Investigation of Training Robustness Techniques in WiFi Sensing Deep Learning Models

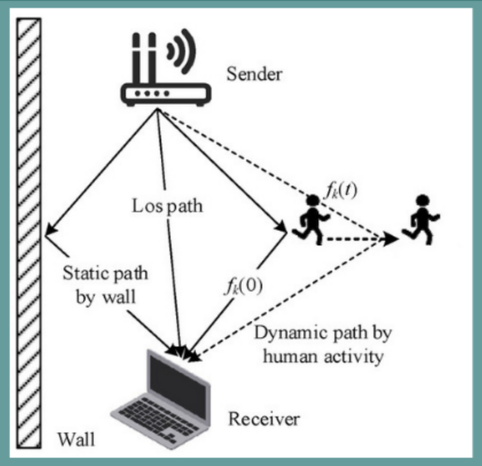
Author: Oleh Grypas
(O.Grypas@student.tudelft.nl)
Supervisors:
Fabian Portner
Dr. Arash Asadi

Why WiFi Sensing?

- Non-intrusive technology
- Very broad applicability: activity recognition, health monitoring, through-wall tracking, etc
- Leverages existing infrastructure

The Problem

- Training on limited data leads to high variability and low accuracy/robustness
- Small environment changes → large performance shifts
- Many training techniques developed. Unknown effect on Wifi Sensing models



Multipath propagation in WiFi sensing (adapted from [1])

How can state-of-the-art learning robustness techniques be leveraged to improve the stability and robustness of deep neural networks for WiFi sensing, particularly under limited data conditions?

Methodology

We began with a deep literature survey to build a clear taxonomy of robustness methods splitting them into training-focused (how a model learns) versus architecture-focused (how the model is made) and to identify best practices for reporting results from limited-data CSI experiments using statistically robust means.

Experimental setup

Our study extends the SenseFi benchmarking framework [2], leveraging its standardized implementations of three widely used model architectures—LeNet (CNN) for spatial pattern recognition, LSTM for temporal modeling, and hybrid CNN+GRU for spatio-temporal learning—along with its preprocessed versions of two public WiFi sensing datasets: Widar [8] (moderate baseline accuracy) and NTU-Fi [2] (near-saturated performance).

Technique Selection

From our taxonomy, we selected five lightweight, training-focused techniques that each address a core small-data challenge:

- **Stability Training** to combat signal noise [3]
- **MixUp** to smooth decision boundaries [4]
- **Weight Decay** to constrain model complexity [5]
- **Early Stopping** to prevent overfitting [6]
- **Label Smoothing** to reduce prediction overconfidence [7]

Rigorous Evaluation

Each technique was applied both individually and in strategic combinations across all architecture-dataset pairs as can be seen in Table 1. Performance was measured through:

- **Repeated trials of stratified 5-fold cross-validation** to guard against lucky initialization and optimistic bias.
- Statistical reporting of **mean accuracy ± standard deviation** to ensure reliable variance estimation in limited-data regimes.

Future work

Possible improvements are to test additional robustness methods—such as adversarial training and self-supervised pretraining—and see how they perform when we use hyperparameter tuning, and vary dataset sizes. Additionally, running more combinations of techniques and increasing repeat trials will also help us better gauge which approaches reliably boost WiFi-CSI model stability.

Method	LeNet	LSTM	CNN+GRU	LeNet	LSTM	CNN+GRU
Baseline	0.6741 ± 0.0062	0.6021 ± 0.0066	0.6799 ± 0.0089	0.9950 ± 0.0038	0.9963 ± 0.0054	0.9764 ± 0.0134
Stability training	0.7055 ± 0.0058	0.6045 ± 0.0046	0.6884 ± 0.0164	0.9963 ± 0.0031	0.9963 ± 0.0046	0.9847 ± 0.0158
Mixup	0.6740 ± 0.0094	0.6091 ± 0.0059	0.6732 ± 0.0145	0.9967 ± 0.0038	0.9867 ± 0.0096	0.9681 ± 0.0146
Weight Decay	0.6849 ± 0.0065	0.6369 ± 0.0054	0.6853 ± 0.0058	0.9942 ± 0.0049	0.9979 ± 0.0029	0.9542 ± 0.0167
Early Stopping	0.6836 ± 0.0090	0.6068 ± 0.0063	0.6821 ± 0.0065	0.9954 ± 0.0060	0.9983 ± 0.0029	0.8333 ± 0.2278
Label Smoothing	0.6754 ± 0.0082	0.5821 ± 0.0068	0.6861 ± 0.0126	0.9979 ± 0.0029	0.9967 ± 0.0038	0.9583 ± 0.0123
Generalization	0.6835 ± 0.0068	0.6238 ± 0.0119	0.6858 ± 0.0079	0.9983 ± 0.0029	0.9925 ± 0.0085	0.8979 ± 0.0688
Stable MixUp	0.6810 ± 0.0091	0.6222 ± 0.0054	0.6931 ± 0.0086	0.9963 ± 0.0041	0.9950 ± 0.0043	0.9344 ± 0.0244
Stable Stopping	0.7041 ± 0.0089	0.6050 ± 0.0067	0.6837 ± 0.0204	0.9938 ± 0.0035	0.9913 ± 0.0077	0.8094 ± 0.0933
Mixed Trio	0.6287 ± 0.0078	0.5311 ± 0.0134	0.4071 ± 0.0289	0.9875 ± 0.0171	0.9433 ± 0.0253	0.8750 ± 0.1197
Full Combination	0.6949 ± 0.0113	0.6848 ± 0.0053	0.6516 ± 0.0197	0.9963 ± 0.0050	0.9654 ± 0.0155	0.7719 ± 0.2766

Table 1 – Classification accuracy (mean ± std) across all model–technique–dataset combinations. Left: Widar dataset (moderate baseline); Right: NTU-Fi dataset (near-saturated baseline).

Results

Our evaluation revealed **architecture and dataset-dependent patterns**:

- On Widar, stability training consistently boosted accuracy with LeNet benefiting the most (**+3.2%**) while weight decay maximized LSTM gains (**+3.7%**). Hybrid CNN+GRU saw limited improvements. Combinations of methods generally showed an increase in LeNet and LSTM performance.
- On NTU-Fi (near-saturated), techniques showed **neutral-to-harmful effects**: early stopping reduced CNN+GRU accuracy by **14.3%**, while label smoothing offered marginal calibration benefits.
- **Combinations proved high-risk high-reward**: Aggressive stacks (e.g., Stability+MixUp+Early Stopping) collapsed CNN+GRU performance by **27.3%** on Widar, though full combinations aided LSTMs (**+8.5%**).

Conclusion

Based on our findings, it is advised to apply robustness techniques with these guidelines:

For Moderate Baselines (e.g., Widar)

- Use Weight Decay for LSTMs to control overfitting
- Test combinations cautiously: synergies are possible but not guaranteed

For Near-Saturated Tasks (e.g., NTU-Fi)

- Minimize intervention. baseline performance often suffices
- If applying techniques, use Label Smoothing for safe calibration

Universal Rules

- Avoid combining Early Stopping with hybrids (CNN+GRU)
- Prioritize Stability Training as your primary defense against noise

[1] Wang Y., Wu K., & Ni L. M. (2019). TensorBeat: Monitoring breathing beats with WiFi. IEEE TMC, 18(3).
[2] Wang Y., et al. (2021). SenseFi: A Benchmark for WiFi Sensing. UbiComp.
[3] Zheng S., et al. (2016). Improving Robustness via Stability Training. arXiv:1604.04326.
[4] Zhang H., et al. (2018). mixup: Beyond ERM. ICLR.
[5] Krogh A. & Hertz J. (1992). A Simple Weight Decay. NIPS.
[6] Prechelt L. (1998). Early Stopping—But When? in *Tricks of the Trade*.
[7] Szegedy C., et al. (2016). Rethinking the Inception Architecture. CVPR.
[8] Pu Q., et al. (2018). Widar: WiFi-based Device-free Gesture Recognition. CCS.