EAI Endorsed Transactions
on Tourism, Technology and Intelligence
Research Article EAI.EU

Efficient Machine Learning for Wi-Fi CSI-based Human Activity Recognition Using Fast Monte Carlo based Feature Extraction

Emelia Logah[1]

[1]Memorial University, Canada, Email: elogah@mun.ca

Abstract

High-dimensional doppler data extracted from Wi-Fi channel state information (CSI) offers distinctive velocity and time patterns that are useful for human activity recognition (HAR), but its scale poses significant challenges for real-time inference and deployment on resource-constrained devices. This work proposes an efficient, fast monte carlo (MC) feature selection framework based on the frieze-kannan-vempala (FKV) algorithm and coefficient estimation to address this bottleneck. The CSI is preprocessed, and doppler traces are computed to encode the velocity and direction of distinct activities. Afterwards, we perform FKV to decompose the doppler data, and the coefficient of the resulting singular vectors is estimated. Using rejection sampling, the topmost features are selected on the basis of their weights, thereby reducing the size of our features. The method identifies a compact set of velocity-time features that preserve critical motion information while significantly reducing computational overhead. The experimental evaluations demonstrated that the decision tree classifier achieved the highest precision at 99.8%, followed by convolutional neural networks (CNN) 96%, the hybrid CNN-long-short-term memory (CNN-LSTM) achieved 87%, while the LSTM model lagged at 53%. These results demonstrated that the integration of fast MC-based feature selection significantly reduced computational overhead without sacrificing classification performance, making it suitable for scalable and real-time HAR applications.

Received on 30 May 2025; accepted on 09 July 2025; published on 16 July 2025

Keywords: Wi-Fi Sensing, Doppler, FKV, CSI, SVD

doi:10.4108/eettti.9442

1. Introduction

Human activity recognition (HAR) has traditionally relied on sensing technologies such as wearable sensors and camera-based systems [1, 2]. Although these approaches can capture human activity, they have many limitations, including user compliance, privacy concerns, and obstructed signals by environmental features [3].

The evolution of WiFi has enabled standards such as the IEEE 802.11ac for WiFi 5 to support higher bandwidth and multi-user multiple-input-multiple-output (MU-MIMO), enabling better performance. Wi-Fi 6 (IEEE 802.11ax), the most recent standard, further optimized network efficiency with orthogonal frequency division multiple access (OFDMA) and enhanced spatial reuse, and integrates sensing and communication functions [4]. In OFDMA, the available bandwidth is divided into smaller sections called resource units (RUs), which allows multiple devices to transmit data over multiple frequency subcarriers simultaneously to ease network congestion and better use the available spectrum [5, 6]. The advancement of wireless sensing technologies has introduced a transformative shift in the field of HAR [7]. Wi-Fi-based sensing has stood out for its ability to seamlessly integrate into existing infrastructure and offer a contactless, privacy-preserving alternative using the received signal strength indicator (RSSI) for activity patterns [7]. Although RSSI provided foundational information, its sensitivity to environmental changes has limited its scalability and overall accuracy.

---

*Corresponding author. Email: elogah@mun.ca

The ubiquitous applications of Wi-Fi-based sensing technology have attracted considerable attention from industry and academic communities through channel state information (CSI). CSI captures both amplitude and phase changes across subcarriers and antenna pairs, offering a fine-grained view of signal propagation affected by human motion. Variations in the CSI matrix encode information about the amplitude and phase of the received signal, which are critical to classifying human activities. The CSI provides spatial and temporal information about signal propagation, reflecting changes caused by human motion [4, 8]. The adoption of these technologies has catalyzed research in various domains, including healthcare, smart homes, and security, demonstrating the versatility of Wi-Fi-based sensing in detecting human activities in various environments [2, 9, 10]. Recent research also attempted to improve model accuracy by analyzing the time-varying components of CSI, particularly the phase information, to extract doppler shifts that reveal velocity-related features of moving subjects. Doppler information visually represents human movement by capturing micro-motion patterns such as arm swings or leg strides, resulting in high-resolution, time-frequency representations called doppler traces. However, using doppler-based CSI features for HAR introduces a new challenge of high-dimensional input data. This increases computational demands, makes models prone to overfitting, and hinders real-time deployment, especially on resource-constrained devices.

This paper proposes an efficient feature selection framework based on the frieze-kannan-vempala (FKV) fast monte carlo (MC) algorithm. This method employs length-squared (LS) sampling to construct a compact submatrix. It performs approximate singular value decomposition (SVD) to extract a low-rank approximation of the doppler CSI matrix, then coefficient estimation enabling dimensionality reduction without sacrificing the temporal and spatial information critical for activity classification. Our approach significantly reduces the number of trainable parameters in machine learning (ML) models, making them more lightweight and deployable in real-world environments.

2. Related Works

This section reviews some relevant literature in three core areas: ML architectures for HAR, dimensionality reduction and feature extraction, and randomized algorithms for matrix sketching.

2.1. Machine Learning for HAR Models

Recent advances in Wi-Fi-based HAR have enabled the use of CSI for device-free activity detection. However, raw CSI is often noisy and unstable due to phase distortions. Phase sanitization techniques have been widely adopted to mitigate these artifacts and improve the consistency of CSI. Studies such as [8, 9] introduced preprocessing pipelines to correct phase offset and enhance motion sensitivity, while [4, 11] demonstrated that phase sanitization leads to robust classification in obstructed or dynamic environments. Several works have incorporated doppler processing to extract motion-specific features from CSI to address environmental variability and occlusions. Doppler-based signatures help encode subtle velocity and gesture differences, enabling more fine-grained activity recognition to better distinguish overlapping or rapid motions [10, 12]. Researchers have applied various ML architectures to model activity from cleaned CSI data. For example, convolutional neural networks (CNN) and the recurrent neural network (RNN) have been used to capture spatial and temporal activity patterns, achieving high accuracy in dynamic scenarios [13]. Hybrid architectures like bidirectional long short-term memory with convolutional layers (BLSTM-CNN) have improved performance by capturing both frequency-localized features and time-dependent behavior [14, 15]. In addition to supervised models, unsupervised and semi-supervised methods have been explored to reduce reliance on labeled data. For instance, [13] proposed an unsupervised deep network to extract discriminative features from CSI without manual annotation. Meanwhile, more advanced structures such as inception-attention networks have been employed to enhance feature extraction in cluttered environments, improving classification accuracy in complex activity scenarios [16, 17].

2.2. Feature Selection and Dimensionality Reduction

Feature extraction and dimensionality reduction are essential to mitigate the curse of dimensionality and enhance the performance of the ML models, hence the need for feature extraction and dimensionality reduction. Feature selection aims to identify the most informative features while eliminating redundant or noisy attributes. Principal component analysis (PCA) has been widely used for dimensionality reduction in domains such as medical diagnostics, imaging, and signal processing, projecting data into directions of maximum variance [18–21]. However, PCA is unsupervised and does not account for class discriminability, often discarding critical features for classification. This makes it suboptimal for HAR tasks where activity-specific motion cues may have low variance but high predictive value. SVD is another tool that has been used in tasks such as image fusion, biometric identification, and collaborative filtering [18, 20, 22]. Advanced variants such as randomized SVD have been developed to reduce computational complexity and improve scalability [23]. CUR decomposition methods follow a similar strategy by approximating a matrix using sampled columns and rows [24]. Another research further extends this by iteratively decomposing matrices into low-rank and sparse components, which proves useful in noisy domains like hyperspectral imaging [25]. These randomized methods have shown effectiveness in reducing the complexity of linear algebra problems and have been explored in applications such as stochastic regression and semidefinite programming [26–28].

2.3. Overview of SVD and FKV

SVD expresses any real-valued matrix $A \in \mathbb{R}^{m\times n}$ as the product of three matrices and is given by

$$A = U\Sigma V^T, \tag{1}$$

where $A \in \mathbb{R}^{m\times n}$ is the original real-valued matrix to be decomposed; $U \in \mathbb{R}^{m\times m}$ is an orthogonal matrix whose columns are the left singular vectors of A, forming an orthonormal basis for its row space; $\Sigma \in \mathbb{R}^{m\times n}$ is a diagonal matrix whose non-negative entries are the singular values of A, representing the magnitude or energy of each component; and $V \in \mathbb{R}^{n\times n}$ is an orthogonal matrix whose columns are the right singular vectors of A, forming an orthonormal basis for its column space. Each matrix plays a distinct role in revealing the structure of A. We define the matrix A

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \in \mathbb{R}^{m\times n}. \tag{2}$$

The matrix $U \in \mathbb{R}^{m\times m}$ is an orthogonal matrix whose columns, denoted $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m$, are the left singular vectors of A. These vectors form an orthonormal basis for the row space of A, and satisfy $\mathbf{u}_i^T\mathbf{u}_j = 0$ for $i \neq j$, and $\mathbf{u}_i^T\mathbf{u}_i = 1$. The matrix has the form

$$U = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1m} \\ u_{21} & u_{22} & \cdots & u_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ u_{m1} & u_{m2} & \cdots & u_{mm} \end{bmatrix}. \tag{3}$$

The matrix $\Sigma \in \mathbb{R}^{m\times n}$ is a diagonal matrix containing the singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$, where $r = \text{rank}(A)$, and all other entries are zero. These singular values quantify the contribution of each singular vector in explaining the variance or energy present in A. The matrix Σ takes the form

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_r \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}. \tag{4}$$

The matrix $V \in \mathbb{R}^{n\times n}$ is an orthogonal matrix whose columns $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ are the right singular vectors of A. These vectors form an orthonormal basis for the column space of A, satisfying $\mathbf{v}_i^T\mathbf{v}_j = \delta_{ij}$, where δ_{ij} is the Kronecker delta. The matrix V has the structure

$$V = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ v_{n1} & v_{n2} & \cdots & v_{nn} \end{bmatrix}. \tag{5}$$

Together, U, Σ, and V provide a geometric interpretation of the original matrix A as a transformation involving rotation, scaling, and another rotation. This makes SVD a powerful tool for dimensionality reduction, denoising, and low-rank approximations. In radar signal processing, for instance, SVD enables clutter suppression by filtering out components corresponding to low-energy singular values [29]. It is also effective for robust target detection in sea clutter environments, where features such as the maximum singular value and spectral norms extracted from Σ are highly discriminative [30]. Together, U, Σ, and V provide a geometric interpretation of the original matrix A as a transformation involving rotation, scaling, and another rotation. Moreover, the scalability of SVD has enabled its deployment in cloud-based frameworks, where it facilitates outsourced computation for large-scale matrix decomposition tasks [31]. This makes SVD a powerful tool for dimensionality reduction, denoising, and low-rank approximations.

Although SVD provides the optimal low-rank approximation of a matrix in terms of the Frobenius norm, its computational complexity typically $\mathcal{O}(mn \cdot \min(m, n))$ makes it computationally expensive for large-scale data. To address this challenge, a study proposed a randomized algorithm that offers a fast and scalable alternative to traditional SVD by approximating the top$k$ singular values using a sampling-based approach [32]. The FKV algorithm assumes that matrix entries can be sampled with probabilities proportional to their squared magnitudes, which is reasonable in many real-world applications after a single data pass. It selects a small number of rows and columns based on this probability distribution, forming a sketch matrix whose low-rank structure approximates that of the original matrix. The core result guarantees that with high probability, the algorithm outputs a rank-k matrix $\widetilde{A}$ such that

$$\|A - \widetilde{A}\|_F \le \min_{\mathrm{rank}(D)\le k} \|A - D\|_F + \varepsilon\|A\|_F, \tag{6}$$

where $A \in \mathbb{R}^{m\times n}$ denotes the original high-dimensional data matrix, and $\widetilde{A} \in \mathbb{R}^{m\times n}$ is its low-rank approximation obtained through a randomized algorithm, $\|\cdot\|_F$ denotes the Frobenius norm, D represents any matrix of rank at most k, and the term $\min_{\mathrm{rank}(D)\le k} \|A - D\|_F$ denotes the best possible error in approximating A by a rank-k matrix, typically achieved via truncated SVD. The scalar $\varepsilon > 0$ is the additive error tolerance, and the term $\varepsilon\|A\|_F$ provides a bound on how far the approximation $\widetilde{A}$ may deviate from the optimal low-rank approximation. This inequality ensures that the approximation error of $\widetilde{A}$ remains within a controlled additive margin of the best achievable rank-k error, with failure probability at most δ, where $\varepsilon > 0$ is the approximation error tolerance. Crucially, the runtime of the algorithm is polynomial in k, $1/\varepsilon$, and $\log(1/\delta)$, but independent of the dimensions of the matrix m and n, making it highly efficient for high-dimensional data. The method essentially reduces the problem to computing the SVD of a small sketch, bypassing the need to fully decompose the original matrix. Given its efficiency and provable accuracy guarantees, the FKV algorithm has become a key component in large-scale matrix approximation pipelines, including applications in feature selection, recommendation systems, and latent semantic indexing [32]. Generally, the algorithm proceeds as follows:

Length-Squared (LS) Probabilities and Row Rescaling. The FKV algorithm uses a two-level sampling strategy to identify the most informative rows and columns in a high-dimensional matrix $A \in \mathbb{C}^{m\times n}$. This approach is based on computing LS probabilities, which assign higher sampling probabilities to rows and columns that contribute more significantly to the overall energy of the matrix.

Row Norm Calculation. The squared euclidean norm also called the ℓ_2 norm of the i-th row of matrix A is computed as

$$\|A_{i,:}\|_2^2 = \sum_{j=1}^{n} |A_{ij}|^2, \tag{7}$$

where $A_{i,:}$ denotes the entire i-th row of the matrix A, n is the number of columns, and $|A_{ij}|^2$ is the squared magnitude of the complex entry in row i, column j.

Frobenius Norm of the Matrix. The total Frobenius norm of matrix A is defined as

$$\|A\|_F^2 = \sum_{i=1}^{m}\sum_{j=1}^{n} |A_{ij}|^2, \tag{8}$$

Here m is the number of rows, n is the number of columns, and $\|A\|_F^2$ represents the total energy (sum of squared magnitudes) of all entries in A.

Row Sampling Probability

Using the above quantities, the probability of selecting the row i is computed as

$$P_{\mathrm{r}}(i) = \frac{\|A_{i,:}\|_2^2}{\|A\|_F^2}. \tag{9}$$

This probability reflects the relative contribution of the row i to the total energy of the matrix. Rows with higher norms are more likely to be selected.

Column Sampling Probability within a Row

Once a row i is sampled, a second-level distribution is computed to sample columns from within that row. The probability of selecting column j from row i is given by

$$P_{ij} = \frac{|A_{ij}|^2}{\|A_{i,:}\|_2^2}, \tag{10}$$

where $|A_{ij}|^2$ is the squared magnitude of the element at position (i, j), and $\|A_{i,:}\|_2^2$ is the squared norm of the sampled row.

Row Rescaling. To correct for the bias introduced by non-uniform sampling, each sampled row is rescaled to maintain an unbiased approximation of the matrix. Let r be the total number of rows sampled. For each sampled row index i_s, the corresponding rescaled row R_s is computed as

$$R_s = \frac{\|A\|_F}{\sqrt{r}\cdot\|A_{i_s,:}\|_2} \cdot A_{i_s,:}, \tag{11}$$

where $R_s \in \mathbb{C}^{1\times n}$ is the rescaled version of the sampled row, $\|A_{i_s,:}\|_2$ is the norm of the sampled row, $\|A\|_F$ is the Frobenius norm of the original matrix, and r is the total number of rows sampled. This normalization ensures that each row in the resulting matrix $R \in \mathbb{C}^{r\times n}$ contributes equally in expectation.

Column Sampling from Matrix R. In the second stage of the FKV algorithm, a uniform sample of one of the r rescaled rows is selected. From this row, c columns are sampled using the same LS method as above. Each selected column is also rescaled similarly, to form a matrix $C \in \mathbb{C}^{r\times c}$. To compute the LS probability for a column directly from matrix A, it is defined as

$$P_{\mathrm{c}}(j) = \frac{\|A_{:,j}\|_2^2}{\|A\|_F^2}, \tag{12}$$

where $A_{:,j}$ is the j-th column of A, and $\|A_{:,j}\|_2^2$ is the squared norm of that column. This reflects how much column j contributes to the total energy of the matrix. This hierarchical sampling and rescaling process allows the FKV algorithm to construct a compressed sketch of the original matrix while preserving its dominant low-rank structure. It enables efficient approximation with strong theoretical guarantees.

Construction of Submatrix C. Given the large and high-dimensional nature of the matrix A, performing SVD directly would be computationally expensive and inefficient. To address this, the sampled rows and columns construct a submatrix C, which serves as a compressed representation of A. This submatrix encapsulates the most informative aspects of A, thereby facilitating efficient computation in subsequent steps:

$$C = A_{\mathrm{r,c}}, \tag{13}$$

where A_r, c represent the rows and columns sampled from A.

Singular Value Decomposition of C. Once the submatrix C is established, SVD is performed to decompose C into its singular vectors and singular values. This decomposition is expressed as

$$C = W \Sigma V^{\dagger}, \tag{14}$$

where $W \in \mathbb{C}^{m \times m}$ is a unitary matrix for the left singular vectors, $\Sigma \in \mathbb{R}^{m \times n}$ is a diagonal matrix with singular values, and $V^{\dagger} \in \mathbb{C}^{n \times n}$ is the conjugate transpose of the matrix for the right singular vectors. The singular values are derived from the SVD of the sampled submatrix C. By focusing on the most significant singular values, we achieve a low-rank approximation of the matrix that preserves critical information while significantly reducing computational complexity.

3. Dataset Aquisition

The dataset used in this study was collected as described in SHARP (sensing human activities through Wi-Fi radio propagation), and comprises seven distinct activities: empty space, arm gym, jumping in place, sitting (standing and sitting for some time), sitting continuously, running, and walking. The experimental setup was implemented in a bedroom with a desk in the middle to introduce natural obstructions, ensuring different signal propagation paths. This study utilized readily available Wi-Fi infrastructure to ensure practical deployment and ease of use. Specifically, two commercial IEEE 802.11ac Asus RTAC86U routers were used, one serving as a transmitter with a single antenna and the other as a receiver with four antennas. The system operated with an 80 MHz channel bandwidth, divided into 246 subcarriers, resulting in complex CSI samples for each subcarrier. Multiple participants were involved, each with varying physical characteristics such as height, weight, and movement styles. The activities were carried out in several sessions, covering different times of the day and environmental conditions to ensure that the data collected are independent of individual participants and the environment [8].

4. CSI Data Preprocessing

This section discusses the concept of CSI and the phase sanitization process. The received signal in an OFDM system $\mathbf{y}$ is expressed as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}, \tag{15}$$

where $\mathbf{H}$ represents the CSI matrix, $\mathbf{x}$ is the transmitted signal, and $\mathbf{n}$ denotes additive noise. Each element of $\mathbf{H}$ represents the amplitude and phase of a specific subcarrier, enabling the precise recognition of activity even in complex environments [8, 9]. In an OFDM-MIMO system, the CSI matrix is represented as

$$H(f,t) = \begin{bmatrix} H_{11}(f,t) & \cdots & H_{1N}(f,t) \\ \vdots & \ddots & \vdots \\ H_{M1}(f,t) & \cdots & H_{MN}(f,t) \end{bmatrix}, \tag{16}$$

where $H_{ij}(f,t)$ is the channel response for the i-th transmitting and j-th receiving antenna at frequency f and time t. This study specifically considers a Wi-Fi sensing setup consisting of one transmit antenna ($N_t = 1$) and four receive antennas ($N_r = 4$). We operate using an 80 MHz channel bandwidth comprising 246 subcarriers, providing detailed CSI characterization for each subcarrier-frequency pair. Consequently, the CSI matrix at any subcarrier frequency f and time t can explicitly be represented as

$$\mathbf{H}(f,t) = [H_{11}(f,t),\ H_{21}(f,t),\ H_{31}(f,t),\ H_{41}(f,t)]^T \in \mathbb{C}^{4\times 1}. \tag{17}$$

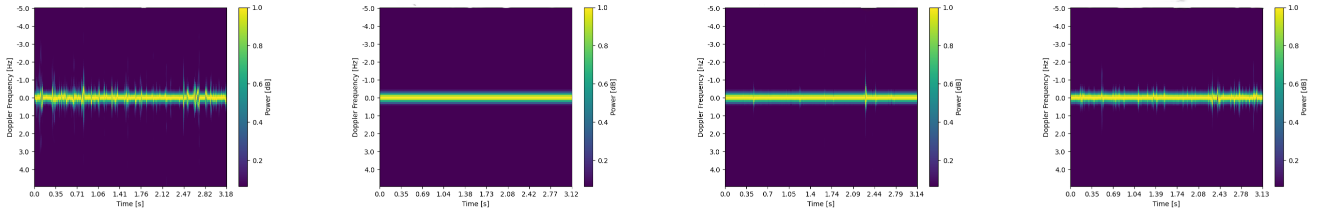
The CSI obtained is illustrated in Fig. 2(a), which shows the amplitude and phase information in four frames as a result of the four receive-transmit antenna pairs.

4.1. Phase Sanitization

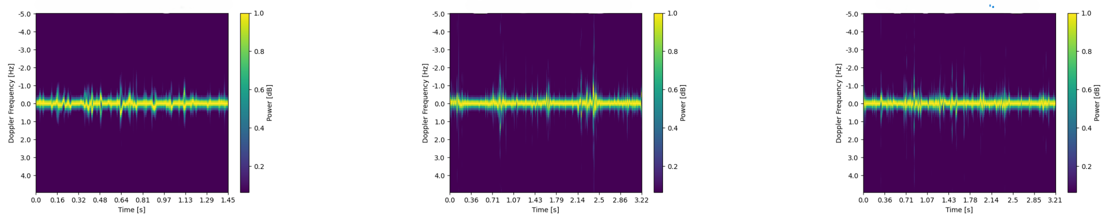
Phase sanitization mitigates offsets caused by signal propagation and hardware imperfections such as channel frequency offset (CFO), phase-locked loop offset (PLO), and phase ambiguity (PA), which are consistent across subchannels. Subchannel-dependent factors include sampling frequency offset (SFO) and packet detection delay (PDD). The steps involved in phase sanitization are expressed as in [8].

The phase offset at subchannel k, $\phi_{\mathrm{offs},k}$, is shown as follows:

$$\phi_{\mathrm{offs},k} = \frac{2\pi k + t_{\mathrm{SFO}} + t_{\mathrm{PDD}}}{T} + \phi_{\mathrm{CFO}} + \phi_{\mathrm{PLO}} + \phi_{\mathrm{PA}}. \tag{18}$$



(a) arm gym (b) empty (c) sitting (d) sitting continuously



(e) jumping in place (f) walking (g) running

Figure 1. Doppler spectrograms for various activities.

Using a reference path with reliably estimated parameters, the common phase offset is removed. The channel frequency response (CFR) vector for k subchannels is given by

$$h = [H_{-\frac{k}{2}}, \dots, H_0, \dots, H_{\frac{k}{2}}]^\top. \tag{19}$$

To separate P multipath contributions, a grid of $P' > P$ potential paths is defined, and P components are selected via

$$h = Tr, \tag{20}$$

where T $(K \times P')$ is defined by adding all delays $T_{p,\text{tot}} = t_P + t_{\text{SFO}} + t_{\text{PDD}}$:

$$T_k = \left[\frac{e^{-j2\pi k t_{o,\text{tot}}}}{T}, \dots, \frac{e^{-j2\pi k t_{p(o-1),\text{tot}}}}{T} \right]. \tag{21}$$

The vector r is modeled as

$$r = \begin{bmatrix} A_0 e^{-j2\pi f_c \tau_0} \\ \vdots \\ A_{P'-1} e^{-j2\pi f_c \tau_{P'-1}} \end{bmatrix}. \tag{22}$$

The minimization problem to estimate r is given by

$$r = \text{argmin}_{\tilde{r}} \|h - T\tilde{r}\|_2^2 + \lambda \|\tilde{r}\|_1. \tag{23}$$

CFR vector h and matrix T are respectively converted as follows:

$$h_{\text{est}} = [\text{Re}(h), -\text{Im}(h)]^\top, \tag{24}$$

$$T_{\text{est}} = \begin{bmatrix} \text{Re}(T) & -\text{Im}(T) \\ -\text{Im}(T) & \text{Re}(T) \end{bmatrix}. \tag{25}$$

The complex vector r is reconstructed as

$$r = r_{\text{ext}}[0 : P' - 1] + j\, r_{\text{ext}}[P' : 2P' - 1]. \tag{26}$$

The position p^* of the strongest path is shown as follows:

$$X_k = e^{-j2\pi k(\tau_{\text{SFO}}+\tau_{\text{PDD}})} e^{j(\phi_{\text{CFO}}+\phi_{\text{PLO}}+\phi_{\text{PA}})} \times \begin{bmatrix} A_0 e^{-j2\pi(f_c+k/T)\tau_0} \\ \vdots \\ A_{P'-1} e^{-j2\pi(f_c+k/T)\tau_{P'-1}} \end{bmatrix}. \tag{27}$$

The sanitized CFR is estimated as

$$\hat{H}_k \approx A_{p^*} e^{j2\pi(f_c+k/T)\tau_{p^*}} H_k, \tag{28}$$

where $\hat{H}_k$ represents the CFR estimate for subchannel k with the phase offset removed.

Fig. 2 illustrates the amplitude and phase responses of the CSI captured across multiple subcarrier indices over four consecutive frames, corresponding to dynamic human activities. The amplitude spectrum exhibits substantial variations and irregular fluctuations in the raw CSI data, as shown in Fig. 2(a). These variations indicate highly dynamic wireless channel conditions, characteristic of rapid human motion such as jumping. The raw phase plot accompanying the amplitude demonstrates significant noise and frequent abrupt discontinuities due to inherent phase wrapping between π and $-\pi$, a common artifact in raw CSI measurements exacerbated by multipath propagation and environmental noise. As seen in Fig. 2(b), the amplitude has been normalized after applying the phase sanitization process used in [8] to clearly highlight periodic peaks. This normalization step makes it easier to observe repetitive patterns in the data. Similarly, the sanitization process improves the phase information by removing discontinuities and reducing noise, resulting in smoother and more consistent trends across subcarriers. The smoother phase variations result from practical phase unwrapping and denoising techniques, significantly reducing discontinuities caused by phase wrapping and multipath reflections. Consequently, the sanitized data provides a more substantial basis for subsequent analysis and classification algorithms in wireless-based HAR systems.

4.2. Computation of Doppler Traces

Human movement introduces activity-specific variations in the CSI, but environmental obstructions can interfere, creating multipath propagation of signals. The delay of a signal along a given path is expressed as

$$t_p(n) = \frac{\ell_p + D_p(n)}{c}, \tag{29}$$

where ℓ_p is the static path length, $D_p(n)$ is the dynamic variation, and c represents the speed of light. The dynamic variation $D_p(n)$, due to moving objects, is approximated as

$$D_p(n) = v_p \cos a_p \cdot nT_c, \tag{30}$$

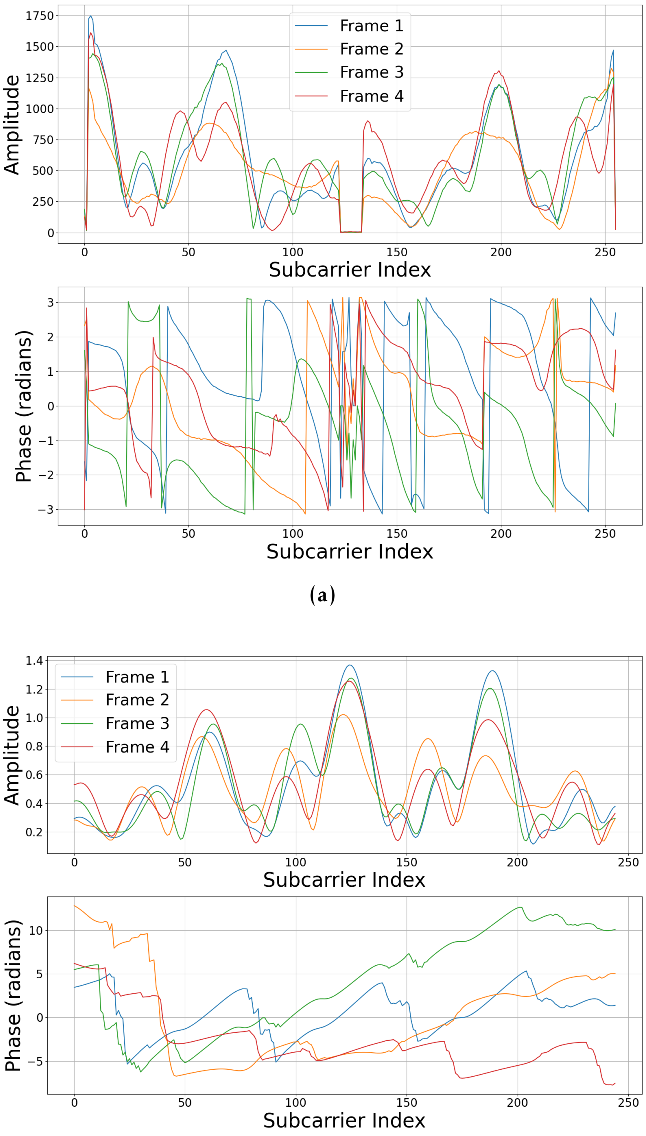
where v_p is the object's velocity, a_p is the angle of motion, and T_c is the sampling period. The velocity component is given by

$$v_p \cos a_p = \frac{u_c}{f_c T_c N_D}, \tag{31}$$

where u_c is the doppler shift, f_c is the carrier frequency, T_c is the sampling interval, and N_D is the number of doppler bins. Doppler traces are computed by dividing the signal into small time windows. For each window, the Doppler vector is defined as

$$d_i(u) = \sum_{k=-K/2}^{K/2-1} \left| F\{H_i\}(k,u) \right|^2, \tag{32}$$

where k represents the signal's frequency components, u is the doppler index, and $F\{\cdot\}$ denotes the fourier transform. By stacking these doppler vectors over time, a doppler spectrogram is generated that offers a detailed representation of how activity evolves [8]. The doppler plots, which distinctively represent each activity, are illustrated in Fig. 1.



(a)

(b)

Figure 2. Amplitude and phase visualizations of an activity for 4 frames across all subcarriers (a) represents the raw CSI and (b) is the sanitized phase.

5. Proposed Method

This study employs a feature extraction method using a fast MC sampling algorithm based on the FKV framework. The objective is to extract the most informative time-velocity features from high-dimensional doppler data while preserving structural information relevant to HAR. The method significantly reduces computational cost by selecting a compact yet discriminative subset of features.

The original doppler radar data is represented as a four-dimensional tensor, which is given by

$$A \in \mathbb{R}^{4356\times4\times100\times340}, \tag{33}$$

where 4356 is the number of Doppler samples, 4 is the number of antennas, 100 is the number of velocity bins, and 340 is the number of time steps. This tensor is reshaped into a two-dimensional matrix for processing

$$A \in \mathbb{R}^{4356\times 136000}, \tag{34}$$

where each sample is flattened across antennas, velocities, and time steps (i.e., $4 \times 100 \times 340 = 136000$). To avoid the computational burden of performing a full SVD on the high-dimensional matrix A we perform a low-rank approximation via FKV, using LS sampling to select rows and columns from A. The number of sampled rows r and columns c required to achieve an approximate low-rank representation with failure probability δ is given by

$$r, c \geq \frac{k}{\epsilon^2} \log\left(\frac{k}{\delta}\right), \tag{35}$$

where k is the target rank of the approximation, ϵ is the allowable approximation error, and δ is the failure probability.

In this work, we chose $r = c = 1000$ for computational efficiency, k is 3, and δ is 0.01. This corresponds to an empirical error of approximately $\epsilon \approx 0.131$. From the sampled rows and columns, we form a submatrix $C \in \mathbb{R}^{1000\times 1000}$ and compute its SVD in order to use the output to reconstruct approximations of the full singular vectors for A, which is given by

$$C = W\Sigma V^T, \tag{36}$$

where $W \in \mathbb{R}^{1000\times k}$ contains the left singular vectors, $V \in \mathbb{R}^{1000\times k}$ contains the right singular vectors, and $\Sigma \in \mathbb{R}^{k\times k}$ is the diagonal matrix of the top k singular values. After computing the SVD of C, we reconstruct the approximations of the singular vectors of A using the information extracted from C. These singular vectors summarize the main patterns in the Doppler features and are used to reconstruct approximations of the original matrix and guide feature selection. Specifically, we approximate the right singular vectors $\mathbf{v}^l \in \mathbb{R}^{136000}$ and the left singular vectors $\mathbf{u}^l \in \mathbb{R}^{4356}$ that would result from a full SVD of A. These are essential for projecting b and approximating x.

The approximate right singular vector $\mathbf{v}^l$ is constructed as a weighted combination of the normalized rows of A that were sampled during the formation of C, and this is expressed as

$$\mathbf{v}^l = \frac{\|A\|_F}{\sqrt{r}\sigma_l} \sum_{s\in\mathcal{S}_r} \frac{w_{s,l}}{\|A_{s,:}\|} A_{s,:}, \tag{37}$$

where $\mathcal{S}_r$ is the set of sampled row indices and $w_{s,l}$ is the s-th component of the l-th left singular vector obtained from the SVD of matrix C. The scaling factor involving σ_l, the l-th singular value of C, ensures consistency with the original magnitude of the data space. This process produces approximate vectors that span a low-dimensional subspace that captures the essential patterns in the original matrix A.

To determine how strongly each singular vector correlates with the label vector b, we compute coefficients λ_l by using

$$\lambda_l = \langle v^l, A^{\dagger} b\rangle, \tag{38}$$

where v^l is the l-th right singular vector, $A^{\dagger}$ is the pseudoinverse of A, and $b \in \mathbb{R}^{4356}$ is the activity label vector. This helps to incorporate the label information to find x, the solution vector as in the linear systems of equations, which is given by

$$Ax = b. \tag{39}$$

We construct a direction vector ω, which will guide the rejection sampling process in selecting the constructions of the solution vector x by emphasizing the directions aligned with the true structure of the label, b. The direction vector ω is then calculated as follows:

$$\omega = \sum_{l=1}^{k} \frac{\lambda_l}{\sigma_l^3} W_l, \tag{40}$$

where σ_l is the l-th singular value and W_l is the l-th column of W.

Rejection sampling is applied to select features that are highly aligned with the vector ω. The probability of accepting the j-th feature is given by

$$P_j \propto \left(\frac{\langle R_j, \omega\rangle}{\|R_j\|\|\omega\|}\right)^3, \tag{41}$$

where R_j is the j-th column vector in the sampled matrix. Features with low alignment are naturally rejected.

To further validate the importance of the selected features, we approximate the solution vector for the system $A\boldsymbol{x} = b$ by using

$$\tilde{x}_j = \sum_{l=1}^{k} \frac{\lambda_l}{\sigma_l^2} v_j^l, \tag{42}$$

where v_j^l is the j-th component of the l-th right singular vector. This approximation highlights the relative contribution of each feature in the reconstruction of the signal. The final reduced matrix $\tilde{A} \in \mathbb{R}^{4356\times 1000}$ is thus constructed by retaining the columns whose indices j are sampled with high probability based on their alignment with ω.

Fig. 4 presents a comparative visualization of the original data set and the data set obtained after

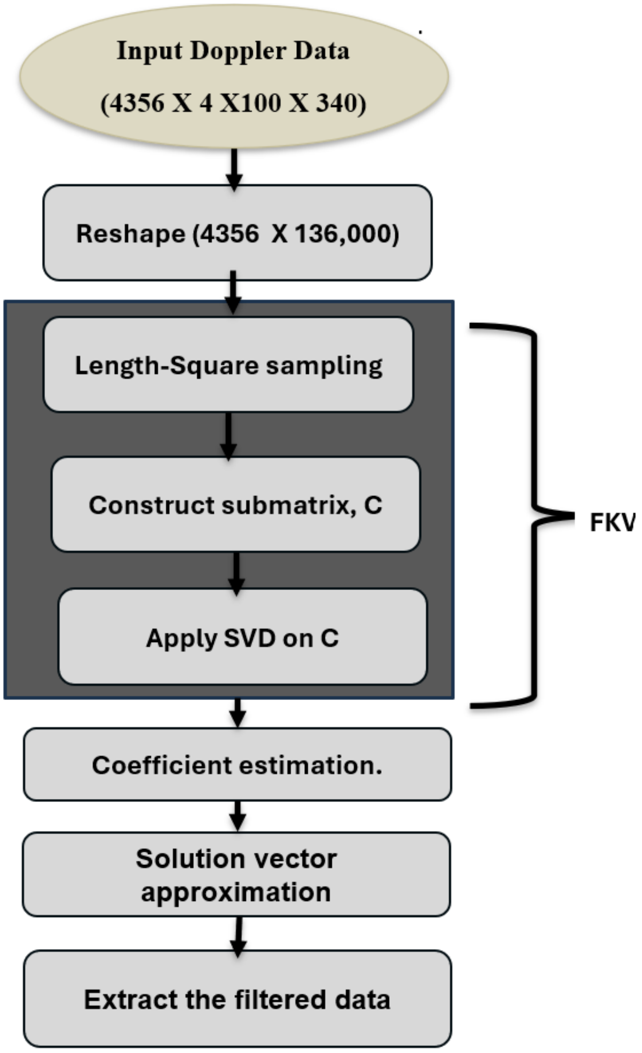
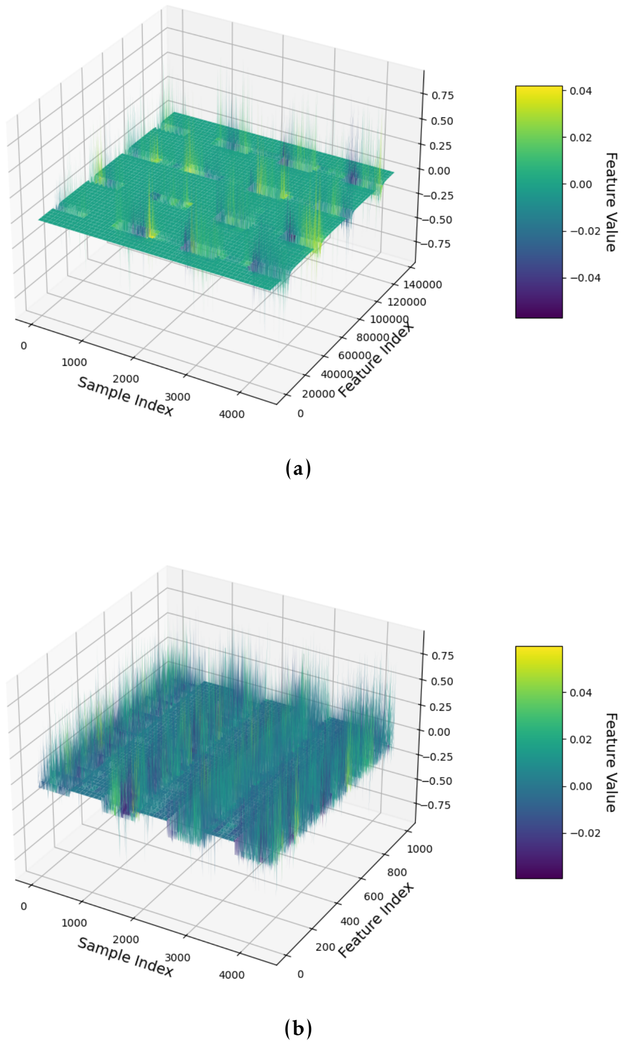


Figure 3. Proposed MC-based feature extraction pipeline.

applying the fast MC feature selection algorithm. The original dataset, shown in Fig. 4(a), consists of approximately 4356 samples with a high-dimensional feature space of around 136,000 features. This dataset exhibits considerable sparsity, indicated by isolated spikes and predominantly negligible feature values distributed across samples. Following the reduction in dimensionality using the fast MC feature selection method, the resulting filtered dataset is shown in Fig. 4(b). This dataset maintains the same number of samples but significantly reduces the dimensionality to approximately 1,000 features. As evidenced by the increased density and frequency of prominent feature values in Fig. 4(b), the fast MC algorithm effectively identifies and preserves highly relevant features while discarding redundant or less informative features. As a result, the filtered dataset becomes more compact and improves the clarity of underlying patterns, suggesting potential enhancements in computational efficiency in resource-constrained environments and being easier to interpret without losing essential information.



(a)

(b)

Figure 4. Visualization of the original Doppler data and the reduced data (a) original data (b) reduced features.

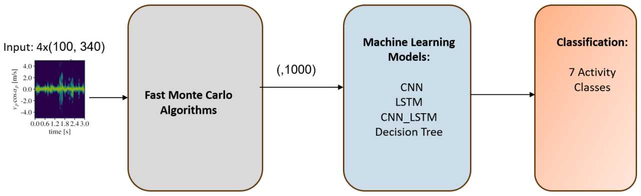


Figure 5. General architecture for HAR, featuring fast MC algorithms and ML models to classify Doppler trace inputs.

5.1. Proposed Machine Learning Architectures

Following feature selection, the resulting reduced feature set is utilized as input for ML models, specifically CNN, long-short-term model (LSTM), a hybrid CNN-LSTM, and a decision tree. These models are chosen to leverage their strengths in pattern recognition, extraction of temporal dependencies, and interpretability. Finally, the trained models classify the data into seven different activities. This section presents the various ML architectures designed to train the selected features as illustrated in Fig. 5.

CNN. To train a CNN, the filtered data was split into 60% training data, 20%, text data, and 20% validation data. The proposed CNN architecture, as shown in Fig. 6, consists of two convolutional blocks and fully connected layers. The first block includes a 2D convolutional layer with 16 filters and a kernel size of 3×1, followed by max pooling and dropout for regularization. The second block repeats this structure with 32 filters. The output is then flattened and passed through a dense layer with 64 ReLU-activated neurons, followed by a final softmax layer for multiclass classification. The model is trained using categorical cross-entropy loss and the Adam optimizer. We apply early stopping and checkpointing based on validation loss to prevent overfitting. This architecture effectively captures local temporal frequency patterns in the Doppler feature space and has demonstrated strong performance across all evaluated activity classes. Combined with the MC-based feature selection method we employ, the overall system provides a robust and computationally efficient pipeline for HAR.

Let the input to the CNN be of shape $(F, 1, 1)$, where F is the number of input features. The model consists of two convolutional layers, a global average pooling layer, a hidden dense layer with D units, and a final output dense layer for C activity classes.

The number of trainable parameters in the first convolutional layer with f_1 filters of size $k_1 \times 1$ and 1 input channel is given by

$$P_1 = (k_1 \cdot 1 \cdot 1 + 1) \cdot f_1 = (k_1 + 1) f_1. \qquad (43)$$

The second convolutional layer applied to the f_1 input channels has f_2 filters of size $k_2 \times 1$, which is shown as follows:

$$P_2 = (k_2 \cdot 1 \cdot f_1 + 1) \cdot f_2 = (k_2 f_1 + 1) f_2. \qquad (44)$$

Following global average pooling, the feature dimension becomes f_2, which feeds into a dense layer with D units:

$$P_3 = f_2 \cdot D + D = D(f_2 + 1). \qquad (45)$$

The final classification layer outputs the probabilities for C classes:

$$P_4 = D \cdot C + C = C(D + 1). \qquad (46)$$

Hence, the total number of trainable parameters in the model is given by

$$P_{\text{total}} = P_1 + P_2 + P_3 + P_4, \qquad (47)$$

where k_1, k_2 denote kernel heights; f_1, f_2 are the number of filters in the respective convolutional layers; D is the number of units in the dense layer; and C is the number of output classes.

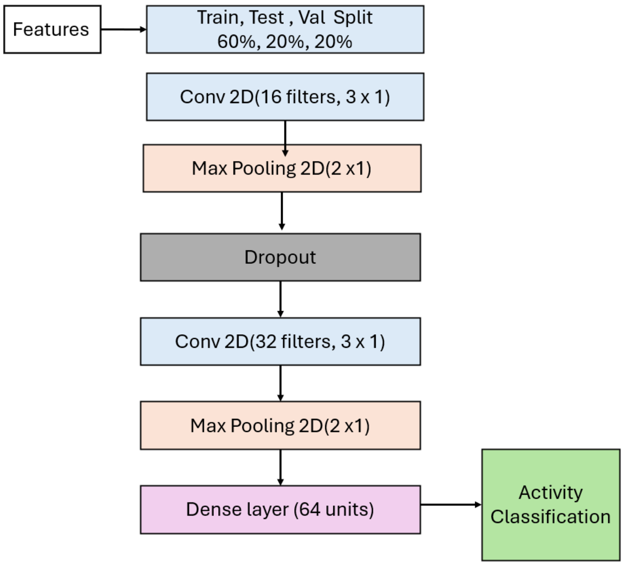


Figure 6. CNN Architecture.

CNN-LSTM Hybrid. The hybrid architecture illustrated in Fig. 7 extends the CNN by adding the LSTM layer. Following the convolutional blocks, the output is reshaped to form a 2D tensor suitable for sequential modeling. This reshaped representation is passed to a unidirectional LSTM layer with 50 units, which captures temporal dynamics and sequential patterns across time windows. The LSTM output is then processed by a dense layer with 128 ReLU-activated neurons and a dropout layer (rate = 0.5), followed by a final softmax layer that outputs class probabilities for multiclass activity recognition.

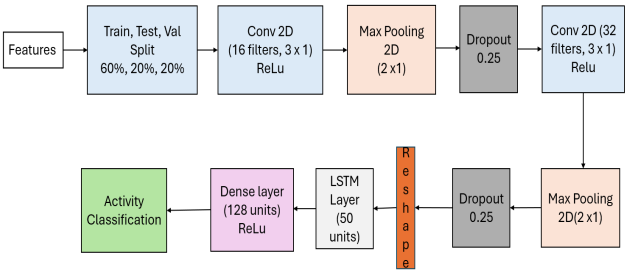


Figure 7. CNN-LSTM Architecture.

LSTM. We also implemented a deep learning model based entirely on LSTM units as illustrated in Fig. 8 to capture the temporal dependencies inherent in Doppler radar time series data. The architecture begins with an LSTM layer consisting of 64 units and configured with `return_sequences=True` to output an entire sequence of hidden states. This allows the subsequent LSTM layer to process the complete temporal evolution of the data. A dropout layer with a rate of 0.25 is used to mitigate overfitting. The second LSTM layer contains 32 units and is set to output only the final hidden state (`return_sequences=False`), effectively summarizing the sequential information into a fixed-length vector. This is again followed by a dropout layer to further regularize the network. The output of the LSTM layers is passed through a fully connected dense layer with 128 ReLU-activated neurons, followed by another dropout layer with a rate of 0.5. The final output layer is a softmax-activated dense layer that outputs class probabilities for multiclass activity classification.

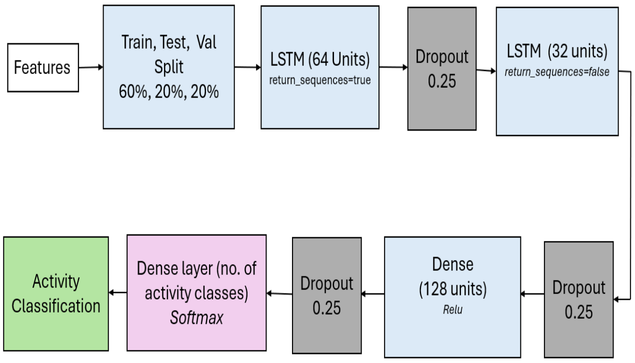


Figure 8. LSTM Architecture.

Decision Tree. In addition to deep learning architectures, we implemented a decision tree classifier as a baseline method for HAR. Decision trees are interpretable, nonparametric models that make predictions by recursively partitioning the feature space based on threshold-based decisions. Fig. 9 shows a truncated decision tree structure, where each internal node applies a feature-based threshold decision, and each leaf node represents a final activity class prediction. Given the filtered and standardized Doppler feature set, the decision tree is trained to learn a hierarchical structure that maps input features to activity labels. To control complexity of and improve generalization, the tree is configured with a maximum depth of 15, a maximum of 50 leaf nodes, a minimum of 10 samples required to split an internal node, and at least five samples per leaf. These hyperparameters were selected to prevent overfitting, especially given the high dimensionality of the Doppler data. The model is trained using the Gini impurity criterion and evaluated on a hold-out test set 20% of the data, with label stratification to preserve class distribution. The resulting model achieved competitive accuracy on the test set and provided interpretable decision paths for each activity class. In addition, the number of nodes and splits was analyzed to quantify the complexity of the model. This baseline is a comparison point for neural network-based methods and highlights the effectiveness of simpler classifiers in HAR using Doppler traces. To calculate the total number of parameters, the following formula was used

$$P_{\text{tree}} \approx S \times F, \tag{48}$$

where P_{tree} denotes the approximate total number of parameters in the decision tree, S is the total number of decision splits or internal nodes, and F represents the number of features available at each split. This estimate assumes that each split evaluates all F features when choosing the best split condition. Although actual implementations may optimize this via heuristics or greedy selection, this upper bound gives insight into the model complexity.

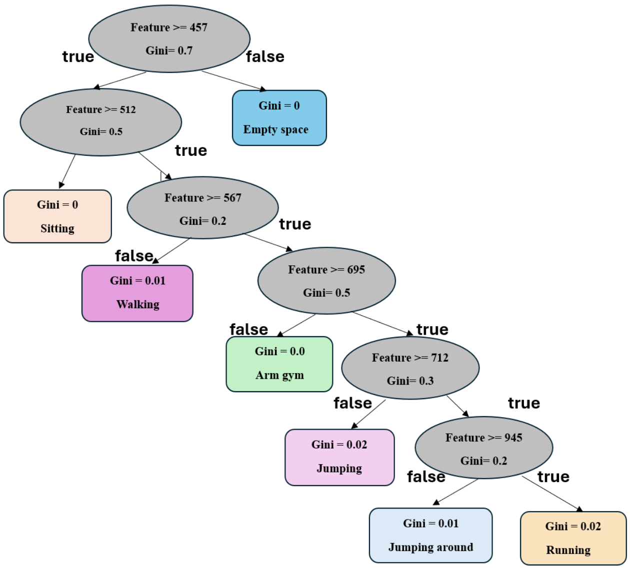


Figure 9. A truncated version of the proposed decision tree structure.

5.2. The Impact of The Number of Selected Features on Accuracy

The plot in Fig. 10 illustrates how our fast MC-based feature selection method impacts CNN performance on a dataset across all activity classes. We evaluated training, validation, and test accuracy at various numbers of feature samples from 10 to 1400. The results show a clear improvement in accuracy as the number of selected features increases from 10 features that achieve an accuracy of 75% to around 97% with 300 features, where the model reaches its maximum generalization performance. Between 200 and 500 features, the validation and test accuracy remain consistently high, indicating that our method successfully retains the most informative components of the original Doppler matrix. Beyond this range, accuracy begins to plateau and eventually declines, particularly after 1000 features, suggesting the inclusion of noisy or redundant information. At the same time, training accuracy gradually decreases as more features are added, which reflects a reduction in overfitting and supports the claim that our approach also encourages better generalization. This performance trend confirms that the fast MC method effectively identifies a compact and high-quality subset of features.

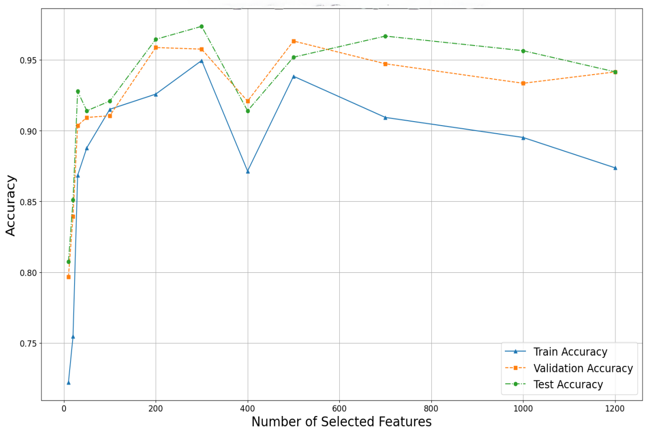


Figure 10. CNN accuracy as a function of the number of selected features. The fast MC method maintains high test and validation accuracy with 200 to 500 features while reducing model complexity, confirming its effectiveness in selecting the most informative Doppler features.

6. Experimental Results and Analysis

This section evaluates the experiment performed for the proposed method compared to the latest models, namely the decision tree, CNN, LSTM, and CNN-LSTM hybrids. Table 1 illustrates the performance of ML models supported by features from the fast MC dimensionality reduction to classify Doppler image features. Model performance is evaluated using standard classification metrics: accuracy, precision, recall, and F1-score. These metrics are defined in a multiclass setting with C activity classes.

Accuracy. Accuracy measures the proportion of correctly predicted instances across all classes and is given by

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{49}$$

where TP is the number of true positives, TN is true negatives, FP is false positives, and FN is false negatives.

Activity	Metric	Models With Fast MC Algorithm			
		CNN	CNN + LSTM	Dec. Tree	LSTM
Empty	Precision	1.00	1.00	0.98	0.86
	Recall	1.00	0.97	1.00	0.78
	F1-score	1.00	0.98	0.99	0.82
Arm Gym	Precision	0.97	0.92	0.98	1.00
	Recall	0.98	1.00	1.00	0.01
	F1-score	0.98	0.96	0.99	0.03
Jumping	Precision	1.00	0.97	1.00	0.92
	Recall	1.00	1.00	0.99	0.89
	F1-score	1.00	0.98	1.00	0.90
Sitting	Precision	1.00	1.00	1.00	0.50
	Recall	1.00	1.00	1.00	1.00
	F1-score	1.00	1.00	1.00	0.66
Sitting Cont.	Precision	0.98	1.00	1.00	1.00
	Recall	0.97	0.91	0.99	1.00
	F1-score	0.98	0.95	1.00	1.00
Running	Precision	0.90	0.75	1.00	0.00
	Recall	0.75	0.95	0.99	0.00
	F1-score	0.82	0.84	1.00	0.00
Walking	Precision	0.78	0.93	0.99	0.46
	Recall	0.91	0.68	0.98	0.99
	F1-score	0.84	0.79	0.99	0.63
Test Accuracy (%)		**96**	**87**	**99.8**	**53**

Table 1. Classification performance of all models.

Per-Class Metrics. Let TP_c, FP_c, and FN_c denote the number of true positives, false positives, and false negatives for class c, respectively. Then, we have

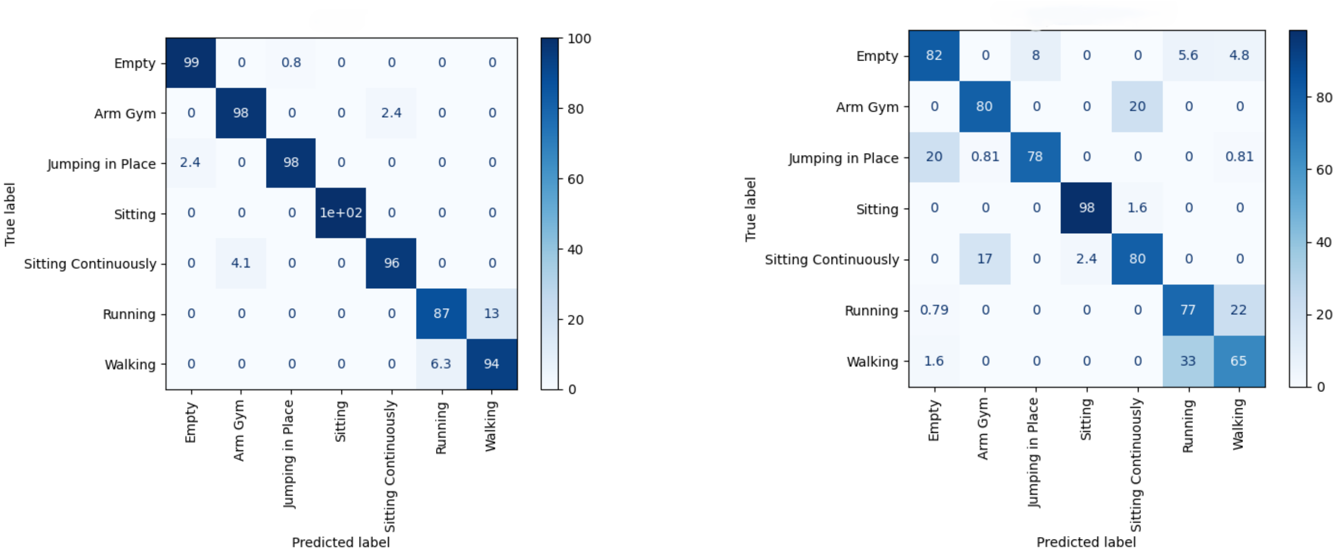
$$\text{Precision}_c = \frac{TP_c}{TP_c + FP_c}, \tag{50}$$

$$\text{Recall}_c = \frac{TP_c}{TP_c + FN_c}, \tag{51}$$

$$\text{F1}_c = 2 \cdot \frac{\text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c}. \tag{52}$$

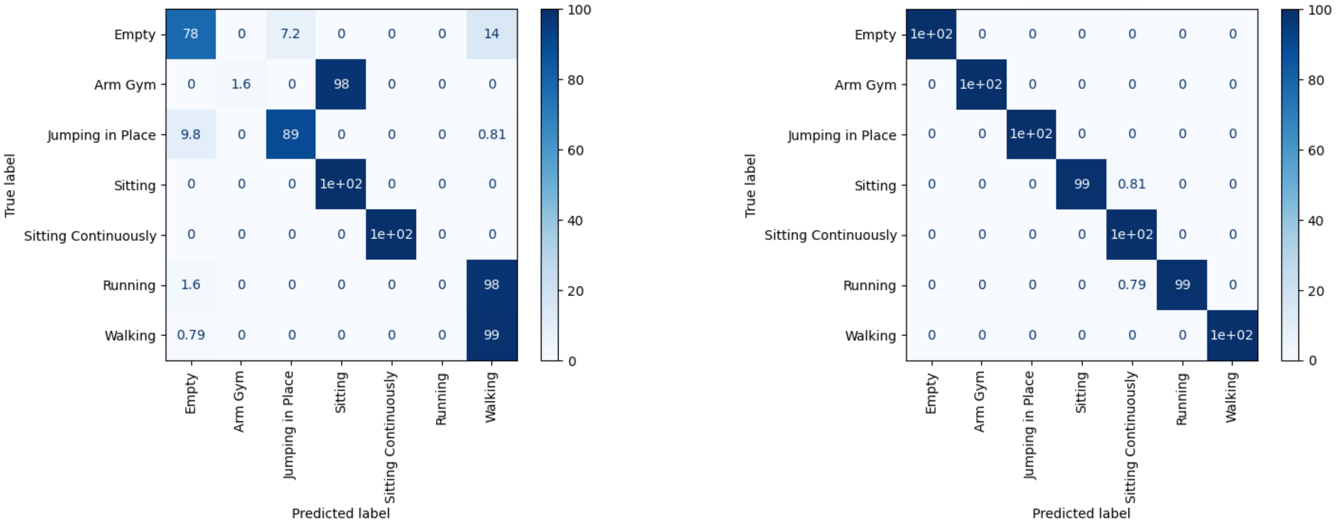
As shown in Fig. 11, the confusion matrices derived from the validation set highlight the comparative classification effectiveness of all models.

The results in Table 2 demonstrate that fast MC algorithms significantly reduce trainable parameters while maintaining or improving accuracy in most models. By applying approximate SVD on this submatrix, only the most informative and discriminative components are retained, effectively reducing the input feature size and dimensionality. The number of trainable parameters is proportional to the number of features, particularly in models with dense layers. Hence, a reduced number of input features minimizes the size of the corresponding weight matrix and biases, causing a cascading effect on the overall complexity of the model and training time. Even in non-neural models like decision trees, input dimensionality significantly impacts the computational cost and model size. Additionally, the method suppresses noise and removes redundant information because it only focuses



(**a**) CNN

(**b**) CNN-LSTM



(**c**) LSTM

(**d**) Decision Tree

Figure 11. Confusion matrices for different models using the fast MC based feature selection.

on relevant subspace projections, improving generalization and training stability. The result is a lightweight and efficient HAR model architecture suitable for real-time inference on resource-constrained edge devices. CNN achieved a substantial parameter reduction from 387,943 to 2,343 with an accuracy constant of 96%. Similarly, decision tree saw a decrease in parameters from 163,2000 to 26,000 and a slight improvement in accuracy from 99.7% to 99.8%. CNN-LSTM also benefited, with parameters reduced from 6,680,903 to 36,767 and accuracy increasing from 80% to 87%. However, LSTM experienced a notable decrease in accuracy from 84% to 53%, suggesting that fast MC techniques may be less effective for sequence-based models. Fast MC optimization proves to be highly effective for CNNs and decision trees, offering substantial efficiency gains without sacrificing performance. The confusion matrices evaluate the performance of the four models for the classification task using the low-dimensional data from the proposed feature selection process.

ML Models	With fast MC		Without fast MC	
	# Params	Acc (%)	# Params	Acc (%)
CNN	2,343	96	387,943	96
Decision Tree	26,000	99.8	1,632,000	99.7
CNN+LSTM	36,767	87	6,680,903	80
LSTM	34,439	53	17,528,071	84

Table 2. Number of parameters and accuracy for each ML model with and without fast MC algorithms.

7. Conclusion

This study introduced a feature extraction method using fast MC sampling based on the FKV framework to tackle the challenge of high-dimensional doppler radar data in HAR. The original four-dimensional doppler tensor was reshaped into a two-dimensional matrix, making it suitable for efficient matrix sampling. By applying LS probability sampling, we selected a subset of informative rows and columns that preserved the essential structure of the signal. A low-rank approximation was performed using SVD on the sampled submatrix. From this, the approximate left and right singular vectors were reconstructed to serve as latent directions in the data. These latent directions were used to guide a rejection sampling step, where the features most aligned with the informative structure of the data were selected. The final filtered data retained only the most relevant 1000 features, achieving significant dimensionality reduction while maintaining class-separable information across activities. Visual analysis revealed that the unique structure of each activity was preserved, while the classification results highlighted the strong ability of the selected features to distinguish between different activities. CNN and decision tree models showed significant parameter reductions without performance loss, making them suitable for resource-efficient implementations. The accuracy of the LSTM model improved slightly with a significant reduction in model complexity, but the standalone model experienced a drop in accuracy, highlighting the need for careful evaluation in sequence-based architectures. Fast MC algorithms generally provide a promising approach for optimizing model efficiency, particularly in computational and resource-constrained scenarios such as real-time HAR in edge deployment, in systems such as elderly monitoring systems, smart homes, and clinics, where reducing the size of the model without sacrificing accuracy is key.

7.1. Future Works

Future research will aim to exploit the potential of quantum technologies to make ML models more efficient and scalable. A key focus will be on exploring quantum SVD, which holds promise for accelerating tasks like dimensionality reduction and feature extraction, particularly for datasets of high complexity. Efforts will also include developing quantum ML (QML) models, such as quantum-enhanced CNNs and decision tree versions, and creating hybrid quantum-classical frameworks. These advances could significantly improve training processes and model accuracy. These initiatives aim to push the limits of ML by integrating quantum innovations into real-world problem-solving.

References

[1] M. J. Bocus, K. Chetty, and R. J. Piechocki, “Uwb and wifi sys. as passive opportunistic activity sensing radars,” in *2021 IEEE Radar Conf.(RadarConf21)*, Atlanta, GA., May 2021, pp. 1–6.

[2] S. P and S. K. Udgata, “Wi-fi sensing enabled violent activity detection in a smart home,” in *2023 IEEE 11th Region 10 Humanitarian Technol. Conf. (R10-HTC)*, Rajkot, India, Oct. 2023, pp. 1274–1279.

[3] G. Apicella, G. D’Aniello, G. Fortino, M. Gaeta, R. Gravina, and L. G. Tramuto, “An adaptive neuro-fuzzy approach for activity recognition in situation-aware wearable systems,” in *2022 IEEE 3rd Int. Conf. Hum.-Mach. Syst. (ICHMS)*, Orlando, FL, Nov 2022, pp. 1–6.

[4] F. Meneghello, C. Chen, C. Cordeiro, and F. Restuccia, “Toward integrated sensing and communications in ieee 802.11bf wi-fi networks,” *IEEE Commun. Mag.*, vol. 61, no. 7, pp. 128–133, Jul 2023.

[5] Y. Dong, X. Cao, Z. Gao, Z. Sun, S. Savovic, X. Zou, W. Pan, L. Yan, and X. Deng, “Resource allocation in multi-user mimo-ofdma vlc systems over low-pass led channels,” *IEEE Photon. Technol. Lett.*, pp. 1–1, Feb 2025.

[6] Y. Golubev, S. Tutelian, V. Loginov, and E. Khorov, “Reducing duration of the mu-mimo csi acquisition procedure in wi-fi networks with ofdma,” in *2023 IEEE Conf. Stand. Commun. Netw. (CSCN)*, Munich, Germany, 06-08 Nov 2023, pp. 395–400.

[7] C.-F. Hsieh, Y.-C. Chen, C.-Y. Hsieh, and M.-L. Ku, “Device-free indoor human activity recognition using wi-fi rssi: Machine learning approaches,” in *2020 IEEE Int. Conf. Consum. Electron.- Taiwan (ICCE-Taiwan)*, Taoyuan, Taiwan, Sep 2020, pp. 1–2.

[8] F. Meneghello, D. Garlisi, N. D. Fabbro, I. Tinnirello, and M. Rossi, “SHARP: Environment and person independent activity recognition with commodity IEEE 802.11 access points,” *IEEE Trans. Mob. Comput.*, vol. 22, no. 10, pp. 6160–6175, Jun. 2023.

[9] B. Tan, Q. Chen, K. Chetty, K. Woodbridge, W. Li, and R. Piechocki, “Exploiting WiFi channel state information for residential healthcare informatics,” *IEEE Commun. Mag.*, vol. 56, no. 5, pp. 130–137, Aug. 2018.

[10] J. Yang, H. Zou, H. Jiang, and L. Xie, “Device-free occupant activity sensing using wifi-enabled iot devices

for smart homes," *IEEE Internet of Things J.*, vol. 5, no. 5, pp. 3991–4002, June 2018.

[11] I. Nirmal, A. Khamis, M. Hassan, W. Hu, R. Li, and A. Kalyanaraman, "Wifi2radar: Orientation-independent single-receiver wifi sensing via wifi to radar translation," *IEEE Internet of Things J.*, vol. 11, no. 9, pp. 15 750–15 766, Jan. 2024.

[12] J. Pegoraro, F. Meneghello, and M. Rossi, "Multiperson continuous tracking and identification from mm-wave micro-doppler signatures," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 4, pp. 2994–3009, Sep 2021.

[13] M. Kim, D. Han, and J.-K. K. Rhee, "Unsupervised view-selective deep learning for practical indoor localization using csi," *IEEE Sens. J.*, vol. 21, no. 21, pp. 24 398–24 408, Sep. 2021.

[14] P. Hu, W. Liu, C. Yang, Y. Sun, and J. Li, "Wifi csi based passive human activity recognition method using blstm-cnn," in *2023 IEEE 23rd Int. Conf. on Commun. Technol. (ICCT)*, Wuxi, China, Oct. 2023, pp. 210–215.

[15] Z. Shi, Q. Cheng, J. A. Zhang, and R. Yi Da Xu, "Environment-robust wifi-based human activity recognition using enhanced csi and deep learning," *IEEE Internet of Things J.*, vol. 9, no. 24, pp. 24 643–24 654, Jul. 2022.

[16] M. H. Kabir, M. H. Rahman, and W. Shin, "Csi-ianet: An inception attention network for human-human interaction recognition based on csi signal," *IEEE Access*, vol. 9, pp. 166 624–166 638, Dec 2021.

[17] P. Chen, F. Liu, S. Gao, P. Li, X. Yang, and Q. Niu, "Smartphone-based indoor fingerprinting localization using channel state information," *IEEE Access*, vol. 7, pp. 180 609–180 619, Dec 2019.

[18] M. M, T. R, V. C, and A. Kanavalli, "Serial vs parallel execution of principal component analysis using singular value decomposition," in *2024 Int. Conf. Intell. Innov. Technol. Comput., Electr. Electron. (IITCEE)*, Bangalore, India, 24-25 Jan. 2024, pp. 1–5.

[19] S. Zhang, F. Qin, J. Xu, and J. Zhang, "Small sample scene cancer classification method based on combined feature dimensionality reduction," in *2024 16th International Conference on Computer and Automation Engineering (ICCAE)*, Melbourne, Australia, 14-16 Mar. 2024, pp. 346–350.

[20] X. Ding, W. Yu, Y. Xie, and S. Liu, "Efficient model-based collaborative filtering with fast adaptive pca," in *2020 IEEE 32nd Int. Conf. Tools Artif. Intell. (ICTAI)*, Baltimore, MD, 9-11 Nov. 2020, pp. 955–960.

[21] YongchangWang and L. Zhu, "Research and implementation of svd in machine learning," in *2017 IEEE/ACIS 16th Int. Conf. Comput. Inf. Sci. (ICIS)*, Kunming, China, 22-24 May 2017, pp. 471–475.

[22] K. J and K. N. Meera, "A hybrid approach for medical image fusion using svd and pca," in *2024 4th Int. Conf. Innov. Pract. Technol. Manag. (ICIPTM)*, Noida, India, 21-23 Feb. 2024, pp. 1–5.

[23] D. Janeković and D. Bojanjac, "Randomized algorithms for singular value decomposition: Implementation and application perspective," in *2021 Int. Symp. ELMAR*, Zadar, Croatia, 13-15 Sep. 2021, pp. 165–168.

[24] P. Drineas, R. Kannan, and M. W. Mahoney, "Fast monte carlo algorithms for matrices ii: Computing a low-rank approximation to a matrix," *SIAM J. Comput.*, vol. 36, no. 1, pp. 158–183, Jan 2006.

[25] H. Cao, X. Shang, Y. Wang, M. Song, S. Chen, and C.-I. Chang, "Go decomposition (GoDec) approach to finding low rank and sparsity matrices for hyperspectral target detection," in *IGARSS 2020 - 2020 IEEE Int. Geosci. Remote Sens. Symp.*, Waikoloa, HI, 26 Sep. - 02 Oct 2020, pp. 2807–2810.

[26] J. M. Arrazola, A. Delgado, B. R. Bardhan, and S. Lloyd, "Quantum-inspired algorithms in practice," *Quantum*, vol. 4, p. 307, Aug. 2020.

[27] A. Gilyén, S. Lloyd, and E. Tang, "Quantum-inspired low-rank stochastic regression with logarithmic dependence on the dimension," 2018.

[28] N.-H. Chia, T. Li, H.-H. Lin, and C. Wang, "Quantum-inspired sublinear algorithm for solving low-rank semidefinite programming," jun 2020.

[29] R. Xiao, Z. Zong, and L. Yang, "Clutter suppression based on singular value decomposition and fast wavelet algorithm," in *IGARSS 2024 - 2024 IEEE International Geoscience and Remote Sensing Symposium*, Athens, Greece, 12-14 Mar. 2024.

[30] Y. Yan, G. Wu, Y. Dong, and Y. Bai, "Floating small target detection in sea clutter based on the singular value decomposition of low rank perturbed random matrices," in *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Chongqing, China, 12-14 Mar. 2021, pp. 527–531.

[31] S. Rath and J. Ramalingam, "Comments on "outsourcing eigen-decomposition and singular value decomposition of large matrix to a public cloud"," *IEEE Access*, vol. 12, pp. 116 013–116 016, Aug. 2024.

[32] A. Frieze, R. Kannan, and S. Vempala, "Fast monte-carlo algorithms for finding low-rank approximations," in *Proc. 39th Annu. Symp. Found. Comput. Sci. (Cat. No.98CB36280)*, Palo Alto, CA, 08-11 Nov 1998, pp. 370–378.