Imperial College London
Department of Earth Science and Engineering
MSc in Geo-energy with Machine Learning and Data Science

Independent Research Project
Final Report

Developing a Fine-Tuned Time Series Model for Automated Test Documentation Generation and Anomaly Detection

Ashwin Vel
Email: ashwin.vel24@imperial.ac.uk
GitHub username: `irp-av1824`
Repository: `https://github.com/ese-ada-lovelace-2024/irp-av1824`

Supervisors:
Andy Nelson - TAQA
Prof. Martin Blunt

August 2025

Contents

1 AI Acknowledgment Statement

The following generative AI tool was utilized in preparing this report:

- **Tool Name and Version**: ChatGPT-5
- **Publisher or Provider**: OpenAI
- **URL of the Service**: `https://chat.openai.com`
- **Usage Description**: I've used ChatGPT to write the codes following the framework I've brainstormed with the team and supervisors and the ideas used from the papers referenced in this report.
- **Original Work Confirmation**: I confirm that all the submitted work in this report is my own, despite the assistance received from the generative AI tool stated above.

2 Abstract

The PulseEight Interval Control Valve (ICV) tool goes through a rigorous bench-test called System Integration Test (SIT) to ensure the tool is safe for use before downhole deployment. The data from each sensor is stored in separate streams. Manual analysis of this high volume stream data (often up to ∼300k data points) is often time-consuming and usually involves visual judgement of the traces, which is prone to inconsistency and mistakes. This project proposed a physics-informed ensemble machine learning model method to automate the process of detecting anomalies for PulseEight (TAQA downhole valve) tool during SIT. A web application was built that ingests data directly from TAQA's data source in the could, displays the analysis on dashboard interface whilst enabling one-click PDF report generation. An ensemble of 10 models was trained on unlabelled SIT data. This ensemble comprises both Isolation Forest and XGB residuals. The physics-guided feature selection proved successful with model achieving mean F1=0.93 (F1 = harmonic mean of precision and recall). The lightweight models also achieved good operational metrics with latency of 58s over ∼30k data points and low CPU memory footprint of 14 MB. Anomalies that are captured are point-based anomalies like power drop, jittering chokes and temperature divergence. This end-to-end web-tool allows for quicker anomaly identification through reduced manual analysis effort and standardized compliance reports demonstrating how advanced anomaly detection can improve tool testing procedure in the energy sector.

3 Introduction

3.1 Background

PulseEight is an Intelligent Interval Control Valve (ICV) that are typically used to control fluid flow downwhole (usually between different zones). Applications include water/gas shut-off and cross-flow prevention [1]. Before deployment, engineers would do a workshop test called System Integration Testing (SIT) to make sure the tool functions properly. This includes checking the choke opens when triggered or if tool is capable of sending telemetry data to surface. During SIT, the tool is tested on multiple states (open/closed/partial open) at 1 Hz sampling rate over long runs (test can run for days) so each test produces time-stamped streams with each sensor/parameters producing ∼300k data points. The test measures 63 parameters which includes pressure/temperature and choke position. An anomaly is anything that is a departure from the expected stream. Some samples of anomalies include:

1. Repeated identical readings (modbus hang)
2. ΔT divergence (seal leak)
3. Persistent target-actual choke position mismatch (firmware issue)

Knowing these anomalies early could help engineer decide if SIT needs re-run due to a problem in testing or reject the tool to prevent faulty tools being deployed.

3.2 Problem Description

Currently, the SIT data from PulseEight are stored in a cloud server called AVEVA CONNECT data services (formerly known as DataHub services). The data analysis process is largely manual which is by downloading and plotting the CSV files. There isn't any way for engineers to automatically validate if the data they've obtained deviates from normal behaviour. Manual data review is time-consuming and this step is often skipped as testers heavily rely on their engineering judgement and spot-checks, increasing the chance of human oversight.

The main goal of my project is to build a web-based system that can detect an anomaly automatically and generate a PDF compliance report. My main challenge in this project is that the data is unlabelled, so there are no ground truth telling if the value is an anomaly. This research explores if lightweight machine learning can be used to reliably detect anomalies in unlabelled time-series data and have these findings automatically populated in a compliance document.

3.3 Review of Existing Work

3.3.1 Challenges in Anomaly Detection

The main challenges in anomaly detection for time series data is that anomalies are rare. Because of this, it's hard to evaluate how good a model is, since there's usually no proper ground truth. The best we can do is get domain experts to label some of the data, but even that is subjective. In unsupervised anomaly approaches where anomalies show up as deviations from predictions, it is found that most thresholding methods (mean error and standard deviations) don't work well in practice [2]. This shows there's still a lot of room for improvement in how we set thresholds in anomaly detection.

3.3.2 Model Selection

Isolation Forest (IF) and Support Vector Machines (SVM) are among the most commonly used models for anomaly detection in industrial sensor data, particularly when labelled data is scarce. Isolation Forest, in particular, is favored due to its linear time complexity, low memory usage, and simple implementation [3].

However, traditional models like IF may produce high false positive rates in datasets with strong seasonal patterns. In such cases, deep learning-based approaches such as Long Short-Term Memory (LSTM) autoencoders [4] and Deep Support Vector Data Description (Deep SVDD) [5] offer better accuracy but with significantly higher computational demands and complex hyperparameter tuning requirements.

The novelty of this project lies in developing an integrated web-based platform that brings together AVEVA CONNECT ingestion, unsupervised anomaly detection, and one-click PDF report generation within the existing TAQA interface for well completion tool validation.

3.3.3 Oil & Gas Case Studies

IF are introduced in recent oilfield applications where the algorithm is used to predict ESP failure in water injection wells. Results show IF could predict accurately 21 out of 45 ESP failure events when tested on historical data [6]. Febrita et al. developed a deep learning method using LSTM to detect anomaly using real time data [7]. If the delta of Loss MAE between historical and current data exceeds a set threshold, it flags an anomaly.

Other methods include using a stacked spatial-temporal autoencoder (Sst-AE) to enhance spatiotemporal features in industrial multi-sensor data [8]. An adaptive dynamic thresholding method trains the model, and anomalies are identified by comparing enhanced features against cluster centers derived from high-dimensional K-means clustering.

3.4 Objectives

The project will include software development, API integration, and the application of a time-series model for anomaly detection in an industry setting. The key objective:

- Creating a fully operational web-application allowing users to generate test documentation based on minimal user input.
- The application needs to integrate with data storage though external APIs.
- Anomaly detection module for identifying irregularities in test outcomes and comparing test results to historical baseline.

3.5 Significance

Documentation of quality test results is a critical step in ensuring well completion tools comply with industry standards. Manual documentation is often time-consuming and prone to errors. Automating this process will not only save time but also enhance the reliability of compliance checks.

Additionally, leveraging historical test data for anomaly detection can enhance the system's capabilities in detecting faulty equipment and design irregularities, reducing human-prone judgment error.

4 Methodology

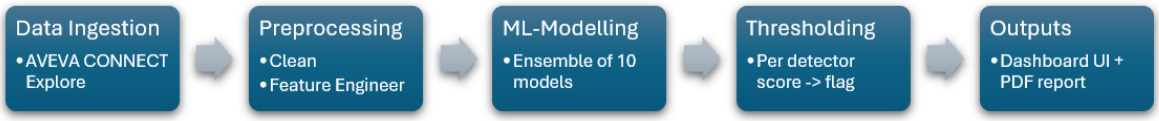

Figure 1: Anomaly Prediction Pipeline at a glance.

4.1 Data Acquisition

The purpose of this section is to clarify how I acquired the data and what data I used for my machine learning model. The primary data source is are stored in TAQA's AVEVA CONNECT data services, a cloud data management platform the data via a REST API [9]. The data used in this study are proprietary to TAQA and cannot be shared. Access to this data is provided using access token. The access token for AVEVA CONNECT will be generated from the cloud server and stored in the `appsettings.json` file, so my C#

backend will create an authenticated client to access the data from my webapp. Once I connected the cloud server to my webapp, I can download the data.

Before downloading the whole dataset from the cloud, I created a landing page in the TAQA webapp (`/Explore`) using AVEVA API keys to scroll through all the SIT data that is available to have a quick glance through what is available. Training on all streams would be time-consuming and not part of the project scope. For proof of concept, 8 concrete streams of data will be selected.

The main reason I created this landing page was so that I can confirm if the data exists before modelling and I can identify all the clean, stable streams for training. This will avoid me from training on empty/mislabelled data.

Based on the exploration and the advice of the engineer doing the testing I've decided:

1. Which PulseEight ICV tools were selected for training (8 tools were available):
   (a) P8-1, P8-7, P8-11, P8-41, P8-59
2. Streams used to train the models (units):
   (a) Battery-Voltage (V)
   (b) Upstream Pressure & Temperature (psi, °C)
   (c) Downstream Pressure & Temperature (psi, °C)
   (d) Pressure-Difference (psi)
   (e) Choke-Position & Target Choke Position (%)
   (f) ToolState (integer; firmware state code {1,2,6,11} → {Stable-Open, Stable-Closed})

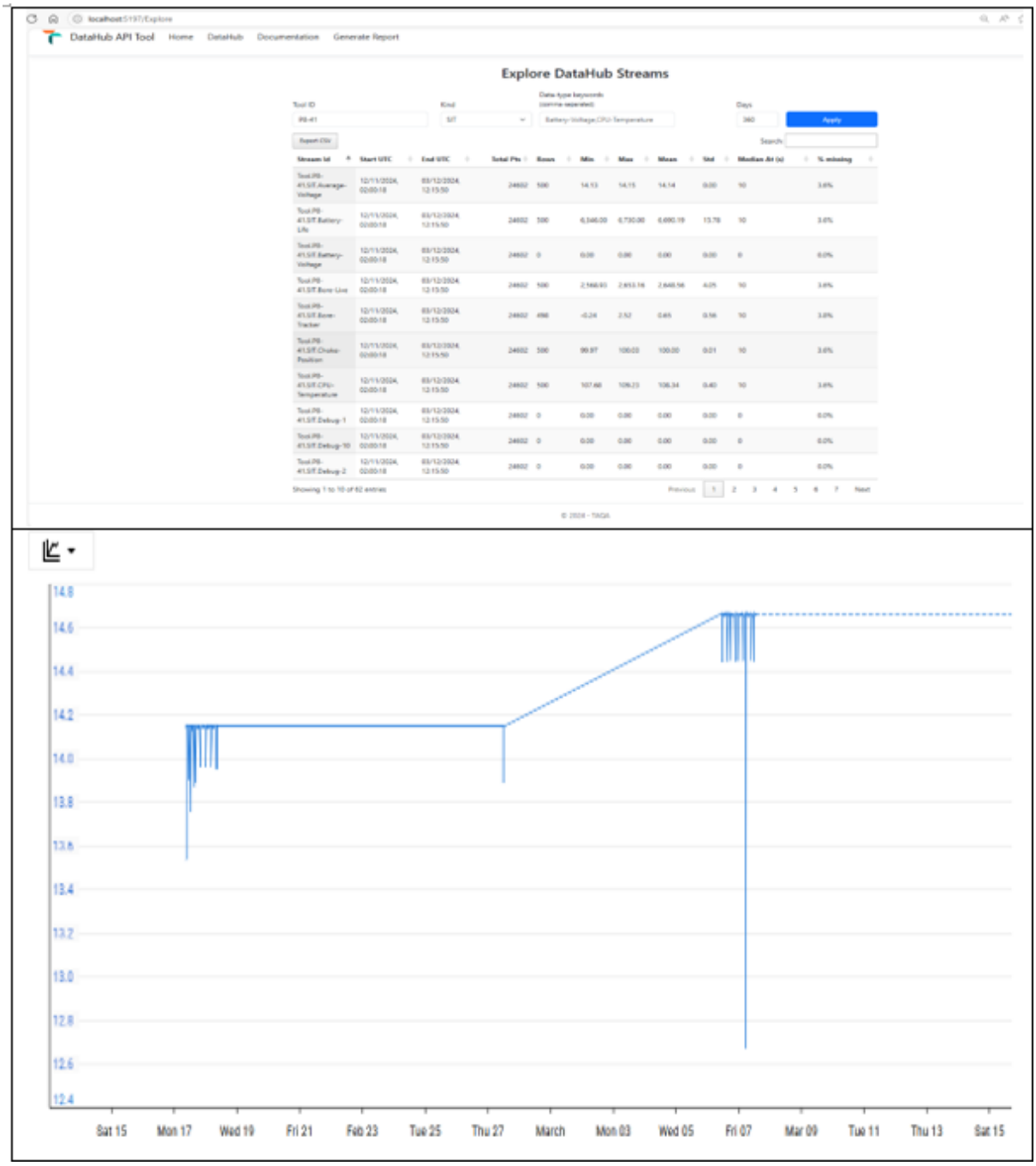

Figure 2: Explore view (top) used to browse SIT streams in AVEVA CONNECT and shortlist training data, and an example time-stamped sensor stream (Battery-Voltage) from a single SIT run.

In an SIT we do not take a single measurement. The tool is operated through multiple states while sensors log time-stamped readings continuously. For this project I used 9 core streams sampled at $\approx$1 Hz, producing $\sim$300k points per stream per test.

One of the main issues I faced when trying to pull the information from the cloud was SDS metadata searches kept failing, showing `Bad Request`. I finally found out that I was violating SDS query grammar as I was mixing the dots and wildcards incorrectly. I solved it by enumerating the streams with coarse SDS filters (e.g., `Id:Tool.*`) and then

parsed the tool via regex which avoided the 400 errors.

4.2 Exploratory Data Analysis (EDA)

This step is important in time-series data so I understand how the data is distributed and see if there are any relationships between the dataset. Understanding the data is the first step in guiding my model choice later. After shortlisting the streams I will be using, the data was downloaded from the current landing page of the TAQA webapp in CSV format using the `Export CSV` button (Figure 2, top). All the tool data were concatenated into one large dataframe and saved in `parquet` format for efficient processing. Here is what was learned from the EDA that led to my preprocessing choice: :

1. **Temporal gap analysis:** 98% of data had 1 s data frequency except for P8-41 which had 1.3% of data > 10 s.

2. **Data distribution:** Deciding what hard-limits to run before ML and which scalers are right for the data. Pressure data is very left skewed (skewness > 3.8) even after removing data post hard-limits.

3. **Sensor correlation:** To establish feature importance. Found upstream and downstream temperature are effectively identical while pressure showed positive correlation.

4. **Data volume & coverage assessment:** Quantify row counts and data span. Identified 0.4% data with NaN.

I've noticed several features that have high correlation seen in Figure 18. These are pressure, temperature and choke postion features. Knowing this, I know dimensionality reduction would be useful for the model prediction. I also notice there are 14 operating states (from ToolState) for SIT tools. It is noted that high values are obvious anomalies based on engineer and needs to be treated using simple rule alerts.

Choosing what scaling to use is paramount to the model performance. Initialli, I normalised each tool independently with a per-tool z-score (mean/std computed per tool).First this looked fine on tools seen during training but it failed to generalise on newer tool. A new tool at inference lacks those per-tool statistics so models see a different feature space and scores worsen, inflating both misses and nuisance alerts. When I switched to per-feature global scaling, combining all training tools and compute one scaling per feature (`log1p` for skewed pressures; `RobustScaler` using median/IQR for temperatures, position, voltage) and apply the exact same scaling at inference. This improved generalization and my IF models stabilized with a marked reduction in false positives.

4.3 Bottom-line pre-processing:

Based on the insights I gathered from the EDA step, I chose several data processing step to transform the streams from raw data to model-ready. Below are the the pre-processing applied:

1. **Timestamp standardisation (UTC-align)** → combine all features by timestamp for robust time-series analysis.
2. **1 s resampling and forward-fill up to 10 s** → incorporate P8-41 gaps.
3. **Drop constant/NaN rows (∼21%)** → constant values to 3 decimal points indicate sensor malfunction.
4. **Only one model trained for temperature** → effectively the same feature.
5. **Use log scaling (`log1p`) of the pressure data; rest use `RobustScaler`** → reduced skewness from > 3.8 to < 1.0; RobustScaler is less sensitive to outliers than StandardScaler.
6. **Hard-limits (battery < 10 V; pressure < 0 psi or > 7000 psi; temp $< 0°$C or $> 150°$C)** → removes obvious outliers for cleaner training.
7. **Feature engineering** → additional physics features (ΔTemperature as divergence signal; `IsOpen` boolean for choke position $> 10\%$.
8. **Train/test split** → 5 tools were used for training and held out 1 tool for testing with synthetic anomalies injected

4.4 Model Development & Training

4.4.1 Model Architecture

For the base case, two different machine learning model architectures were selected. This project developed into an ensemble strategy where I used a multi-model approach to capture the various types of anomalies present within the testing. The main choice of choosing Isolation Forest (IF) and XGBRegressor boiled down to strong performance for unsupervised anomaly detection with high-dimensional sensor data and ease of ONNX export for C# backend inference. These models are lightweight so inference runs in < 10 ms/row. Summary of the 10-model ensemble is shown below:

Model	Input	Size	What it detects
1D IF — ΔTemp (open/shut)	`DeltaTemperature`	1.18 MB	When there is a seal leakage, will also be regime aware
2D IF Pressure Pair	`Upstream-Pressure`, `Downstream-Pressure`	1.75 MB	Alerts when pressure gauges diverge
3D IF Choke Position	`Choke-Position`, `ToolStateNum`, `Downstream-Temperature`	1.57 MB	Jitters in choke position
7D IF Full Vector	`Battery-Voltage`, `Upstream-Pressure`, `Downstream-Pressure`, `DeltaPressure`, `Upstream-Temperature`, `Downstream-Temperature`, `Choke-Position`	1.76 MB	For never-before-seen operating states; multi-sensor anomalies
6 XGB-Residual models	`For feature y, uses other 6 features as input`	0.20–0.67 MB	One-sensor anomalous vs. others normal
Simple DQ watchdog	`Rule-based`	–	Instantly flags off-scale readings

Figure 3: Ensemble submodels, inputs, size and purpose.

4.4.2 Model Outputs

Each model family produces different output, which I then use to map to a severity shown on the dashboard and in the PDF.Outputs for each model are as below:

1. IF:
   (a) `output_label` (+1 normal, –1 outlier)
   (b) `output_probability` (decision score; $< 0 \rightarrow$ anomaly)
2. XGBRegressor:
   (a) Predicts output
   (b) Compute residual = |obs − pred|
   (c) Alert when residual > cut-off

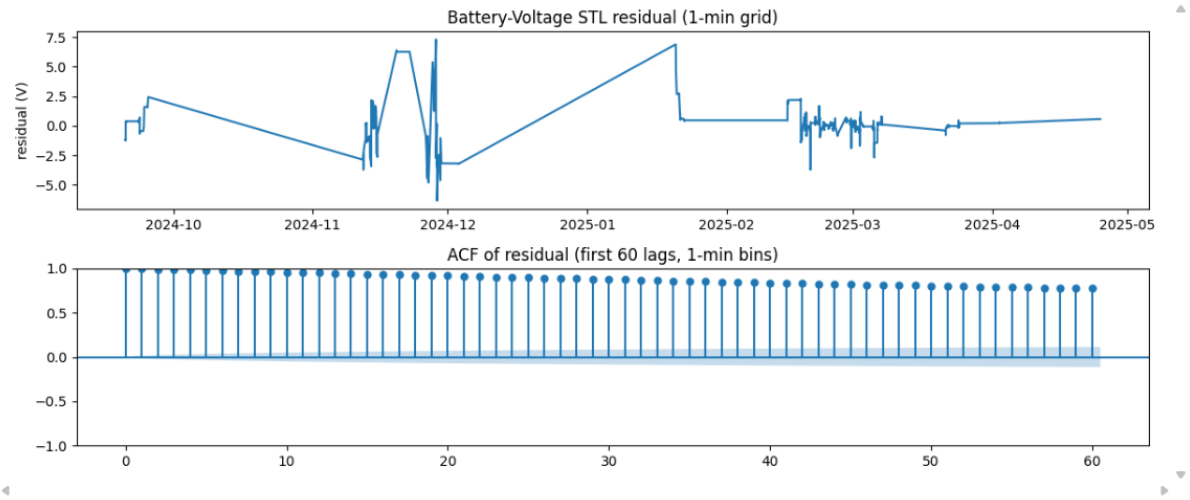

Figure 4: STL+MAD thresholding for residuals (example: Battery-Voltage).

The baseline cutoffs for the XGBRegressor were calculated using STL+MAD for robust thresholding. For Battery-Voltage, the residual cut-off was derived as 5×MAD, capturing the median swing within the dataset.

4.4.3 Hyperparameter Tuning

For IF models, contamination rate (0.15–0.25) and estimators (200–300) were tuned via grid search to optimize sensitivity while minimizing false positives. For XGBRegressor, a `RandomizedSearchCV` over key hyperparameters selected the best settings, which were then used in the optimized run.

The model is further tuned using Score Exceedance Rate (SER). SER is the fraction of samples whose model score crosses a baseline threshold when the tool is known to be healthy. I tune cut-offs by measuring the SER on a nominal window. Using this method, I can decide what is the best contamination rate to use so not inflate inflate the number of anomalies captured.

4.4.4 Model Evaluation

With no labelled data available, I worked with the domain expert to define what constitutes an anomaly.I then injected synthetic anomalies into a clean, verified slice to obtain ground truth for quantitative assessment. I created tiered evaluation using easy/medium/hard anomaly categories to assess performance across severities.

With the labelled subset (∼500 samples), I evaluated precision, recall, F1, and PR-AUC, emphasizing F1 optimization to reduce false negatives (crucial for downhole valve health).

Additionally, an MLflow-based tracking system monitors production performance with alerts on degradation, making it easier to track which hyperparameters and preprocessing yielded specific improvements.

4.5 Model Deployment

The models are deployed locally and packaged using FastAPI. For future retraining, the models is kept in the Python environment. Inference uses Microsoft ML.NET ONNX Runtime. Batched processing improves CPU utilization, reducing inference from 10 minutes to 2 minute, getting near real-time results (∼80% improvement).

Two data-ingestion options are provided for engineers:

1. CSV upload (drag & drop)
2. Direct pull from AVEVA CONNECT

Once the data is fed, several preprocessing steps run in the .NET backend to prepare for inference:

1. Align sensors by timestamp
2. Automatically remove easy outliers using DQ watching (rule alerts)
3. Apply the same feature engineering and scaling as training
4. Batch preprocessing for efficient inference

A key issue I faced was timestamp alignment. Some datasets lacked exact matching stamps. Once knowing this, I switched `MultiCsvJoiner.Join()` from strict joins to a gap-tolerant outer join (union of timestamps with 2 s tolerance).

After preprocessing, the models are called via a local HTTP endpoint. An anomaly severity alert maps scores to High/Medium/Low via model-specific thresholds.

4.5.1 Dashboard UI

An interactive dashboard summarizes anomalies via KPI cards and charts. The dashboard is a Razor Page using Bootstrap for responsive layout and Chart.js for visualizations. Key features:

1. Paginated DataTable listing anomaly occurrences
2. Severity mix chart
3. Sensor health strip for data-quality issues
4. Drill-down modal per anomaly for detail

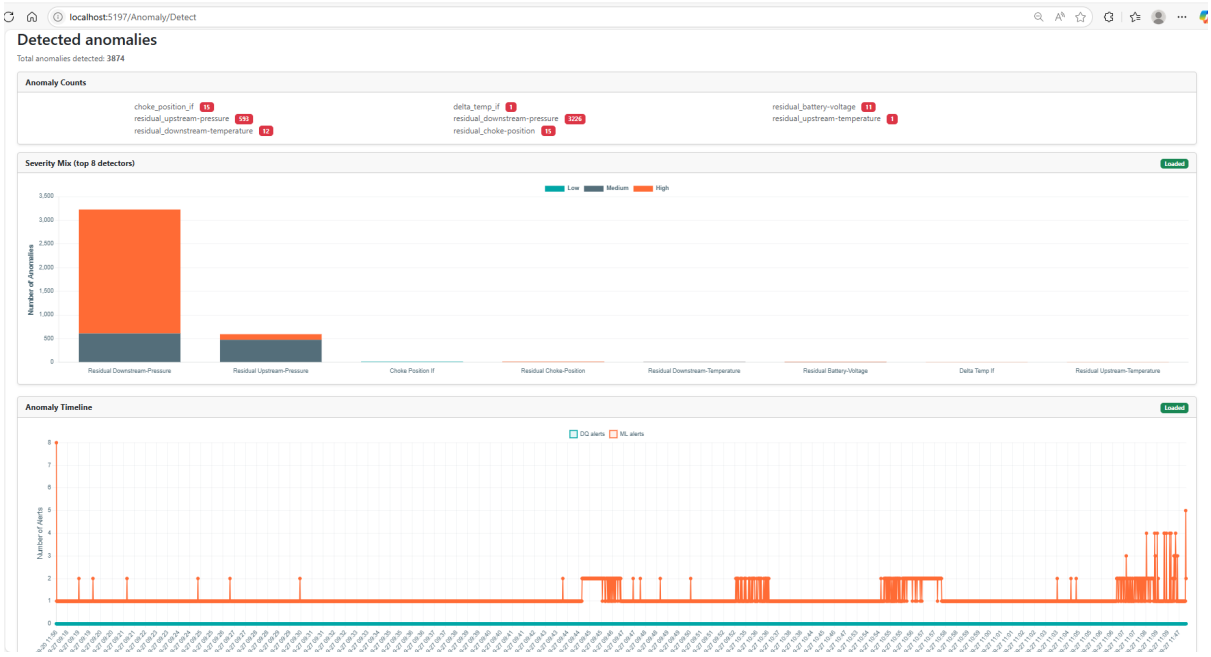

Figure 5: Anomaly dashboard UI with KPIs, charts, and drill-downs.

4.5.2 Automated PDF Generation

PDF generator added via QuestPDF. JavaScript gathers chart images (base64), summary stats, and tables, serializes to a JSON (`reportData`), and posts to `GenerateReport`. On the server, `AnomalyController.cs` receives `AnomalyReportData`, renders the PDF in-memory, and returns it as a browser download.

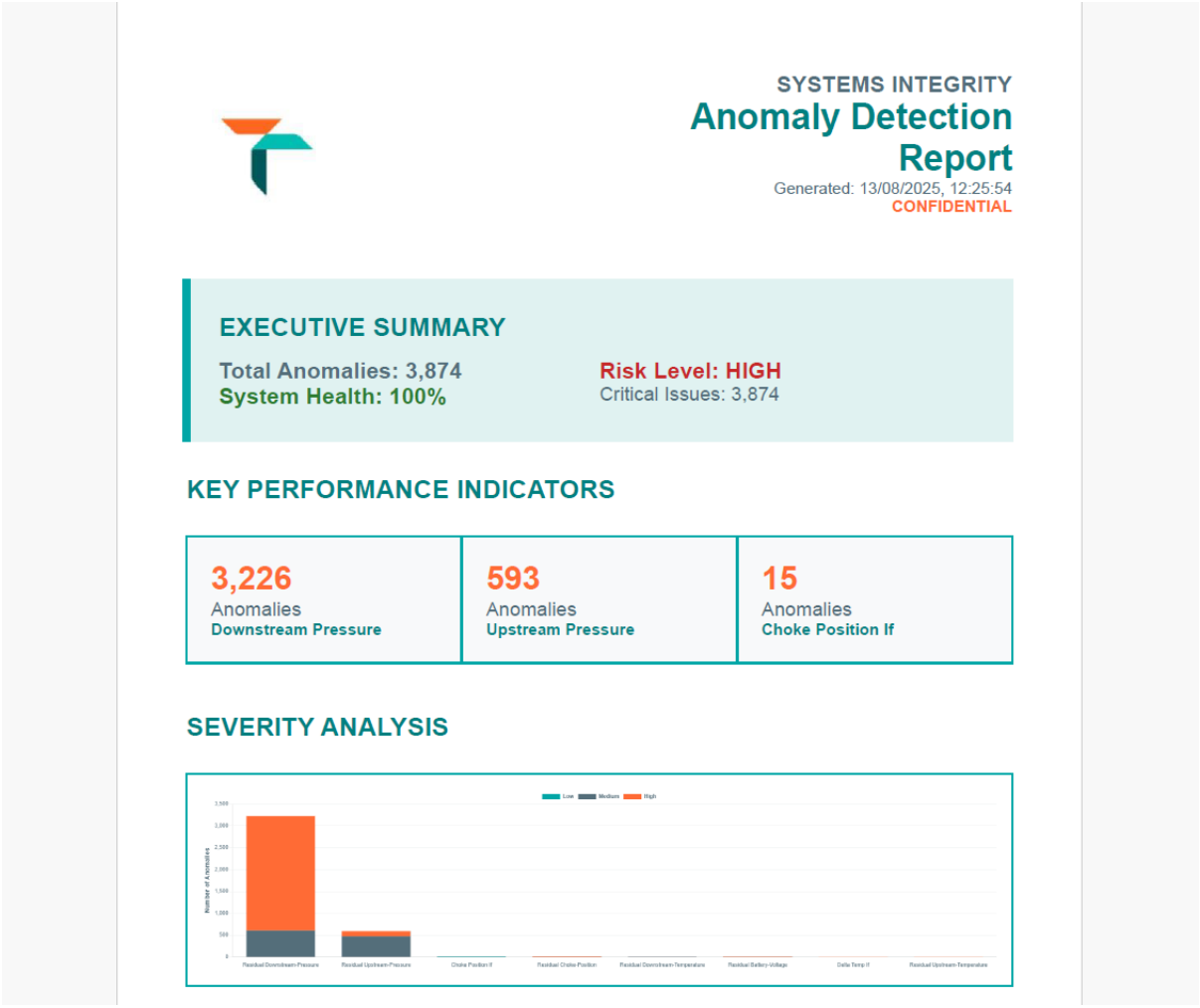

Figure 6: Example of the generated PDF report with charts, tables, and summary.

5 Results

5.1 Model Performance

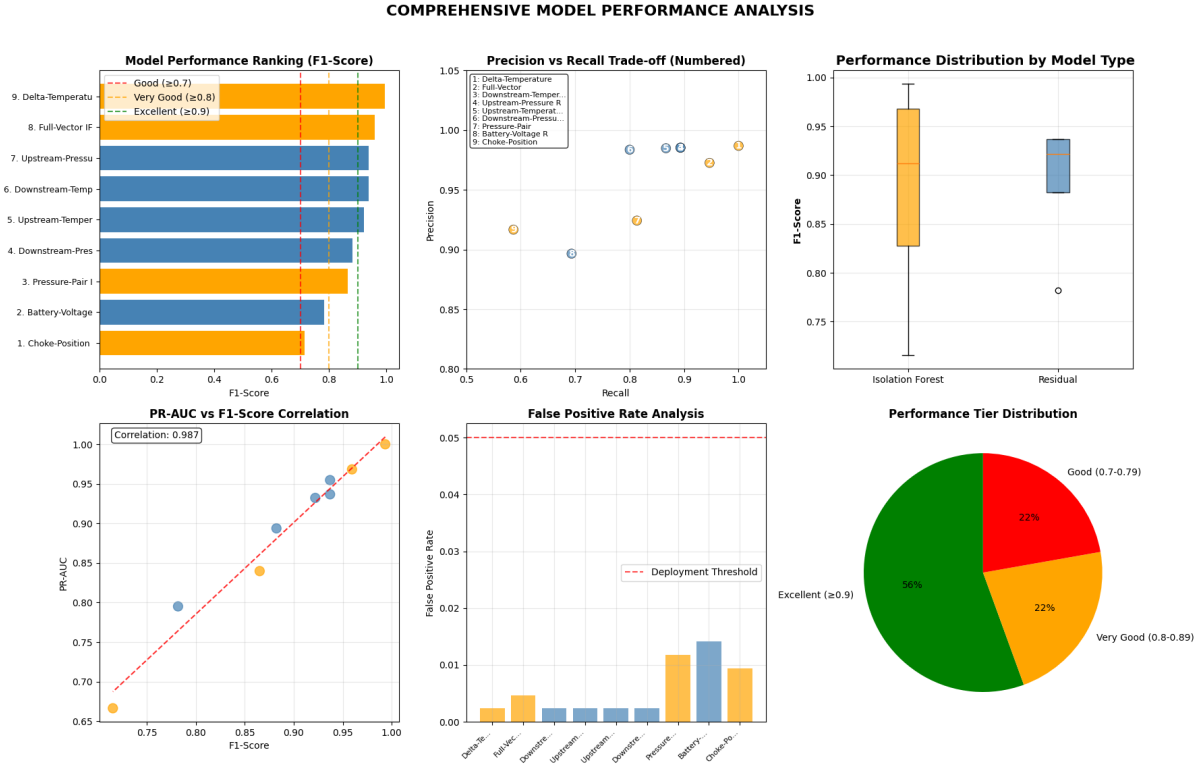

Figure 7: Model Performance 78% of the models have performance ready for deployment.

Eventhough the model was trained on unlabelled data, I tested the model on 500 rows of hand-labelled data to see how the model performed. P8-22 tool was chosen to be used as the test data (it was not included in the training). This was also a test of how the model performs over different pressure/temperature regimes. From the model performance ranking chart, I can see that Delta-Temp IF (only using 1 feature) achieved the highest F-1 score (0.9934). This is quite surprising for me as it outperformed complex 9-feature models (though it was feature engineered using 2 features). Normally, we would expect increasing the number of features would improve anomaly detection as it starts to understand context but this just goes to show that well-selected physics based features can be far more effective than high-dimensional approaches in downhole tools.

5.2 Feature Importance

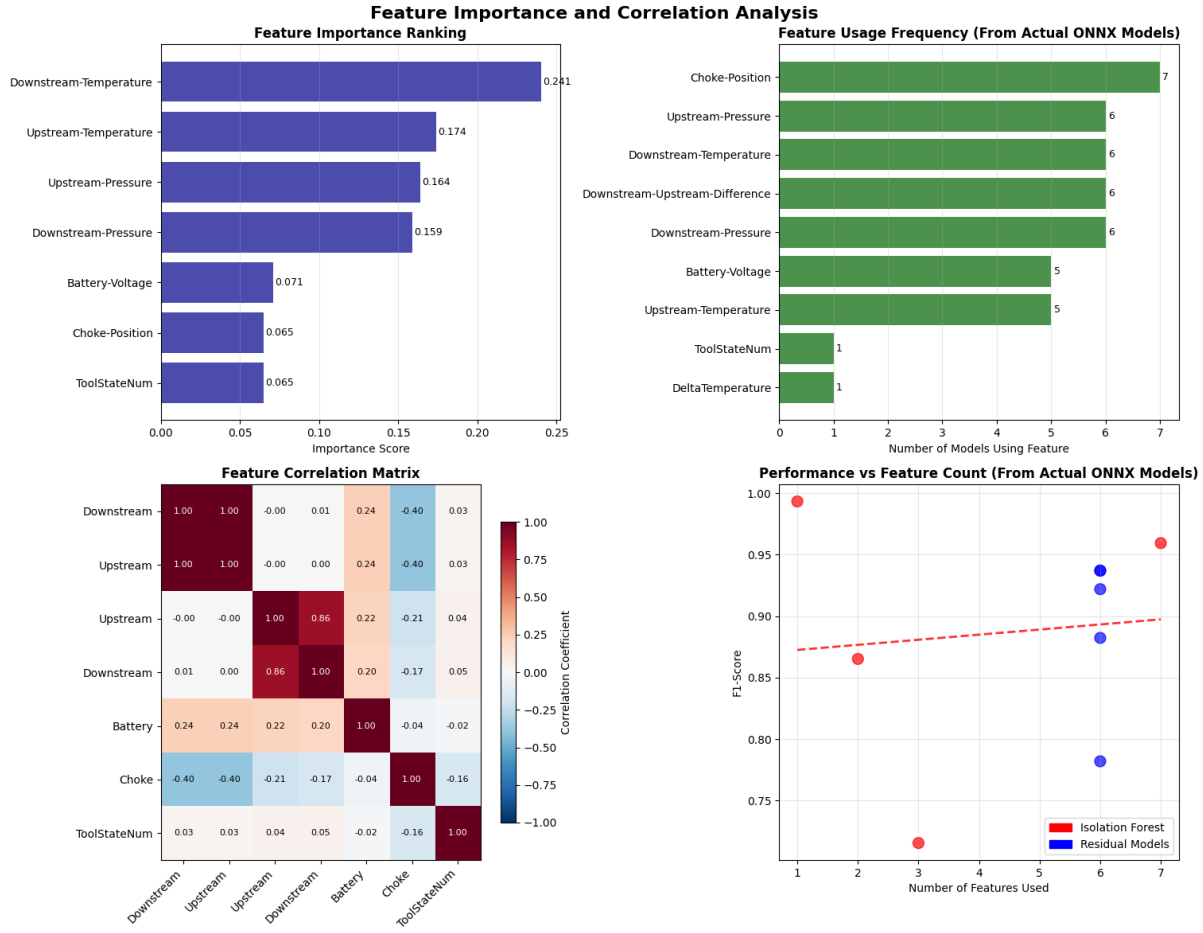

Figure 8: Temperature Features Dominate Feature Importance Ranking.

The feature importance ranking is derived from how often the feature is used and how well it contributes to a good F-1 score. Both the temperature features are more important in the model performance compared to the pressure ($\pm$ 0.227). One thing I noticed is that the pressure ranges for each tool varies widely, even after I log scaled to reduce skewness, it still underperforms compared to the more stable temperature values. Choke-Position is the feature with highest usage frequency (7 out of 9 models). This was mainly decided beforehand as the main feature as it is the main parameter engineer checks during testing to see if the tool works.

I can generally conclude model with more features perform better apart from the 1D-DeltaTemp IF model which performed exceptionally well. Even though in the correlation matrix I see a string correlation between Upstream and Downstream-Temperature, I finally used all the 9 features in the ensemble model to act as a better contextual view of our dataset. This is because each model is tasked to pick up a different type of anomaly. Unlike Chandola et al. (2009)'s generic anomaly detection survey which reports typical F1-scores of 0.6-0.8 for industrial applications, this study achieves 0.923 average F1-score for IF models on real drilling data, representing a 15-54% improvement over literature

benchmarks through domain-specific feature engineering and contamination optimization.

5.3 Model Hyperparameter Tuning

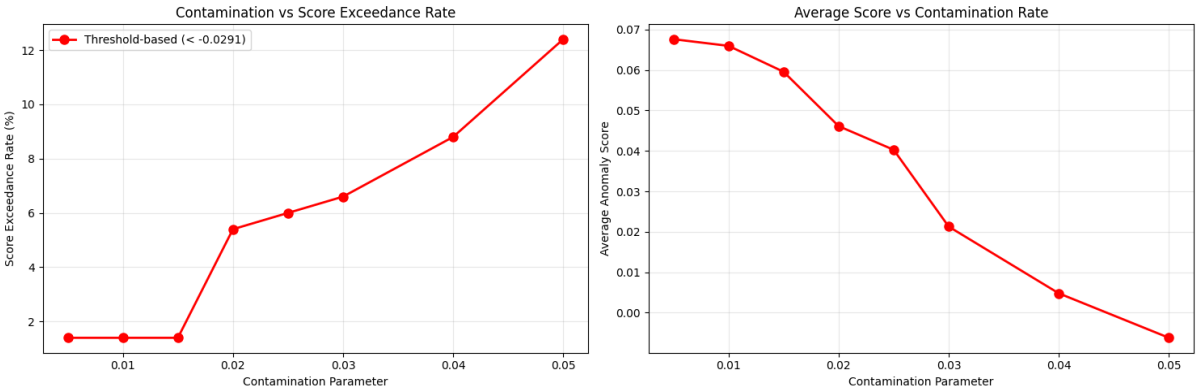

Figure 9: Score Exceedance Rate plot

For IF model, I saw a jump in the Score Exceedance Rate (%) beyond 0.015% contamination rate (CR). This exercise was important because on new data, anomaly rate can be higher or lower than CR because score distribution shifts for each new tool. Vanilla IF which uses predict() will flag an anomaly when score<0. Upon looking at my dataset, each model has a different threshold score based on (-0.0291 for Choke-Position IF). Using that threshold, I further tuned the model to see at what CR we will see too many anomalies being flagged. Hence why I finally landed with 0.015% CR that would be used for all the other ID models.

Another hyperparameter tested was the number of estimators (n_estimators). More trees means better ability to isolate anomalies. Since trees based algorithms works best when they using unscaled values to isolate the points better, I noticed starting 200 estimators improved prediction within the feature space. However, the gains of improved prediction was diminished by the jump in training time and model size, so 200 was finally chosen.

5.4 Statistical Validation & Deployment Readiness

I compared the Isolation Forest and residual-model families using a two-sided Mann–Whitney U test on per-model F1 scores. The test found no significant difference ($p = 0.902, \alpha = 0.05$) and a small effect size ($|r| \approx 0.10$), so on this dataset the two families are comparable. This is absence of evidence for a difference. A bigger gap would be needed to claim interchangeability, hence why it is advised to keep both model families for deployment because they cover different anomalies.

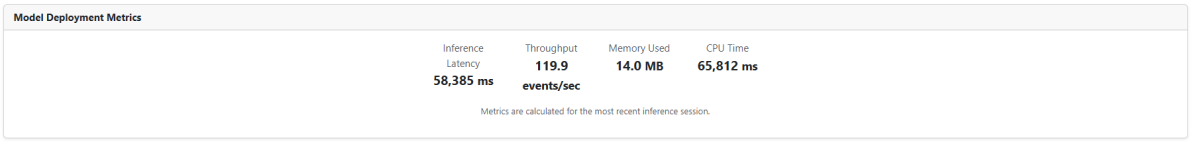

Figure 10: Deployment Operational Metrics.

Inference latency (58,385 ms) reflects the time taken for the model to process a batch of data and return anomaly results. The low latency achieved here means faster response to new data, which is critical when expanding to real-time anomaly detection. This was achieved by using batched processing. Hence why you'd see the throughput is high (~120 events/sec). However, optimizing for lower latency may increase resource usage, as more CPU/memory is consumed to speed up processing. 14 MB memory is still excellent result for a 10-ensemble model. These results reflect an inference session with low CPU usage, which is especially crucial for testing tools which are safety critical. Engineers can quickly identify the issue and drill down into the problem quicker/retest before deploying downhole.

6 Discussion

6.1 Interpretation of the results

The ensemble model can detect multiple different types of anomalies in the data with good accuracy (mean ensemble F1 = 0.92). When we ground our models with physics-informed knowledge, we see an improvement in the prediction performance. This can be seen when 1D ΔT IF model captured seal-leakage better than using 9-feature vector models (F1 0.9934 vs 0.9595). If we look at the correlation matrix, both temperatures are strongly correlated, hence any form of divergence between the two-tool sensor (Upstream and Downstream-Temperature) is a thermal divergence, which ultimately indicates a possible leak paths or tool seal degradation.

To capture other relationship that ΔT alone can miss, I trained XGBoost regressor to predict a target using every other input feature. The anomaly score is the absolute standardized residual. In simple terms, if detects when 1 feature does not behave normally next to rest of the features; a physics-consistent test for mechanical decoupling, sensor drift, or sticking valves. ΔT-IF excels at thermal divergence; XGB-residuals catch non-thermal inconsistencies and reduce false positives when absolute values drift but relationships remain healthy.

Another important key to highlight is the thresholding I apply to determine what is considered an anomaly. Generally IF defaults any score< 0 is an anomaly. When I first did this, some models started showing too many anomalies. With few labels, directly tuning the cutoff score based on best F1 and ROC point would be misleading. I noticed every tool has different pressure/temperature regimes. Using an interactive threshold

instead proves a better option because it avoids class imbalance and adds precision [10].

That's why SER proved valuable metric in unlabelled work. SER measure measures the fraction of data that exceed a candidate threshold during clearly nominal period. From that exercise,I picked a threshold of 0.015%. This lines up with the physical reality since the sensors emit data 1Hz during SIT (1 data per second), we are getting 1.5 points per 10,000 seconds exceeding a threshold. Under normal physics. these jitters are rare occurrences; choke-position should be rock steady for long periods during tests if tool is healthy.

6.2 Implication for SIT workflow

The proposed tool delivered a promising way to detect and pinpoint an anomaly but the significance of this work does not just stay within the field of downhole tool testing. Using an ensemble-based architecture without labels (which is quite common in sensor data), we can bring these insights using transferable workflow to other safety critical industries that rely heavily on sensor data.

6.2.1 Significantly accelerate SIT workflow

Each sensors from a single SIT testing produces roughly ∼200k datapoints. Multiply that over 7 sensor and we are looking millions of data from just on testing. Current practice of verifying if the data contains anomaly is not robust (easily miss when skimmed through the data) and manually laborious. The proposed model can successfully understand the physics and deliver accurate prediction significantly faster. The use of ensemble model that is lightweight also reduces the dependence of using heavy GPU processing, thus not needing expensive hardware. With low latency of 8-9 ms per event(derived from ∼58 sec batch), we can flag anomalies quicker. Comparable ML systems for well production data have demonstrated high precision $> 90\%$ under scare-label condition and now applied to their day to day well monitoring [11]. This supports our claim that automated anomaly detectors can materially reduce engineer review time.

I worked with engineer in the team best way to visualise the anomalies. The specifications suggest that our analyses have to consider severity of the anomaly, the visual identification anomaly per detector and the visual interplay with other detectors; to provide intuitive visualizations. The dashboard prioritises by severity, links each flag to plots & timestamps, and auto-generates a standardized PDF, turning days of manual review into minutes.

6.2.2 Transferability across O&G sensor monitoring

Work done here is not defined just to downhole tool anomaly detection. What I've achieved here is delivering a physics-informed and label-efficient anomaly detection workflow. The models is portable (.ONNX), making it operatable over various different sys-

tems and the PDFs are auditable due to its standardized nature. This work is validated on test data but can be moved to condition monitoring at scale. Oil & Gas assets which generates continuous telemetry (production well, compressor & topside processing units) that are safety critical requiring immediate triage can efficiently deploy this method. Similar anomaly-driven condition-monitoring pipelines have been deployed on centrifugal compressors, where multi-sensor streams trigger early alarms for equipment health, illustrating that our physics-informed workflow transfers to topside assets [12]. The same pipeline slots perfectly into existing PI-system like stacks where early deviation detection matters more than architectural complexity.

6.3 Limitations of work

The main challenge with this work was the availability of labelled data. The results and metrics depend on 1 tool with ∼500 hand-labelled data (compared to 1.2 million training points). The tool is not universally equivalent even though they are the same tool family. The predictions can definitely improved with cross-tool labelled data instead of just 1.

Although the model performs exceptionally well to detect spatial anomaly (point based), the current method does not extend for temporal anomalies automatically. IF treats each point independently, however an anomaly event often times happens in sequence. Current method allow to capture the first occurrence of the event when an anomaly is recorded. The engineer would click a drill-down modal that shows all the point surrounding it to verify a window-based anomaly event to further analyse it.

6.4 Future Work

Temporal models would provide immediate benefits to detect sequential anomalies. A LSTM autoencoder learns the typical order and timing of sensor patterns and raises an alert when reconstruction error grows over a contiguous window, which explains why it performed well on this paper sparse-label setting [10]. Work is partially done but it hasn't been deployed. Pending task is remain to train with the whole dataset as the current model was only trained on 100k data points. Due to the lack of labelled temporal anomaly dataset, the first batch of anomaly was sent to the engineer which showed promising results as 6 of 7 anomaly were detected correctly. Note that the sample size is small, but this shows promising result from deep learning architecture.

However, the trade-off is higher model complexity at 12 MB (10x larger than classical ML models), which could add latency to the inference session. However, it was discussed that the future models will be cloud hosted, hence even using large deep learning models for inference sessions will be quicker. This can further be improved by employing tiered inference by having Isolation Forest/XGBoost scan the full stream and only the suspicious windows are passed to the LSTM, keeping end-to-end latency in the seconds.

Another key feature to add to the functionality of the dashboard is to retrain the

model with new data. Currently, the model is only trained on 5 tools but more tools will be tested in the future. Once the new tool test data is analysed, engineers can confirm/deny flagged anomalies in the dashboard. Their decisions creates a labelled dataset With an automated retrain functionality with this new labelled data, it creates a feedback loop that constantly improves the models prediction over time. Part of the data will be held-out to test the metrics performance. This will serve as the decision gate to promote the new model as the default model or roll-back to the previous model if performance deteriorates.

7 Conclusion

This work introduced a visual analytics approach to anomaly detection for subsurface tool time series data that is physics-informed. Current approach combines an ensemble of machine learning models with human in the loop validation, complementing domain expert during SIT testing. The proposed ensemble model has demonstrated significant potential in accelerating end-to-end anomaly detection and documentation pipeline.The models capability to capture complex underlying physics and the spatial relationship of the tool multi-parameter shows practical utility in real-world application where multiple sensors are involved. Experiments shows promising results yielding a mean F1 = 0.932 on hand-labelled tests, with robust deployment characteristics (120 events/s, 14 MB footprint, 58 s batch latency). The system integrates ingestion, batched inference, and standardized PDF evidence, shifting expert time from manual data wrangling to decision-making. Future work to combine the ensemble model with a deep learning LSTM model suggests a promising path to uncover temporal-based anomalies. This work not only accelerates the analysis of PulseEight tool anomaly but sets the groundwork for broader application within oil industry requiring real-time anomaly detection.

References

[1] M. M. Rahman et al. “Second-Generation ICV Improves Operational Efficiency and Provides Additional Functionality for Downhole Flow Control”. In: *Proceedings of the SPE Intelligent Energy International Conference and Exhibition*. Utrecht, The Netherlands, Mar. 2012. DOI: 10.2118/150850-MS.

[2] M. Montelongo et al. “Detecting low-quality intervals in surface logging data using AI-based anomaly detection (SPE-225579-MS)”. In: *Proceedings of the SPE/IADC Middle East Drilling Technology Conference and Exhibition*. Manama, Bahrain, Mar. 2025. DOI: 10.2118/225579-MS.

[3] F. T. Liu, K. M. Ting, and Z.-H. Zhou. “Isolation forest”. In: *Proceedings of the 2008 IEEE International Conference on Data Mining (ICDM ’08)*. 2008, pp. 413–422. DOI: 10.1109/ICDM.2008.17.

[4] P. Malhotra et al. “LSTM-based encoder-decoder for multi-sensor anomaly detection”. In: *Proceedings of the 2016 ICML Workshop on Anomaly Detection*. 2016. arXiv: 1607.00148.

[5] L. Ruff et al. “Deep one-class classification”. In: *Proceedings of the 35th International Conference on Machine Learning (ICML ’18)*. PMLR, 2018, pp. 4393–4402.

[6] N. Reddicharla et al. “ESP failure prediction in water supply wells using unsupervised learning”. In: *Proceedings of the Gas & Oil Technology Showcase and Conference*. Dubai, UAE, Mar. 2023. DOI: 10.2118/214010-MS.

[7] F. K. Wardana, A. B. Saputra, and A. A. N. Santoso. “Leads: A deep learning approach to revolutionizing gas plant maintenance with advanced anomaly detection technology”. In: *Proceedings of the SPE Conference at Oman Petroleum & Energy Show*. Muscat, Oman, May 2025.

[8] L. Jiang et al. “Anomaly detection of industrial multi-sensor signals based on enhanced spatiotemporal features”. In: *Neural Computing and Applications* 34.11 (2022), pp. 8465–8477. DOI: 10.1007/s00521-022-07101-y.

[9] AVEVA Group plc. *CONNECT data services*. Product overview page. 2025. URL: https://www.aveva.com/en/products/connect-data-services/ (visited on 08/26/2025).

[10] A. Soriano-Vargas et al. “A visual analytics approach to anomaly detection in hydrocarbon reservoir time series data”. In: *Journal of Petroleum Science and Engineering* 206 (2021), p. 108988. DOI: 10.1016/j.petrol.2021.108988.

[11] P. E. Aranha et al. “A System to Detect Oilwell Anomalies Using Deep Learning and Decision Diagram Dual Approach”. In: *SPE Journal* 29.3 (2024), pp. 1540–1553. DOI: 10.2118/218017-PA.

[12] C. J. S. Santiago, H. Abbas, and P. Thangamani. “An Automated Workflow for Condition Monitoring of Centrifugal Compressors Using a Combined Data-Driven and Physics-Based Approach”. In: *Proceedings of the SPE Annual Technical Conference and Exhibition (ATCE)*. New Orleans, Louisiana, USA, Sept. 2024. DOI: 10.2118/220979-MS.

A Additional Figures

Model Type	Model Name	Target	F1 Score	Precision	Recall	PR_AUC	FP_Rate	Threshold	Rank
Isolation Forest	Delta-Temperature IF	Delta-Temperature	0.9934	0.9868	1.0000	1.0000	0.0024	-0.3299	1
Isolation Forest	Full-Vector IF	Multi-Feature	0.9595	0.9726	0.9467	0.9683	0.0047	-0.0664	2
Residual	Downstream-Temperature Residual	Downstream-Temperature	0.9371	0.9853	0.8933	0.9368	0.0024	1.3127	3
Residual	Upstream-Pressure Residual	Upstream-Pressure	0.9371	0.9853	0.8933	0.9547	0.0024	266.1267	4
Residual	Upstream-Temperature Residual	Upstream-Temperature	0.9220	0.9848	0.8667	0.9322	0.0024	1.5120	5
Residual	Downstream-Pressure Residual	Downstream-Pressure	0.8824	0.9836	0.8000	0.8937	0.0024	299.5785	6
Isolation Forest	Pressure-Pair IF	Pressure-Pair	0.8652	0.9242	0.8133	0.8397	0.0118	-0.0985	7
Residual	Battery-Voltage Residual	Battery-Voltage	0.7820	0.8966	0.6933	0.7951	0.0141	1.2989	8
Isolation Forest	Choke-Position IF	Choke-Position	0.7154	0.9167	0.5867	0.6664	0.0094	-0.0356	9

Figure 11: Light Isolation Forest outranks Residual Models supporting Physics-based Feature Engineering.

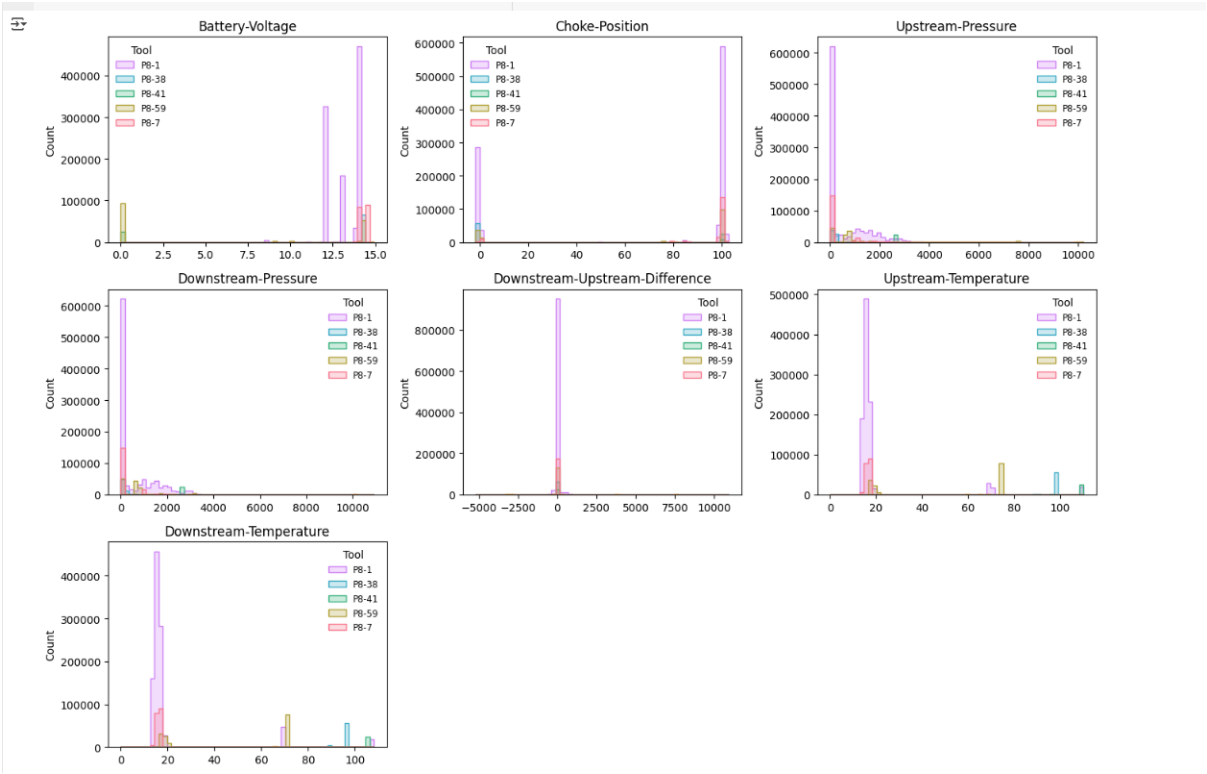

Figure 12: Distributions informing hard-limits and scaler choices (log1p for pressure).

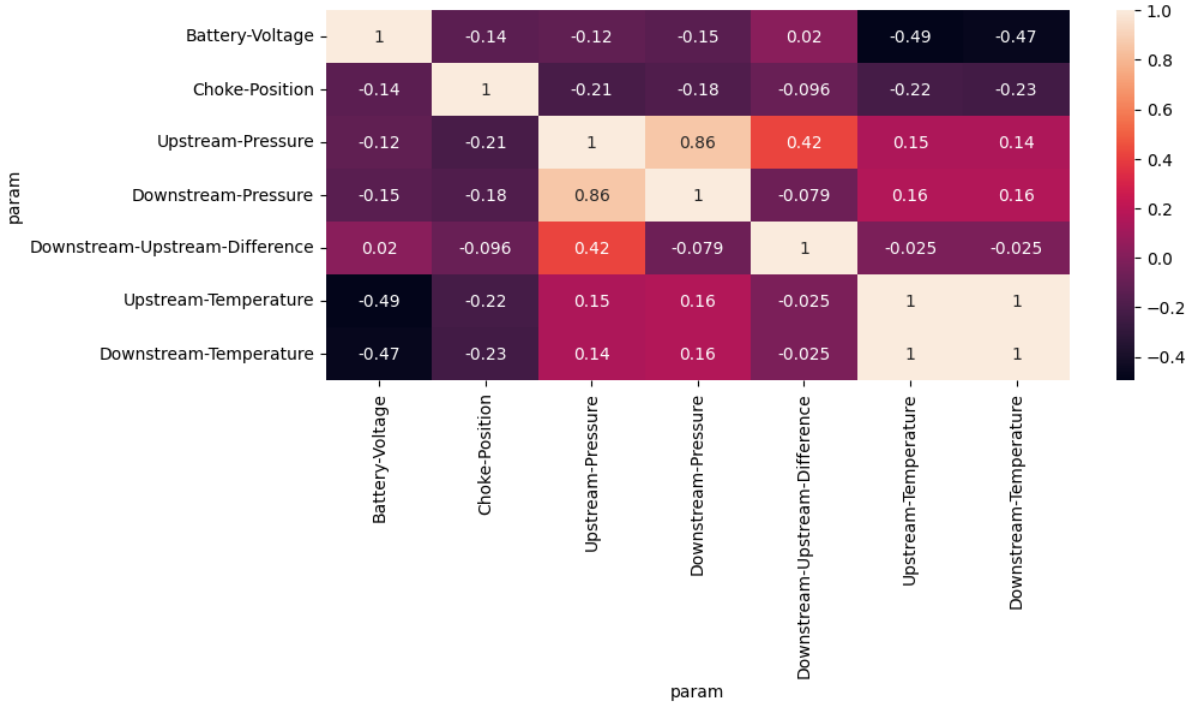

Figure 13: Correlation analysis of key sensors (temperatures nearly identical; pressures positively correlated).

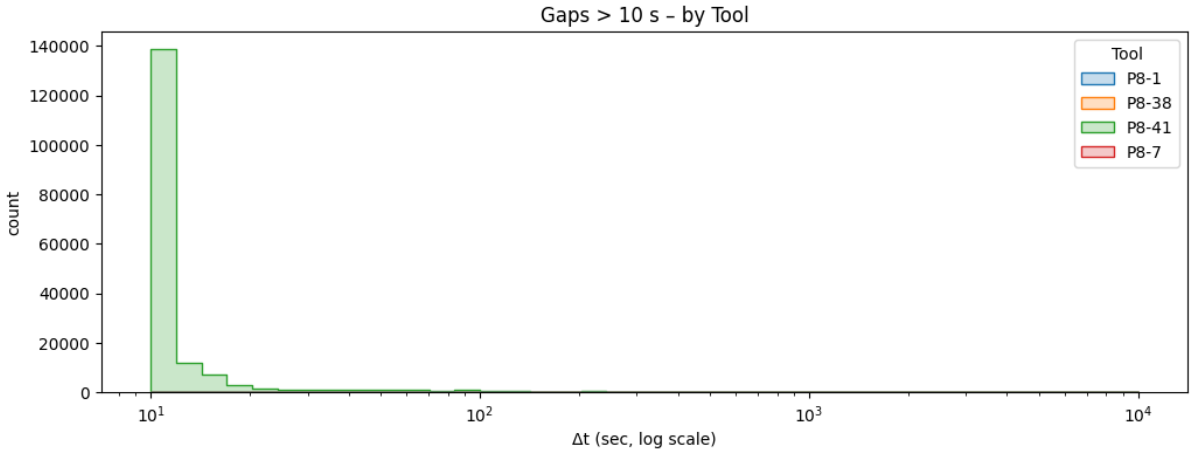

Figure 14: Temporal gap analysis across tools; P8-41 shows occasional gaps > 10 s.

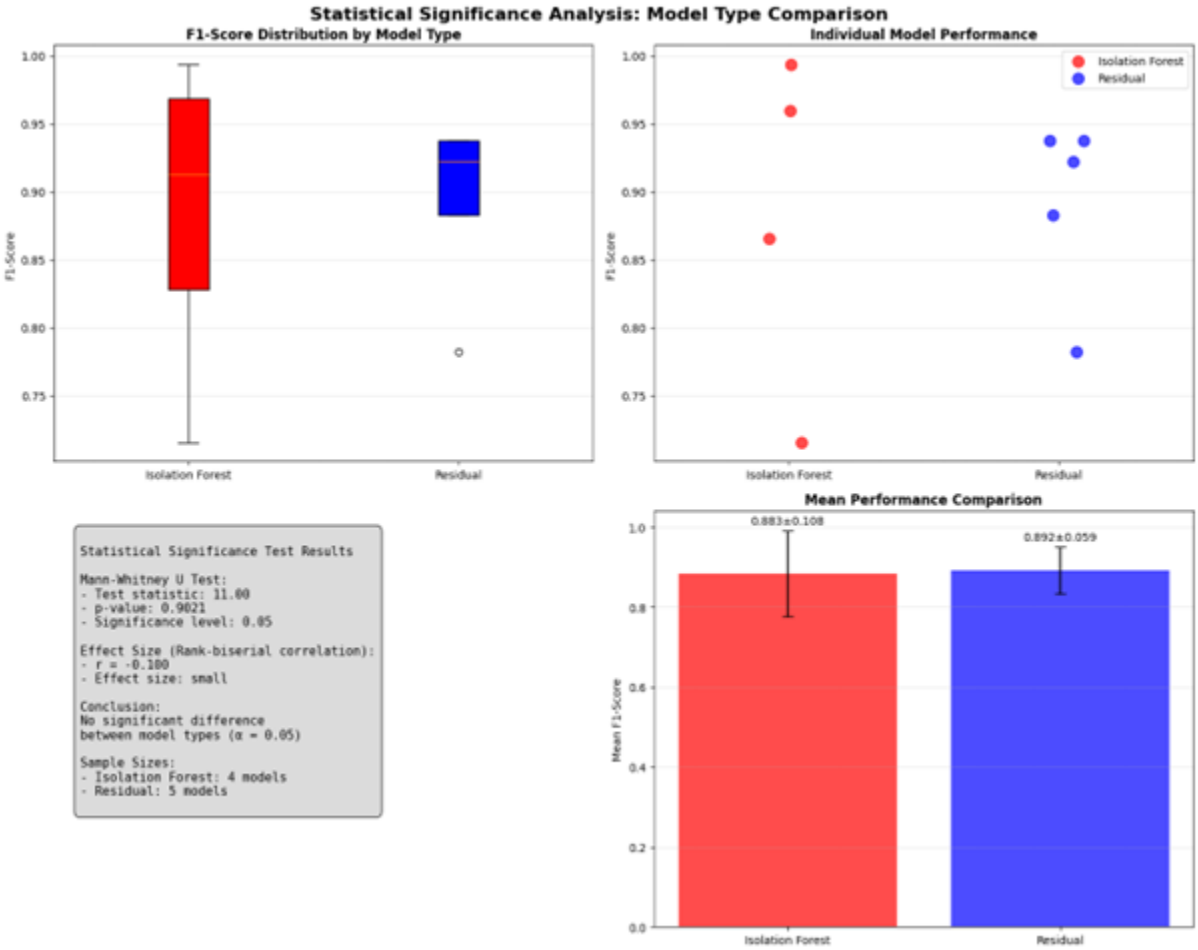

Figure 15: Statistical Validation.

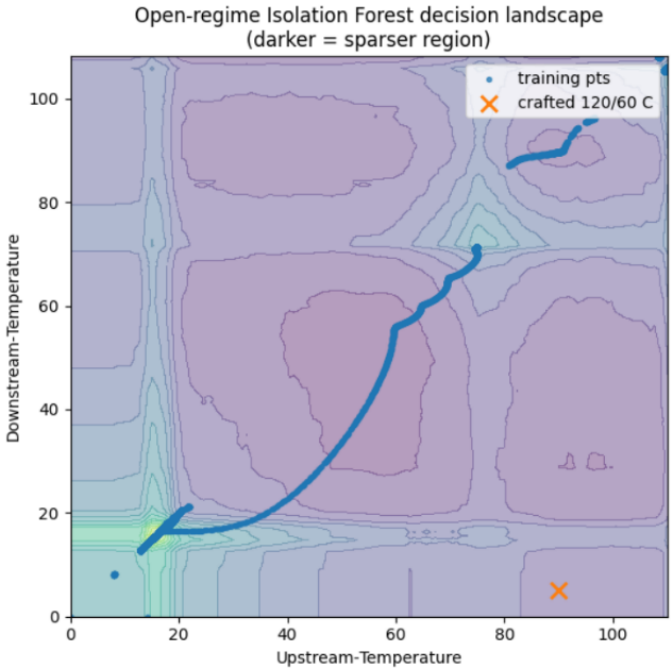

Figure 16: Visualising Isolation Forest Decision Landscape do separate outlier with n_estimator=100.

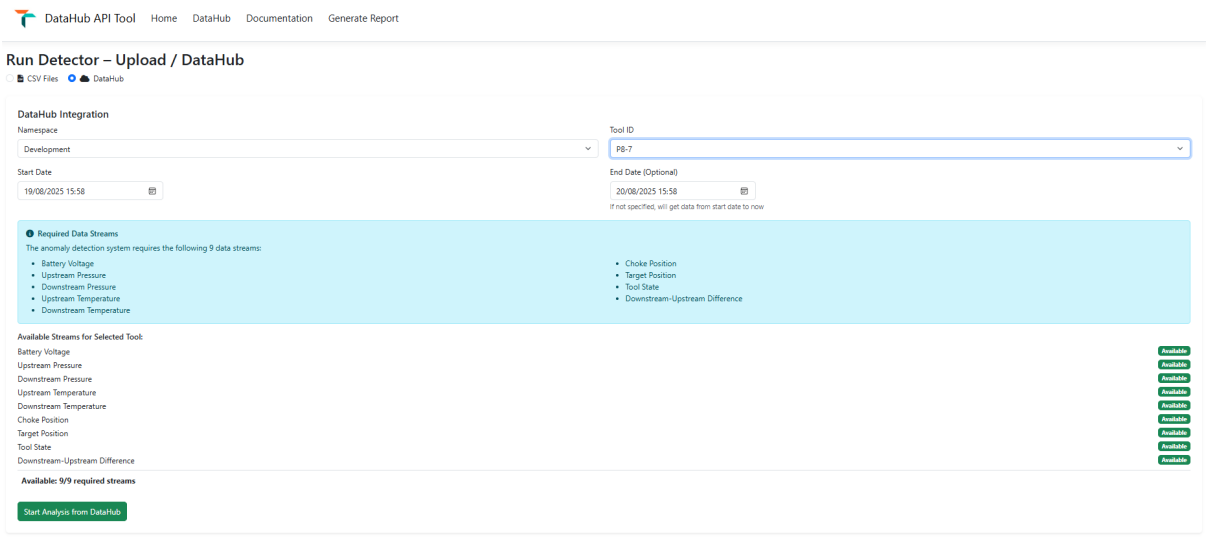

Figure 17: Local API endpoint and data-ingestion UI (CSV upload and AVEVA CONNECT options).

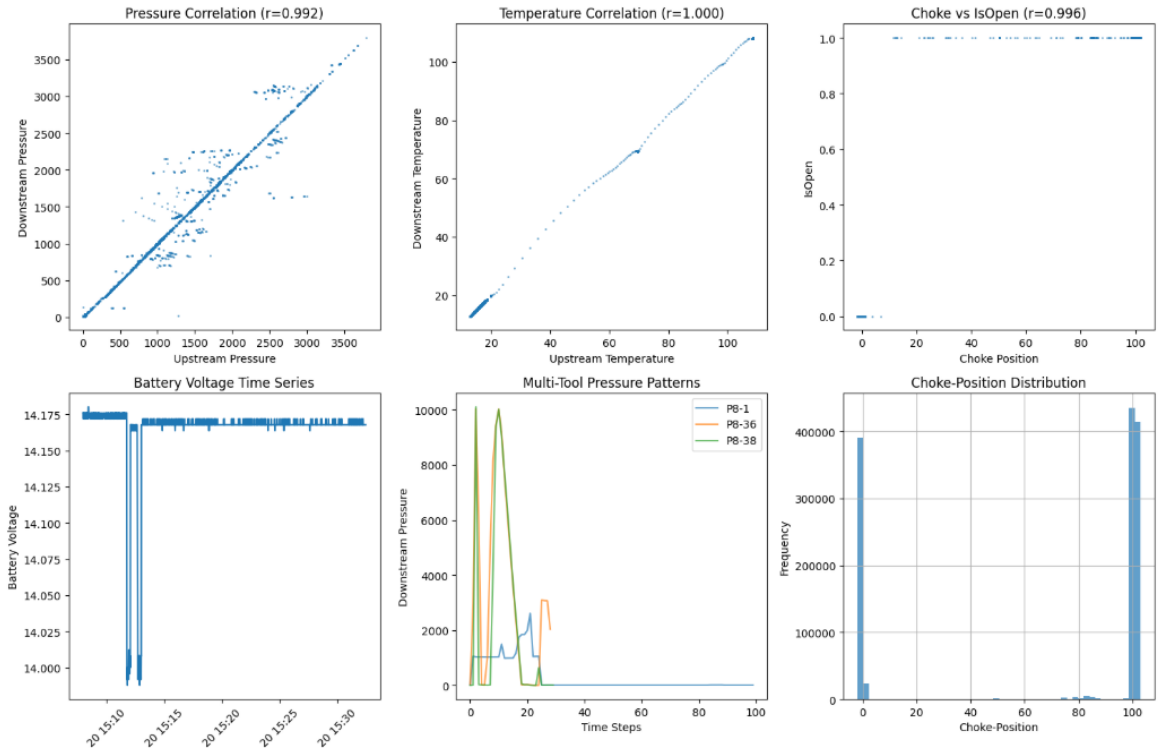

Figure 18: Pressure regime distribution acorss the tool.