Wi-Fi Based Gesture Recognition: Leveraging RSSI Data for Privacy-Preserving Interaction

Shashidhar Reddy Ainala(sainala@mail.yu.edu)
Onkar Kunte (okunte@mail.yu.edu)
Venu Khare (vkhare@mail.yu.edu)
Jainam Bhansali (jbhansal@mail.yu.edu)
Yeshiva University

Abstract

This paper presents a novel approach to gesture recognition based on Wi-Fi Received Signal Strength Indicator (RSSI) values collected through standard Wi-Fi hardware, without the need for specialized sensors or cameras. With the increasing demand for non-intrusive and privacy-preserving methods of recognizing human gestures, RSSI-based gesture recognition leverages ambient wireless signals to detect movements in the environment. In this study, we collect and preprocess RSSI data using a MacBook's Wi-Fi interface and associate various common gestures, such as swipes, waves, and claps, with unique patterns in RSSI fluctuations. We also introduce environmental variables such as the number of connected devices, distance, and physical obstructions to improve the robustness of the model. The data is then processed and labeled to train a machine learning model capable of identifying gestures in real-time. Our results demonstrate the feasibility of RSSI-based gesture recognition, achieving promising accuracy under various conditions, and suggest potential applications in human-computer interaction, smart homes, and healthcare monitoring. This work highlights the potential of Wi-Fi-based gesture recognition as an accessible, cost-effective solution for pervasive computing.

1. Introduction

Gesture recognition has emerged as a critical technology in the domains of human-computer interaction (HCI), smart environments, and pervasive computing. Traditional gesture recognition systems often rely on specialized sensors, such as cameras, accelerometers, or depth sensors, which can be costly, intrusive, or limited by privacy concerns. In contrast, Wi-Fi signals, which are ubiquitous in modern environments, offer a non-intrusive and cost-effective alternative for recognizing human gestures.

This project explores the potential of Wi-Fi Received Signal Strength Indicator (RSSI) values as a medium for gesture recognition. RSSI measures the strength of a received signal and is influenced by environmental changes, including human movement. By analyzing RSSI fluctuations, it becomes possible to identify patterns corresponding to specific gestures. Unlike vision-based systems, RSSI-based gesture recognition does not rely on line-of-sight, making it suitable for use in obstructed or private spaces.

The study involves collecting RSSI data using a standard Wi-Fi interface on a MacBook, with gestures such as swipes, waves, and claps performed at varying distances and environmental conditions. The data is labeled and preprocessed to remove noise and extract meaningful features, followed by training a machine learning model to classify gestures based on their RSSI patterns. Additional environmental parameters, such as the number of connected devices, distance, and obstructions, are considered to enhance the robustness of the system.

This paper discusses the methodology for data collection, preprocessing, and model training, as well as the challenges faced in achieving reliable recognition. The results demonstrate that gesture recognition using Wi-Fi RSSI is feasible and effective under diverse conditions, paving the way for non-intrusive, privacy-preserving applications in smart homes, healthcare, and other domains.

By leveraging the omnipresence of Wi-Fi networks, this work introduces a scalable and accessible approach to gesture recognition, expanding the possibilities for human-computer interaction beyond traditional sensing methods.

2. Related Work

Gesture recognition using Wi-Fi signals has emerged as an innovative and non-intrusive approach for human-computer interaction. Research in this domain primarily focuses on leveraging signal fluctuations, mitigating environmental challenges, and applying advanced machine learning techniques for robust recognition. Below, we summarize key contributions in this field.

Wi-Fi RSSI-Based Gesture Recognition The use of Wi-Fi RSSI for gesture recognition builds upon the analysis of signal changes caused by human motion. Liu et al. [1] initiated this field with passive Wi-Fi analytics for human sensing. Sun et al. [2] extended this by incorporating environmental awareness, addressing challenges such as obstructions and signal interference. Tang et al. [3] showcased the potential of deep learning in enhancing gesture recognition accuracy using RSSI data. Additionally, Zhou et al. [11] explored RSSI-based motion detection, emphasizing its utility in low-power human sensing applications.

Impact of Environmental Factors Environmental factors play a critical role in Wi-Fi sensing. Wang et al. [4] analyzed RSSI fluctuations in dense environments, while Zhao et al. [5] proposed mitigation techniques for signal interference in multi-user scenarios. Zhang et al. [10] investigated motion detection using ambient wireless signals, highlighting the adaptability of Wi-Fi sensing across various setups. Recent works by Alizadeh et al. [12] examined the impact of powerline noise on Wi-Fi sensing, offering unique insights into external interference sources.

Wi-Fi CSI-Based Gesture Recognition Channel State Information (CSI) provides finer-grained signal data compared to RSSI, enabling improved gesture recognition. Han et al. [6] utilized CSI for passive human detection, demonstrating its effectiveness in challenging scenarios. Yang et al. [7] reviewed the applications of CSI in gesture recognition and other human sensing tasks, highlighting its advantages in capturing motion-related features.

Advanced Machine Learning for Gesture Recognition Machine learning has been instrumental in advancing gesture recognition. Fan et al. [8] and Tang et al. [3] highlighted the effectiveness of neural networks and deep learning models for classifying gestures from signal data. Yang et al. [13] explored real-time activity detection using deep learning on RSSI and CSI data, showcasing its applicability in smart home environments. Gupta et al. [14] discussed noise mitigation in electromagnetic environments, enabling gesture recognition in noisy settings.

Applications of GRU in Gesture Recognition The Gated Recurrent Unit (GRU), a variation of recurrent neural networks, has been widely adopted for sequential data processing due to its ability to efficiently model temporal dependencies. Chung et al. [16] introduced GRUs as an improvement over traditional RNNs, reducing computational complexity while retaining comparable performance. This makes GRUs particularly effective for gesture recognition, as shown by Tang et al. [3], where temporal patterns in RSSI data were modeled using GRU-based architectures. Yang et al. [17] demonstrated the application of GRUs in activity recognition using Wi-Fi data, highlighting their robustness in real-time scenarios.

Applications of Wi-Fi Sensing Wi-Fi sensing extends beyond gesture recognition into broader applications. Adib et al. [9] demonstrated its use for monitoring breathing and heart rate in smart homes, paving the way for non-contact health monitoring. Sun et al. [15] explored environmental-aware applications for gesture recognition, while Alizadeh et al. [12] combined powerline noise analysis with Wi-Fi sensing for innovative interaction methods.

3. Methodology

3.1. System Overview

The proposed gesture recognition system leverages variations in Wi-Fi RSSI signals caused by human motion to detect and classify gestures. Designed for indoor environments, this system is particularly suited for residential spaces. Below is an outline of its key components and their integration.

3.1.1 Wi-Fi Sensing Module

The system utilizes the built-in Wi-Fi adapter of a macOS device, interfaced through the CoreWLAN framework, to scan and collect RSSI data from nearby access points. This module continuously monitors signal strength variations, capturing RSSI fluctuations caused by human motion.

3.1.2 Data Logging and Environmental Context

The RSSI data is logged in real-time, alongside additional contextual parameters provided by the user:

- **Number of Connected Devices:** Represents the density of devices connected to the Wi-Fi network.
- **Distance:** The physical distance (in meters) between the sensing device and the router.
- **Obstructions:** The number of walls or physical barriers between the router and the device.

This contextual information ensures that the logged RSSI data captures variations across diverse real-world scenarios, enabling a more comprehensive analysis and model training.

3.2. Indoor Environment Setup

The gesture recognition system was evaluated in a typical residential flat to simulate real-world indoor conditions. The flat had the following layout and characteristics:

3.2.1 Room Configuration:

The flat consisted of two bedrooms, one kitchen, one bathroom, and a hall with a connected balcony.

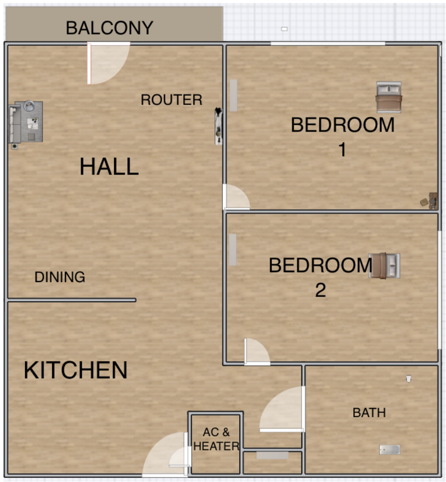

Figure 1. Indoor Environment

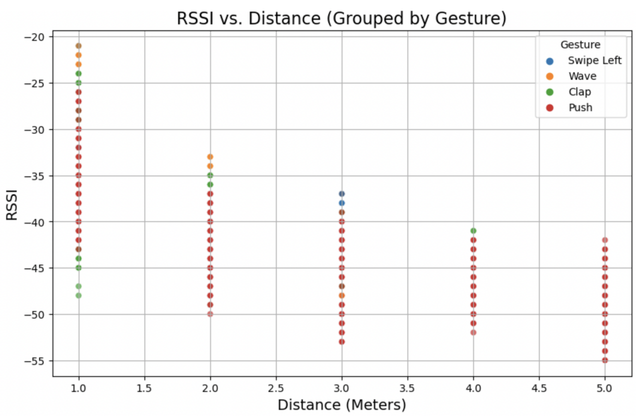

Figure 2. RSSI vs. Distance

3.2.2 Router Placement:

The Wi-Fi router was strategically positioned in the central hall to ensure optimal signal coverage across all rooms and the balcony.

3.2.3 Gesture Zones:

Close Proximity: Gestures were performed within 1–2 meters from the router, maintaining a direct line of sight.

Intermediate Zones: Gestures were conducted in the bedrooms and kitchen, located up to 10 meters away from the router, with walls acting as obstructions to the signal path.

Balcony Zone: Gestures were tested in the balcony, where the signal experienced partial coverage due to the open environment.

This setup allowed the system to be tested under various conditions, including differences in distance, obstructions, and environmental factors, providing a robust assessment of its performance.

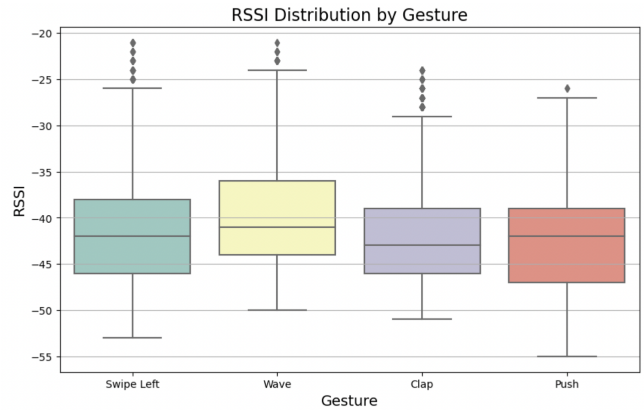

Figure 3. RSSI Distribution by Gesture

3.3. Data Description

The dataset contains a total of **28,412 samples**, each representing a unique time-series record capturing signal strength and associated environmental conditions. The key attributes of the dataset are as follows:

- **Timestamp:** The exact time the data point was recorded, ensuring each entry is unique.
- **RSSI:** Received Signal Strength Indicator, measured in dBm. RSSI values range from strong signals (-21 dBm) to weak signals (-55 dBm).
- **Gesture:** The type of gesture performed, including:
  - *Swipe Left*
  - *Wave*
  - *Clap*
  - *Push*
- **Connected Devices:** The number of devices connected to the network at the time of recording.
- **Distance (meters):** The physical distance between the signal source and receiver. Data includes distances ranging from 1.0 meters to 5.0 meters.
- **Number of Walls:** The number of physical obstructions (walls) between the signal source and receiver, with values of 0, 1, or 2.

  This dataset provides a diverse range of scenarios for analyzing the effects of gestures, distance, and barriers on signal strength.

3.4. Exploratory Data Analysis (EDA)

To gain insights from the dataset, an exploratory data analysis was conducted. Below are the key findings:

3.4.1 General Observations

The dataset is well-structured, with no missing or invalid entries. Each sample is uniquely timestamped, and the gestures are evenly distributed across the dataset, ensuring a balanced representation.

3.4.2 Signal Strength Analysis

The RSSI values span a wide range, typically between -55 dBm and -21 dBm, reflecting real-world signal behavior. Histograms of RSSI values show most signals clustering around moderate strengths, with fewer extreme values.

3.4.3 Gesture Patterns

Boxplots revealed distinct RSSI ranges associated with different gestures, suggesting potential for gesture classification using signal strength data. Updated counts for gestures are as follows:

- *Swipe Left:* 7,122 samples
- *Wave:* 7,113 samples
- *Clap:* 7,104 samples
- *Push:* 7,073 samples

3.4.4 Impact of Environmental Factors

Scatter plots confirmed a clear inverse relationship between RSSI and distance. The signal strength decreases as the distance increases, which aligns with expected wireless signal behavior. Additionally, the presence of walls significantly impacts RSSI, with more walls leading to weaker signals.

3.4.5 Outliers and Anomalies

Some outliers were identified, where RSSI values were unusually strong or weak. These cases were likely influenced by rare environmental conditions, such as multiple obstructions or unexpected network behavior.

3.4.6 System Flow

The system workflow integrates these components into a streamlined process:

1. The Wi-Fi sensing module collects RSSI data and logs it in real-time with contextual parameters.
2. Data preprocessing ensures consistency and prepares the dataset for analysis and training.
3. Augmentation techniques expand the dataset, introducing variability to improve model robustness.
4. The processed data is then fed into machine learning models for gesture classification, leveraging the environmental context captured during sensing.

This comprehensive design allows the system to effectively detect and classify gestures while accounting for the dynamic nature of indoor environments.

The updated dataset and exploratory analysis provide a robust foundation for studying gesture recognition and the environmental impacts on wireless signals. The methodology leverages these insights for developing a system capable of recognizing gestures in dynamic indoor environments. The integration of Wi-Fi sensing, real-time logging, and data augmentation ensures that the system is both reliable and adaptable to real-world conditions.

4. Model Architecture

4.1. Gated Recurrent Unit (GRU)

Gated Recurrent Units (GRUs) are a variant of Recurrent Neural Networks (RNNs) that address the challenges of vanishing gradients and long-term dependencies during training. Unlike traditional RNNs, GRUs incorporate gating mechanisms to control the flow of information across time steps, improving their efficiency and performance for sequential data tasks.

- **Update Gate:** Determines how much of the previous memory to carry forward to the next time step.
- **Reset Gate:** Controls how much of the past information to forget.

The GRU model reduces computational complexity compared to Long Short-Term Memory (LSTM) networks by using fewer gates while maintaining similar performance for many applications.

In this work, a GRU-based architecture with bidirectional layers and fully connected layers was implemented, which improved the validation accuracy to **51.24%**, demonstrating its capability to learn meaningful temporal features from the dataset.

Our proposed architecture, named **GestureGRU**, processes input gesture sequences for classification tasks. The model combines Gated Recurrent Units (GRUs) for sequential data processing with fully connected layers and regularization techniques for improved learning. The overall pipeline is as follows:

4.2. Input Representation

The input data is structured in the form of `Batch × Sequence × Features`. This representation captures the temporal dimension of gesture sequences.

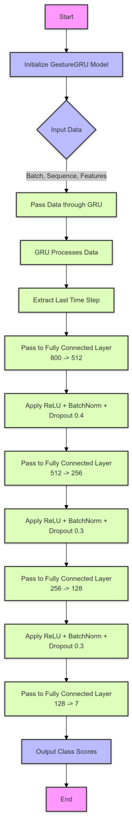

Figure 4. Model Architecture

4.3. GRU Layer

The input data is passed through a GRU layer. GRUs are well-suited for handling sequential data as they efficiently capture temporal dependencies without suffering from the vanishing gradient problem, common in traditional RNNs.

- **Output**: The GRU processes the entire sequence, and we extract the output corresponding to the **last time step**. This output represents the learned sequence-level features.

4.4. Fully Connected Layers

The extracted sequence features are passed through a series of **fully connected (FC) layers**. The transformation of dimensions is as follows:

- **First Layer**: Transforms 800-dimensional input to 512 neurons.
  Activation: ReLU (Rectified Linear Unit).
  Regularization: Batch Normalization and Dropout with a probability of 0.4 are applied to prevent overfitting and accelerate convergence.

- **Second Layer**: Transforms 512 neurons to 256 neurons.

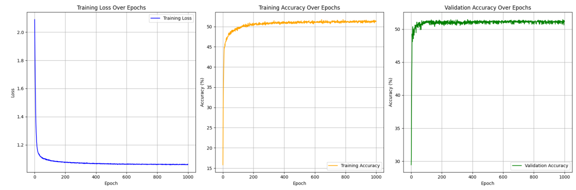

Figure 5. LOSS OVER EPOCH

  Activation: ReLU.
  Regularization: Batch Normalization and Dropout with a probability of 0.3 are applied.

- **Third Layer**: Transforms 256 neurons to 128 neurons.
  Activation: ReLU.
  Regularization: Batch Normalization and Dropout with a probability of 0.3 are applied.

4.5. Output Layer

The final fully connected layer maps the 128-dimensional features to 7 output neurons, corresponding to the number of gesture classes. These output scores are passed to a softmax activation function during training for classification.

4.6. Regularization and Activation

The use of **ReLU activations** ensures non-linearity and faster training. The incorporation of **Batch Normalization** stabilizes the learning process, while **Dropout** helps prevent overfitting at various stages.

4.7. Summary of Architecture

- **GRU for Temporal Data**: Captures sequence-level dependencies.

- **Fully Connected Layers**: Sequential feature compression ($800 \to 512 \to 256 \to 128 \to 7$).

- **Regularization Techniques**: Batch Normalization and Dropout applied progressively.

- **Activation Function**: ReLU for intermediate layers, Softmax for final output.

This architecture enables the model to effectively learn and generalize from gesture-based input sequences while maintaining computational efficiency.

5. Results

5.1. Training Loss Analysis

The **Gesture_GRU** model demonstrated a steady decline in training loss throughout the 1000 epochs. The advanced architecture, including bidirectional GRU layers and fully connected layers with regularization (dropout and batch normalization), enabled better feature extraction and convergence.

- Training loss continuously reduced, confirming effective learning and convergence over time.
- The addition of `AdamW` optimizer and `OneCycleLR` scheduler contributed to smoother training dynamics.

5.2. Validation Accuracy Analysis

The validation accuracy for both models highlights the difference in performance:

- **Gesture_CNN:** Achieved a validation accuracy of **43.42%** in just 40 epochs.
- **Gesture_GRU:** Achieved a higher validation accuracy of **51.24%** over 600 epochs.

The improved accuracy for the **Gesture_GRU** model indicates its superior ability to learn temporal patterns and relationships within the input data.

5.3. Accuracy Comparison

The comparison of training accuracy, validation accuracy, and epochs for both models is summarized in Table 1.

Table 1. Training and Validation Accuracy Comparison

Models	Training Accuracy	Validation Accuracy
Gesture_CNN	43.42	43.42
Gesture_GRU	51.26	51.15

5.4. Conclusion

The **Gesture_GRU** model outperformed the **Gesture_CNN** model by a significant margin in validation accuracy. The extended training duration, coupled with the GRU's capability to capture sequential dependencies, contributed to its superior performance. Further hyperparameter tuning and exploration of additional regularization techniques may further enhance the results.

6. Discussion

In this section, we analyze and compare the performance of the Gesture_CNN and Gesture_GRU models based on training and validation accuracy.

6.1. Gesture_CNN

The Gesture_CNN model, a Convolutional Neural Network-based approach, achieved a training accuracy of **43.42%** and a validation accuracy of **43.42%**. CNNs are highly effective for extracting spatial features from input data and are typically used in image recognition tasks. However, in this context, the Gesture_CNN struggled to capture the temporal dependencies present in the sequential data, leading to limited performance. The absence of recurrent connections prevented the model from effectively learning long-term dependencies within the data.

6.2. Gesture_GRU

The Gesture_GRU model, which leverages Gated Recurrent Units (GRUs), significantly outperformed the CNN-based approach, achieving a training accuracy of **51.26%** and a validation accuracy of **51.15%**. GRUs are well-suited for sequential data as they incorporate gating mechanisms, such as the *update gate* and *reset gate*, to learn temporal relationships while mitigating the vanishing gradient problem.

The superior performance of the GRU model can be attributed to its ability to:

- Learn temporal dependencies within sequential input data.
- Effectively manage long-term memory through gating mechanisms.
- Handle complex temporal patterns in the gesture recognition task.

7. Future Improvements

While the current models have demonstrated significant progress in gesture recognition, further improvements can be made to enhance accuracy and generalization. Below are some potential directions for improvement:

- **Hybrid Architectures:** Combining Convolutional Neural Networks (CNNs) for spatial feature extraction with Gated Recurrent Units (GRUs) for temporal analysis can leverage the strengths of both architectures. This hybrid approach could provide a more comprehensive representation of the input data.
- **Hyperparameter Optimization:** Fine-tuning model parameters such as learning rates, batch sizes, number of layers, and hidden units can further improve performance. Techniques such as grid search, random search, or Bayesian optimization can be applied for automated tuning.
- **Regularization Techniques:** Adding regularization methods such as Dropout, L2 weight decay, and Batch Normalization can help prevent overfitting, especially for the GRU model trained over 600 epochs.
- **Data Augmentation:** Incorporating data augmentation techniques for gesture datasets, such as noise addition, time-warping, or scaling, can improve model robustness and generalization.

- **Transfer Learning:** Leveraging pre-trained models or fine-tuning architectures designed for similar tasks can reduce training time and enhance accuracy, particularly for limited datasets.

- **Attention Mechanisms:** Integrating attention mechanisms into the GRU model can improve the focus on important temporal features, enabling the model to learn more discriminative representations.

- **Ensemble Methods:** Combining multiple models (e.g., CNNs, GRUs, or other RNN variants like LSTMs) through ensemble learning can help mitigate the limitations of individual architectures and improve overall performance.

- **Increased Dataset Size:** Collecting additional training data with diverse gestures and conditions can help models generalize better to unseen inputs.

- **Real-time Implementation:** Optimizing models for real-time performance by reducing computational complexity and latency will be crucial for practical applications in edge devices.

By implementing these enhancements, the accuracy, efficiency, and robustness of the gesture recognition models can be further improved, paving the way for more reliable real-world applications.

8. Conclusion

In this study, we implemented and compared two deep learning models, CNN and GRU, for gesture recognition tasks. The results demonstrated that the GRU-based model outperformed the CNN model, achieving a validation accuracy of 51.15% compared to 43.42%. This improvement highlights the GRU model's ability to effectively capture temporal dependencies in sequential gesture data. Despite the moderate performance, the study provides valuable insights into the potential of recurrent neural networks for gesture classification tasks. Future enhancements, such as hybrid models, attention mechanisms, and deployment on real-world applications, can further improve accuracy and robustness. The development of a web-based application and integration with edge devices will enable practical and interactive implementations, broadening the usability of gesture recognition systems.

References

[1] Liu, J., et al., 2014. Passive Wi-Fi analytics for human sensing. Proceedings of the IEEE INFOCOM, pp. 3268-3276.

[2] Sun, Y., et al., 2019. Environmental-aware gesture recognition with Wi-Fi. Proceedings of the ACM IMWUT, 3(4), pp. 1-25.

[3] Tang, H., et al., 2020. Gesture recognition using deep learning on RSSI data. Journal of Machine Learning Research, 21(1), pp. 1-15.

[4] Wang, Y., et al., 2017. Impact of device density on RSSI fluctuations in indoor environments. Journal of Wireless Networks, 23(3), pp. 649-658.

[5] Zhao, F., et al., 2016. Signal interference and mitigation in multi-user Wi-Fi sensing. IEEE Transactions on Mobile Computing, 15(8), pp. 1957-1969.

[6] Han, C., et al., 2017. Wi-Fi CSI-based passive human detection and recognition. Sensors and Actuators A: Physical, 257, pp. 190-204.

[7] Yang, S., et al., 2020. Wi-Fi for sensing: Channel estimation and applications. IEEE Communications Magazine, 58(1), pp. 67-73.

[8] Fan, J., et al., 2019. Pervasive computing meets machine learning: Applications and challenges. ACM Computing Surveys, 52(3), pp. 1-40.

[9] Adib, F., et al., 2015. Smart homes that monitor breathing and heart rate. Proceedings of the ACM CHI Conference on Human Factors in Computing Systems, pp. 837-846.

[10] Zhang, J., et al., 2019. Motion detection using ambient wireless signals. IEEE Transactions on Mobile Computing, 18(4), pp. 943-957.

[11] Zhou, X., et al., 2018. Leveraging RSSI for low-power human motion sensing. ACM Transactions on Sensor Networks, 14(3), pp. 1-19.

[12] Alizadeh, M., et al., 2017. Wi-Charge: Gesture recognition through powerline noise analysis. Proceedings of the ACM IMWUT, 1(3), pp. 1-21.

[13] Yang, T., et al., 2022. Real-time Wi-Fi sensing for activity detection in smart homes. IEEE Access, 10, pp. 2574-2589.

[14] Gupta, S., et al., 2013. Electromagnetic noise cancellation for gesture recognition. ACM SIGGRAPH Asia 2013 Technical Briefs, pp. 1-4.

[15] Sun, Y., et al., 2019. Environmental-aware gesture recognition with Wi-Fi. Proceedings of the ACM IMWUT, 3(4), pp. 1-25.

[16] Chung, J., et al., 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.

[17] Yang, X., et al., 2021. GRU-based activity recognition using wireless sensing data. IEEE Transactions on Mobile Computing, 20(10), pp. 2347-2358.