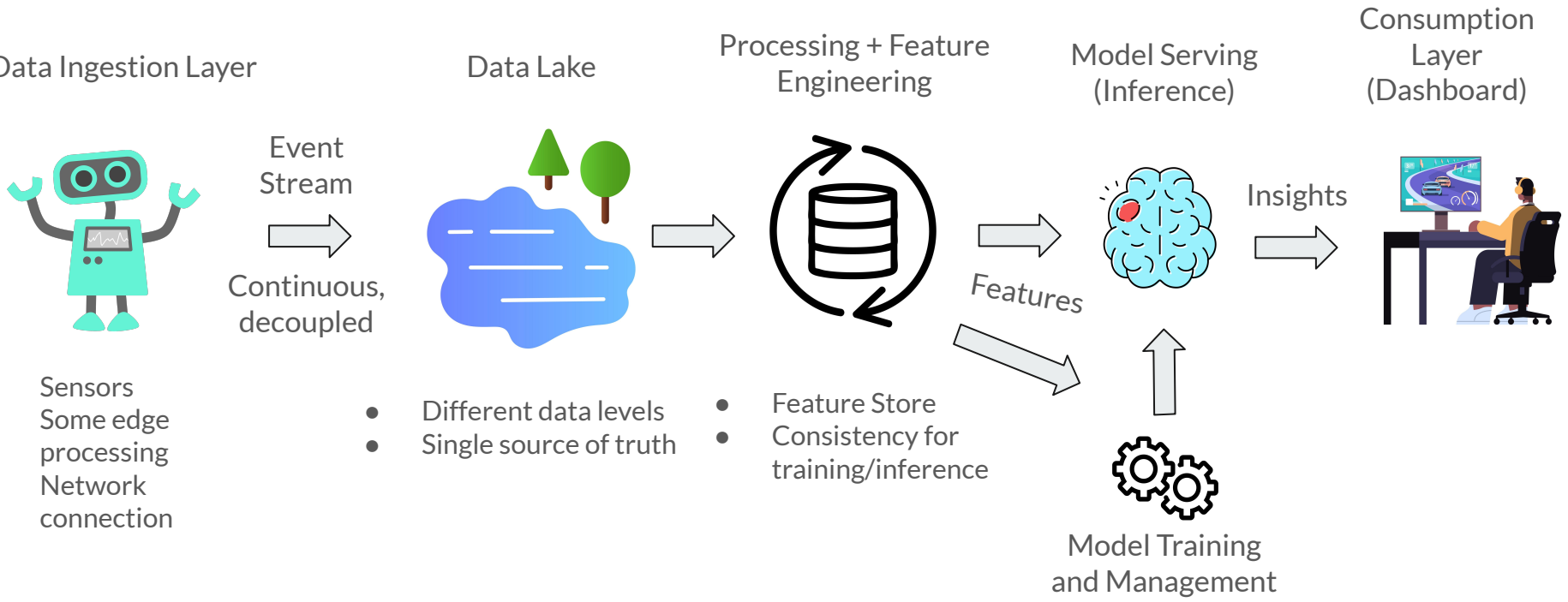
Warehouse Intelligence System

Samuel Bennett

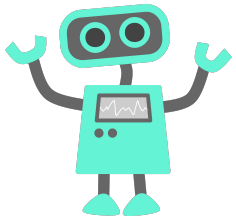
Contents

Architecture: Layers



Architecture: Near Real Time Adaption

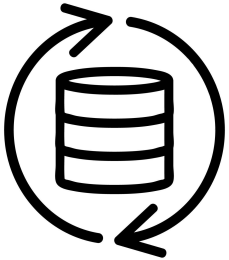
Data Ingestion Layer



Data Lake

Some data streams directly into Processing Layer

Processing + Feature Engineering



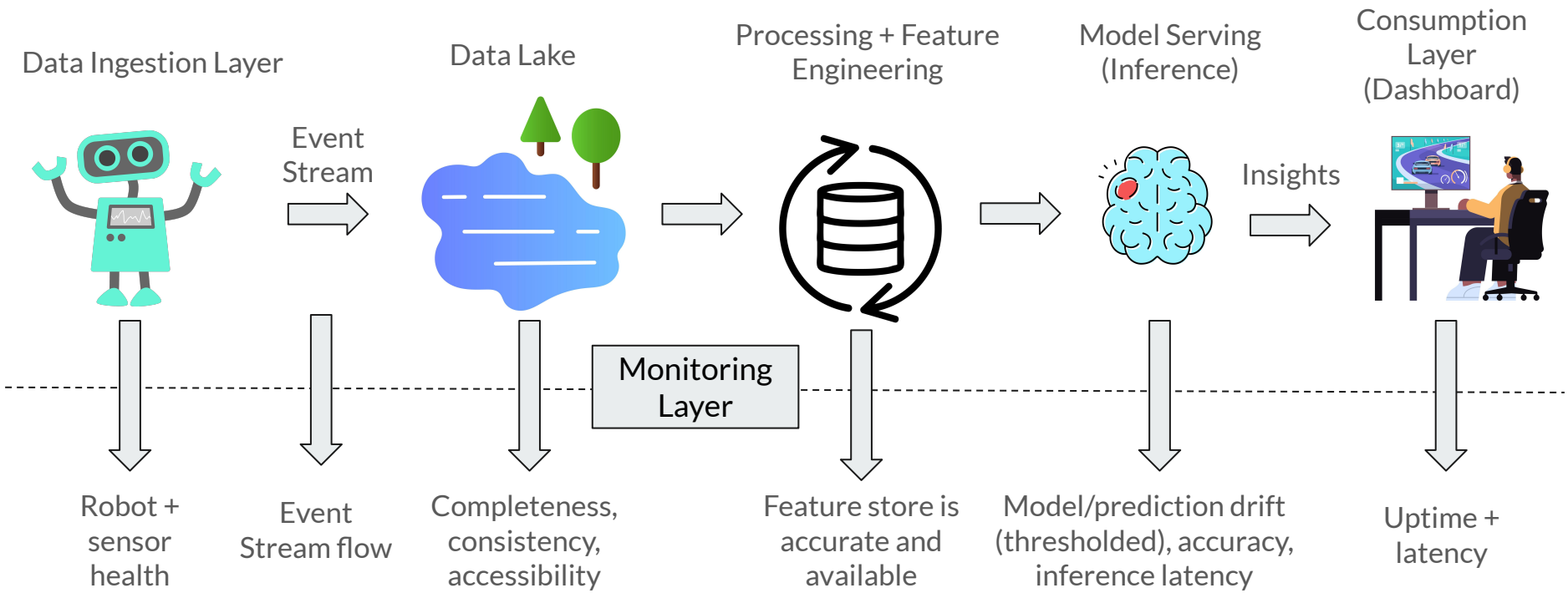
Model Serving (Inference)

Insights

Consumption Layer (Dashboard)

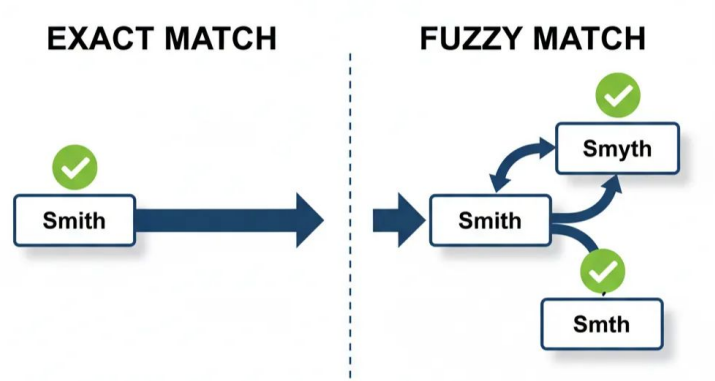
- Alerts
- Dashboard for near real-time monitoring

Architecture: Monitoring



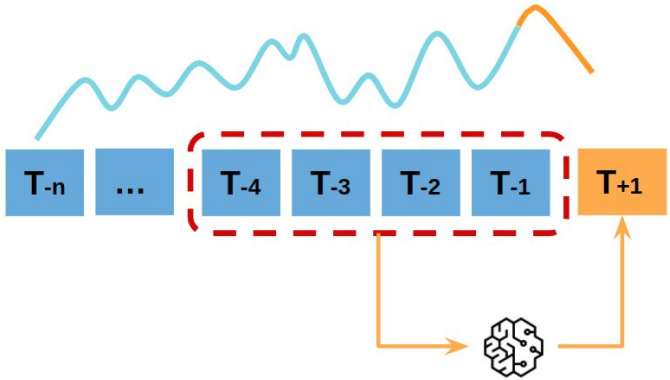
Modelling: Preprocessing

- Merging multiple days of scan data
- Fuzzy text match with rapidfuzz
- Column type conversions
- Configurable CSV output with YAML
- Parquet for categorical variables
- Logging for traceability
- Non-numeric/categorical features dropped for some of the modelling



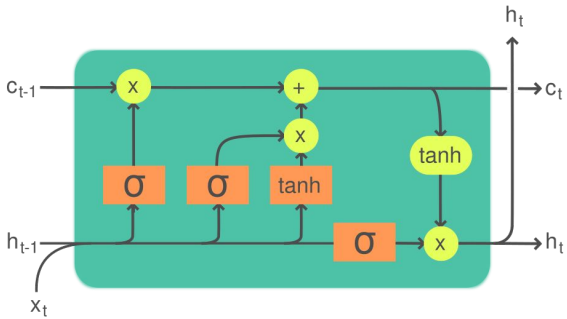
Modelling: Error Prediction Problem Context

- Next-day category prediction from sequence data
- Both numeric and categorical features, with missing values
- Medium sized data set - 30k pts
- Strong temporal and feature correlation
- Potential for “sliding window” of future data over more timesteps



Modelling: Error Prediction LSTM

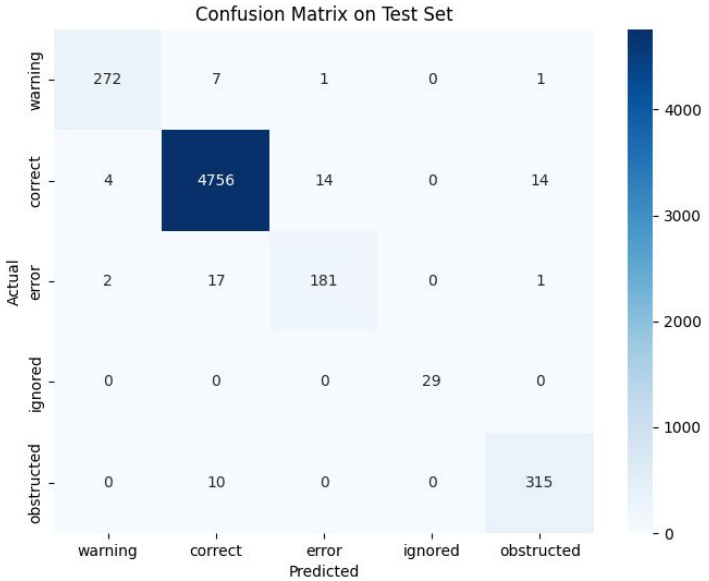
- Numeric feature scaling
- Categorical features as embeddings, with masking/padding for unknowns
- **LSTM** chosen, as can
  - Capture temporal patterns
  - Model complex inter feature dependencies
  - Handle feature types/missing values
- Early stopping with a validation data set to avoid overfitting
- Multi-class prediction, to better encode the feature dependencies



Modelling: Error Prediction LSTM

Pros	Cons
Performance: high accuracy, can capture complex relationships well	**Explainability**: black-box model, difficult to interpret
Scalable: compatible with “sliding-window” of data over time	**Latency**: more processing than a simple feed-forward
Robustness: against missing data	**Compute Cost**: LSTM with embeddings more expensive, so higher train/inference cost

Modelling: Error Prediction LSTM Performance



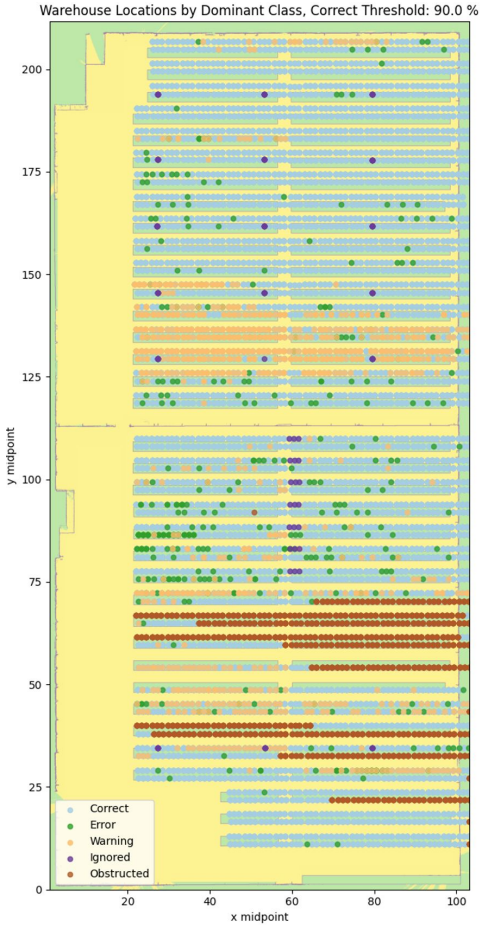
Binary Prediction

	precision	recall	f1-score	support
correct	0.96	0.96	0.96	836
not correct	0.99	0.99	0.99	4788
accuracy			0.99	5624
macro avg	0.98	0.98	0.98	5624
weighted avg	0.99	0.99	0.99	5624

Test Accuracy: 98.83%

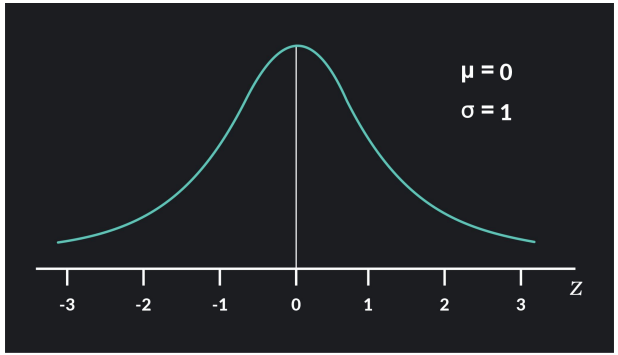
Modelling: Spatial Error Clustering

- Spatial data (x, y, z)
- Potential to have clusters of arbitrary shape/density
- Number of clusters unknown
- Class proportions as features
- Based on this, **HDBSCAN** chosen
- Data averaged over time
- Extension: could consider spatio-temporal clustering directly



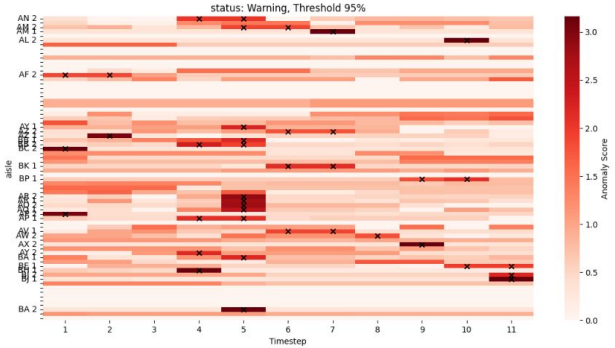
Modelling: Time Series Anomaly Detection

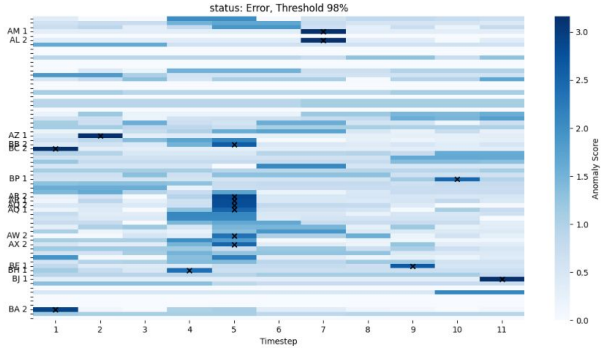
- Compute the average and standard deviation for each class occurrence rate over the time window
- Compute the z-score (statistical distance) of a given point
- If past a threshold - classify as an anomaly

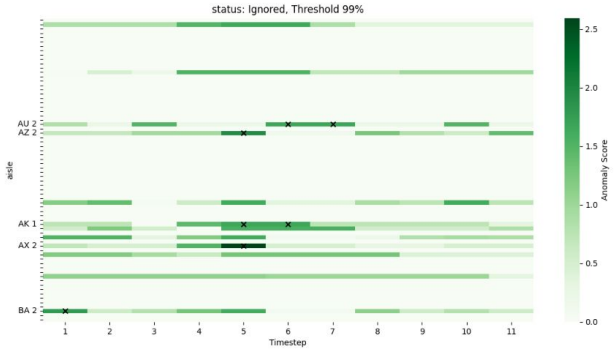


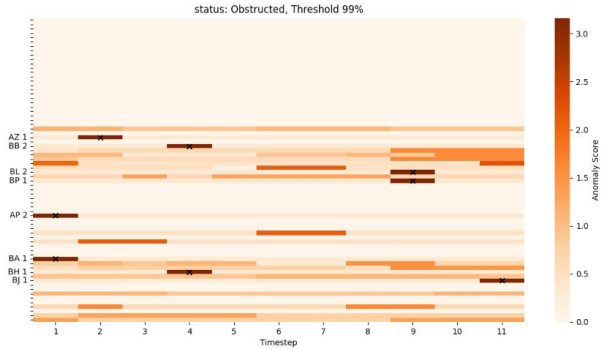
$$z_{i,t,c} = \frac{|p_{i,t,c} - \mu_{i,c}|}{\sigma_{i,c}}$$

Modelling: Time Series Anomaly Detection



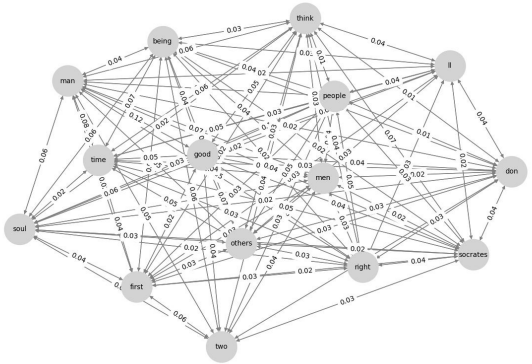






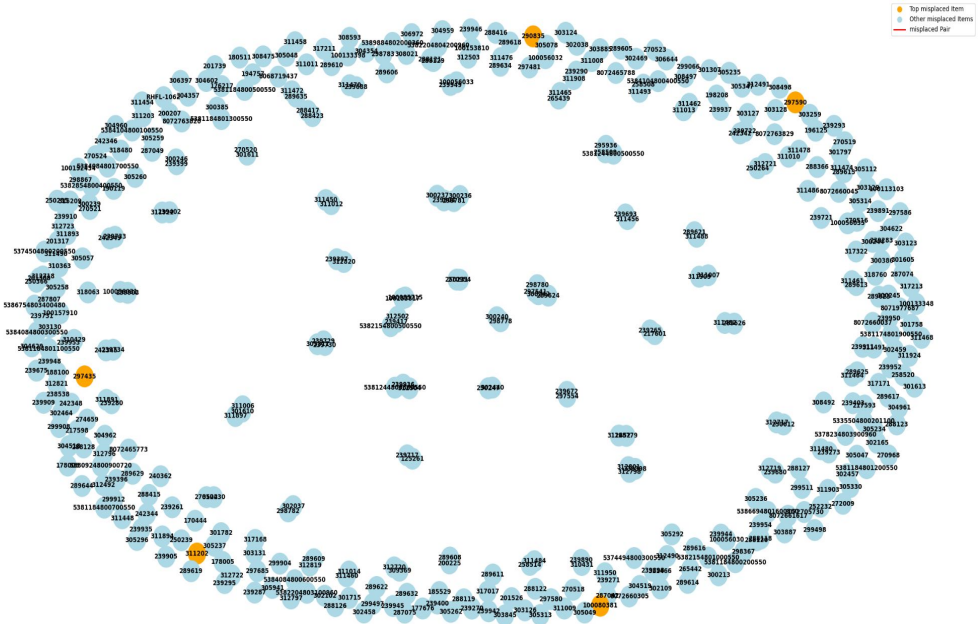
Modelling: Item Co-location and Pick-Path Analysis

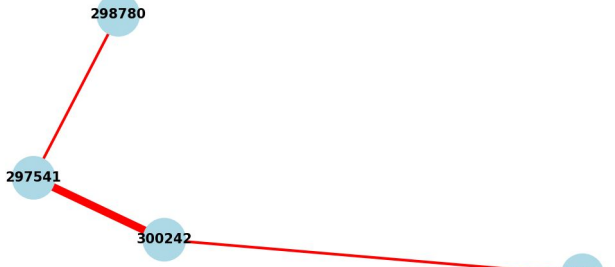
- **Co-located pairs**: Items appearing together in the same location at the same timestep.
- **Misplaced pairs**: Items flagged as misplaced together at the same location/timestep.
- **Picked-together pairs**: Items that move together across timesteps (temporal co-occurrence).
- Count each case for and build association graph



“Store these 2 items together”

Modelling: “Misplaced” Association Graph





Summary

- Presented high-level architecture
- Discussed how to account for near real-time decision making and how to monitor system
- Outlined modelling:
    - Error prediction with LSTM
    - Spatial Error Clustering with HDBSCAN
    - Time series Anomaly Detection by computing z-scores
    - Barcode association graph analysis to generate insights

Contact:

samuelbennett1020@gmail.com