(n)\, e^{j\phi_{\text{offs},m}}, \tag{1}$$

where $\phi_{\text{offs},m} = -2\pi m(\tau_{\text{SFO}} + \tau_{\text{PDD}})/T + \phi_{\text{CFO}} + \phi_{\text{PPO}} + \phi_{\text{PA}}$ combines sampling frequency offset (τ_{SFO}), packet detection delay (τ_{PDD}), carrier frequency offset (ϕ_{CFO}), packet phase offset (ϕ_{PPO}), and power amplifier distortion (ϕ_{PA}).

The channel can be modeled as a sum of L propagation paths:

$$H_m(n) = \sum_{p=1}^{L} A_p(n)\, e^{-j2\pi(f_c + m/T)\tau_p(n)} \tag{2}$$

(a) Home environment (b) Office environment

Fig. 1. Floor plan layouts of the two data collection environments. (a) Home: a bedroom with a bed, desk, and two seating positions used for HAR and occupancy detection tasks. (b) Office: a workspace with six numbered desk positions (1–6) used for HAR and indoor localization tasks. The ESP32-C6 transmitter and receiver are positioned to cover the activity areas in each environment.

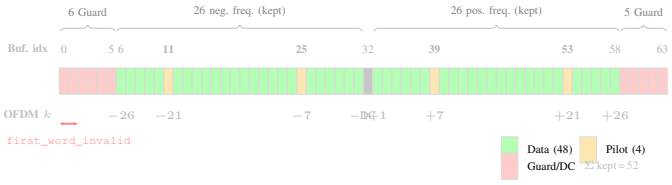

Fig. 2. ESP32 HT20 LLTF subcarrier mask (64-point CSI buffer). Buffer index b maps to OFDM index $k = b - 32$. Green cells: 48 data subcarriers retained.

TABLE I
DATASET OVERVIEW AND COLLECTION SUMMARY.

Dataset	Env.	Task	# Cls	Split	Recorded	Packets	Dur.
Home HAR (train)	Home	HAR	7	Session holdout	Oct 2025	2.7 M	232 min
Home HAR (test)	Home	HAR	7	∼3.5 mo gap	Feb 2026	2.7 M	233 min
Home Occ. (train)	Home	Occ.	3	Temporal split	Feb 2026	1.1 M	100 min
Home Occ. (test)	Home	Occ.	3	same-session	Feb 2026	0.6 M	50 min
Office HAR	Office	HAR	4	% split	Oct 2025	0.8 M	66 min
Office Loc. (train)	Office	Loc.	4	File holdout	Oct 2025	0.9 M	67 min
Office Loc. (test)	Office	Loc.	4	File holdout	Oct 2025	0.7 M	57 min

III. Data Collection and Datasets

Our data collection system employs two ESP32-C6 microcontrollers operating in 802.11n mode at $f_c = 2.412$ GHz with HT20 configuration. The ESP-IDF CSI driver reports 64 complex-valued subcarrier coefficients per LLTF packet (128 bytes: imaginary followed by real, per subcarrier). Of these, 52 correspond to valid LLTF data and pilot tones (OFDM indices -26 to -1 and $+1$ to $+26$, spaced at $\Delta f = 312.5\,\text{kHz}$); the remaining 12—six lower guard-band, one DC null, and five upper guard-band subcarriers—are excluded by a boolean mask. Additionally, the ESP32-C6 hardware may invalidate the first four bytes (buffer indices 0–1) due to a known limitation (`first_word_invalid` flag).

We collected four datasets spanning two environments (home, office) and three sensing tasks. Tables I and II summarize the key characteristics.

TABLE II
ACTIVITY LABEL TAXONOMY.

Dataset	Label	Semantic Cat.
Home HAR	drink	Hand-to-mouth
	eat	Hand-to-mouth
	smoke	Hand-to-mouth
	watch	Sedentary
	work	Sedentary
	sleep	Away
	empty	Away
Home Occ.	empty	Away
	sleep	Away
	work	Sedentary
Office HAR	eat	Hand-to-mouth
	empty	Away
	watch	Sedentary
	work	Sedentary
Office Loc.	empty	Away
	five	Localization
	one	Localization
	two	Localization

TABLE III
COMPARISON OF COMMUNICATION SETUP PARAMETERS: SHARP METHOD [1] VS OUR ESP32-BASED SYSTEM.

Parameter	SHARP	Ours
Monitored channel	802.11ac ch. 42	802.11n HT20
OFDM sample duration, T	3.2×10^{-6} s	3.2×10^{-6} s
No. OFDM sub-channels, M	256 (245 used)	64 (52 used)
Subcarrier spacing, Δf	312.5 kHz	312.5 kHz
No. monitoring antennas, N_{ant}	4	1

The Home HAR dataset contains three semantically similar "hand-to-mouth" activities (drink, eat, smoke) that share nearly identical gross body posture, making WiFi-based discrimination challenging. One of the observation during data collection sessions is the volatile sampling rate of CSI data. Additionally The ESP32 Sampling rate varies across recordings, motivating the need for a uniform resampling step as a common data cleaning process before applying the proposed data processing, shown in the system overview figure 3.

IV. Methods

In this section we present a CSI data processing method to be used in ML learning setting to enhance performance. First, we explain SHARP-based phase sanitization which is used as a baseline in following ML/DL experiments. While the original method applied phase sanitization to remove hardware offsets in a different setup as detailed in table III. The authors used a multi antenna setup (4 rx, 4 tx antennas) using Asus RTAC86U IEEE 802.11ac Wi-Fi routers enabeld by Nexmon project [2], [3], we use an ESP32 c6 six both as a sender and a receiver (1 rx, 4 tx antennas). However, the theoretical justification for this method still holds.

In vector form across K subcarriers, $\mathbf{h} = \mathbf{T}\mathbf{r}$, where $\mathbf{T} \in \mathbb{C}^{K \times P_0}$ is the delay dictionary with entries $\mathbf{T}_{k,p} = e^{-j2\pi f_k t_p}$ and $\mathbf{r} \in \mathbb{C}^{P_0}$ bundles path complex gains.

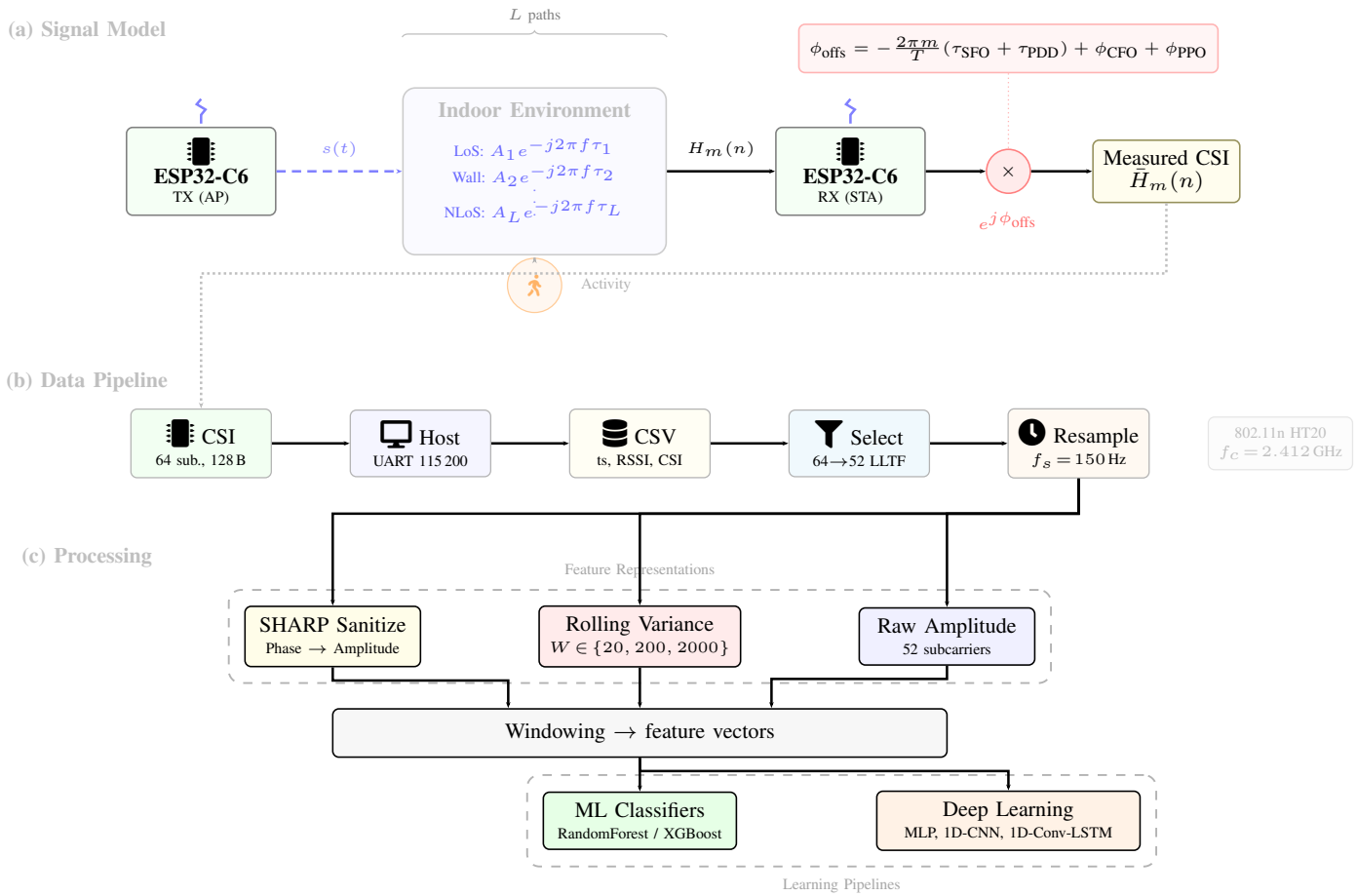

Fig. 3. End-to-end system overview. (a) Signal model: the ESP32-C6 transmitter emits $s(t)$ through an indoor environment with L multipath components modulated by human activity; the receiver measures $H_m(n)$ corrupted by hardware phase offsets ϕ_{offs} (SFO, PDD, CFO, PPO), yielding $\bar{H}_m(n)$. (b) Data pipeline: 64-subcarrier CSI packets are logged via UART, stored as CSV, filtered to 52 valid LLTF tones, and resampled to 150 Hz [**?**]. (c) Three preprocessing pipelines (SHARP sanitisation, rolling variance, raw amplitude) produce feature vectors classified by ML models (RandomForest, XGBoost) or DL architectures (MLP, 1D-CNN, 1D-Conv-LSTM).

to recover a clean phase signal and solve the ℓ_1-regularised inverse problem in the delay domain. We explicitly model the hardware-induced error in the measured CFR on subcarrier m:

$$\bar{H}_m(n) = H_m(n)\, e^{j\,\phi_{\text{offs},m}} \tag{3}$$

A. SHARP Inspired Phase Sanitization

SHARP-inspired sparse delay-domain reconstruction approach [1]. The key idea is to solve an ℓ_1-regularised inverse problem to recover sparse path coefficients $\mathbf{r}$, then use the dominant path as a phase reference to cancel common hardware offsets.

SHARP models the channel as a sparse superposition of L paths and builds a delay dictionary $\mathbf{T} \in \mathbb{C}^{K \times P_0}$ to recover sparse path coefficients via Lasso:

$$\hat{\mathbf{r}}(n) = \arg\min_{\mathbf{r}} \|\mathbf{h}(n) - \mathbf{T}\,\mathbf{r}\|_2^2 + \lambda\,\|\mathbf{r}\|_1 \tag{4}$$

The dominant path $p^* = \arg\max_p |\hat{r}_p|$ provides a phase reference; conjugate multiplication cancels the common hardware offset, yielding a sanitised CFR $\hat{\mathbf{H}}(n)$. In our implementation we set $\Delta t = 100$ ns, $t_{\min} = -300$ ns, $t_{\max} = 500$ ns ($P_0 = 9$ delay bins), and use every second subcarrier ($K_{\text{sel}} = 26$).

B. Proposed rolling variance feature representation

We pro- pose rolling variance as an alternative to both raw amplitude and phase sanitisation. The key insight is that human activity modulates the temporal variability of CSI amplitude rather than its absolute level. By computing a local variance over a sliding window, we (i) suppress slow-varying environmental trends that cause domain drift, (ii) amplify the transient fluctuations caused by human motion, and (iii) avoid the computational overhead and instability of sparse delay-domain reconstruction required by phase sanitisation.

For a CSI amplitude time series $x[n]$ on a single subcarrier, the rolling variance with window W is

$$\sigma_W^2[n] = \frac{1}{W}\sum_{i=n-W+1}^{n} x[i]^2 - \left(\frac{1}{W}\sum_{i=n-W+1}^{n} x[i]\right)^2 \tag{5}$$

The rolling variance acts as a high-pass envelope detector: slow baseline drift (period $\gg W/f_s$) is suppressed because

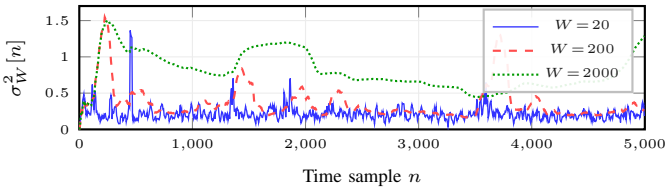

Fig. 4. Rolling-variance profiles from real ESP32-C6 CSI data (Office HAR, "work" activity, subcarrier 16). $W = 20$ (≈133 ms at 150 Hz) preserves fast transients and gesture boundaries; $W = 200$ (≈1.33 s) captures sustained energy envelopes; $W = 2000$ (≈13.3 s) over-smooths and loses temporal detail.

it contributes negligibly to within-window variance, while activity-induced fluctuations at the gesture time-scale are amplified.

We evaluate three window sizes: $W = 20$ (≈133 ms at 150 Hz) to capture fast transient motions and gesture boundaries, $W = 200$ (≈1.33 s) to capture sustained posture changes, and $W = 2000$ (≈13.3 s) to test the effect of extreme smoothing. In practice, we compute σ^2_W via cumulative sums of $x[n]$ and $x[n]^2$, yielding $O(N)$ complexity per subcarrier. Fig. 4 shows real rolling-variance profiles from the Office HAR dataset (subcarrier 16, "work" activity).

V. Analysis and Evaluation

1) Experiment Pipelines: We evaluate two families of classifiers with increasing model capacity:

a) Traditional ML: Four feature pipelines—amplitude, amplitude+phase, amplitude+sanitised phase, and rolling variance—are classified by RandomForest (100 trees) and XGBoost (500 estimators).

b) Deep Learning: Three architectures—MLP, 1D-CNN (Fig. 8), and 1D-Conv-LSTM—are trained with rolling-variance features using $W \in \{20, 200, 2000\}$ and window lengths $L \in \{500, 1000, 2000\}$ to study the joint effect of variance window size and temporal context on classification performance.

A. Dataset Characterization

Table IV profiles each CSI dataset alongside reference benchmarks using Silhouette score (cluster cohesion, $[-1, +1]$) and Fisher Discriminant Ratio ($\mathrm{tr}(\mathbf{S}_B)/\mathrm{tr}(\mathbf{S}_W)$), computed after PCA-50 projection. Fig. 5 visualises the 2D projections.

Office Localization is the easiest task (Sil. 0.80, Fisher 23.45), while Home HAR is the hardest (Sil. −0.16, Fisher 0.16), with class overlap comparable to CIFAR-10/100. This spread implies that simpler classifiers suffice for well-separated tasks, while fine-grained activity recognition requires deeper models.

B. Evaluation Metrics

We report two families of metrics: *data-characterisation* metrics (computed on the feature space to quantify dataset difficulty) and *prediction-analysis* metrics (computed on classifier outputs to quantify model quality).

1) Data-Characterisation Metrics:

TABLE IV
Dataset characterization (amplitude, $W = 500$, stride $= 100$, 150 Hz). Metrics computed on balanced training data after PCA-50. Sil. (↑), Fisher (↑).

Dataset	# Cls	Sil.	Fisher
Home HAR	7	−0.155	0.164
Home Occ.	3	0.412	0.594
Office HAR	4	0.407	4.030
Office Loc.	4	0.800	23.446
Iris	3	0.513	6.632
MNIST	10	0.063	0.334
CIFAR-10	10	−0.058	0.088
CIFAR-100	100	−0.073	0.031

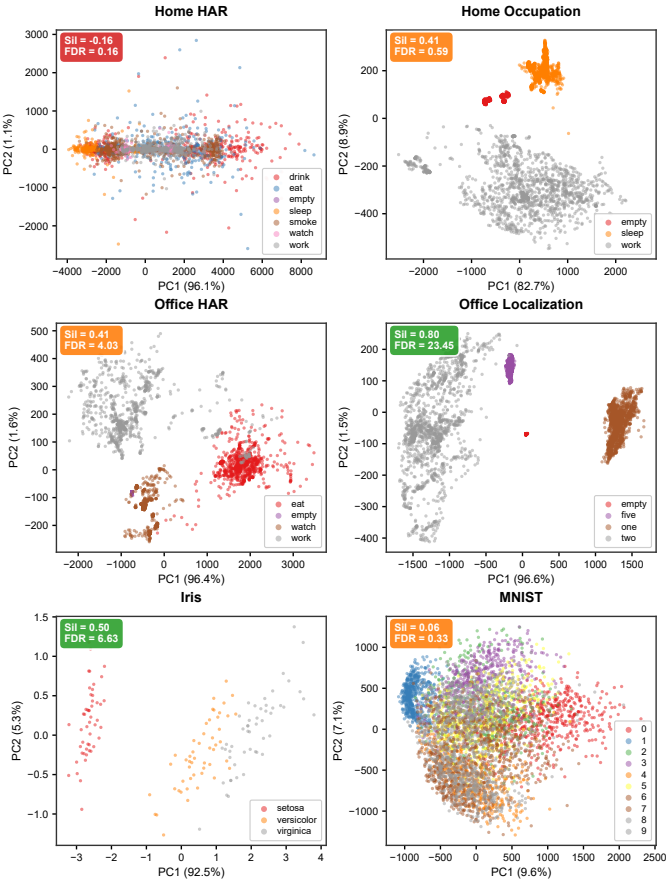

Fig. 5. 2D PCA projections of all CSI and reference datasets, coloured by class label. Each subplot is annotated with the Silhouette score (Sil., range $[-1, +1]$) and Fisher Discriminant Ratio (FDR).

a) Silhouette Score: For sample i in cluster C_i, let $a(i)$ be its mean intra-cluster distance and $b(i)$ the smallest mean

inter-cluster distance. The Silhouette score is

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i),\, b(i)\}}, \quad \bar{s} = \frac{1}{N}\sum_{i=1}^{N} s(i) \tag{6}$$

Range: $[-1, +1]$; where higher means better-separated clusters.

b) Fisher Discriminant Ratio:

$$\text{FR} = \frac{\text{tr}(\mathbf{S}_B)}{\text{tr}(\mathbf{S}_W)} \tag{7}$$

where $\mathbf{S}_B$ is the between-class scatter matrix and $\mathbf{S}_W$ the within-class scatter. Where higher indicates greater linear separability.

2) Prediction-Analysis Metrics: We report accuracy.

VI. Results and Discussion

In this section, we present the results for the conducted experiments. First, we compare the proposed rolling variance based feature representation method with SHARP's phase sanitization and report the accuracy of ML models trained on data preprocessing using each of the two methods. Second, having established that ML models combined with the proposed method achieve better performance compared with other preprocessing methods, we focus our attention to choosing the best learning paradigms. We compare three different DL architectures: Multi-Layer Perceptron (MLP), Convolutional Neural Network (CNN), and 1-dimensional Convolutional Layer(Conv1D)+ Long Short-Term Memory (LSTM). Next, we perform ablation experiments on the best-performing model, varying both the rolling variance window W and the CSI window length L. Fourth, we evaluate the resource consumption for each preprocessing method, showing the significant overhead introduced by phase sanitization compared to our lightweight method. We also compare the training time across the considered learning paradigms (ML/DL). Finally, we discuss and summarize our findings.

A. Feature Representation

We evaluate our proposed rolling variance-based feature representation method against SHARP phase sanitization [1] and unprocessed data (raw amplitude and raw amplitude + raw phase). In this first experiment, only two ML models are considered: Random Forests and XGBOOST, as they are powerful models that usually outperform other ML models. The rolling variance window length was fixed at 20, but three CSI window length are considered: 500, 1000, and 2000. Table V shows the accuracy of ML models across the four CSI datasets. Only the best accuracy across the three considered CSI window lengths is shown. On the hardest dataset, Home HAR, rolling variance yields a 1.6x accuracy improvement over other methods (46.3% vs 29.2%) and +10% improvement on the Office HAR dataset (93.3% vs 83.3%). For the datasets Home Occupancy and Office Localization, our proposed method perform slighly worse than other methods (2% accuracy loss). Surprisingly, SHARP-based sanitization does not provide consistent gain over raw amplitude despite the strong theoretical foundation and the impressive gains reported by the authors. However, the SHAPR sanitization method was proposed for devices with MIMO support, since the ESP32 device used in this study has only one antenna, it is possible that this descrepency in hardware properties causes phase sanitization to almost provide no performance advantage over raw amplitude.

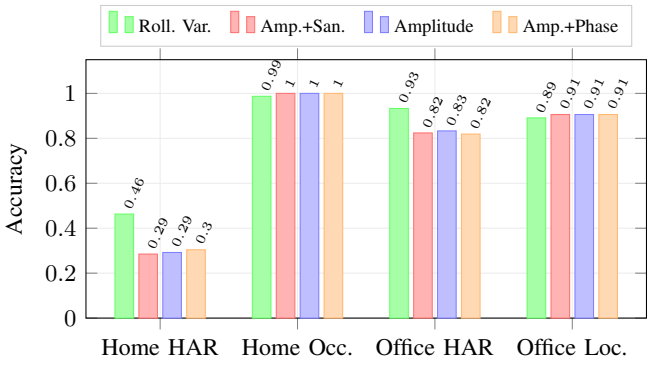

Fig. 6. Best ML accuracy per feature pipeline across all four datasets (best of RandomForest/XGBoost across all window lengths L).

Figure 6 shows the performance comparison between the proposed rolling variance-based processing and SHARP-based phase sanitization [1]. Rolling variance outperforms all phase-based pipelines on Home HAR (+17.1% over amplitude) and Office HAR (+10.0%), while performing comparably on the easier tasks. Notably, SHARP-based sanitisation provides no consistent gain over raw amplitude.

B. Learning Paradigm Selection

In the previous section, we discussed the accuracy achieved by two ML models for each data processing pipeline. In this section, we test DL models' performance, focusing on rolling variance processing only. Three DL model architectures are considered: MLP, which accepts flattened input data, 1D-CNN, and 1D-Conv-LSTM, both of which accepts sequential input data. We apply each of the three model architectures to three variance window sizes: $W \in \{20, 200, 2000\}$ crossed with three CSI window lengths: $L \in \{500, 1000, 2000\}$, yielding 27 different configurations per dataset. Table VI shows the full DL results (best window length L per configuration. We observe that DL models achieve higher performance across all datasets compared with ML models, as shown in Figure 7. The best overall model is the 1D-CNN, depicted in figure 8, which achieves the best accuracy on Home HAR (52%).

C. Ablation Study

Based on the results reported in the previous section, we conduct ablation experiments using the 1D-CNN model, which is the best model overall, where we vary both the rolling variance window W and the CSI window length L. Part of this comparison is already reported in table VI, where we show the performance per variance window length W across datasets.

Tables VII show the performance of the 1D-CNN model using different variance window lengths, and table VIII shows the accuracy of the same model using different CSI window lengths while fixing variance window $W = 20$.

TABLE V
ML RESULTS: BEST ACCURACY ACROSS WINDOW LENGTHS $L \in \{500, 1000, 2000\}$ PER DATASET, PIPELINE, AND MODEL.

Home HAR (7 cls)				Home Occ. (3 cls)				Office HAR (4 cls)				Office Loc. (4 cls)			
Pipeline	**Mdl**	L^*	**Acc**	**Pipeline**	**Mdl**	L^*	**Acc**	**Pipeline**	**Mdl**	L^*	**Acc**	**Pipeline**	**Mdl**	L^*	**Acc**
Amplitude	RF	2k	.269	Amplitude	RF	500	**1.00**	Amplitude	RF	500	.815	Amplitude	RF	500	**.906**
Amplitude	XGB	1k	.292	Amplitude	XGB	500	**1.00**	Amplitude	XGB	2k	.833	Amplitude	XGB	1k	.904
Amp.+Phase	RF	2k	.289	Amp.+Phase	RF	500	**1.00**	Amp.+Phase	RF	500	.811	Amp.+Phase	RF	500	**.906**
Amp.+Phase	XGB	1k	.304	Amp.+Phase	XGB	500	**1.00**	Amp.+Phase	XGB	500	.819	Amp.+Phase	XGB	500	.844
Amp.+San.	RF	2k	.285	Amp.+San.	RF	500	**1.00**	Amp.+San.	RF	500	.815	Amp.+San.	RF	500	**.906**
Amp.+San.	XGB	500	.274	Amp.+San.	XGB	500	**1.00**	Amp.+San.	XGB	500	.824	Amp.+San.	XGB	500	.844
Roll. Var.	RF	500	**.463**	Roll. Var.	RF	2k	.978	Roll. Var.	RF	2k	**.933**	Roll. Var.	RF	500	.891
Roll. Var.	XGB	2k	.446	Roll. Var.	XGB	500	.987	Roll. Var.	XGB	1k	.908	Roll. Var.	XGB	500	.868

TABLE VI
DL RESULTS (ROLLING-VARIANCE PIPELINE): BEST ACCURACY ACROSS WINDOW LENGTHS $L \in \{500, 1000, 2000\}$ PER DATASET, VARIANCE WINDOW W, AND ARCHITECTURE.

		Home HAR		**Home Occ.**		**Office HAR**		**Office Loc.**	
W	**Arch.**	L^*	**Acc**	L^*	**Acc**	L^*	**Acc**	L^*	**Acc**
20	1D-CNN	1k	**.529**	1k	.993	1k	**.942**	2k	.957
	1D-Conv-LSTM	500	.510	2k	**1.00**	500	.933	1k	**.967**
	MLP	1k	.351	500	.958	500	.836	1k	.890
200	1D-CNN	2k	.513	2k	**1.00**	500	.941	1k	.957
	1D-Conv-LSTM	2k	.525	1k	**1.00**	500	.933	500	.889
	MLP	1k	.350	500	.989	500	.882	1k	.924
2000	1D-CNN	500	.355	1k	.996	500	.824	1k	.934
	1D-Conv-LSTM	500	.468	500	.994	500	.924	500	.916
	MLP	500	.143	500	.989	500	.483	1k	.935

Fig. 7. Comparing learning paradigms (ML vs DL). Best architecture and window length L for each rolling-variance window W.

We observe that $W = 2000$ consistently underperforms the shorter window lengths across all datasets, confirming that excessive smoothing destroys temporal structure. As for the CSI window lengths L, we see that the intermediate window length $L = 1000$ achieves the best accuracy, with the other two window lengths also achieving close accuracy. This suggests that the impact of varying the CSI window length L on performance is less critical than changing the variance window length W.

D. Resource Consumption

Table IX profiles the computational cost of the proposed rolling variance CSI processing vs SHARP-based phase sanitization. Rolling variance is consistently faster regardless of

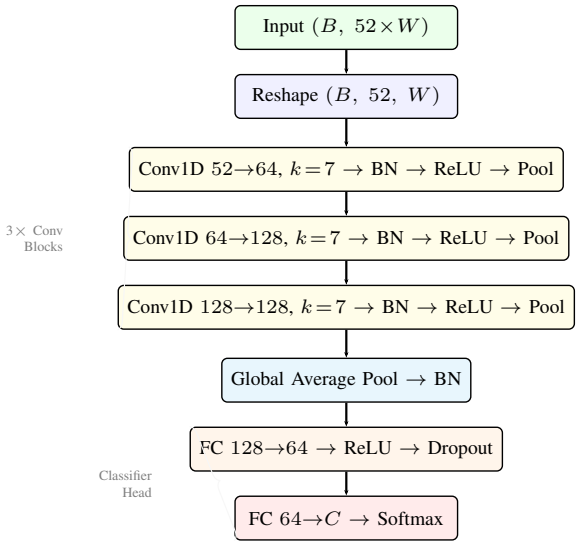

Fig. 8. Conv1D architecture. The flat input is reshaped to (subcarriers, time), processed by three stacked Conv1D blocks (each: convolution, batch normalisation, ReLU, max-pooling), globally average-pooled, and classified by two fully connected layers.

TABLE VII
ABLATION: ROLLING-VARIANCE WINDOW W (1D-CNN, BEST L).

W	Home HAR	Home Occ.	Off. HAR	Off. Loc.
20	**0.529**	0.993	**0.942**	**0.957**
200	0.512	**1.000**	0.941	0.957
2000	0.355	0.996	0.824	0.933

TABLE VIII
ABLATION: WINDOW LENGTH L (1D-CNN, $W=20$).

L	Home HAR	Home Occ.	Off. HAR	Off. Loc.
500	0.507	0.984	0.937	0.943
1000	**0.529**	**0.993**	**0.942**	0.953
2000	0.504	0.982	0.917	**0.957**

TABLE IX
PREPROCESSING BENCHMARK (52 SUBCARRIERS, 10,000 PACKETS).

Method	Wall (s)
Rolling Var. ($W=20$)	0.031
Rolling Var. ($W=200$)	0.027
Rolling Var. ($W=2000$)	0.027
SHARP Sanitisation	3.508

window size W. SHARP-based phase sanitization is over 250x slower, highlighting the light footprint of rolling variance as a CSI processing method, especially for real-time analysis. Table X compares the training and inference times for four representative models on the Home HAR dataset. Notably, XGBoost is the slowest at 1,796 s due to boosting 500 estimators. DL models achieve moderate training times at 185 and 245 s for 1D-CNN and MLP, respectively. All models achieve sub-second inference.

TABLE X
LEARNING BENCHMARK ON HOME HAR (ROLLING VARIANCE, $W=20$, $L=300$).

Model	Train (s)	Infer (s)
RandomForest	20.38	0.36
XGBoost	1795.80	0.34
1D-CNN	185.12	0.89
MLP	245.90	0.39

E. Discussion

Our results show that the proposed rolling variance-based CSI processing method achieves a 1.6x accuracy improvement over other methods (46.3% vs 29.2%) on the hardest dataset (Home HAR).

We make the following observations on WiFi sensing model training:

1) Architecture-wise, the performance comparison shows that 1D-CNN and 1D-Conv-LSTM consistently outperform MLP by exploiting temporal structure.
2) Increasing model capacity yields accuracy gains on harder tasks: ML → DL improves Home HAR from 46.3 to 52.9%. On easier tasks (Home Occ., Office datasets),simpler models perform well.

VII. CODE AVAILABILITY

The collected datasets are made available as well as the code for dataset analysis, machine learning and deep learning training. All of the files can be accessed on the project page: https://gadm21.github.io/WifiSensingESP32HAR/

VIII. CONCLUSION

We presented a robust, resource-efficient WiFi sensing data processing method based on a rolling-variance feature transform applied to CSI amplitude. To evaluate the performance of this method, four WiFi sensing labeled datasets are collected spanning two environments and three challenging smart home tasks. Our proposed method showed superior performance compared to previous resource-intensive phase sanitization and baseline methods, achieving up to 1.6x higher accuracy at a 250x computational discount.

Our analysis highlight the importance of choosing the variance window length to strike a balance between smoothing out environment-dependent noise while maintaining the discriminative temporal structure.

Compared to ML models, the moderate training cost of DL models, combined with their superior performance, makes them a better choice for WiFi sensing applications deployment on edge devices.

Future work shall investigate devising better processing and learning pipelines, and improving cross-environment drift mitigation using techniques like transfer learning and Test Time Adaptation (TTA).

REFERENCES

[1] F. Meneghello, D. Garlisi, N. Dal Fabbro, I. Tinnirello, and M. Rossi, "Sharp: Environment and person independent activity recognition with commodity ieee 802.11 access points," *IEEE Transactions on Mobile Computing*, vol. 22, no. 10, pp. 6160–6175, 2022.

[2] J. Schäfer, B. R. Barrsiwal, M. Kokhkharova, H. Adil, and J. Liebehenschel, "Human activity recognition using csi information with nexmon," *Applied Sciences*, vol. 11, no. 19, p. 8860, 2021.

[3] T. Li, C. Shi, P. Li, and P. Chen, "A novel gesture recognition system based on csi extracted from a smartphone with nexmon firmware," *Sensors*, vol. 21, no. 1, p. 222, 2020.