ATLAS Evaluation Framework: Academic Documentation and Reproducibility Guide

Shreyas Sreenivas

002825934

ATLAS - Adaptive Triage and Local Advisory System
IE7945 - Data Analytics Engineering Capstone Project
Master of Science Program, Northeastern University

`Note: Larger Documentation available at the Github Repository`

Executive Summary

This document provides comprehensive documentation for the ATLAS (Adaptive Triage and Local Advisory System) evaluation framework, designed to assess AI-powered clinical decision support systems in resource-limited healthcare environments. The framework implements rigorous evaluation methodologies combining technical performance benchmarks, clinical validation protocols, expert assessment surveys, and implementation science frameworks to provide evidence-based evaluation of healthcare AI systems.

Primary Contribution: A reproducible, multi-dimensional evaluation framework that bridges the gap between technical AI system assessment and real-world healthcare implementation readiness, specifically designed for resource-constrained clinical environments.

1. Introduction and Research Context

1.1 Problem Statement

Healthcare providers in resource-limited settings serve approximately 3.6 billion people worldwide without reliable internet connectivity, advanced hardware infrastructure, or consistent access to specialist clinical knowledge. Existing clinical decision support systems are either too sophisticated (requiring continuous connectivity and expensive infrastructure) or too limited (lacking AI capabilities and sophisticated clinical reasoning). This creates a critical gap in healthcare equity and clinical decision support availability.

1.2 Research Objectives

The ATLAS evaluation framework addresses the following research objectives:

1. **Technical Validation**: Assess Progressive Web Application (PWA) performance, offline functionality, and system reliability in resource-constrained environments
2. **Clinical Validation**: Evaluate AI recommendation accuracy and alignment with World Health Organization (WHO) clinical guidelines across diverse medical scenarios
3. **Implementation Assessment**: Apply implementation science frameworks (NASSS and RE-AIM) to determine deployment readiness and complexity
4. **Expert Validation**: Incorporate clinical expert assessments using standardized usability and appropriateness metrics

1.3 Methodological Approach

The evaluation framework employs a mixed-methods approach combining:

- **Quantitative Assessment**: Technical performance metrics, clinical accuracy measurements, statistical validation
- **Qualitative Assessment**: Expert clinical evaluations, thematic analysis of user feedback
- **Implementation Science**: NASSS complexity assessment and RE-AIM implementation readiness evaluation
- **Reproducible Research**: Open-source evaluation tools and synthetic datasets for validation

2. System Under Evaluation

2.1 ATLAS System Overview

ATLAS is a Progressive Web Application that provides AI-enhanced clinical decision support specifically designed for resource-limited healthcare settings. Key technical specifications:

- **Architecture**: Next.js 14 with React 18, offline-first design
- **AI Integration**: Google Gemini 2.5 Flash API with hybrid fallback architecture
- **Data Persistence**: Client-side IndexedDB storage via Dexie.js
- **Offline Capability**: Service Worker implementation with intelligent caching
- **Clinical Knowledge**: WHO-aligned protocols and evidence-based guidelines

Live Application: https://atlas-clinical-git-main-lambdabypis-projects.vercel.app/
Complete Codebase: https://github.com/lambdabypi/ATLAS

2.2 Technical Innovation

The system implements a novel three-tier AI architecture:

1. **Primary**: Google Gemini API (online connectivity)
2. **Secondary**: Clinical RAG with semantic embeddings (offline capability)
3. **Tertiary**: Rule-based emergency protocols (guaranteed availability)

This hybrid approach ensures 95% system availability while maintaining clinical decision quality across varying infrastructure conditions.

3. Evaluation Framework Architecture

3.1 Framework Components

The evaluation framework comprises four integrated assessment dimensions:

3.1.1 Technical Performance Assessment

- **PWA Compliance**: Lighthouse audit scores across performance, accessibility, and best practices
- **Offline Functionality**: Service worker effectiveness, cache performance, data persistence
- **System Reliability**: Error rates, response times, resource utilization
- **Network Adaptability**: Performance across 4G/3G/2G/offline conditions

3.1.2 Clinical Validation Protocol

- **Synthetic Scenario Generation**: 100 WHO-aligned clinical cases across four domains
- **AI Accuracy Assessment**: Diagnostic appropriateness, treatment recommendations, safety protocols
- **Guideline Compliance**: Alignment with WHO SMART Guidelines and evidence-based protocols
- **Resource Awareness**: Appropriateness of recommendations for resource-limited settings

3.1.3 Expert Clinical Assessment

- **Structured Evaluation**: Clinical appropriateness, technical implementation, implementation feasibility
- **System Usability Scale (SUS)**: Standardized usability assessment with healthcare professional validation
- **Thematic Analysis**: Qualitative assessment of expert feedback and recommendations
- **Safety Assessment**: Clinical safety protocols and risk mitigation evaluation

3.1.4 Implementation Science Evaluation

- **NASSS Framework**: Seven-domain complexity assessment (Technology, Value Proposition, Adopters, Organization, Wider System, Embedding, Adaptation)
- **RE-AIM Framework**: Five-dimension implementation readiness (Reach, Effectiveness, Adoption, Implementation, Maintenance)
- **WHO MAPS Assessment**: Maturity and scale readiness evaluation

3.2 Evaluation Methodology

The framework employs rigorous evaluation methodology with the following characteristics:

- **Reproducibility**: All evaluation scripts, synthetic datasets, and analysis methods are open-source and documented
- **Statistical Rigor**: Confidence intervals, significance testing, and correlation analysis using both Python and R
- **Multi-dimensional Integration**: Combined technical, clinical, and implementation perspectives
- **Evidence-based Thresholds**: Performance targets based on healthcare industry standards and WHO recommendations

4. Implementation and Technical Specifications

4.1 Directory Structure

```
evaluation/
├── config/
│   └── test_config.json                  # Configuration parameters
├── data/                                 # Manual assessment data
│   ├── expert_surveys/                   # Clinical expert evaluations
│   ├── framework_assessments/            # NASSS & RE-AIM assessments
│   └── synthetic_scenarios/              # WHO-aligned test cases
├── results/                              # Organized final outputs
│   ├── clinical_validation/              # Clinical analysis results
│   ├── performance_metrics/              # Technical performance data
│   └── reports/                           # Comprehensive analysis reports
└── scripts/                               # Evaluation implementation
    ├── atlas_data_analysis.py              # Primary analysis framework
    ├── atlas_evaluation_analysis.R         # Statistical analysis
    ├── atlas_live_testing.py               # Live system testing
    │
    ├── [Generated Outputs]
    ├── evaluation_results/                 # Primary analysis outputs
    │   ├── atlas_evaluation_dashboard.png  # Visualization dashboard
    │   ├── atlas_evaluation_report.json    # Comprehensive report
    │   ├── clinical_validation_results.csv # Clinical accuracy metrics
    │   ├── nasss_assessment.csv           # NASSS complexity scores
    │   ├── performance_metrics.csv        # Technical performance data
    │   └── reaim_assessment.csv           # RE-AIM readiness scores
    │
    └── live_test_results/                 # Browser automation outputs
        ├── atlas_live_test_report.json    # Live testing comprehensive report
        └── test_summary.csv               # Test execution summary
```

4.2 Evaluation Scripts

4.2.1 `atlas_live_testing.py` - Automated System Testing

Purpose: Comprehensive automated testing of the live ATLAS system using browser automation and performance monitoring.

Technical Implementation:

- **Browser Automation**: Selenium WebDriver with Chrome headless mode
- **Multi-viewport Testing**: Responsive design validation across mobile (375×667), tablet (768×1024), and desktop (1920×1080) viewports
- **PWA Validation**: Automated testing of service worker registration, manifest accessibility, and offline storage capabilities
- **Network Simulation**: Chrome DevTools Protocol integration for offline condition simulation
- **Performance Integration**: Lighthouse CLI automation with error handling for multiple installation methods

Key Testing Modules:

```
class ATLASLiveTester:
    def test_application_accessibility(self)      # HTTP endpoint validation
    def test_pwa_functionality(self)             # PWA compliance testing
    def test_user_interface_responsiveness(self) # Multi-device compatibility
    def test_patient_management_workflow(self)   # Clinical workflow validation
    def test_consultation_workflow(self)         # AI feature integration testing
    def test_offline_functionality_simulation(self) # Network condition simulation
    def run_lighthouse_audit(self)              # Automated PWA performance audit
```

Outputs Generated:

- `live_test_results/atlas_live_test_report.json`: Comprehensive browser automation results with health scoring
- `live_test_results/test_summary.csv`: Test execution summary with success/failure analysis
- `lighthouse_results.json`: PWA performance metrics (if Lighthouse CLI available)

4.2.2 `atlas_data_analysis.py` - Comprehensive Evaluation Framework

Purpose: Implementation of multi-framework evaluation methodology combining performance analysis, clinical validation, expert assessment, and implementation science frameworks.

Framework Implementation:

ATLASPerformanceAnalyzer:

- Lighthouse data processing and PWA metric analysis
- Network condition simulation (4G/3G/2G/offline scenarios)
- Performance threshold comparison against healthcare industry standards

ClinicalScenarioAnalyzer:

- Generation of 100 synthetic WHO-aligned clinical scenarios across four domains:
  - WHO IMCI (Integrated Management of Childhood Illness) - 25 scenarios
  - Maternal Health protocols - 25 scenarios
  - General Medicine cases - 25 scenarios
  - Emergency Medicine protocols - 25 scenarios
- WHO guideline alignment analysis with statistical validation
- Clinical accuracy metrics calculation (precision, recall, F1-score)

ExpertEvaluationAnalyzer:

- Structured expert survey template generation
- System Usability Scale (SUS) score analysis with healthcare-specific interpretation
- Thematic analysis of qualitative expert feedback using keyword-based categorization
- Clinical safety assessment protocols

FrameworkAssessmentAnalyzer:

- NASSS Framework: Seven-domain complexity assessment with automated scoring
- RE-AIM Framework: Five-dimension implementation readiness with evidence-based thresholds
- Implementation recommendation generation based on framework scores
- Automated deployment readiness decision matrix

ATLASIntegratedAnalyzer:

- Multi-dimensional analysis integration with cross-framework correlation
- Comprehensive reporting with evidence-based conclusions
- Statistical validation and confidence interval calculation

Visualization Generation:

```
```

```
def generate_evaluation_visualizations():
    # 6-panel comprehensive dashboard:
    # 1. Technical performance vs targets
    # 2. Clinical scenario WHO alignment
    # 3. Expert evaluation SUS scores
    # 4. NASSS complexity heatmap
    # 5. RE-AIM readiness radar chart
    # 6. Integrated assessment summary
```

4.2.3 `atlas_evaluation_analysis.R` - Statistical Analysis Framework

Purpose: Advanced statistical modeling and research-grade analysis using R statistical computing environment.

Statistical Methods Implemented:

- Confidence interval calculation for clinical accuracy metrics
- Correlation analysis between technical and clinical performance indicators
- Implementation framework score correlations and predictive modeling
- Significance testing for performance improvements and clinical outcomes

Key Statistical Functions:

```
perform_clinical_accuracy_analysis()    # Binomial confidence intervals
analyze_framework_correlations()        # Multi-dimensional correlation matrices
main_r_analysis()                      # Integrated statistical modeling
```

4.3 Datasets and Data Sources

4.3.1 Synthetic Clinical Scenarios

- **Volume**: 100 structured clinical cases
- **Domains**: WHO IMCI, Maternal Health, General Medicine, Emergency Protocols
- **Structure**: Patient demographics, clinical presentation, expected WHO-aligned management
- **Validation**: Expert clinical review and WHO guideline alignment verification

4.3.2 Performance Benchmark Data

- **PWA Metrics**: Lighthouse audit scores across performance, accessibility, best practices
- **Response Times**: AI query response times across online/offline conditions
- **System Reliability**: Error rates, uptime statistics, offline functionality metrics

4.3.3 Expert Assessment Data

- **Clinical Expert Surveys**: Structured evaluations from healthcare professionals
- **System Usability Scale**: Standardized usability assessment scores
- **Implementation Assessments**: NASSS and RE-AIM framework evaluations

4.3.4 Framework Assessment Data

- **NASSS Complexity Scores**: Seven-domain implementation complexity assessment
- **RE-AIM Readiness Scores**: Five-dimension deployment readiness evaluation
- **WHO MAPS Maturity**: System maturity and scale readiness assessment

5. Dependencies and Requirements

5.1 Python Environment

Core Analysis Libraries:

```
pandas>=1.5.0                  # Data manipulation and CSV processing
numpy>=1.24.0                  # Numerical computing and array operations
scipy>=1.10.0                  # Statistical analysis and significance testing
matplotlib>=3.6.0              # Data visualization and figure generation
seaborn>=0.12.0                # Statistical visualization and styling
plotly>=5.15.0                 # Interactive visualization dashboard
scikit-learn>=1.2.0            # Machine learning and validation metrics
statsmodels>=0.14.0            # Advanced statistical modeling
```

Web Testing and Automation:

```
selenium>=4.15.0               # Browser automation for live system testing
requests>=2.31.0               # HTTP requests and API endpoint testing
beautifulsoup4>=4.12.0         # HTML parsing for web content analysis
```

Installation Protocol:

```
# Create dedicated virtual environment
python -m venv atlas_evaluation_env
atlas_evaluation_env\Scripts\activate  # Windows
# source atlas_evaluation_env/bin/activate  # Linux/MacOS

# Install core dependencies
pip install pandas numpy scipy matplotlib seaborn plotly
pip install scikit-learn statsmodels selenium requests beautifulsoup4

# Verify installation
python -c "import pandas, selenium, matplotlib; print('Environment ready')"
```

5.2 External Dependencies

Chrome Browser and ChromeDriver:

- Chrome Browser (latest stable version)

- ChromeDriver matching Chrome version
- Alternative: WebDriver Manager for automatic driver management

  ```
  pip install webdriver-manager
  ```

Lighthouse CLI (Optional but Recommended):

```
# Install Lighthouse for automated PWA auditing
npm install -g lighthouse

# Verify installation
lighthouse --version
```

5.3 R Statistical Environment

Required R Packages:

```
install.packages(c("tidyverse", "ggplot2", "plotly", "corrplot",
                   "psych", "effsize", "broom", "knitr", "rmarkdown"))
```

Statistical Analysis Capabilities:

- Advanced correlation analysis with visualization
- Confidence interval calculation for clinical metrics
- Implementation framework statistical modeling
- Research-quality report generation

6. Execution Protocol and Reproducibility

6.1 Systematic Execution Procedure

Phase 1: Environment Preparation

```
# Navigate to evaluation framework
cd evaluation

# Activate Python virtual environment
atlas_evaluation_env\Scripts\activate

# Verify dependencies
pip list | grep -E "(pandas|selenium|matplotlib)"
```

Phase 2: Live System Evaluation

```
# Execute comprehensive live testing
cd scripts
python atlas_live_testing.py

# Expected outputs:
# - live_test_results/atlas_live_test_report.json
# - live_test_results/test_summary.csv
# - lighthouse_results.json (if Lighthouse available)
```

Phase 3: Comprehensive Analysis

```
# Execute multi-framework evaluation
python atlas_data_analysis.py

# Generated outputs:
# - evaluation_results/atlas_evaluation_dashboard.png (6-panel visualization)
# - evaluation_results/atlas_evaluation_report.json (integrated analysis)
# - evaluation_results/clinical_validation_results.csv (clinical metrics)
# - evaluation_results/performance_metrics.csv (technical benchmarks)
# - evaluation_results/nasss_assessment.csv (complexity analysis)
# - evaluation_results/reaim_assessment.csv (readiness assessment)
# - test_scenarios.csv (100 synthetic clinical scenarios)
# - ai_testing_results.csv (AI performance benchmarks)
```

Phase 4: Statistical Analysis

```
# Execute advanced statistical modeling
Rscript atlas_evaluation_analysis.R

# Statistical outputs:
# - Advanced correlation matrices
# - Confidence interval calculations
# - Significance testing results
# - Research-quality statistical reports
```

6.2 Expected Research Outcomes

Technical Performance Validation:

- PWA Lighthouse scores with healthcare-specific performance thresholds
- Offline functionality reliability metrics (target: >95% availability)
- Multi-device responsiveness validation across clinical workflow scenarios

Clinical Decision Support Validation:

- WHO guideline alignment percentage across 100 synthetic scenarios (target: >75%)
- Clinical safety assessment scores with expert validation

- AI recommendation appropriateness for resource-limited settings

Implementation Readiness Assessment:

- NASSS complexity analysis with seven-domain scoring
- RE-AIM implementation readiness with evidence-based deployment recommendations
- Expert clinical assessment with System Usability Scale validation

Statistical Validation:

- Confidence intervals for all clinical accuracy metrics
- Correlation analysis between technical performance and clinical outcomes
- Significance testing for system performance improvements

6.3 Quality Assurance and Validation

Reproducibility Measures:

- All evaluation scripts are version-controlled and documented
- Synthetic datasets are generated using deterministic methods with documented seed values
- Statistical analyses include confidence intervals and significance testing
- Expert assessment protocols follow standardized healthcare evaluation frameworks

Validation Protocols:

- Clinical scenarios validated against WHO SMART Guidelines
- Technical performance benchmarks aligned with healthcare industry standards
- Expert assessment surveys follow established usability and clinical appropriateness protocols
- Implementation framework assessments use peer-reviewed evaluation criteria

7. Research Contributions and Academic Significance

7.1 Primary Contributions

Methodological Innovation:

- Novel multi-dimensional evaluation framework specifically designed for healthcare AI systems in resource-limited settings
- Integration of technical performance, clinical validation, and implementation science frameworks
- Reproducible evaluation methodology with open-source implementation

Clinical Validation Advancement:

- Systematic approach to WHO guideline alignment assessment for AI clinical decision support
- Evidence-based clinical scenario generation with expert validation protocols
- Resource-awareness evaluation criteria for healthcare AI recommendations

Implementation Science Application:

- First application of NASSS and RE-AIM frameworks to AI-powered clinical decision support systems
- Evidence-based deployment readiness assessment methodology
- Integration of technical performance with implementation complexity analysis

7.2 Academic and Practical Impact

Academic Contributions:

- Reproducible research framework for healthcare AI evaluation
- Evidence-based methodology for clinical decision support system assessment
- Integration of technical and clinical evaluation perspectives

Practical Healthcare Impact:

- Evaluation framework applicable to diverse healthcare AI systems
- Evidence-based approach to deployment readiness assessment
- Quality assurance methodology for resource-limited healthcare settings

Open Science Impact:

- Complete evaluation framework available as open-source software
- Reproducible research methodology with detailed documentation
- Community-accessible evaluation protocols and synthetic datasets

8. Conclusion and Future Directions

The ATLAS evaluation framework provides a comprehensive, evidence-based methodology for assessing AI-powered clinical decision support systems in resource-limited healthcare environments. The framework's integration of technical performance validation, clinical appropriateness assessment, expert evaluation, and implementation science frameworks offers a rigorous approach to healthcare AI system evaluation that bridges the gap between technical capability and real-world deployment readiness.

Key Academic Contributions:

1. **Reproducible Multi-dimensional Evaluation**: Systematic methodology combining technical, clinical, and implementation perspectives
2. **Healthcare-Specific Validation Protocols**: WHO-aligned clinical scenario generation and expert assessment frameworks
3. **Implementation Science Integration**: Novel application of NASSS and RE-AIM frameworks to healthcare AI systems
4. **Open Science Methodology**: Complete framework implementation available for community use and validation

Future Research Directions:

- Extension to additional clinical domains and healthcare settings
- Integration with real-world clinical workflow evaluation
- Longitudinal deployment impact assessment methodology
- Cross-cultural adaptation and validation protocols

The evaluation framework demonstrates that rigorous, multi-dimensional assessment of healthcare AI systems is both feasible and essential for evidence-based deployment decisions in resource-limited healthcare settings.

References and Resources

Complete System Implementation: https://github.com/lambdabypi/ATLAS

Live System Demonstration: https://atlas-clinical-git-main-lambdabypis-projects.vercel.app/

WHO SMART Guidelines: https://www.who.int/teams/digital-health-and-innovation/smart-guidelines

Implementation Science Frameworks:

- NASSS Framework: Non-adoption, Abandonment, Scale-up, Spread, Sustainability
- RE-AIM Framework: Reach, Effectiveness, Adoption, Implementation, Maintenance

Technical Standards:

- Progressive Web App Standards: https://web.dev/progressive-web-apps/
- Web Content Accessibility Guidelines (WCAG): https://www.w3.org/WAI/WCAG21/Understanding/

Submitted for: IE7945 - Data Analytics Engineering Capstone
Institution: Northeastern University, Master of Science Program
Submission Date: December 5, 2025
Academic Supervisor: Dr. Sivarit Sultornsanee