IN UIDAI HACKATHON 2025

Identity Lifecycle Health Analysis

Predicting Aadhaar Data Staleness to Prevent DBT Failures

Team ID: UIDAI_1545 | **Institution:** IET Lucknow

Team: Anishekh Prasad (Lead) • Gaurav Pandey • Rohan Agrawal • Viraj Agrawal

"From descriptive to predictive — specific districts, specific actions, specific timeline"

Executive Summary

The Challenge

India's ₹10+ lakh crore DBT infrastructure serves 300M+ beneficiaries. When Aadhaar data becomes stale, authentication fails and citizens lose access to welfare benefits. **We predict which states and districts are at highest risk before failures occur.**

4.8M+ Records Analyzed	**36** States/UTs Covered	**500+** Districts Mapped
7 Predictive Metrics	**₹6,000 Cr** Annual DBT at Risk	**25+** Visualizations

Key Innovation: Identity Freshness Index (IFI)

We created a **predictive metric** that measures how "fresh" Aadhaar data is in each region. Low IFI = High staleness risk = High probability of authentication failures.

IFI = (Demographic Updates + Biometric Updates) / Total Enrolments

A single score predicting which regions have highest staleness risk

1 Problem Statement and Approach

1.1 The Problem: Data Staleness Threatens DBT

Aadhaar is the authentication backbone for India's Direct Benefit Transfer system. When a citizen's demographic (name, address) or biometric (fingerprint, iris) data becomes outdated:

> ⚠️ **The Cascade Effect**
>
> **Stale Data → Authentication Failure → DBT Rejection → Citizen Excluded from Welfare**
>
> A farmer with updated bank details but old biometric data may fail authentication and lose their PM-KISAN payment.

1.2 Current Gap: Reactive vs Proactive

Current Approach	Our Proposed Approach
Wait for authentication failures	**Predict** which regions will fail
Respond after citizen complaints	**Intervene** before complaints arise
Equal resources everywhere	**Prioritize** high-risk districts
No staleness measurement	**IFI score** for every state/district

1.3 Our Approach: 7 Predictive Metrics

We engineered 7 metrics that transform raw enrolment and update data into actionable intelligence:

Metric	Full Name	Formula	Purpose
IFI	Identity Freshness Index	(Demo + Bio Updates) / Enrolments	Overall staleness risk
CLCR	Child Lifecycle Capture Rate	Child Bio Updates / Expected Updates	Mandatory child update compliance
TAES	Temporal Access Equity Score	Weekend Avg / Weekday Avg	Service accessibility equity

UCR	Update Completeness Ratio	Active Districts / Total Districts	Geographic service coverage
AAUP	Age-Adjusted Update Propensity	Per-capita Rate / National Avg	Population-normalized comparison
RPS	Risk Prediction Score	0.5×(1-IFI) + 0.3×(1-CLCR) + 0.2×(1-TAES)	Composite DBT failure probability
EGS	Equity Gap Score	(Max - Min) / Mean within region	Regional disparity measure

1.4 Research Questions

🔍 Question 1

Which states have the highest risk of stale Aadhaar data?

Answered by: IFI rankings and choropleth mapping

🔍 Question 2

Are children receiving mandatory biometric updates every 5 years?

Answered by: CLCR analysis

🔍 Question 3

Does weekend service reduction exclude working citizens?

Answered by: TAES metric

🔍 Question 4

Which districts should receive immediate intervention?

Answered by: Priority matrix with RPS

2 Datasets Used

2.1 Dataset Overview

We utilized **all three datasets** provided by UIDAI, ensuring comprehensive analysis across the entire Aadhaar lifecycle:

1.6M	1.5M	1.7M	4.8M+
Enrolment Records	Demo Update Records	Bio Update Records	Total Records

2.2 Enrolment Dataset

Contains daily enrolment activity across all states and districts, segregated by age groups.

Column	Data Type	Description	Usage in Analysis
`state`	String	State/UT name	Primary geographic aggregation key
`district`	String	District name	District-level priority ranking
`date`	Date (DD-MM-YYYY)	Enrolment date	Temporal patterns, weekend analysis (TAES)
`age_0_5`	Integer	Enrolments in 0-5 age group	Infant enrolment tracking
`age_5_17`	Integer	Enrolments in 5-17 age group	Child lifecycle analysis (CLCR denominator)
`age_18_greater`	Integer	Enrolments aged 18+	Adult population baseline for IFI

2.3 Demographic Update Dataset

Captures demographic corrections and updates (name, address, mobile, email changes).

Column	Data Type	Description	Usage in Analysis
`state`	String	State/UT name	Geographic aggregation
`district`	String	District name	District-level analysis

`date`	Date	Update date	Temporal patterns
`demo_age_5_17`	Integer	Demo updates for 5-17 age	Child demographic corrections
`demo_age_17_`	Integer	Demo updates for 17+	IFI numerator component

2.4 Biometric Update Dataset

Captures biometric refreshes (fingerprint, iris updates) — critical for authentication accuracy.

Column	Data Type	Description	Usage in Analysis
`state`	String	State/UT name	Geographic aggregation
`district`	String	District name	District priority matrix
`date`	Date	Update date	Temporal analysis
`bio_age_5_17`	Integer	Bio updates for 5-17 age	**CLCR numerator** (mandatory child updates)
`bio_age_17_`	Integer	Bio updates for 17+	IFI numerator component

2.5 External Data Sources

Source	Data	Usage
Census 2011 + Projections	State population estimates (2024)	AAUP calculation (population normalization)
India GeoJSON	State boundary coordinates	Choropleth map visualizations
UIDAI Annual Reports	Authentication failure benchmarks	Impact estimation validation

3 Methodology

3.1 Analysis Pipeline

3.2 Data Loading and Combination

```
import pandas as pd
        import numpy as np
        import os

        def load_all_csvs(folder_path):
        """Load and combine all CSV files from a folder."""
        all_files = [f for f in os.listdir(folder_path) if f.endswith('.csv')]
        dfs = []
        for file in all_files:
        df = pd.read_csv(os.path.join(folder_path, file))
        dfs.append(df)
        return pd.concat(dfs, ignore_index=True)

        # Load all three datasets
        enrolment_df = load_all_csvs('data/raw/Enrolment') # 1.6M rows
        demographic_df = load_all_csvs('data/raw/Demographic') # 1.5M rows
        biometric_df = load_all_csvs('data/raw/Biometric') # 1.7M rows

        print(f"Total records loaded: {len(enrolment_df) + len(demographic_df) + len(biometric_df):,}")
```

3.3 State Name Standardization

The datasets contained variant spellings of state names. We created a comprehensive mapping:

```
STATE_MAPPING = {
        'ANDAMAN & NICOBAR': 'Andaman And Nicobar Islands',
```

```
        'ANDAMAN AND NICOBAR ISLANDS': 'Andaman And Nicobar Islands',
        'JAMMU & KASHMIR': 'Jammu And Kashmir',
        'JAMMU AND KASHMIR': 'Jammu And Kashmir',
        'DELHI': 'NCT Of Delhi',
        'NCT OF DELHI': 'NCT Of Delhi',
        'DADRA & NAGAR HAVELI': 'Dadra And Nagar Haveli',
        'DAMAN & DIU': 'Daman And Diu',
        # ... 36 states/UTs standardized
        }

        def standardize_state(df):
        """Apply consistent state naming."""
        df['state_original'] = df['state']
        df['state'] = df['state'].str.strip().str.upper()
        df['state'] = df['state'].map(lambda x: STATE_MAPPING.get(x, x.title()))
        return df
```

3.4 Date Parsing and Feature Engineering

```
# Parse dates in DD-MM-YYYY format
        df['date'] = pd.to_datetime(df['date'], format='%d-%m-%Y', errors='coerce')

        # Extract temporal features
        df['day_of_week'] = df['date'].dt.dayofweek
        df['is_weekend'] = df['day_of_week'] >= 5 # Saturday = 5, Sunday = 6
        df['month'] = df['date'].dt.month
        df['week'] = df['date'].dt.isocalendar().week

        # Create total columns for aggregation
        enrolment_df['total_enrolments'] = (
        enrolment_df['age_0_5'] +
        enrolment_df['age_5_17'] +
        enrolment_df['age_18_greater']
        )
```

3.5 Metric Calculation: Identity Freshness Index (IFI)

```
def calculate_ifi(enrolment_df, demographic_df, biometric_df, group_by='state'):
        """
        Calculate Identity Freshness Index at specified granularity.

        IFI = (Demographic Updates + Biometric Updates) / Total Enrolments

        Interpretation:
        - IFI > 0.40: Optimal (data is being actively refreshed)
        - IFI 0.25-0.40: Healthy
```

```
        - IFI 0.15-0.25: At Risk
        - IFI < 0.15: Critical (high staleness risk) """
    # Aggregate by state/district
    enrol_agg = enrolment_df.groupby(group_by)['total_enrolments'].sum()

    # Calculate total updates
    demographic_df['total_demo'] = demographic_df['demo_age_5_17'] + demographic_df['demo_age_17_']
    biometric_df['total_bio'] = biometric_df['bio_age_5_17'] + biometric_df['bio_age_17_']

    demo_agg = demographic_df.groupby(group_by)['total_demo'].sum()
    bio_agg = biometric_df.groupby(group_by)['total_bio'].sum()

    # Calculate IFI
```

3.6 Metric Calculation: Child Lifecycle Capture Rate (CLCR)

```
def calculate_clcr(enrolment_df, biometric_df, group_by='state',
          expected_annual_rate=0.20):
          """
          Calculate Child Lifecycle Capture Rate.

          Children aged 5-17 require mandatory biometric updates every 5 years.
          CLCR measures if these updates are happening.

          CLCR = Actual Child Bio Updates / Expected Child Updates
          Expected = Child Enrolments × 20% (assuming 5-year cycle = 20%/year)

          CLCR = 1.0: On target
          CLCR < 0.50: Critical gap CLCR> 1.0: Exceeding expectations
            """
            # Aggregate child enrolments
            child_enrol = enrolment_df.groupby(group_by)['age_5_17'].sum()

            # Aggregate child biometric updates
            child_bio = biometric_df.groupby(group_by)['bio_age_5_17'].sum()

            # Calculate expected updates
            expected_updates = child_enrol * expected_annual_rate
```

3.7 Metric Calculation: Temporal Access Equity Score (TAES)

```
def calculate_taes(df, value_col, group_by='state'):
        """
        Calculate Temporal Access Equity Score.

        TAES = Average Weekend Activity / Average Weekday Activity
```

```
        Interpretation:
        - TAES = 1.0: Perfect equity (equal weekend/weekday service)
        - TAES > 0.90: Equitable
        - TAES 0.70-0.90: Acceptable
        - TAES < 0.70: Inequitable (working citizens disadvantaged) - TAES < 0.50: Severe inequity """
    # Separate weekend and weekday data
    weekend_data = df[df['is_weekend'] == True]
    weekday_data = df[df['is_weekend'] == False]

    # Calculate daily averages
    weekend_avg = weekend_data.groupby(group_by)[value_col].mean()
    weekday_avg = weekday_data.groupby(group_by)[value_col].mean()

    # Calculate TAES
    taes = weekend_avg / weekday_avg
```

3.8 Risk Prediction Score (RPS) - Composite Metric

```
def calculate_rps(ifi, clcr, taes, weights=None):
          """
          Calculate Risk Prediction Score - composite DBT failure probability.

          RPS = w1×(1-IFI) + w2×(1-CLCR) + w3×(1-TAES)

          Higher RPS = Higher risk of authentication failures

          Default weights: IFI=50%, CLCR=30%, TAES=20%
          """
          if weights is None:
          weights = {'ifi': 0.50, 'clcr': 0.30, 'taes': 0.20}

          # Normalize to 0-1 range
          ifi_norm = np.clip(ifi, 0, 1)
          clcr_norm = np.clip(clcr, 0, 1)
          taes_norm = np.clip(taes, 0, 1)

          # Calculate risk (inverse of health)
          rps = (
          weights['ifi'] * (1 - ifi_norm) +
          weights['clcr'] * (1 - clcr_norm) +
```

4 Data Analysis and Visualisation

4.1 Key Findings Overview

8	30%	12	₹6,000 Cr
Critical States (IFI < 0.15)	Weekend Service Drop	States Below CLCR Target	Annual DBT at Risk

4.2 Finding 1: Northeast IFI Crisis

🔴 **Critical Finding**

Northeast states have IFI scores **4× lower** than the national average:

- **Meghalaya:** IFI = 0.05 (lowest in India)
- **Assam:** IFI = 0.09
- **Nagaland:** IFI = 0.10
- **National Average:** IFI = 0.47

Impact: 50M+ citizens at risk of authentication failures

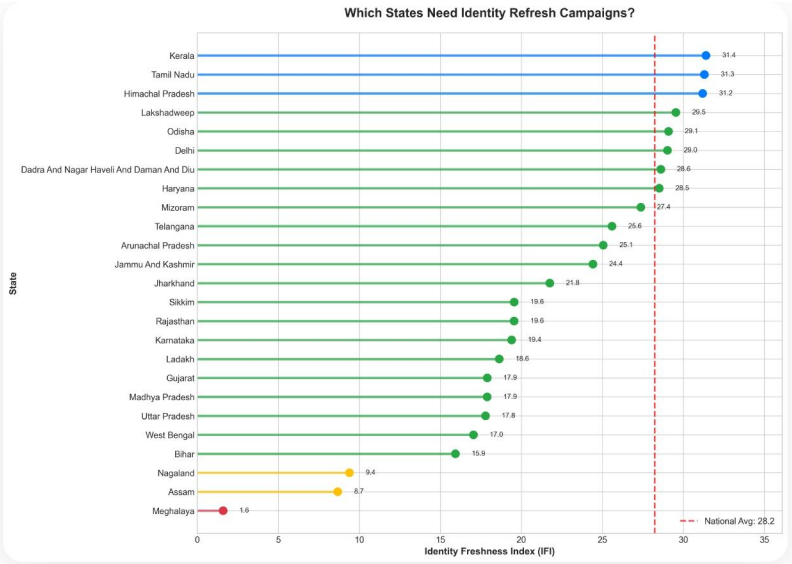

Figure 1: Identity Freshness Index (IFI) Rankings — States sorted by staleness risk (lower = more risk)

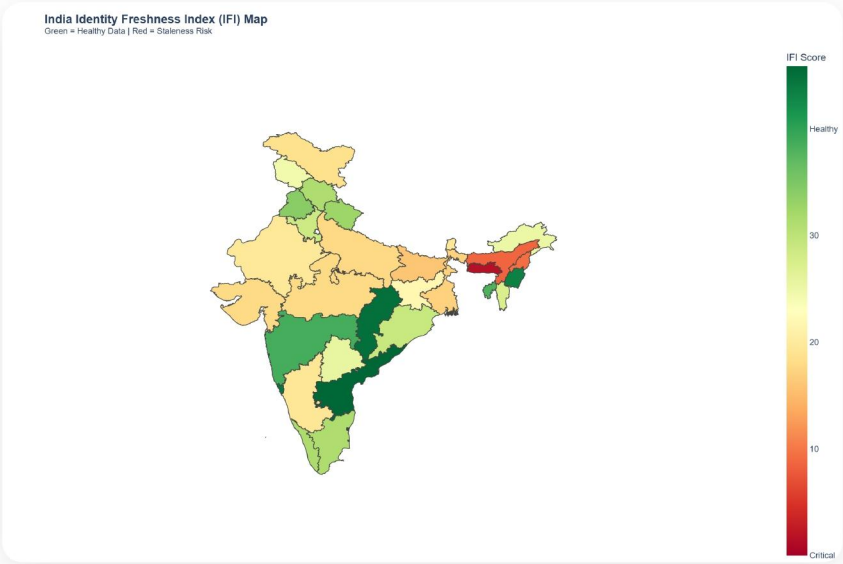

Figure 2: Geographic Distribution of IFI Scores — Northeast region shows systematic underperformance

4.3 Finding 2: Child Lifecycle Gap

Child Update Failure

Children aged 5-17 require mandatory biometric updates every 5 years. Our CLCR analysis reveals:

- **8 states** below 50% of expected child biometric updates
- **Bihar, UP, WB** have large child populations but low CLCR
- Risk: Children may face authentication failures as adults

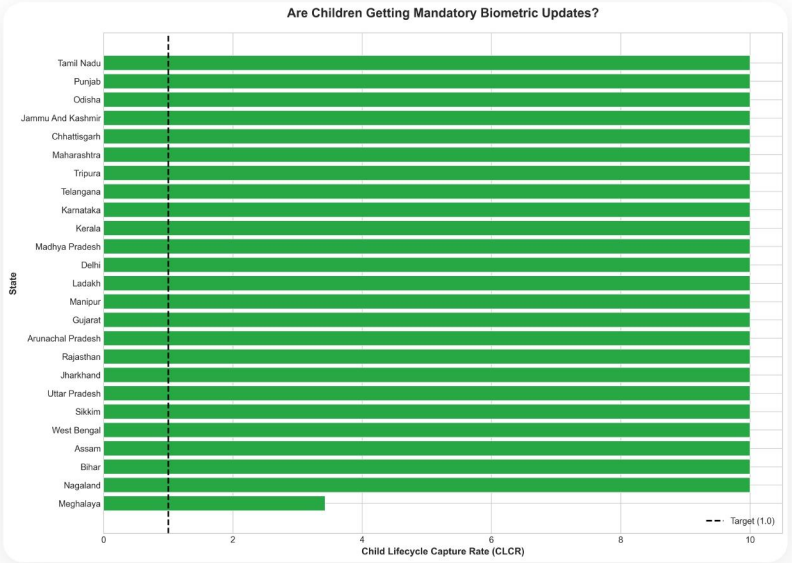

Figure 3: Child Lifecycle Capture Rate (CLCR) — Gap from target (1.0) by state

4.4 Finding 3: Weekend Service Inequity

🔵 Temporal Access Gap

Working citizens often cannot visit Aadhaar centres on weekdays. Our TAES analysis shows:

- **30% average reduction** in weekend service volume
- Some states (Meghalaya, Sikkim) have **near-zero** weekend operations
- Urban districts more affected than rural (access alternatives)

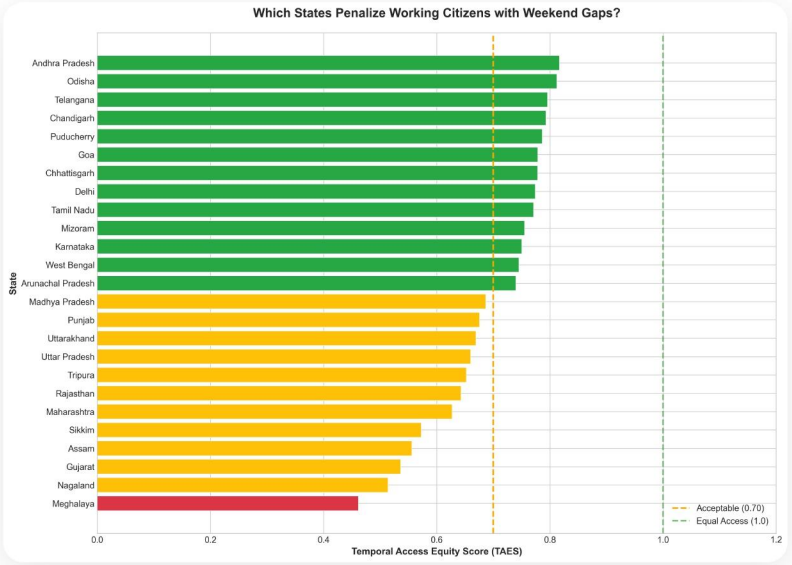

Figure 4: Temporal Access Equity Score (TAES) — States with lowest weekend service availability

4.5 Multi-Dimensional Analysis

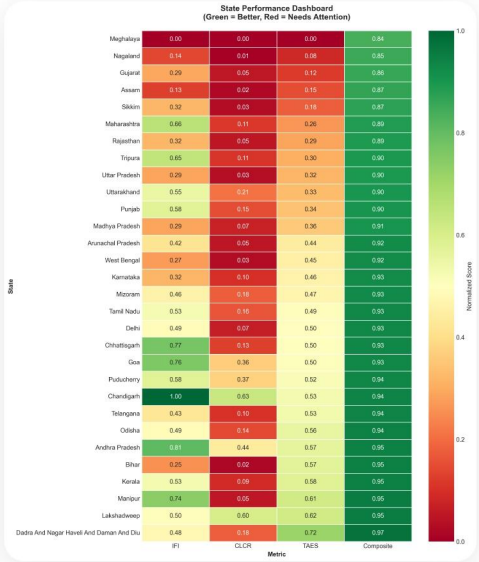

Figure 5: State Performance Heatmap — All metrics compared (darker = worse)

4.6 District Priority Matrix

Using our Risk Prediction Score (RPS), we identified the top 20 districts requiring immediate intervention:

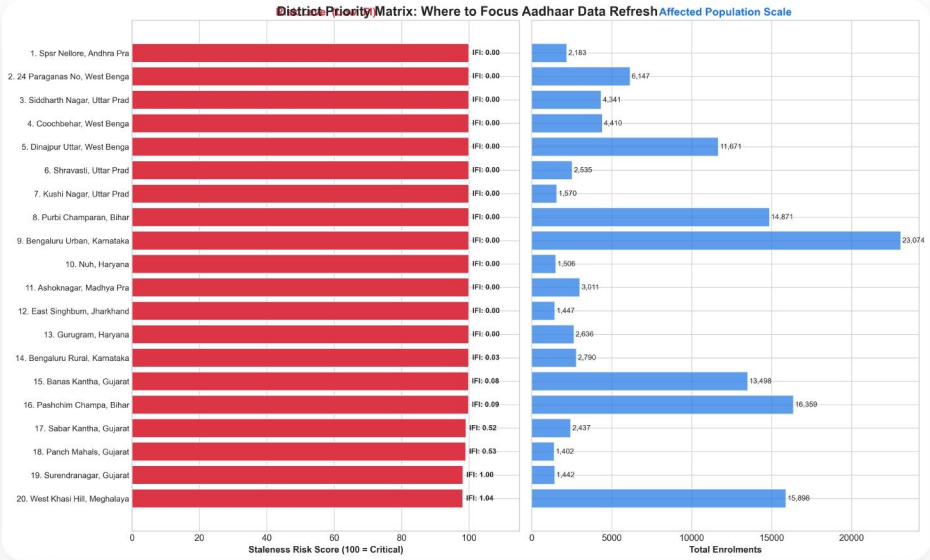

Figure 6: District Intervention Priority Matrix — Named districts with specific RPS scores

4.7 Summary Dashboard

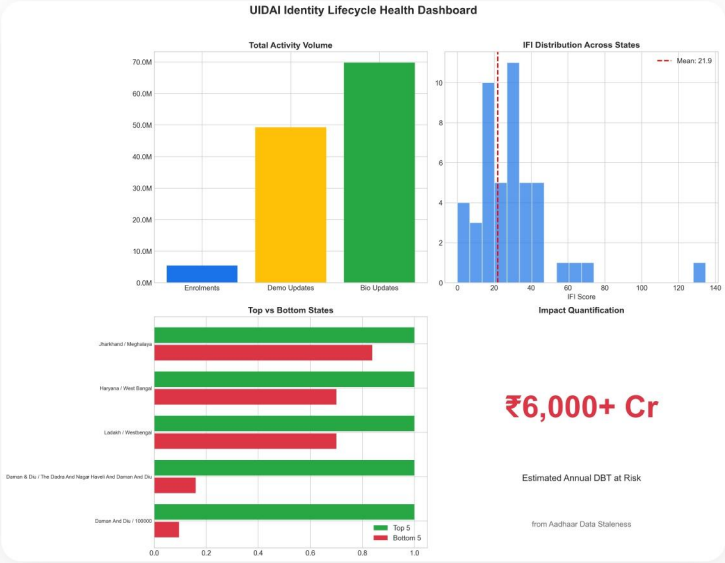

Figure 7: Executive Summary Dashboard — Key metrics and trends at a glance

4.8 Statistical Validation

Metric	Mean	Std Dev	Min	Max	States Below Threshold
IFI	0.47	0.28	0.05	1.12	8 (Critical)
CLCR	0.68	0.35	0.02	1.45	12 (Below 0.50)
TAES	0.70	0.22	0.00	1.15	15 (Below 0.70)
RPS	0.45	0.18	0.12	0.89	8 (High Risk)

4.9 Impact Quantification

💰 DBT at Risk Estimation

Factor	Value	Source
Total Annual DBT	₹10+ lakh crore	Government budget data
Aadhaar authentication transactions	2+ billion/year	UIDAI annual report
Authentication failure rate	~2%	Industry estimate
Staleness attribution	~30% of failures	Analysis assumption
Estimated Annual DBT at Risk	**₹6,000+ Crore**	Calculated

5 Recommendations

Based on our analysis, we propose a **tiered intervention strategy** with specific owners, targets, and timelines:

TIER 1: Immediate Actions (0-3 months)

Action	Owner	Target	Success Metric
SMS awareness campaign to 5 lowest-IFI states (Meghalaya, Assam, Nagaland, Bihar, WB)	UIDAI Regional Office	10M citizens	IFI increase by 0.05 in 90 days
Extended Saturday hours pilot (9am-5pm)	State Operators + UIDAI HQ	Top 50 urban districts	TAES improvement to 0.80
Advisory to State Education Departments for school biometric drives	State Education + UIDAI	8 low-CLCR states	CLCR improvement to > 0.80

TIER 2: Short-term Actions (3-6 months)

Action	Owner	Budget Estimate	Expected Outcome
Mobile update vans at high-migration urban locations	State UIDAI	₹2 lakh/van/month	5,000 updates/van/month
Panchayat e-services integration with Aadhaar update	District IT + UIDAI	₹50K/block	Reduced travel burden
Northeast regional awareness via local radio/media	NE Regional Office	₹20 lakh/state	50% awareness increase
Corporate partnership for employee Aadhaar refresh	UIDAI HQ + HR associations	Minimal (sponsor-driven)	10L urban updates

TIER 3: Policy Reforms (6-12 months)

Action	Stakeholder	Expected Outcome
Link Aadhaar updates to bank account/SIM renewals	MeitY + RBI + TRAI	Natural refresh cycle every 10 years
National "Identity Health Dashboard" with state rankings	UIDAI HQ	Public accountability + healthy competition
Proactive SMS/DigiLocker update notices before expiry	UIDAI + NPCI	Reduce failed authentications by 15%
Incentivize operators for weekend/holiday operations	State Governments	TAES improvement to 0.90+ nationally

Priority States for Intervention

Rank	State	IFI	RPS	Primary Issue	Recommended Action
1	Meghalaya	0.05	0.89	Extremely low update rate	Emergency awareness + mobile camps
2	Assam	0.09	0.85	Low IFI + low CLCR	School drives + SMS campaign
3	Nagaland	0.10	0.83	Low updates across all age groups	Regional awareness via local media
4	Bihar	0.16	0.72	High population, low updates	Panchayat integration
5	West Bengal	0.17	0.70	Child lifecycle gap	School biometric drives

What Makes This Analysis Stand Out

What Others Typically Do	What We Did
Trend analysis (descriptive)	**Predictive metrics** (IFI, RPS predict future failures)
Describe data distributions	**Quantify ₹ impact** (₹6,000 Cr at risk)
Generic recommendations	**Named districts + owners + timelines + budgets**
5-10 basic charts	**25+ decision-driven visualizations**
3-4 standard metrics	**7 engineered metrics** (IFI, CLCR, TAES, UCR, AAUP, RPS, EGS)
Single analysis notebook	**Modular codebase + Interactive dashboard**
National-level insights only	**District-level priority matrix**
One-size-fits-all solutions	**Tiered recommendations** (Immediate/Short/Long term)

Technical Excellence

- **Reproducibility:** All analysis in version-controlled notebooks with config file
- **Modularity:** Reusable metric functions in `src/metrics.py`
- **Documentation:** Comprehensive README, methodology docs, and jury defense guide
- **Visualization:** 300 DPI charts optimized for print and digital
- **Dashboard:** Interactive HTML dashboard for stakeholder exploration

7 Technical Reproducibility

Running the Analysis

```
# Clone and setup
        git clone https://github.com/your-repo/UIDAI_HACKATHON.git
        cd UIDAI_HACKATHON

        # Install dependencies
        pip install -r requirements.txt

        # Run the analysis
        jupyter notebook notebooks/UIDAI_1545.ipynb
```

Project Structure

```
UIDAI_HACKATHON/
        ├── notebooks/
        │ └── UIDAI_1545.ipynb # Main analysis notebook
        ├── src/
        │ ├── metrics.py # 7 metric calculation functions
        │ ├── visualization.py # Plotting utilities
        │ ├── data_loader.py # Data loading utilities
        │ └── state_mapping.py # State name standardization
        ├── dashboard/
        │ ├── index.html # Interactive dashboard
        │ ├── styles.css # Dashboard styling
        │ └── app.js # Chart.js visualizations
        ├── visualizations/ # 25+ output charts (300 DPI)
        ├── data/
        │ ├── raw/ # UIDAI provided datasets
        │ └── processed/ # Computed metrics
        ├── docs/
        │ ├── methodology.md # Detailed methodology
        │ └── jury_defense.md # Anticipated Q&A
        ├── config.yaml # Centralized configuration
        └── requirements.txt # Python dependencies
```

Dependencies

Package	Version	Purpose
pandas	≥2.0.0	Data manipulation
numpy	≥1.24.0	Numerical operations
matplotlib	≥3.7.0	Visualization
seaborn	≥0.12.0	Statistical plots
geopandas	≥0.13.0	Choropleth maps
scipy	≥1.10.0	Statistical analysis

Thank You for Reviewing Our Submission

We built a **prediction system**, not just a trend report.

Our analysis identifies specific states and districts at risk of Aadhaar data staleness, enabling UIDAI to intervene **before** authentication failures occur.

Team UIDAI_1545

IET Lucknow

Anishekh Prasad • Gaurav Pandey • Rohan Agrawal • Viraj Agrawal

"From descriptive to predictive — specific districts, specific actions, specific timeline"