An AI-Based Low-Cost System for Detecting Fake 5G Base Stations Using Open 5G Signal Measurements

The research adopts a quantitative, experimental design in a controlled laboratory environment, consistent with established FBS detection studies that use measurement data and ML classification

Sayem Akther Talukdar
Student ID: 2502919
MSc in Cybersecurity & Digital Forensics
Research Methods and Professional Development

Research Context

01

Fake Base Stations (FBS), also known as IMSI-catchers or Stingrays, impersonate legitimate cellular towers.

02

4G/5G early broadcast messages are not fully authenticated.

03

Attackers can:

- Extract IMSI or SUCI
- Force downgrade to insecure 2G/3G
- Launch denial-of-service attacks

04

5G improves security but early handshake remains vulnerable [2], [4].

Key Issue:
Current detection systems are accurate but expensive and not suitable for low-cost deployment.

Research Problem & Literature Gap

Research Problem

There is a lack of:

- Low-cost, edge-deployable FBS detection systems
- Open datasets for reproducibility
- Real-time latency and resource evaluation on embedded devices

Literature Gap

Existing Work	Limitation
SMDFbs [1]	Complex protocol modelling
FBSDetector [2]	Requires large datasets & powerful hardware
SeaGlass [7]	Needs large infrastructure

❖ No affordable Raspberry Pi–based solution with open dataset and latency profiling

Research Aim & Objectives

Aim

To design, implement, and evaluate a low-cost FBS detection system using Raspberry Pi and SDR.

Objectives

- ✓Conduct literature review
- ✓Set up hardware & simulation
- ✓Collect ≥5,000 labelled samples
- ✓Train 5 ML models (Decision Tree, RF, k-NN, SVM, LightGBM)
- ✓Deploy best model on Raspberry Pi
- ✓Achieve inference latency <200ms
- ✓Compare with SMDFbs [1] & FBSDetector [2]

Literature Review Summary

1.Specification-Based Detection

- SMDFbs achieves ~98% accuracy [1]
- Uses protocol state machine
- Not suitable for low-cost edge deployment

2. Machine Learning Detection

- ML on RSRP features shows 75–96% accuracy [2], [3], [6]
- Requires large datasets
- Not evaluated on Raspberry Pi

3. Crowdsourced Detection

- SeaGlass & FBS-Radar [7], [10]
- City-scale monitoring
- Requires fleet infrastructure

➢ Identified Gap

High accuracy exists — but low-cost edge deployment is missing.

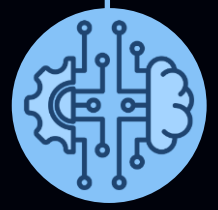
Research Methodology

Research Design

- Quantitative
- Experimental
- Controlled lab environment

Tools & Hardware

- Raspberry Pi 4
- RTL-SDR / HackRF
- srsRAN / ns-3
- Python (scikit-learn, LightGBM)



ML Models

- Decision Tree
- Random Forest
- k-NN
- SVM
- LightGBM

Evaluation Metrics

- Accuracy
- Precision
- Recall
- F1-score
- ROC-AUC

Data Collection & Deployment

Data Collection

- Simulated genuine & fake gNodeBs
- Features collected:
  - RSRP
  - RSRQ
  - SNR
  - Timing Advance
  - PCI
  - MNC
- Target: ≥5,000 labelled samples

Deployment

- Best model exported using joblib
- Raspberry Pi inference pipeline
- Target latency: <200ms
- CPU & memory profiling

Project Plan

Phase 1
Literature Review

Weeks 1 & 2 (March 1 – March 14)

- ✓ Review recent journal papers & conference articles
- ✓ Identify research gaps
- ✓ Study related methodologies & tools
- ✓ Prepare literature summary document

Phase 2
Experimental Setup

Week 3 (March 15 – March 21)

- ✓ Install required software & frameworks
- ✓ Configure hardware
- ✓ Set up dataset storage
- ✓ Environment testing

Phase 3
Data Preparation

Weeks 4–5 (March 22 – April 4)

- ✓ Collect raw datasets
- ✓ Data cleaning & preprocessing
- ✓ Data labeling (if required)
- ✓ Split dataset (train/test/validation)

Phase 4
Model Development

Weeks 6–7 (April 5 – April 18)

- ✓ Select model architecture
- ✓ Hyperparameter tuning
- ✓ Training & validation
- ✓ Save trained models

Phase 5
Deployment & Performance Analysis

Weeks 8–9 (April 19 – May 2)

- ✓ Deploy model (local/cloud/edge)
- ✓ Measure latency & resource usage
- ✓ Performance optimization
- ✓ Profiling report

Phase 6
Evaluation & Comparison

Week 10 (May 3 – May 9)

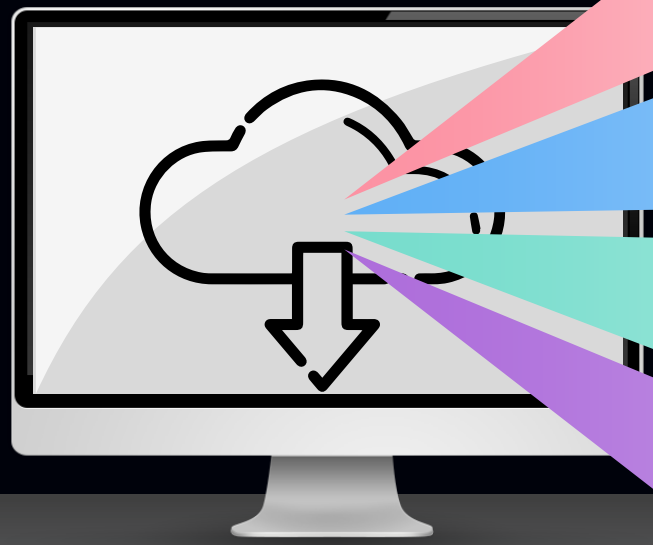
- ✓ Compare with baseline models
- ✓ Statistical performance analysis
- ✓ Generate graphs & tables
- ✓ Interpretation of results

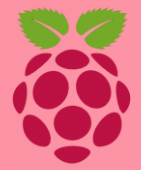
Phase 6
Final Documentation & Thesis Writing

Weeks 11–15 (May 10 – 2nd Week of August)

- ✓ Final Report Writing & Revision

Resources





Raspberry Pi 4

A low-cost single-board computer used as the edge device for deploying and evaluating the detection model. It allows real-time inference and resource profiling in a constrained environment.

SDR (Software-Defined Radio)

Used to capture and analyse 4G/5G signal measurements. It enables flexible and low-cost radio signal monitoring without specialised telecom hardware.

Ubuntu laptop

Acts as the base-station simulation environment. It runs srsRAN and/or ns-3 to generate both genuine and fake gNodeB transmissions under controlled laboratory conditions.

Open-source software

This project uses a Raspberry Pi 4 to deploy the detection model, an SDR to capture 4G/5G signals, and an Ubuntu laptop to simulate base stations. All tools are open source to ensure low cost and reproducibility.

Impact

Impact & Limitations

01 Academic

- Open labelled dataset
- Reproducible research

02 Professional

- Low-cost monitoring for small operators

03 Societal

- Protection against IMSI-catchers

Limitations

Impact & Limitations

01 Simulated environment

02 Limited model complexity

03 Blockchain component optional

04 Focus on broadcast-level detection only

References

[1] H. Park, P. V. B. Astillo, Y. Ko, Y. Park, T. Kim, and I. You, "SMDFbs: Specification-based misbehaviour detection for false base stations," Sensors, vol. 23, no. 23, p. 9504, 2023. doi: 10.3390/s23239504.

[2] K. S. Mubasshir, I. Karim, and E. Bertino, "Gotta detect 'em all: Fake base station and multi-step attack detection in cellular networks," in Proc. 34th USENIX Security Symposium, 2025. [Preprint].

[3] P. K. Nakarmi, J. Sternby, and I. Ullah, "Applying machine learning on RSRP-based features for false base station detection," arXiv preprint arXiv:2207.10999, 2022. [Online].

[4] M. Chlosta, D. Rupprecht, C. Pöpper, and T. Holz, "5G SUCI-catchers: Still catching them all?" in Proc. 14th ACM Conf. Security Privacy Wireless Mobile Netw., 2021, pp. 359–364. doi: 10.1145/3448300.3467826.

[5] A. Islam, S. Parvez, and S.-Y. Chang, "Anomaly detection against fake base station threats using machine learning," J. Cybersecurity Privacy, vol. 5, no. 4, p. 94, 2025. doi: 10.3390/jcp5040094.

[6] J. Jin, C. Lian, and M. Xu, "Rogue base station detection using a machine learning approach," in Proc. 28th Wireless Opt. Commun. Conf. (WOCC), 2019, pp. 1–5. doi: 10.1109/WOCC.2019.8770554.

[7] P. Ney, I. Smith, G. Cadamuro, and T. Kohno, "SeaGlass: Enabling city-wide IMSI-catcher detection," Proc. Privacy Enhancing Technol., vol. 2017, no. 3, pp. 39–56, 2017. doi: 10.1515/popets-2017-0027.

[8] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for IoT security and privacy: The case study of a smart home," in Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops, 2017, pp. 618–623. doi: 10.1109/PERCOMW.2017.7917634.

[9] A. Ali and G. Fischer, "Enabling fake base station detection through sample-based higher order noise statistics," in Proc. 42nd Int. Conf. Telecommun. Signal Process. (TSP), 2019, pp. 695–700. doi: 10.1109/TSP.2019.8769046.

[10] Z. Li et al., "FBS-Radar: Uncovering fake base stations at scale in the wild," in Proc. 2017 Netw. Distributed Syst. Security Symp. (NDSS), 2017. [Online].

THANK YOU