# Use-case: Retail Loss Prevention with Wi-Fi sensing

Here's a concrete, engineer-level playbook: how to *actually* realize Wi-Fi sensing for catching someone lifting a high-end display or shoplifting, what the system would detect, how you build it, and realistic limits.

## 1) What the system *can* detect (intuition → measurable signal)

- **Object removal / lift**: removing a display changes the multipath environment (one or more strong static reflectors disappear). This produces a *sustained* change in the CSI (amplitude & phase) and in the power-delay profile.
- **Human interaction / reach-and-grab**: transient Doppler shifts and short-duration spectrogram patterns as hands move near/around the display.
- **Manipulation vs browsing**: brief hand-motions (browsing) produce short, low-energy perturbations; deliberate lifting tends to create: (a) approach signature, (b) grasp signature (distinct spectrotemporal pattern), (c) object removal (sustained baseline shift).
- **Tampering / concealment**: shielding or covering causes characteristic slow changes in amplitude/phase across subcarriers.

# 2) Concrete hardware & deployment patterns

- **Receivers / transmitters**:
  - Use commodity APs / routers that expose CSI (Intel 5300 NIC, Atheros, or Broadcom with Nexmon on Raspberry Pi). For early prototyping, Intel 5300 + laptop is easiest; for product, use mesh APs with firmware (prpl/ OpenWRT approach ).
  - Place **2–4 radios** as a small geometry around the high-value display (front, sides, ceiling) to get spatial diversity and angle sensitivity.
- **Form factor**:
  - Routers / APs mounted on ceiling or shelf; receivers can be locked in place (stationary) to maximize repeatability.
- **Optional fusion sensors (recommended for robustness)**:
  - Thin load cell / strain gauge under the display base or shelf (if allowed) — provides ground truth/backup.
  - Magnetic/contact switch on mount (discrete, cheap).
  - RFID / BLE-tagged commodity items (when allowed).
  - CCTV snapshot-on-alert (privacy-preserving: only on suspected theft).
- **Edge compute**: small SBC (RPi4/Compute Module) or router SoC runs preprocessing + streaming to central inference if needed. Notes: cloud vs local tradeoffs.

# 3) Signal processing pipeline (practical steps)

1. **Raw capture**: collect CSI (per subcarrier amplitude & phase) at >100–500 Hz if possible for hand movements; lower is OK for slow changes.

2. **Preprocess**:
   - Phase sanitization (remove linear phase offset).
   - Amplitude normalization, smoothing, median filtering to remove burst noise.
   - Remove static component (background subtraction) for activity detection; but *retain* long-term baseline for object-presence checks.

3. **Feature extraction** (examples):
   - Short-Time Fourier Transform (STFT) spectrograms per link (Doppler features).
   - Principal Component Analysis (PCA) or SVD to reduce dimensionality and remove common-mode noise.
   - Time-domain envelope and delta of per-subcarrier amplitude (to detect sustained baseline shift).
   - Correlation across antennas/links to localize the perturbation.
     (DSP → PCA → STFT → spectrogram pipeline).

4. **Event detection**:
   - *Transient detector* (sliding-window energy on STFT → candidate "reach/grasp").
   - *Change-point detector* (CUSUM or Bayesian online change detection) to detect sustained baseline change indicating object removal.

5 **Classification / scoring**:
   - Train a classifier on labeled actions (browse, pick, lift, place) using spectrogram CNN or LSTM/Transformer on time-series.
   - For low-data regimes use anomaly detection / one-class models for "normal browsing" and flag outliers (useful for shopfloors with varied behavior).

6 **Decision fusion**:
   - Fuse outputs across multiple radios and sensors using voting/weighted score.
   - Require corroboration: e.g., transient (hand) + sustained baseline shift + localization near the display ⇒ high-confidence theft alert.

# 4) Example detection sequence — "someone lifts a display"

1 **Approach**: small low-frequency Doppler, localized to side link A.

2 **Grab**: short high-energy spectrotemporal burst across links near the item (hand motion).

3 **Lift**: change-point — sustained amplitude/phase baseline shift on links whose dominant reflection included the display. This persists until the object is replaced.

4 **Confirm**: fusion rule triggers: (grab + lift + sustained baseline shift) from same spatial cluster ⇒ alert.

5 **Action**: snapshot camera, local alarm, security notification with timestamp + audio snapshot (if policy allows).

This is realistic: object removal modifies the static multipath and causes a measurable persistent delta — not just transient noise.

# 5) ML models & training strategy

- **Dataset**: collect many trials per display: normal browsing, picking up and replacing, full removal, long-term shelf changes, many different people/heights/angles. Aim for 200–1000 labeled events per class for supervised CNN approach; fewer for anomaly/one-class approaches.
- **Model choices**:
  - **CNN on spectrograms** for action classification (works well for Doppler-like features).
  - **LSTM/Transformer** for sequential context if you want temporal reasoning (approach → grab → lift).
  - **One-class / autoencoder** for environments where labeled theft data is scarce — model "normal" and flag deviations.
- **Evaluation metrics**: precision, recall, F1, false alarm rate per hour (FAR/h), ROC/AUC. Retail deployments prioritize **low false alarms** (so require high precision at reasonable recall).
- **Transfer / adaptation**: environment-adaptive models

(fine-tune per store) or domain-adaptive layers to handle furniture changes.

# 6) Practical performance expectations & limitations

- **Can catch many lift events** especially for medium-to-large objects that meaningfully alter multipath.
- **Hard cases**:
  - tiny, concealed thefts (e.g., slipping a small expensive part into pocket) — signal weak.
  - crowded conditions: multiple simultaneous movers create overlapping signals (degrades accuracy).
  - very slow/gradual removal that imitates normal browsing (can be reduced via fusion with weight/contact sensors).
- **False positives**: reconfiguration, staff restocking, moving a cart. Mitigation: staff-authenticated windows, scheduled maintenance mode, or sensitivity zoning.
- **Latency**: real-time detection with edge processing possible (hundreds of ms to seconds).
- **Privacy**: no images required until a corroboration event (privacy-friendly). But check local privacy laws and store policy.

# 7) Evaluation plan (how you prove it works)

Design an experiment:

1. Controlled trials: 20 people, multiple sessions, actions: browse, pick-and-replace, lift-and-walk-away. Record CSI + video ground truth + any weight sensors.
2. Metrics: per-event detection rate, false alarm rate/hr, time-to-detect.
3. Ablation: single-AP vs multi-AP, with/without phase features, different sampling rates.
4. Stress tests: busy background (30 people), different times of day, furniture moved, new AP positions.
5. Calibration: show ROC curves and choose operating point that balances recall vs false alarms acceptable for store.

# 8) Hybrid deployment (practical commercial recipe)

Because Wi-Fi sensing alone is probabilistic, real deployments typically use **sensor fusion**:

- Wi-Fi sensing (wide-area, non-intrusive) → early detection / anomaly flag
- Weight/load cell or contact switch (discrete verification) → confirm removal
- Snapshot camera on alert (privacy policy controlled) → visual verification
  This combination yields high precision and acceptable recall while respecting privacy.

# 9) Quick prototype steps you could run on

# Raspberry Pi (when you resume prototyping)

1. Get a Broadcom-based Pi (Raspberry Pi 4) and use **Nexmon** to extract CSI from Broadcom Wi-Fi chips (there are guides). Or use an Intel 5300 NIC with Ubuntu laptop for easier CSI collection.
2. Place two Wi-Fi radios opposite a small display (or small object) on a table.
3. Record baseline CSI for 30 minutes (normal browsing).
4. Record labeled events: pick-and-replace, lift-off, approach, etc.
5. Compute STFT spectrogram per link, run PCA, visualize differences. Train a small CNN on spectrograms.
6. Evaluate and iterate. ("next steps"- hands-on prototype testing).

# 10) Concrete example rule for in-store alert (practical)

- If: (Transient energy > T1) on links within 2-second window *AND* (Change-point magnitude > T2) on same spatial cluster for > 5 seconds *AND* (localization within display zone)
- Then: escalate to "suspected removal" → snapshot + notify guard. Tune T1/T2 from validation dataset to set FAR ≤ desired threshold (e.g., <1 false alarm per 24 hours).

# 11) Next steps

- **Short term**: run a small RPi or Intel5300 experiment at home with a single display and a few labeled events to see the baseline signals.

- **Medium term**: build multi-AP prototype, add one load cell under a sample display for ground truth.
- **Long term**: evaluate in live store (pilot) with staff-auth mode and privacy controls.