Topic: Engineering Resilient Reproducible Analytical Pipelines (RAP): A Semantic-Based Self-Healing Framework for High-Velocity Heterogeneous Data Streams

Keywords: Data engineering, Reproducible Analytical Pipelines (RAP), Autonomous Agents, Data Provenance, Schema Drift, Self-Healing, BERT, Telemetry

Data Availability: https://github.com/tarek-clarke/Resilient-RAP-Framework

Research Abstract:
Mission-critical telemetry systems, including sports performance teams and clinical monitoring systems all face critical limitations in data availability, veracity and velocity. High-frequency data pipelines break easily when upstream schemas shift, sensors fail or interfaces change.

Traditional pipelines rely on brittle selectors or rigid schemas. When these fail, organizations experience data blackouts, delayed decision-making and loss of situational awareness at critical points.

This research implemented a self-healing Reproducible Analytical Pipeline (RAP) designed to autonomously mitigate schema drift without manual intervention. Leveraging a containerized Python ecosystem and BERT Large Language Model Processing, the model replaces static schema changes with a dynamic semantic embedding-driven reconciliation.

Grounded in the software reliability principles of the Pareto distribution (Fenton & Neil, 1999), and tamper-evident processing (Simmhan et al., 2005), the agent uses a cross-domain generalizable model to work in various industries.

This framework introduces a domain-agnostic ingestion interface supported by modular domain adapters that implement industry specific-extraction, validation and normalization logic. This approach enables a unified, cross-domain approach to resilient data ingestion, while reducing pipeline fragility and ensuring the stability of critical, high-velocity analytical workflows in mission-critical environments.

To test this framework, it has been assessed in two distinct telemetry environments where schema drift is possible. It was tested in a Formula 1 driver biometric and car performance telemetry stream, and a healthcare/ICU telemetry stream for patient vitals.

1. **Introduction and Problem Statement**
    a. **Context:** High-velocity telemetry data streams often operate with ingestion pipelines where the upstream data schema evolves rapidly and can create instability in these pipelines. Since these pipelines are often brittle and schema-bound, even minor changes in the schema structure can break the pipeline, leading to “data blackouts”. In high-stakes environments such as motorsport or ICU monitoring, data blackouts can have serious implications on the well-being of their subjects. This research will focus on mitigating the impact of schema drift on these environments and engineering a resilient, self-healing analytical pipeline that is domain agnostic and remains fully reproducible throughout its life-cycle.

    b. **Technical Failure Point:** Current data ingestion pipelines are brittle. They typically rely on static schema structures that break when the source data structure is updated. In high-stakes telemetry environments, an ill-timed schema update can break the data ingestion pipeline at critical moments. This leads to a

"data blackout", limiting the reporting of vital data. Following the research of Pareto analysis of software defects established by Gittens, Kim and Godwin (2005), this research focuses on automating the recovery of the "vital few" structural failures that represent the highest risk to the integrity of longitudinal data.

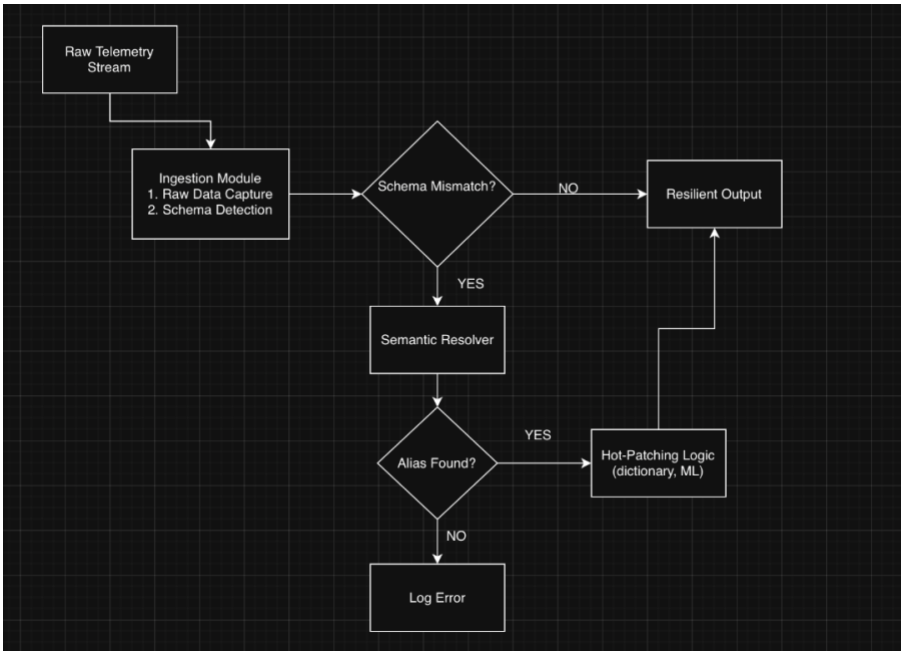
c. **Research Question:** To what extent can semantic embedding models (BERT) autonomously reconcile schema drift in high-velocity telemetry environments without compromising data integrity or traceability?
d. **Theoretical Framework:** This research is grounded in software reliability theory, Pareto defect concentration, and tamper-evident data provenance models, which together inform the design of resilient analytical pipelines.
e. **Outcome:** The outcome of this project is the creation of a reproducible, domain-agnostic self-healing analytical pipeline framework that is able to autonomously adapt to schema drift in high-stakes telemetry environments as the drift magnitude approaches 1.0, which signifies total mutation.

2. Methodology: The Resilient RAP Framework

This project moves beyond ad-hoc scripts to a formalize semantic mapping layer to address schema drift dynamically.

a. Dynamic Semantic-Based Remapping: Moving from brittle regex or manual based schema remapping, research applies a BERT based semantic translation layer to dynamically address schema drift in real-time. This remapping process ensures no data is lost from telemetry streams in critical, high-stakes environments.

b. **The Pipeline as Code:** To ensure the system is logged forensically, hashlib will be used to generate deterministic checksums for auditability.
    i. **Containerisation:** The environment is encapsulated using Docker using the DirectX GPU-PV (/dev/dxg interface), torch-directml and Docker with pinned dependencies. This guarantees the engine built in Year 1 remains functional in Year 3, solving the academic "reproducibility crisis." By containerizing the translation logic, the system ensures that the environment remains stable throughout the entire research and testing process.
    ii. **Self-Healing Logic:** Upon identifying a schema mismatch, the data label is referred to the semantic resolver. The resolver will identifies a suitable alias and autonomously initiates a vector space search to change the variable name to the "Gold-Standard" name. If it is unable to identify an acceptable alias, it will log an error.



iii. **Auditability:** hashlib generates checksums throughout the process to ensure the auditability of each mapping event.

c. **Data Validation:** The validation layer integrates an ensemble learning approach to distinguish between pipeline failures and genuine telemetric volatility. Drawing on methodologies for detecting “Structural Breaks” in time-series data, the system evaluates anomalies not as mere statistical outliers, but as potential indicators of exogenous shocks. By applying Random Forest classifiers to metadata features, such as variable/unit inconsistencies versus missing data packets, the pipeline can autonomously categorise a data spike as either a “Technical Schema Drift” requiring a repair or an event requiring reporting.

d. **Hardware and Resources:** This research uses a combination of my local hardware and cloud resources. It uses a Windows PC with an Intel 12600k CPU and AMD 7900XT 20GB GPU, leveraging PyTorch with a DirectML backend for local hardware acceleration, and Macbook M4 Pro for lighter work. The architecture also uses a serverless data lake (Amazon S3 and Athena) to ensure that the data pipeline is cost-efficient and capable of maintaining an immutable audit trail for all scraped data. AWS is also used for large-scale validation of the self-healing logs.

3. Technical Implementation Stack

a. **Core Architecture:** Python environment, sentence-transformers (semantic mapping), polars (data processing), rich (TUI visualization), requests (HTTP/API client)

b. **ML/AI:** torch (deep learning framework), transformers (transformer model), scikit-learn (ML integration), numpy (numerical processing)

c. **Testing:** pytest (test framework), FastF1 (for F1 environment)

d. **GPU Acceleration:** CUDA or DirectML (for BERT integration)

4. **Scientific Contribution and Relevance**
    a. **Contribution to Computer Science and Systems Design:** The primary technical contribution of this framework is to move from rigid, key-value matching to Semantic Reconciliation. This research contributes a novel semantic-driven schema repair engine enabling Zero-Shot adaptation to upstream structural changes. Unlike a traditional pipeline that relies on brittle regex patterns or static mapping variables, this framework uses a BERT-based semantic translator to map variables that are experiencing schema drift in real-time. The system will convert incoming unknown telemetry variable names (such as vehicle_kph) into high-dimensional vector embeddings. The autonomous repair agent will then calculate the cosine similarity between the unknown tag and the "Gold Standard" schema name (such as Speed (km/h).) The Zero-Shot Adaptability of the model allows the pipeline to ingest data from various hardware vendors or clinical sensors that have variations in schema without manual code changes or retraining, provided the semantic meaning of the available tags remain consistent.

    b. **Relevance to Industry and Development:**
        i. **For Healthcare:** This research provides a method to integrate healthcare telemetry data from various sources that do not require specified software or firmware for each source or update.
        ii. **For Sports:** This research allows for analytics teams for high-performance sports franchises (F1, NHL, Premier League) to maintain longitudinal athlete databases without manual intervention in the event of a source change. For example, if an API changes its schema, the framework allows for the ingestion of the data regardless of the new schema names, provided they have the same semantic meaning.

5. **Risk Mitigation**
    a. **Environmental and Infrastructure Drift:** The use of containerization ensures that the environment remains static regardless of OS or hardware updates.
    b. **Silent Data Corruption:** Data is uploaded to a cloud server on a regular basis to cold storage for long term backup. Deterministic hashing is conducted immediately upon data processing to ensure data integrity.
    c. **Local Hardware Failure:** The environment is containerized using Docker to ensure reproducibility in any hardware environment.

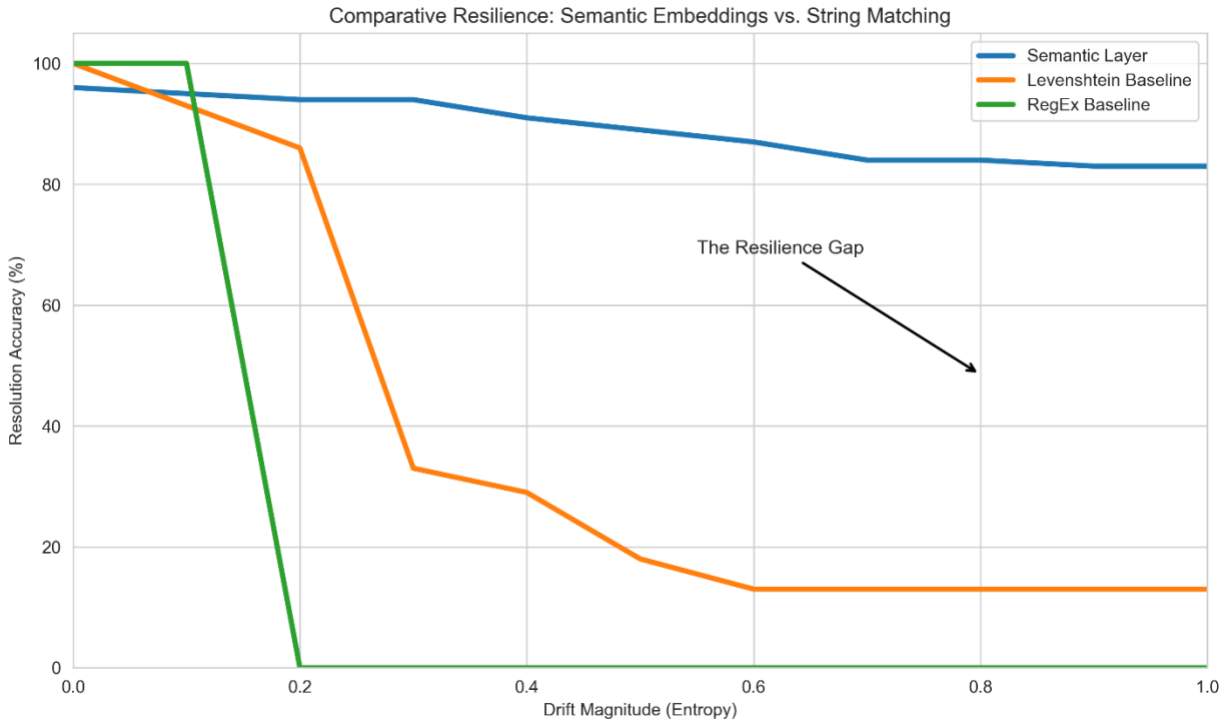
6. **Research Ethics and Institutional Considerations**
    a. **Privacy Consideration:** In order to maintain patient privacy and adhere to all laws, F1 health and ICU data are simulated via a JSON structure and streamed from an internal environment. Formula 1 car telemetry data is streamed via the OpenF1 Python package.
    b. **Algorithmic Accountability:** To ensure the algorithm does not negatively impact the schema, a Human-In-The-Loop system has been implemented. Drawing on the evaluation metrics for Remote Human-In-The-Loop (Goodrich & Schultz, 2007), the dashboard prioritizes "tele-presence" for the end user to validate.
    c. **Data Sovereignty:** All synthetic data generation occurs on the host computer within the secure containerized environment.

7. Results

Benchmarks simulating the schema drift reveal a quantified “Resilience Gap”.

In a low-drift environment (drift magnitude < 0.2), the RegEx baseline maintains 100% accuracy, while the Levenshtein baseline maintains a 95% accuracy. Comparatively, the Semantic Layer maintains a 98% accuracy.

In a high-drift environment (drift magnitude > 0.2), the RegEx baseline drops to 0% accuracy, while the Levenshtein baseline drops below 15% accuracy. The Semantic Layer maintains 85%+ accuracy, even as the drift magnitude approaches 1.0.



8. References

Fenton, N. E., & Neil, M. (1999). A critique of software defect prediction models. IEEE Transactions on Software Engineering, 25(5), 675–689. https://doi.org/10.1109/32.815326

Gittens, M., Kim, Y., & Godwin, D. (2005). The vital few versus the trivial many: Examining the Pareto principle for software. In Proceedings of the 29th Annual International Computer Software and Applications Conference (COMPSAC’05) (pp. 187–192). IEEE.

Goodrich, M. A., & Schultz, A. C. (2007). Human-robot interaction: A survey. Foundations and Trends in Human-Computer Interaction, 1(3), 203–275. https://doi.org/10.1561/1100000005

Simmhan, Y. L., Plale, B., & Gannon, D. (2005). A survey of data provenance in e-science. SIGMOD Record, 34(3), 31–36. https://doi.org/10.1145/1084805.1084812

UK Government Digital Service. (2017). Reproducible Analytical Pipelines (RAP): Strategy for official statistics. https://dataingovernment.blog.gov.uk/2017/03/27/reproducible-analytical-pipeline/