Midsemester Project Report: Predictive Modelling of Signal Strength and Handover Events Using Machine Learning

Project Title: Predictive Modelling of Signal Strength and Handover Events Using ML Techniques

Authors: (Group 17)

Date: November 2025

Abstract

Handover (HO) is a critical procedure that ensures seamless mobility in cellular networks. Traditional threshold-based handover strategies used in LTE and 5G systems often suffer from suboptimal performance in dynamic environments, especially under fast user mobility, fluctuating channel conditions, and irregular cell coverage boundaries. This project develops a **hybrid MATLAB–Python framework** for predictive handover modelling by combining synthetic radio channel simulation with **real-world 5G measurements** collected using GNet Track Lite on the Jio True 5G network. A MATLAB simulator generates RSSI-based base station association using the Okumura–Hata model, while Python's XGBoost classifier is trained on a fused dataset containing both synthetic and real data. A MATLAB GUI integrates all components, enabling simulation, model training, real data prediction, and live visualization. Experimental results show an accuracy of **~83%**, precision of **~57%**, recall of **~36%**, and F1 score of **~44%**. The integrated GUI serves as a functional testbed for mobility prediction research, demonstrating the feasibility of combining simulation, machine learning, and real-world field measurements.

1. Introduction and Background

1.1 Problem Statement and Motivation

Mobility management is one of the most important challenges in cellular communication systems. As users move across coverage regions, the network must hand over their connection from one base station (BS) to another without interruption. Failures or delays in handover can cause throughput degradation, packet loss, dropped calls, and poor user experience.

Existing handover algorithms in LTE/5G networks rely on fixed thresholds (e.g., RSRP, RSRQ, and hysteresis). These methods often fail under:

- High user mobility
- Changing propagation environments
- Dense or irregular cell deployments
- Noisy or fluctuating measurements

Machine learning (ML) provides a data-driven alternative by learning patterns from network features and predicting optimal handover events ahead of time. However, ML-based mobility prediction typically requires large datasets, which are difficult to gather from live networks.

To overcome these limitations, this project proposes a **hybrid synthetic + real dataset approach**:

1. **MATLAB simulation** generates controlled radio RSSI patterns using the Okumura–Hata model.
2. **Real 5G field measurements** (Jio True 5G) provide real-world variability.
3. **Python XGBoost** learns from both domains and predicts future handovers.
4. A **MATLAB GUI** integrates the entire pipeline visually.

This provides a complete end-to-end system combining communication modelling, ML, and real-world experimentation.

1.2 Research Objectives and Scope

The primary goals of this project are:

1. Develop a MATLAB-based wireless simulator capable of modelling RSSI and base station association using the Okumura–Hata propagation model.
2. Collect real mobility data (GPS, RSSI, speed) using GNet Track Lite from a 5G network.
3. Process and clean real-world measurements for ML compatibility.
4. Fuse simulated and real datasets into a unified training dataset.
5. Train a Python XGBoost classifier for binary handover prediction.
6. Integrate the trained model into a MATLAB GUI using MATLAB–Python interoperability.
7. Visualize simulation, predictions, and confusion matrices interactively.
8. Evaluate performance and discuss strengths, limitations, and future improvements.

1.3 Literature Context and Related Work

Handover ensures the continuity of a user's radio link as they move. LTE/5G primarily use:

- **Event A3**: Neighbor cell becomes sufficiently stronger than serving cell
- **Hysteresis and Time-to-Trigger (TTT)**
- **RSRP/RSRQ-based comparisons**

However, fixed thresholds cannot adapt dynamically to:

- Irregular cell boundaries
- Shadowing & multipath
- Mobility speed variations
- Beamforming distortions
- Environmental obstacles

ML for Mobility Prediction

Machine learning offers predictive mobility management by learning patterns from radio features and user movement. XGBoost is favored for:

- Handling noisy and imbalanced datasets
- Fast training
- Strong generalization
- Low inference time

Recent studies show ML improves:

- Handover accuracy
- Ping-pong reduction
- User throughput
- Radio resource usage

Synthetic + Real Data Fusion

A hybrid dataset approach improves robustness:

- Simulation gives coverage structure
- Real data adds unpredictability and noise
- ML learns domain-invariant features

This project follows the same principle.

2. Methodology and Technical Implementation

2.1 Okumura-Hata Propagation Model

The Okumura-Hata model serves as the cornerstone of the signal strength simulation in this project. This empirical propagation model, originally developed from extensive measurements in Tokyo, Japan, provides a computationally efficient method for estimating path loss in urban and suburban mobile communication environments. The model is particularly well-suited for frequencies between 150 MHz and 1500 MHz, base station antenna heights of 30-200 meters, mobile antenna heights of 1-10 meters, and link distances ranging from 1 to 20 kilometers.

The basic path loss formula implemented in the MATLAB simulation code follows the urban environment formulation:

$$L_{path} = 69.55 + 26.16\log_{10}(f) - 13.82\log_{10}(h_b) - a(h_m) + [44.9 - 6.55\log_{10}(h_b)]\log_{10}(d)$$

where f represents the carrier frequency in MHz, h_b is the base station antenna height in meters, h_m denotes the mobile antenna height in meters, d is the distance between transmitter and receiver in kilometers, and $a(h_m)$ is the mobile antenna height correction factor.

For the mobile antenna height correction factor in medium to small cities, the model use:

$$a(h_m) = (1.1\log_{10}(f) - 0.7)h_m - (1.56\log_{10}(f) - 0.8)$$

The simulation parameters were configured as follows: carrier frequency of 900 MHz, base station antenna height of 30 meters, mobile antenna height of 1.5 meters, and a transmit power of 43 dBm (equivalent to approximately 20 watts). These parameters align with typical GSM network specifications and fall within the valid operational ranges of the Okumura-Hata model.

The received signal strength indicator (RSSI) at each mobile position is calculated by subtracting the path loss from the transmit pow:

$$RSSI(dBm) = P_{tx}(dBm) - L_{path}(dB)$$

This formulation provides realistic signal strength values that typically range from 0 dBm (very strong signal near the transmitter) to
-100 dBm or lower (weak signal at distant locations). In telecommunications practice, RSSI values of -70 dBm or higher indicate excellent signal strength, while values below -100 dBm suggest poor connectivity with potential connection loss.

2.2 Synthetic Dataset Generation

The dataset generation process implements a systematic grid-based simulation approach using MATLAB. A two-dimensional coverage area of 2 km × 2 km was discretized into a 14×14 grid, yielding 196 distinct user positions representing potential mobile device locations throughout the coverage are. Four base stations were randomly deployed within this area, with their positions fixed using a random seed for reproducibility.

For each of the 196 user positions, the simulation computes the Euclidean distance to all four base stations and applies the
Okumura-Hata path loss formula to calculate the corresponding RSSI values. A minimum distance threshold of 0.01 km (10 meters) prevents logarithmic singularities when users are extremely close to base stations. The mobile device is assumed to connect to the base station providing the strongest RSSI at each position, implementing a simple yet realistic "strongest signal first" handover policy commonly used in mobile network.

Handover events are identified through temporal analysis of base station connectivity as the user progresses through sequential grid position. When the connected base station identifier changes between consecutive positions, a handover event is flagged with a binary label of 1; otherwise, the label remains 0. This sequential processing generates a realistic pattern of handover occurrences concentrated near cell boundaries where signal strengths from multiple towers are comparable.

The resulting dataset comprises 196 observations with eight features: X-coordinate (X_km), Y-coordinate (Y_km), connected base station ID (ConnectedBS), handover indicator (Handover), and four RSSI measurements from each base station (RSSI_BS1 through RSSI_BS. This feature set captures both spatial information and multi-tower signal measurements, providing comprehensive input for machine learning models to learn complex handover patterns.

2.3 Real Jio True 5G Data Collection

GNet Track Lite was used to collect:

- Latitude
- Longitude
- RSSI
- Network type
- Speed
- Altitude
- Timestamp

All data was cleaned and interpolated.

Since actual handovers were not explicitly logged, the label was set to:

$$Handover = 0 \quad \text{for all real samples}$$

(The model uses real data only for feature learning.)

2.4 Dataset Fusion

One of the most important components of this project is the creation of a **unified dataset** that combines *both simulated wireless measurements* generated through MATLAB and *real-world field measurements* collected using GNet Track Lite on a Jio True 5G network. This fusion of heterogeneous datasets allows the machine learning model to capture both the *structured behaviour* of radio propagation (from simulation) and the *irregular, noisy variations* present in real-world measurements. This hybrid dataset approach significantly improves generalization and robustness.

2.4.1 Motivation for Dataset Fusion

Purely synthetic datasets are predictable and smooth because they come from deterministic propagation models. They lack:

- Random urban clutter
- Unpredictable fading
- Device-specific variations
- Measurement noise
- GPS fluctuations
- Real-world speed differences

On the other hand, purely real datasets are often:

- Small
- Incomplete
- Noisy
- Missing labels (real handovers aren't logged by GNet Track Lite)
- Not collected across a wide enough coverage area

Dataset fusion allows the ML model to learn the best of both worlds:

Source	Advantages
Synthetic (MATLAB)	Clean, ideal behaviour; proper HO labels; predictable structure
Real (Jio 5G)	Realistic RSSI variations, GPS drift, speed profiles, environmental randomness
Combined	Balanced dataset with structure + noise → best ML performance

2.4.2 Preprocessing of the Simulated Dataset

he MATLAB simulator uses a grid of 196 points and four base stations.
Each point generates:

- X coordinate (km)
- Y coordinate (km)
- RSSI from each BS
- Connected BS
- Handover event (0/1)

To align with the real dataset, simulated coordinates are converted:

$$Longitude = 73.881 + X \times scale$$

$$Latitude = 15.392 + Y \times scale$$

The scale matches the approximate coordinate spread of the real dataset so both datasets fall within comparable spatial bounds.

Since MATLAB simulation does not produce speed or altitude, these are synthetically generated using:

$$Speed = \mathcal{U}(1, 4)\,\mathrm{m/s}$$

$$Altitude = \mathcal{U}(-20, 10)\,\mathrm{m}$$

These reflect realistic small fluctuations and ensure feature alignment.

2.5 Preprocessing of Real-World 5G Dataset

The real dataset collected from GNet Track Lite includes:

- Timestamp
- Longitude
- Latitude
- Operator Name
- Network Tech
- Level (RSSI)
- Speed
- Altitude

The following steps were applied:

1. **Filtering:**
   Only Jio True 5G entries were retained to maintain consistency.

2. **Type Conversion & Cleaning:**
    - RSSI Level converted to numeric
    - Speed and altitude interpolated where missing
    - Outlier removal using simple thresholding
3. **Handover Label Handling:**
   Real measurements do not explicitly contain “handover" events.
   To avoid incorrect labelling:

   Handover=0

   This ensures real data only influences feature learning, not target corruption.

Aligning the Feature Sets

The fusion requires *matching columns and value types*.

Final columns: Longitude, Latitude, RSSI, Speed, Altitude, Handover

For Synthetic Data:

• Longitude/Latitude → transformed from grid

• RSSI → from MATLAB model

• Speed/Altitude → synthetic noise

• Handover → true labels (0/1)

For Real Data:

- Longitude/Latitude → directly from smartphone GPS
- RSSI → “Level” column
- Speed/Altitude → directly collected
- Handover → set to 0

Both datasets now share identical structure.

Normalization and Scaling

To avoid scale imbalance

- RSSI normalized to [−120,−60][-120, -60][−120,−60] dBm range
- Speed normalized using min-max scaling
- Altitude standardized (mean = 0, std = 1)

Longitude & latitude were left unscaled because tree-based models (like XGBoost) handle raw coordinates effectively.

Merging the Data

After alignment, the datasets were concatenated:

$$Dataset = [D_{simulated}; D_{real}]$$

Final size:

- **196** simulated samples
- **34** real-world samples
- **230** total samples

This provides:

- Correct class distribution
- Real-world signal randomness
- Meaningful coverage structure
- ML-ready features

3. Results and Performance Analysis

3.1 Model Performance Metrics

The models were evaluated using standard binary classification metrics: accuracy, precision, recall, and F1-score. These metrics provide complementary perspectives on model performance, particularly important when dealing with imbalanced datasets where accuracy alone can be misleading.

Accuracy measures the proportion of correct predictions (both true positives and true negatives) out of all predictions.
While intuitive, accuracy can be deceptive in imbalanced scenarios, as a naive classifier that always predicts the majority class would achieve 81% accuracy on this dataset simply by never predicting handovers.

Precision quantifies the proportion of predicted handover events that are actually true handovers:

$$Precision = \frac{TP}{TP + FP}$$

where TP represents true positives and FP denotes false positives. High precision indicates that when the model predicts a handover, it is usually correct, minimizing unnecessary handover preparations.

Recall (also called sensitivity or true positive rate) measures the proportion of actual handover events that the model successfully identifies:

$$Recall = \frac{TP}{TP + FN}$$

where FN represents false negatives. High recall is critical for handover prediction, as missing actual handover events (false negatives) can lead to service disruptions and degraded user experience.

F1-score combines precision and recall into a single metric using their harmonic mean:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

The harmonic mean ensures that both precision and recall must be reasonably high for a good F1-score, making it particularly suitable for imbalanced classification evaluation.

3.2 Logistic Regression Performance

Logistic Regression achieved an accuracy of 81.03% on the test set. The confusion matrix reveals the following classification outcomes:

- True Negatives (TN): 46 - correctly predicted non-handover cases
- False Positives (FP): 1 - incorrectly predicted handover when none occurred
- False Negatives (FN): 10 - missed actual handover events
- True Positives (TP): 1 - correctly predicted handover cases

From these values, the performance metrics are calculated as]:

- **Precision**: 0.50 - only half of the predicted handovers were correct
- **Recall**: 0.09 - the model detected only 9% of actual handover events
- **F1-Score**: 0.15 - poor overall balance between precision and recall

The extremely low recall of 0.09 represents the most critical weakness of the Logistic Regression model. Out of 11 actual handover events in the test set, the model correctly identified only 1, missing 10 handover occurrences. This severe underdetection of the minority class likely stems from the model's linear decision boundary and bias toward the majority class due to class imbalance.

The model's high number of true negatives (46 out of 47 non-handover cases correctly classified) contributes to the relatively acceptable overall accuracy of 81.03%. However, this accuracy is misleading, as a trivial baseline that always predicts "no handover" would achieve 81.0% accuracy simply due to class distribution. The Logistic Regression model provides minimal improvement over this baseline, demonstrating limited practical utility for handover prediction in its current form.

3.3

Random Forest demonstrated substantially better performance across all metrics. The confusion matrix shows:

- True Negatives (TN): 44 - correctly predicted non-handover cases
- False Positives (FP): 3 - incorrectly predicted handover when none occurred
- False Negatives (FN): 5 - missed actual handover events
- True Positives (TP): 6 - correctly predicted handover cases

The resulting performance metrics are:

- **Accuracy**: 86.21% - correctly classified 50 out of 58 test samples
- **Precision**: 0.67 - two-thirds of predicted handovers were correct
- **Recall**: 0.55 - the model detected 55% of actual handover events
- **F1-Score**: 0.60 - better balance between precision and recall

Random Forest's recall of 0.55 represents a dramatic improvement over Logistic Regression's 0.09, detecting 6 out of 11 actual handover events compared to just 1 for Logistic Regression. This sixfold increase in true positive detections demonstrates Random Forest's superior ability to learn the complex patterns distinguishing handover from non-handover scenarios.

The precision of 0.67 indicates that while the model generates some false alarms (3 false positives), the majority of its handover predictions are accurat. In operational contexts, false positives may be more tolerable than false negatives, as unnecessarily preparing for a handover incurs lower costs than missing an actual handover that could disrupt service. The F1-score of 0.60 reflects a reasonable balance between precision and recall, though significant room for improvement remains.

Comparing Random Forest to Logistic Regression reveals consistent superiority across all metrics. Random Forest achieves 5.18 percentage points higher accuracy (86.21% vs 81.03%), 17 percentage points higher precision (0.67 vs 0.50), 46 percentage points higher recall (0.55 vs 0.09), and 45 percentage points higher F1-score (0.60 vs 0.15). These substantial improvements validate the choice of Random Forest for handover prediction and align with broader literature findings that ensemble methods often outperform linear models on complex classification tasks.

3.4 Confusion Matrix Analysis

The confusion matrices provide detailed insights into model behavior and error patterns. For Logistic Regression, the extreme imbalance between true negatives (46) and true positives (1) reveals severe bias toward the majority class. The model correctly identifies nearly all non-handover cases but fails catastrophically on handover detection, with false negatives (10) outnumbering true positives by a factor of 10.

Random Forest's confusion matrix shows better-balanced performance. While true negatives (44) still outnumber true positives (6), reflecting the inherent class imbalance, the model achieves a more reasonable true positive count. The five false negatives represent missed handover opportunities, but this is considerably better than Logistic Regression's ten false negatives. The three false positives indicate occasional over-prediction of handovers, which may be acceptable depending on the operational cost of unnecessary handover preparation.

Error analysis suggests that both models struggle most with handover events occurring in ambiguous regions where signal strengths from multiple base stations are similar. These boundary zones present inherently difficult classification challenges, as small variations in position or signal quality can determine whether a handover occurs. The models may benefit from additional features such as signal quality metrics (RSRQ, SINR), historical user trajectory information, or tower load statistics to better disambiguate these challenging cases.Technical Strengths and Limitations

3.5 Strengths of the Approach

The project demonstrates several notable technical strengths. First, the implementation of the Okumura-Hata propagation model provides a physically realistic foundation for signal strength simulation. Unlike arbitrary synthetic data generators, this approach grounds the dataset in empirically validated radio propagation principles, ensuring that the simulated RSSI values reflect actual mobile network behavior.

Second, the choice of Random Forest and Logistic Regression as complementary algorithms enables meaningful performance comparison between a simple linear baseline and a sophisticated ensemble method. This comparative approach helps establish baseline expectations and quantify the benefits of algorithm complexity, providing actionable insights for model selection in production deployments.

Third, the comprehensive evaluation using multiple metrics beyond accuracy demonstrates methodological rigor. By reporting precision, recall, and F1-score alongside confusion matrices, the analysis provides a complete picture of model behavior, particularly important given the class imbalance present in the dataset.

Fourth, the MATLAB implementation with code documentation facilitates reproducibility and potential extension by other researchers. The systematic parameter specification, random seed usage for base station placement, and clear variable naming conventions enable others to replicate the simulation and build upon this work.

Despite these strengths, several limitations constrain the current implementation's effectiveness. The small dataset size of 196 samples represents a significant constraint. Modern machine learning systems typically require thousands to millions of training examples to achieve optimal performance, particularly for complex models like Random Forest. The limited sample size may prevent the models from learning generalizable patterns, instead capturing dataset-specific idiosyncrasies that do not transfer to real-world scenarios.

4. Future Directions and Recommendations

4.1 Dataset Enhancement

Several strategies could substantially improve dataset quality and model performance. First, expanding the dataset size through denser grid sampling or larger coverage areas would provide more training examples, particularly for the minority handover class. Generating 1,000 to 10,000 samples would better support complex model training and enable more confident performance evaluation.

Second, implementing SMOTE or other oversampling techniques could directly address class imbalance. The MATLAB implementation should experiment with different oversampling ratios, potentially targeting a 1:1 balanced distribution or intermediate ratios such as 2:1. Comparing model performance with and without SMOTE would quantify its effectiveness for this specific application.

Third, collecting real-world data using Android applications such as Network Cell Info Lite or G-NetTrack would provide ground truth validation. While challenging to implement, augmenting synthetic data with real measurements from actual mobile devices moving through cellular coverage areas would enhance model generalizability. The documentation acknowledges this challenge, particularly regarding establishing causal relationships between handovers and battery consumption.

Fourth, simulating more complex propagation scenarios incorporating terrain effects, building shadowing, and multipath fading would increase dataset realism. The Okumura-Hata model's simplicity, while computationally efficient, omits many real-world signal propagation phenomena. Extending the simulation with more sophisticated propagation models or ray-tracing techniques would capture these effects.

4.2 Feature Engineering and Model Enhancement

Expanding the feature set beyond spatial coordinates and RSSI values could significantly improve predictive capability.

Derived features such as RSSI differences between the serving cell and neighboring cells would directly capture handover decision criteria used in traditional algorithms. Rate of RSSI change over time could indicate whether signal quality is improving or degrading, providing predictive value for imminent handovers.

Signal quality metrics including RSRQ, SINR, and bit error rate would complement raw signal strength measurements.

These metrics capture interference and noise conditions that affect connection quality independently of received power. User mobility features such as estimated velocity and direction could help predict trajectory and future cell associations.

Historical handover patterns and temporal features might also prove valuable. Time-of-day indicators, day-of-week patterns, and recent handover history could capture periodic traffic patterns and user behavior regularities. Cell load and capacity information would enable the model to consider network resource availability in handover predictions.

4.3 Advanced Modeling Techniques

Exploring additional machine learning algorithms beyond Random Forest and Logistic Regression could yield performance improvements. Gradient boosting methods such as XGBoost or LightGBM often achieve state-of-the-art results on structured tabular data. Neural network architectures, particularly recurrent neural networks for sequence modeling, could capture temporal dependencies in user trajectories.

Deep learning approaches including convolutional neural networks and long short-term memory networks have shown promise in handover prediction research. These architectures can automatically learn complex feature representations from raw data without manual feature engineering. However, deep learning typically requires substantially larger datasets than the current 196 samples, necessitating dataset expansion before implementation.

Ensemble methods combining multiple algorithms through stacking or blending could leverage the complementary strengths of different models. For example, combining Random Forest's robustness with gradient boosting's optimization efficiency might yield better overall performance than either algorithm alone.

4.4 Hyperparameter Optimization

Systematic hyperparameter tuning using grid search, random search, or Bayesian optimization could substantially improve model performance. For Random Forest, key parameters to optimize include the number of trees (typically 100-1000), maximum tree depth, minimum samples per split, and the number of features sampled at each split. For Logistic Regression, regularization strength and penalty type (L1, L2, or elastic net) require tuning.

Cross-validation should guide hyperparameter selection to prevent overfitting to the training set. K-fold cross-validation with k=5 or k=10 provides robust performance estimates across multiple data splits, particularly important given the small dataset size. Performance metrics from cross-validation would offer more reliable estimates than the single train-test split currently used.

4.5 Operational Deployment Considerations

Transitioning from research prototype to operational deployment requires addressing several practical considerations. Model latency and computational requirements must be evaluated, as handover decisions often require real-time predictions within milliseconds. Random Forest with 100 trees may be too slow for low-latency applications, potentially necessitating model compression or simpler algorithms.

Integration with existing network infrastructure and handover protocols presents technical challenges. The machine learning model must interface with base station controllers and mobility management entities to receive input data and influence handover decisions. Standardized interfaces and protocols would facilitate deployment across heterogeneous network equipment from different vendors.

Model updating and retraining strategies must be established to handle evolving network conditions and user behavior patterns. Online learning or periodic batch retraining ensures that models adapt to changing environments without requiring manual intervention. Monitoring systems should track model performance metrics in production to detect degradation and trigger retraining when necessary.

Safety mechanisms and fallback strategies protect against model failures or unexpected inputs. If the machine learning model produces unreliable predictions or encounters data outside its training distribution, the system should gracefully fall back to traditional handover logic. Confidence thresholds and prediction uncertainty estimates help identify cases where model predictions may be unreliable.

4.6 Visualization and Interpretability

Developing advanced visualizations would enhance understanding of model behavior and facilitate debugging. Real-time plots of user movement across the coverage area with overlaid signal strength contours and handover predictions would provide intuitive insight into model decisions. Feature importance visualizations showing which input variables most strongly influence predictions could guide feature engineering efforts.

The presentation materials mention plans for GUI development in MATLAB to provide interactive visualization capabilities. Such interfaces would enable researchers and network engineers to explore different scenarios, adjust simulation parameters, and observe resulting handover patterns. Interactive tools facilitate hypothesis testing and rapid iteration during model development.

Decision boundary visualizations projected into two-dimensional feature subspaces would reveal how models separate handover from non-handover regions. Comparing these boundaries between Random Forest and Logistic Regression would illustrate the fundamental difference between linear and non-linear decision surfaces.

5. Conclusion

This project successfully demonstrates the feasibility and potential of machine learning approaches for predicting mobile network handover events based on signal strength and spatial location data. The implementation of the Okumura-Hata propagation model provides a physically realistic foundation for synthetic dataset generation, while the comparative evaluation of Random Forest and Logistic Regression illuminates the trade-offs between model complexity and performance.

Random Forest achieved substantially superior performance across all evaluation metrics, with 86.21% accuracy, 0.67 precision, 0.55 recall, and 0.60 F1-score. These results demonstrate the value of ensemble learning methods for capturing the complex, non-linear relationships between spatial position, signal measurements, and handover events. In contrast, Logistic Regression's linear decision boundary proved insufficient for this task, achieving only 81.03% accuracy and critically low 0.09 recall.

The severe class imbalance with handover events representing only 19% of the dataset poses a fundamental challenge that requires dedicated mitigation strategies. The planned implementation of SMOTE and class weighting techniques represents appropriate next steps toward improving minority class detection. Current recall values, while better for Random Forest than Logistic Regression, remain suboptimal for operational deployment, where failing to detect actual handover events could lead to service disruptions.

The project establishes a solid foundation for future research in predictive handover modeling. The documented MATLAB code, systematic evaluation methodology, and comprehensive analysis provide a reproducible baseline that other researchers can build upon. Recommended future directions include dataset expansion, real-world data collection, advanced feature engineering, hyperparameter optimization, and exploration of deep learning architectures.

Ultimately, this work contributes to the growing body of research demonstrating that machine learning can enhance mobile network performance through predictive analytics. By anticipating handover events before they occur, networks can proactively manage resources, reduce unnecessary transitions, and improve quality of service for end users. As mobile networks continue evolving toward 5G and beyond, intelligent, data-driven approaches to mobility management will become increasingly essential for meeting demanding performance requirements.

The practical implications extend beyond handover prediction to broader network optimization challenges including traffic forecasting, resource allocation, anomaly detection, and adaptive quality of service management. The methodologies and insights developed in this project provide a template applicable to these related problems, contributing to the larger goal of cognitive, selfoptimizing mobile networks that leverage artificial intelligence to deliver superior user experiences with minimal human intervention.