



ORIGINAL CONTRIBUTION

Uncovering Irregularities in National Identity Frameworks: An Algorithmic Approach to Anomaly Detection and LLM Based Predictive Auditing

¹Sounak Sarkar, ²Soumyashis Ghosh, ³Sourodeep Kar, ⁴Dr. Tirthadip Sinha

^{1,2}UG student, Dept. of Computer Science & Engineering, Haldia Institute of Technology, Haldia, West Bengal

³UG student, Dept. of Electronics & Communication Engineering, Haldia Institute of Technology, Haldia, West Bengal

⁴Assistant Professor, Dept. of Electronics & Communication Engineering, Haldia Institute of Technology, Haldia, West Bengal

ABSTRACT

National biometric platforms like India's Aadhaar generate massive daily transaction volumes, making traditional reporting too slow to catch operational anomalies or sudden regional demand surges. To shift from reactive auditing to intelligent real-time monitoring, we introduce a predictive analytics engine using real-world UIDAI operational data (March–December 2025). Built on a decoupled React and FastAPI architecture, the platform features an automated time-series forecasting tournament. Using rolling-origin cross-validation with split-conformal prediction intervals, it delivers reliable capacity forecasts with precise statistical bounds. Concurrently, unsupervised Isolation Forests flag localized anomalies across state and district levels without requiring pre-labeled fraud data. A local LLM translates these technical insights into actionable, plain-language briefs backed by a human-in-the-loop audit trail—enabling swift, targeted field responses at national scale.

KEYWORDS: Isolation Forest, Split-Conformal Prediction, Unsupervised Anomaly Detection, Time-Series Forecasting, LLM, Predictive Auditing, Geospatial Intelligence, Data Governance.

1. INTRODUCTION

Managing a national biometric system involves complex daily hurdles in securing data, distributing resources, and mitigating localized risks. The Unique Identification Authority of India (UIDAI) oversees Aadhaar, the world's largest biometric ID system, which processes millions of daily transactions including new enrolments, biometric updates, and demographic modifications. As the sheer volume and velocity of this data grow, traditional static reporting dashboards become insufficient and inefficient. There is a clear need for an intelligent, scalable system that can surface operational anomalies, predict future workloads with quantified uncertainty, and support transparent data governance.

This paper introduces the Aadhaar Intelligence Engine (AIE), an AI-integrated end-to-end analytical platform built for real-time monitoring and predictive decision-making to address these operational challenges. The system uses a modern, decoupled architecture with a fast frontend and a high-throughput backend. Far

beyond a simple visualization dashboard, the backend serves as a dedicated statistical and machine learning engine for the identity infrastructure.

The primary dataset driving this engine was collected from UIDAI, capturing daily operational volumes for new enrolments, demographic updates, and biometric updates across Indian states and districts. The dataset spans the operational period from March 2, 2025 to December 31, 2025. We applied strict data cleaning and normalization pipelines to ensure our statistics were valid. This included the deduplication of anomalous records, normalization of district aliases, reallocation of legacy state jurisdictions, and the quarantining of records with invalid dates or corrupted pin-codes. The system leverages the following primary tech stack and methodologies.

- The backend API is served via FastAPI.
- Data manipulation and numerical operations are handled using Pandas and NumPy.

- Time-series forecasting and anomaly detection models are implemented using Scikit-learn.
- Natural language insight generation is powered by Ollama, serving the Qwen2.5 Large Language Model locally.
- The user interface is built with React and bundled using Vite.
- Data visualization is achieved through Recharts, while complex 2D and 3D geospatial mappings are rendered using Deck.GL layered over a MapLibre base map.

2. LITERATURE REVIEW

The analytical methodologies and infrastructure choices employed in the AIE build directly upon foundational research across machine learning, time-series forecasting, data storage, and large language models.

- **Unsupervised Anomaly Detection:** To address the challenge of flagging operational deviations without pre-labeled fraud datasets, the system relies on Isolation Forest [1]. Unlike conventional distance or density-based anomaly detection algorithms that compute normal regions, Liu et al. (2008) introduced a tree-structured approach that explicitly isolates anomalies by randomly partitioning feature spaces. Because anomalies are sparse and distinct, they require significantly fewer splits to isolate, resulting in shorter path lengths in the trees. This method provides low computational complexity $O(n)$ and linear memory requirements, making it suitable for national identity transaction logs.
- **Time-Series Forecasting & Uncertainty Quantification:** Evaluating model performance across geographically diverse regions requires a metric robust to varying data scales. Hyndman and Koehler (2006) introduced Mean Absolute Scaled Error (MASE) to overcome the numerical instabilities of percentage-based metrics (such as MAPE) when encountering values close to zero. By scaling the forecast errors against the in-sample mean absolute error of a naive benchmark, MASE provides a scale-free comparison across diverse datasets [2]. To handle model diversity, the system employs stacked generalization principles [3]. Wolpert's ensemble framework demonstrates that combining distinct learning algorithms via a meta-model minimizes generalization bias and yields superior predictive performance compared to individual base models. Furthermore, traditional point forecasts fail to convey risk without valid confidence limits. Xu and Xie (2023) developed sequential predictive conformal prediction frameworks tailored for non-exchangeable time-series data. By adaptively estimating quantiles over non-conformity scores, split-conformal methods yield distribution-free, statistically guaranteed uncertainty bounds without relying on Gaussian residual assumptions [4].
- **Columnar Data Analytics and Local LLM Inference:** For high-speed real-time querying, the platform draws on the columnar data storage paradigm originally formalized by Melnik et al. (2010) in the Dremel system (which forms the basis for Apache Parquet). Storing transactional records in a columnar format significantly reduces I/O footprint and speeds up aggregations over large datasets [5]. Finally, generating qualitative briefs while ensuring data privacy is achieved through local large language model deployment. Bai et al. (2023) presented the Qwen technical architecture, establishing high-capability open-weight language models optimized for reasoning and structured text generation. Local inference avoids transmitting sensitive operational data to external APIs while converting deterministic analytical outputs into narrative intelligence [6].

3. DESIGN METHODOLOGY

The AIE is built on a modular architecture, pairing a fast Python backend with a responsive, hardware-accelerated frontend. The methodology driving the system is divided into four primary pipelines: Data Pre-processing, Time-Series Forecasting, Unsupervised Anomaly Detection, and LLM-Driven Governance.

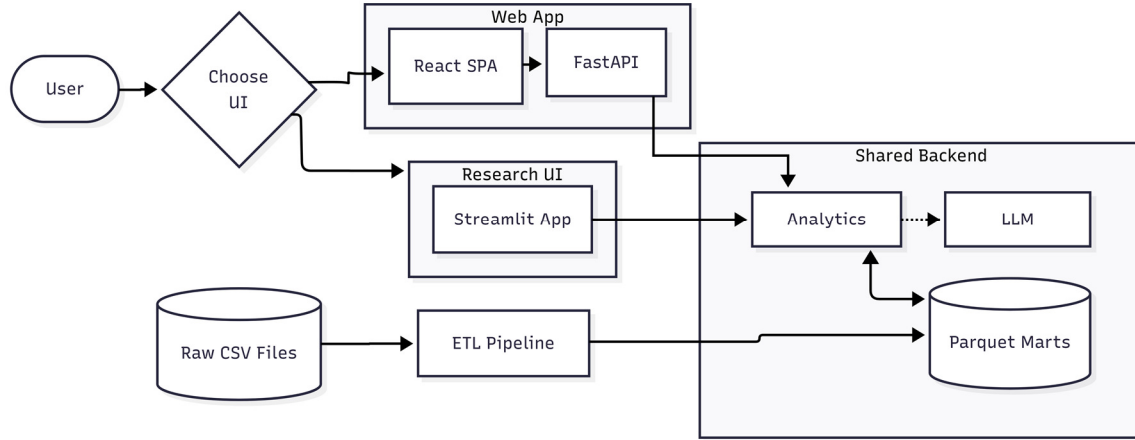


Figure 1: High-level system architecture illustrating the integration of dual user interfaces with the shared analytical engine.

To handle high-velocity data, the system utilizes a strict Extract, Transform, Load (ETL) pipeline resulting in highly optimized Parquet data marts, leveraging high-performance columnar storage principles [5]. Raw CSV shards representing enrolments, demographic updates, and biometric updates are scanned and parsed. Records are grouped by a natural composite key (Date, State, District, Pin-code), and metric duplicates are collapsed via summation to prevent double-counting. The system automatically fixes geographic data errors by applying a strict set of cleaning rules:

- It repairs records where districts were mistakenly entered as states.
- Normalizes thousands of local district aliases to official UIDAI nomenclature.
- Reallocates legacy state borders (e.g., Andhra Pradesh to Telangana splits).

Cleaned DataFrames are persisted as columnar Parquet files (fact_enrol_daily, dim_geo, etc.), allowing the analytics engine to execute in-memory aggregations in milliseconds without re-parsing CSVs.

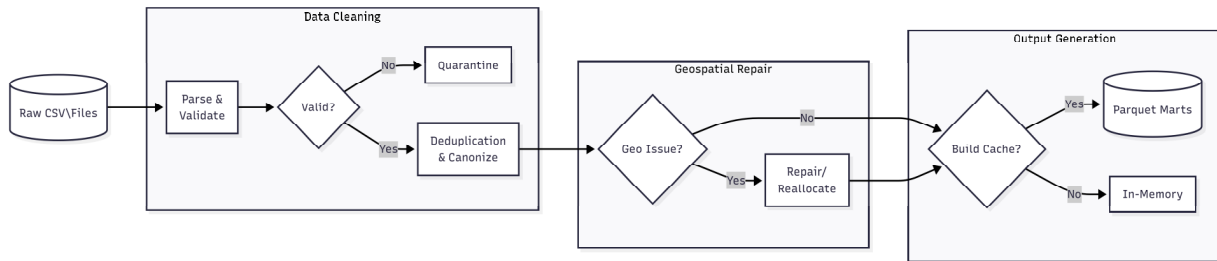


Figure 2: The automated data processing pipeline that cleans, standardizes, and converts raw CSV data into optimized storage.

Risk management relies on proactive, unsupervised machine learning to surface multidimensional outliers. To keep the analysis consistent, all calculations are performed at the state–district level. The engine calculates derived features for each district, including volume volatility (coefficient of variation), demographic-to-enrolment ratios, biometric stress, and week-over-week growth. These features are scaled and fed into an Isolation Forest [1] ensemble. The model flags multidimensional anomalies without needing labeled fraud data, assigning each flagged district a normalized risk score from 10 to 100. The engine aggregates these district-level

flags to generate a "State Risk Radar," prioritizing states by their total anomalous volume.

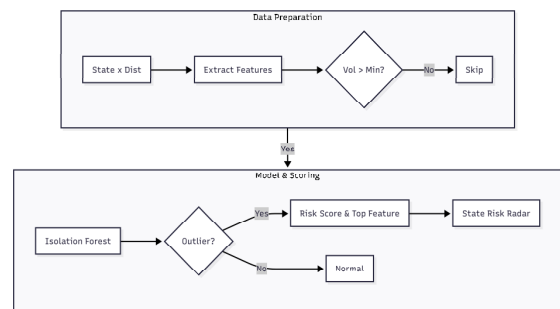


Figure 3: Two-stage anomaly detection process identifying regional outliers without relying on pre-labeled data.

The system includes a data governance service that requires human review while maintaining a continuous record of changes. Rather than mutating the source Parquet or CSV files, administrators review data discrepancies in the UI. Actions such as "Merge" or "Delete" are appended to a JSON patch store and an immutable CSV audit log. These patches are dynamically applied at runtime, ensuring data lineage remains transparent and reproducible.

Capacity planning is driven by a rigorous, rolling-origin cross-validated model tournament. The engine evaluates a diverse registry of models:

Moving Averages, damped Drift models, intelligent Ensembles based on stacked generalization concepts [3], and Seasonal Naive baselines (accounting for Indian holidays). Models are back tested using rolling-origin cross-validation. The winning model is selected based on the lowest Mean Absolute Scaled Error (MASE) [2], provided it strictly beats the naive baselines. Instead of assuming Gaussian residuals, the engine applies split-conformal prediction to generate statistically guaranteed uncertainty bounds (upper and lower prediction intervals), classifying them into operational bands (Tight, Directional, or Exploratory).

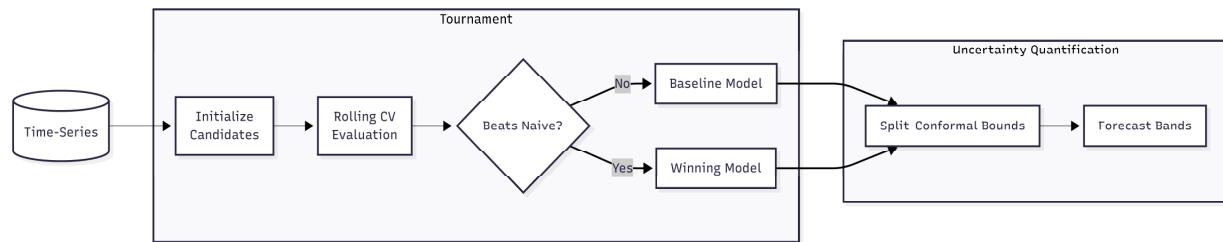


Figure 4: Automated forecasting tournament that pits multiple models against baselines to ensure reliable capacity predictions.

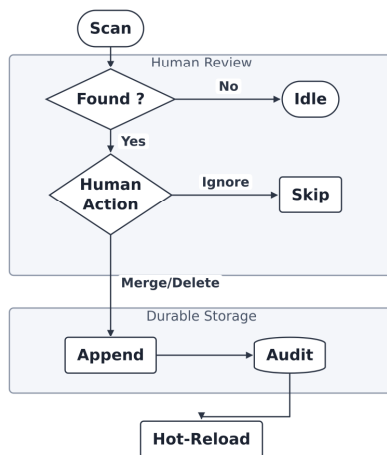


Figure 5: Human-in-the-loop data governance workflow for auditing, resolving, and durably patching operational dataset discrepancies.

To translate complex statistical outputs into actionable intelligence, the engine integrates a local LLM [6]. The Python engine deterministically compiles "evidence" such as winning forecast models, specific anomaly ratios, total volume, etc. The LLM is strictly prompted to process this evidence and generate a structured markdown brief, ensuring it cannot invent or alter underlying mathematical metrics.

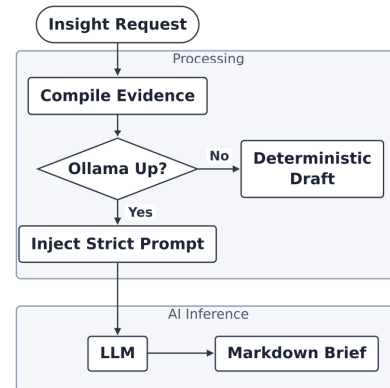


Figure 6: LLM integration framework for transforming quantitative engine outputs into qualitative research briefs.

4. OUTCOMES

A core accomplishment of this work lies in converting massive daily transactional CSV files (Enrolment, Biometric, and Demographic) into an optimized, unified format capable of real-time querying. The automated Extract, Transform, Load (ETL) pipeline reliably handled deduplication, geographic repairs, and Parquet aggregation in sequence with high-speed data processing.

Table 1: Performance of the different processing stages in automated data pipeline, highlighting significant data compression and geographic corrections.

Processing Stage	Action / Metric	Volume (Rows)	Operational Impact
Ingestion	Raw transactions processed	4,938,837	Established the total operational volume.
Deduplication	Natural keys collapsed	738,146	Prevented statistical inflation by summing overlapping metric keys.
Geospatial Repair	District aliases normalized	205,657	Standardized thousands of inconsistent spellings.
Geospatial Repair	Boundary reallocations (AP→TS)	92,491	Corrected legacy state assignments.
Mart Generation	Final aggregated Parquet rows	218,287	Compressed 4.2M rows into a lightweight matrix.

Nearly 300,000 geographic errors were algorithmically corrected upon ingest, ensuring that regional aggregations and geospatial maps remain perfectly accurate for administrators without requiring tedious manual data-entry corrections. Compressing 4.9 million transactions into 218k daily Parquet rows allows the FastAPI backend to load all the necessary analytics data directly into memory. This architectural decision guarantees that all filtering and chart rendering on the React Web App executes with sub-second latency.

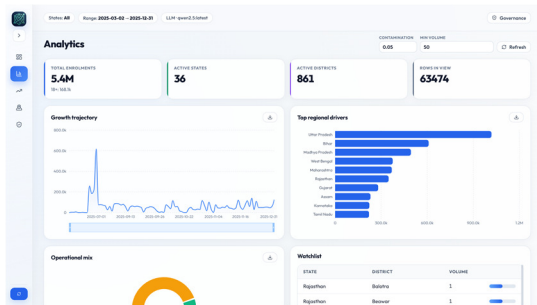


Figure 7: React Web App Dashboard displaying clean, unified State and District demographic data.

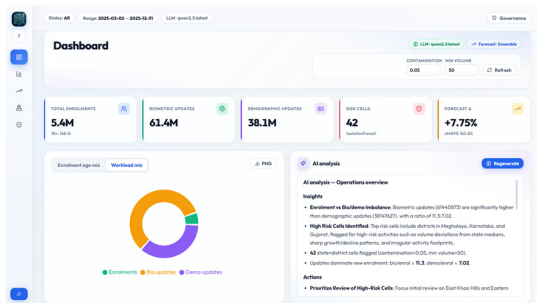


Figure 8: Analytics tab demonstrating sub-second rendering of over 218,000 aggregated Parquet rows.

Rather than relying on rigid, rule-based alerts, the engine detects complex and hidden nationwide anomalies using unsupervised learning specifically the Isolation Forest algorithm. The following radar output aggregates the highest-risk districts, prioritizing state-level interventions based on flagged transaction volume. Karnataka and West Bengal emerged as the highest priority jurisdictions, jointly accounting for over 120,000 anomalous transactions across 9 combined districts.

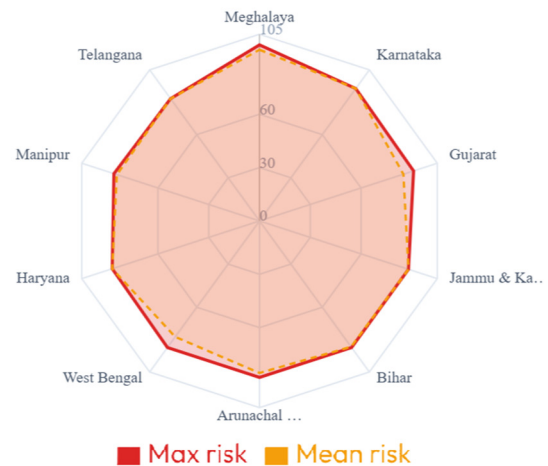


Figure 9: Polar Risk Radar charting the top jurisdictions requiring immediate operational audits.

Table 2: The top 5 highest-risk states automatically flagged by the AIE for immediate operational audits.

State	Flagged Districts	Flagged Volume	Max Risk Score	Mean Risk Score	Primary Anomaly Driver
Karnataka	2	61,274	96.0	87.5	Extreme volume volatility
West Bengal	7	60,139	82.0	77.6	High demographic-to-enrolment ratio
Bihar	2	31,230	84.0	80.5	Extreme volume volatility

State	Flagged Districts	Flagged Volume	Max Risk Score	Mean Risk Score	Primary Anomaly Driver
Telangana	1	30,830	88.0	88.0	High biometric update stress
Meghalaya	3	29,773	99.0	89.3	Volume outlier vs state median

Effective operational planning requires highly reliable capacity forecasting method like Time-Series Forecasting. Instead of relying on a single fixed algorithm, the forecasting backend runs a rolling-origin cross-validation comparison across

several statistical models. Models are ranked by Mean Absolute Scaled Error (MASE). More complex models must outperform simple baselines such as Seasonal Naive or Moving Average on the data being examined.

Table 3: Ensemble model outperforms single-architecture baselines.

Rank	Model Architecture	MASE	MAPE (%)	RMSE	Decision Band
1	Ensemble (Median Stack)	0.9010	188.6	34,916	Directional
2	Moving Average	0.9863	182.4	36,185	Directional
3	Drift	0.9945	200.0	36,542	Directional
4	Seasonal Naive (Baseline)	1.0383	190.0	40,658	Directional

The automated Ensemble model utilizing median stacking strategies consistently outperformed all single-architecture baselines, reducing the scaled error (MASE) below 1.0, indicating it actively learned the volume patterns rather than just echoing recent history. Split-conformal prediction intervals applied to the residuals of the winning model give a statistically valid 90 % confidence band around the forecasts. Using the upper prediction limits, administrators can safely allocate servers, update kits, and staff for upcoming traffic spikes without over-allocating

the budget. While the Ensemble model dominated at the national aggregate level, the tournament proved crucial at the micro-level. For example, when predicting capacity for Bihar (characterized by extreme volume volatility) and Sikkim (a low-volume, sparser dataset), the system automatically discarded the complex ensemble in favor of a robust drift model. This demonstrates the engine's ability to guard against overfitting by dynamically reverting to simpler linear trendlines when they mathematically out-score complex architectures in noisy regions.

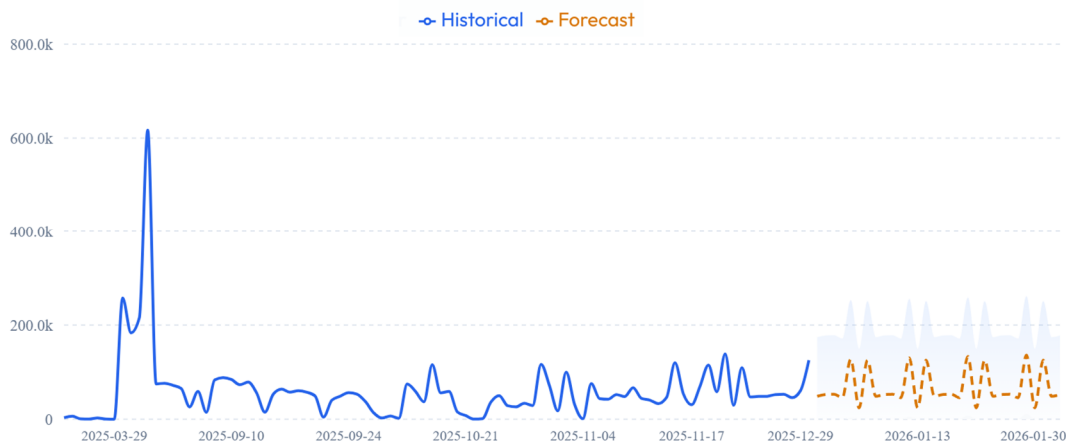


Figure 10: Time-series forecasting shows the Ensemble model's trajectory and 90% confidence bounds.

Deploying the model on-premise safeguards sensitive operational data from third-party exposure, aligning strictly with national security standards. Constraining the LLM to pre-computed JSON payloads eliminates numerical

hallucinations in the generated summaries from the final report. Operators interacting with the React Web App instantly receive human-readable, actionable summaries of a state's health

and accelerating response times without requiring dedicated data science expertise.

Table 4: Complex mathematical payloads are shifted into actionable operator directives in processing stages.

Processing Stage	Output Example (Karnataka Enrolments)	Operational Impact
Statistical Engine Output (Deterministic)	{"anomaly": true, "vol": 61274, "risk": 96.0, "driver": "volatility"}	Calculates the exact statistical metrics.
LLM Output Brief (Qualitative)	"Karnataka has been flagged for a critical anomaly (Risk Score: 96.0). Our engine detected extreme volume volatility resulting in 61,274 flagged transactions. An immediate regional audit is recommended."	Translates complex statistics into clear, actionable instructions for administrators.

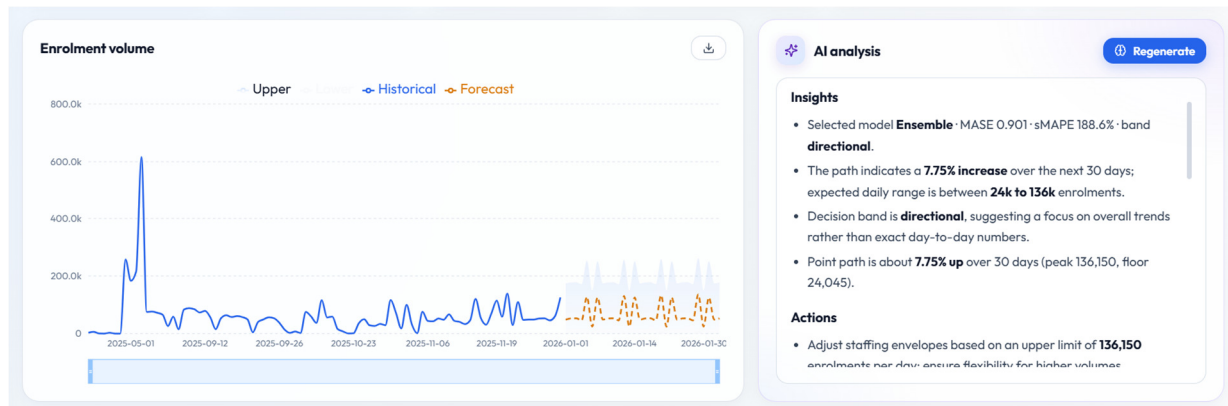


Figure 11: LLM-generated operational brief providing zero-shot narrative insights directly on the dashboard.

5. CONCLUSION

This work marks a significant step forward in national identity management, shifting the paradigm from slow, reactive spreadsheet reporting to an agile, automated, and AI-driven intelligence platform. By seamlessly uniting automated data engineering, unsupervised anomaly detection, and dynamic forecasting, the Aadhaar Intelligence Engine empowers administrators to proactively monitor millions of daily transactions with sub-second responsiveness. Crucially, by running Large Language Models locally on the ground, the system safeguards strict data privacy while instantly translating complex statistical patterns into clear, plain-language directives for everyday operators.

While the current platform demonstrates high operational efficiency, we recognize several realistic boundaries. First, because models currently process data as daily totals, brief intra-day or hourly capacity spikes and bottlenecks can still go undetected. Secondly, the forecasting algorithms rely primarily on historical transaction volumes. As a result, unexpected external events—such as sudden local holidays, severe

weather disruptions, or regional internet outages—are not yet factored into workload predictions. Finally, while local LLM deployment ensures absolute data privacy, running these models locally introduces a hardware bottleneck on dashboard performance in environments lacking dedicated GPU hardware. To build a more resilient system, future iterations will focus on our three key enhancements. We plan to integrate external dataset streams such as local infrastructure health indices, census demographics, and regional event calendars, to make workload forecasts significantly more adaptive. Extending geospatial analytics down to the pin-code level will allow administrators to pinpoint under-served areas and deploy physical assets, such as mobile enrolment vans, directly where demand is highest.

Also, transitioning the architecture to a streaming framework (e.g., Apache Kafka) will enable the system to catch transactional irregularities the exact moment they occur.

Acknowledgement

Due to applicable national privacy, security, and data-protection regulations, public-sharing of the raw data is restricted in India. The authors

gratefully acknowledge the Unique Identification Authority of India (UIDAI) for providing the real-world data used in this work.

REFERENCES

- [1] Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation Forest. *2008 Eighth IEEE International Conference on Data Mining*, 413–422. <https://doi.org/10.1109/ICDM.2008.17>
- [2] Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4), 679–688. <https://doi.org/10.1016/j.ijforecast.2006.03.001>
- [3] Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, 5(2), 241–259. [https://doi.org/10.1016/S0893-6080\(05\)80023-1](https://doi.org/10.1016/S0893-6080(05)80023-1)
- [4] Xu, C., & Xie, Y. (2023). Sequential Predictive Conformal Inference for Time Series. *Proceedings of the 40th International Conference on Machine Learning*, 38707–38727. <https://doi.org/10.48550/arXiv.2212.03463>
- [5] Melnik, S., Gubarev, A., Long, J. J., Romer, G., Shivakumar, S., Tolton, M., & Vassilakis, T. (2010). Dremel: Interactive analysis of web-scale datasets. *Proceedings of the VLDB Endowment*, 3(1-2), 330–339. <https://doi.org/10.14778/1920841.1920886>
- [6] Bai, J., Bai, S., Chu, Y., Cui, Z., Dang, K., Deng, X., Fan, Y., Ge, R., Han, Y., Fei, H., & Zhu, T. (2023). *Qwen Technical Report*. arXiv. <https://doi.org/10.48550/arXiv.2309.16609>